УЧЕБНОЕ ПОСОБИЕ
МГТУ им. Н.Э. БАУМАНА

В.В. Бондарев

АНАЛИЗ ЗАЩИЩЕННОСТИ И МОНИТОРИНГ КОМПЬЮТЕРНЫХ СЕТЕЙ

Методы и средства

BAUMANPRESS
ИЗДАТЕЛЬСТВО
МГТУ им. Н. Э. БАУМАНА

В.В. Бондарев

Анализ защищенности и мониторинг компьютерных сетей

Методы и средства

Учебное пособие

Москва
ИЗДАТЕЛЬСТВО
МГТУ им. Н. Э. Баумана
2017

УДК 681.326
ББК 32.973
Б81

Издание доступно в электронном виде на портале *ebooks.bmstu.ru*
по адресу: http://ebooks.bmstu.ru/catalog/117/book1712.html

Факультет «Информатика и системы управления»
Кафедра «Информационная безопасность»

Рекомендовано Редакционно-издательским советом
МГТУ им. Н.Э. Баумана в качестве учебного пособия

Бондарев, В. В.

Б81 Анализ защищенности и мониторинг компьютерных сетей. Методы и средства : учебное пособие / В. В. Бондарев. — Москва : Издательство МГТУ им. Н. Э. Баумана, 2017. — 225, [3] с. : ил.

ISBN 978-5-7038-4757-2

Изложены теоретические вопросы, связанные с архитектурой и принципами работы систем обнаружения и предотвращения атак в компьютерных сетях. Приведены методы, приемы и инструменты, применяемые при защите компьютерных систем и сетей от атак. Содержание учебного пособия соответствует программе и курсу лекций, читаемых автором в МГТУ им. Н.Э. Баумана.

Для студентов, обучающихся по направлению подготовки «Информационная безопасность автоматизированных систем», а также слушателей факультета повышения квалификации. Может представлять интерес для специалистов в области использования современных средств и методов обеспечения информационной безопасности.

УДК 681.326
ББК 32.973

ISBN 978-5-7038-4757-2

Предисловие

В настоящее время деятельность многих организаций зависит от состояния их информационных систем. При этом инфраструктура информационных систем часто содержит узлы и системы, нарушение безопасности которых может привести к нанесению значительного ущерба для ведения бизнеса в организации.

Для предотвращения таких случаев, как правило, после соответствующего анализа формируется перечень актуальных угроз и разрабатывается комплекс мер по их нейтрализации. В конечном итоге строится система управления информационной безопасностью, которая включает в себя различные средства защиты, реализующие необходимые защитные механизмы. В состав данной системы может входить подсистема управления уязвимостями, представляющая собой комплекс организационно-технических мероприятий, направленных на предотвращение использования известных уязвимостей, потенциально существующих в защищаемой системе или в сети. В частности, в рамках управления уязвимостями проводятся такие мероприятия, как периодический мониторинг защищенности информационных систем и устранение обнаруженных уязвимостей.

В последнее время большое внимание уделяется новому направлению в области защиты информации — адаптивной безопасности компьютерной сети. Это направление включает в себя две основные технологии: анализ защищенности (Security Assessment) и обнаружение атак (Intrusion Detection).

Целью данного учебного пособия является ознакомление студентов с теоретическими вопросами, связанными с архитектурой и принципами работы систем обнаружения атак злоумышленников, а также приемами и инструментами, применяемыми при защите компьютерных систем и сетей от атак.

Пособие предназначено для студентов, обучающихся по направлению подготовки «Информационная безопасность» (комплексное обеспечение информационной безопасности автоматизированных систем — КОИБАС, организация и технология защиты информации и т. д.), и слушателей факультета повышения квалификации по этому направлению. Может представлять интерес для студентов и аспирантов других специальностей, занимающихся вопросами использования современных средств и методов обеспечения информационной безопасности компьютерных систем.

В рамках данного учебного пособия рассматриваются следующие темы:

- необходимость и актуальность разработки и внедрения технологий обнаружения и предотвращения атак (IDS/IPS);

• понятийный аппарат в области мониторинга информационной безопасности в части обнаружения и предотвращения вторжений;

• архитектура системы IDS/IPS (источники данных, признаки атак, методы обнаружения атак, механизмы реагирования);

• специализированные технологии IDS/IPS;

• централизованное управление сетевыми и хостовыми технологиями IDS/IPS разных производителей и их взаимодействие с другими механизмами защиты;

• обзор и направления развития перспективных IDS/IPS.

Предлагаемый материал может быть использован при изучении следующих дисциплин: методы и средства обеспечения информационной безопасности, технология защиты компьютерных систем, программно-аппаратные средства защиты информации, аудит информационной безопасности, мониторинг информационных систем и т. д.

Усвоение материала, приведенного в учебном пособии, позволит студентам применять полученные знания в области мониторинга и управления информационной безопасностью и технологий обнаружения атак, распознавать признаки атак, оперировать источниками данных для IDS/IPS, использовать методы обнаружения атак, методы сбора информации о сети, механизмы реагирования и специализированные системы для обнаружения и предотвращения атак на компьютерные системы управления и методы их восстановления, а также иметь навыки идентификации сетевых объектов, определения топологии сети, идентификации статуса порта, сервисов и приложений, операционных систем, централизованного управления уязвимостями, централизованного управления технологиями IDS/IPS и организации их взаимодействия с другими механизмами защиты информационных сетей.

Обозначения, используемые в книге

В книге используются следующие пиктограммы для обозначения сетевых устройств различных типов:

Глава 1. ПОСТАНОВКА ЗАДАЧИ АНАЛИЗА ЗАЩИЩЕННОСТИ КОМПЬЮТЕРНОЙ СИСТЕМЫ

1.1. Корпоративная сеть как объект защиты

Главная задача корпоративной сети — централизованное управление предприятием или объединением предприятий. Сеть обеспечивает передачу информации между различными приложениями, используемыми в данной организации. Взаимодействующие приложения могут быть расположены в разных филиалах организации, территориально удаленных один от другого и соединенных между собой выделенными каналами связи. Обмен информацией осуществляется посредством глобальной сети Интернет.

Типичная конфигурация корпоративной сети представлена на рис. 1.1.

Для защиты компьютерных систем от неправомерного вмешательства в процессы их функционирования и несанкционированного доступа (НСД) к информации используются различные защитные механизмы, например:

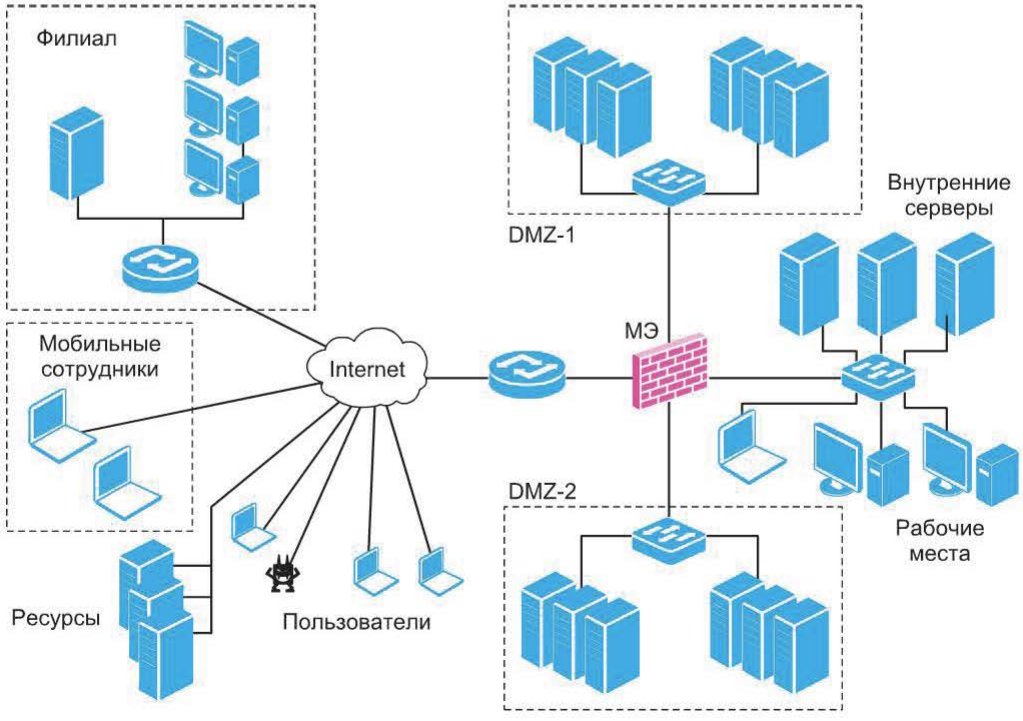


Рис. 1.1. Типичная конфигурация корпоративной сети (МЭ — межсетевой экран)

- идентификация и аутентификация;
- разграничение доступа (и изоляция);
- регистрация событий и аудит;
- контроль целостности;
- шифрование данных и электронная подпись (ЭП);
- резервирование и резервное копирование;
- затирание остаточной информации;
- обнаружение и обезвреживание вирусов;
- фильтрация трафика и трансляция адресов;
- выявление и устранение уязвимостей;
- обнаружение вторжений (атак);
- маскировка и создание ложных объектов;
- страхование рисков.

Приведенные механизмы защиты применяются в конкретных технических средствах и системах защиты в различных комбинациях и вариациях. Наибольший эффект достигается при их системном использовании в комплексе с другими видами мер защиты.

Часто для простоты эти механизмы подразделяют на следующие группы.

Превентивные — направлены на предотвращение нарушений безопасности, например, к этой категории можно отнести разграничение доступа.

Цели применения превентивных механизмов защиты: упреждающее выявление и предотвращение проблемных ситуаций, создание барьеров на пути реализации угроз, резервирование, разделение ролей, использование средств разграничения доступа, контроль доступа в помещения и т. д.

Детективные — позволяют своевременно обнаруживать факт нарушения, например, к ним относятся такие механизмы, как регистрация событий и обнаружение атак.

Цели применения детективных механизмов защиты: выявление проблемных ситуаций и нарушений безопасности во время или после их появления, мониторинг событий безопасности, обнаружение сетевых атак, антивирусное сканирование, проверка контрольных сумм файлов, процедуры внутреннего аудита и т. д.

Коррективные — позволяют за приемлемый срок разрешить проблемные ситуации, выявленные с помощью детектирующих механизмов контроля.

Цели применения корректирующих механизмов защиты: восстановление системы в случае атаки, страхование рисков информационной безопасности, реагирование на нарушения безопасности, ликвидация последствий осуществления угроз и минимизация ущерба, разработка, внедрение и реализация плана восстановления после аварии (план обеспечения непрерывной работы и восстановления), резервное копирование и восстановление данных и т. д.

Теоретически превентивный подход к обеспечению информационной безопасности представляется идеальным, так как позволяет предотвратить реализацию угроз, поскольку самый лучший закон — тот, который невозможно нарушить. Но на практике все не так очевидно. В некоторых случаях

затраты на построение системы обеспечения безопасности, основанной на превентивных мерах, могут оказаться чересчур высокими, что будет противоречить одному из базовых принципов информационной безопасности — принципу разумной достаточности защиты информации.

Контроль состояния защищенности как раз и относится к категории превентивных защитных механизмов. Его главное назначение — своевременно «заметить» слабость (уязвимость) в защищаемой системе и тем самым помочь предотвратить возможные атаки с ее использованием.

Работу приведенных трех групп защитных механизмов можно рассмотреть на простом примере (рис. 1.2). Допустим, нарушитель, воспользовавшись уязвимостью, позволяющей просматривать содержимое файлов на web-сервере, выполнил в отношении него успешную атаку и, получив к нему доступ, модифицировал имеющуюся там информацию.

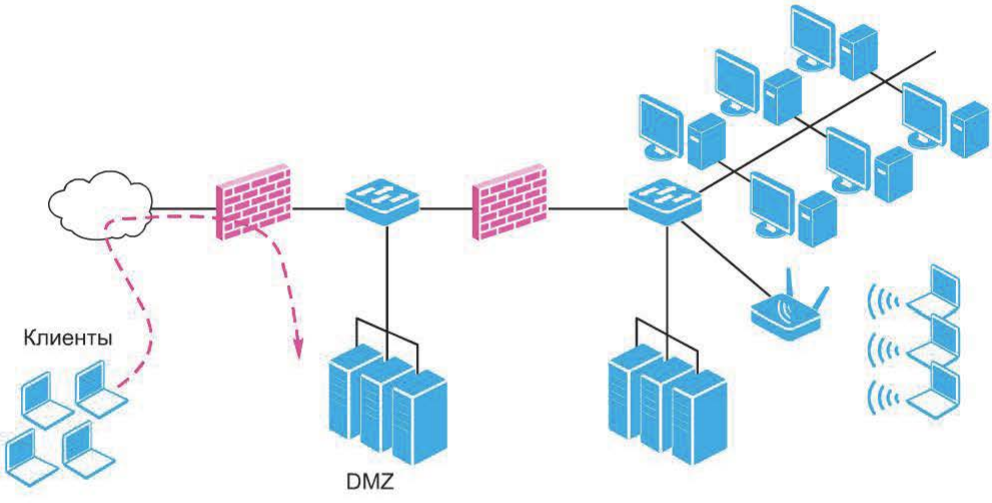


Рис. 1.2. Атака (штриховая линия) на корпоративный web-сервер

Роль защитных механизмов в этом случае могла бы быть следующей:

• функции превентивных механизмов защиты заключаются в анализе защищенности web-сервера перед вводом его в эксплуатацию, что позволяет выявить уязвимость и сделать невозможной атаку с ее использованием;

• детективные механизмы защиты позволяют обнаружить факт атаки с помощью системы обнаружения атак, установленной, например, непосредственно на сервере;

• функции коррективных механизмов защиты сводятся к восстановлению содержимого модифицированных файлов и, возможно, устранению уязвимости, явившейся причиной атаки.

Из приведенного примера следует, что в некоторых случаях наиболее эффективны именно превентивные меры защиты, позволяющие своевременно выявить уязвимость и сделать невозможным ее использование нарушителем.

1.2. Событие безопасности

Во время функционирования узлов сети происходят различные события, изменяющие их состояние. *Событие безопасности* — минимальная единица, которой оперируют современные средства защиты. Примерами событий безопасности могут быть: доступ к файлу, перезагрузка системы, вход в систему и т. д. Другое определение события информационной безопасности — идентифицированное возникновение состояния системы, услуги или сети, указывающее на возможное нарушение политики информационной безопасности, отказ защитных мер, а также возникновение ранее не известной ситуации, которая может быть связана с безопасностью.

Инцидент информационной безопасности — любое непредвиденное или нежелательное событие, которое может нарушить деятельность или информационную безопасность. Примерами инцидентов информационной безопасности могут быть:

- утрата услуг, оборудования или устройств;
- системные сбои или перегрузки;
- ошибки пользователей;
- несоблюдение политик или рекомендаций;
- нарушение физических мер защиты;
- неконтролируемые изменения систем;
- сбои программного обеспечения (ПО) и отказы технических средств;
- нарушение правил доступа.

Активы (*ресурсы*) — вся информация, имеющая ценность для организации и вследствие этого подлежащая защите. Среди значительного числа примеров классификации активов можно выделить следующие:

- материальные (физические) активы;
- информация (данные, контракты, документация);
- ПО;
- способность производить продукт или предоставлять услугу;
- служащие и их квалификация;
- нематериальные ресурсы (престиж фирмы, репутация).

1.3. Понятие уязвимости

Понятие «уязвимость» нельзя рассматривать отдельно от таких понятий, как «угроза» и «атака». Рассмотрим эти три понятия с системных позиций.

Уязвимость (Vulnerability) — некоторая характеристика (свойство) чего-либо (узла сети, службы, протокола), которая может быть использована нарушителем при проведении атаки и привести к реализации угрозы.

Угроза (Threat) — потенциально возможное событие, явление или процесс, которое посредством воздействия на компоненты информационной системы может привести к нанесению ущерба.

Атака (Attack) — любое действие нарушителя, которое приводит к реализации угрозы путем использования уязвимостей информационной системы.

Взаимосвязь этих трех понятий иллюстрирует рис. 1.3.

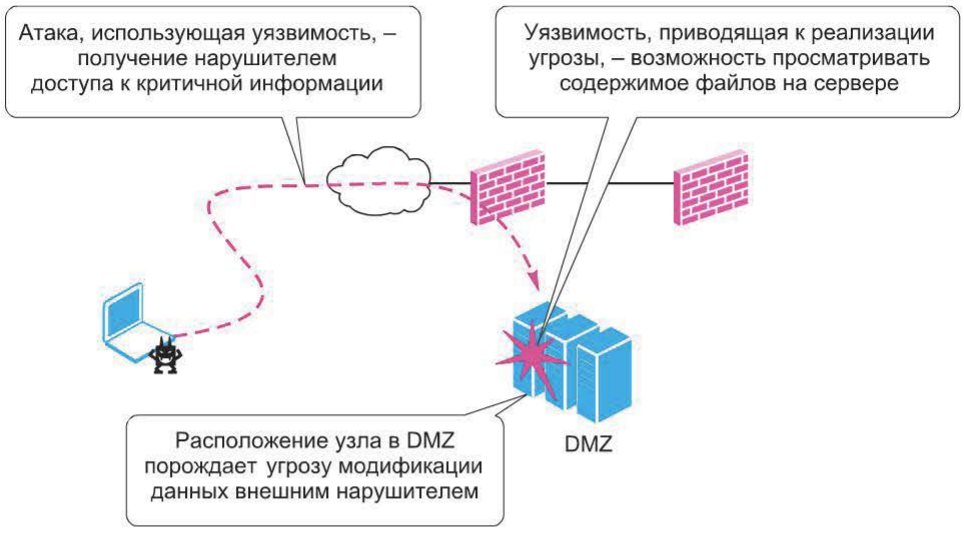


Рис. 1.3. Взаимосвязь угрозы, уязвимости и атаки

Обнаружение вторжений (*атак*) (Intrusion detection) — процесс мониторинга событий, происходящих в компьютерной системе или сети с целью поиска признаков возможных инцидентов.

При этом инциденты могут иметь самую разную природу, например, попытки неавторизованного доступа к ресурсам корпоративной сети или попытки повышения привилегий и т. д.

Система обнаружения атак IDS (Intrusion Detection System) — программное (или программно-аппаратное) обеспечение, автоматизирующее процесс обнаружения атак.

В принципе обнаружение атак может выполняться вручную. Например, в результате анализа журнала аудита ОС могут быть выявлены попытки подбора пароля. Та же самая процедура может быть выполнена и с помощью программного средства, предназначенного для анализа журналов.

Расположение узла в демилитаризованной зоне (Demilitarized Zone, DMZ) (в данном случае — web-сервера) обусловливает угрозу нарушения целостности (например, в результате модификации) данных внешним нарушителем. Уязвимость, приводящая к реализации угрозы, — возможность просмотра содержимого файлов на сервере. Атака, использующая уязвимость, — получение нарушителем доступа к критичной информации, которая позволила ему впоследствии получить доступ к узлу.

В принципе безопасность — это защищенность от возможного ущерба, наносимого при реализации «опасностей» (угроз). Конечной целью обеспе-

чения безопасности корпоративной сети является защита всех категорий субъектов (как внешних, так и внутренних), прямо или косвенно участвующих в процессах информационного взаимодействия, от нанесения им ощутимого материального, морального или иного ущерба в результате случайных или преднамеренных нежелательных воздействий на информацию и системы ее обработки и передачи.

Соответственно одним из возможных вариантов защиты можно считать своевременное выявление слабостей (уязвимостей), которые могут привести к реализации угроз и в результате — нанесению ущерба.

1.4. Классификация уязвимостей

По мере накопления информации об уязвимостях возникали и различные варианты их классификации. В настоящее время информация об обнаруженных уязвимостях достаточно систематизирована, существует несколько общеизвестных источников, где эта информация представлена. Ниже приведены примеры возможных вариантов (критериев) классификации уязвимостей.

Один из самых удачных вариантов классификации уязвимостей — по источнику возникновения. Данный вариант классификации связан с этапами жизненного цикла системы и часто указывает на причину возникновения той или иной уязвимости.

Уязвимости проектирования. Часть уязвимостей возникает на этапе проектирования. Например, значительная часть прикладных сервисов стека TCP/IP (TELNET, FTP и др.) не предусматривает шифрования данных при передаче по сети. В результате критичная информация (например, имя пользователя и пароль) передается по сети в открытом виде (рис. 1.4).

Как правило, уязвимости, возникшие на этапе проектирования, с трудом поддаются устранению. Например, в случае с сервисами прикладного уровня для устранения уязвимостей можно либо отказаться от использования

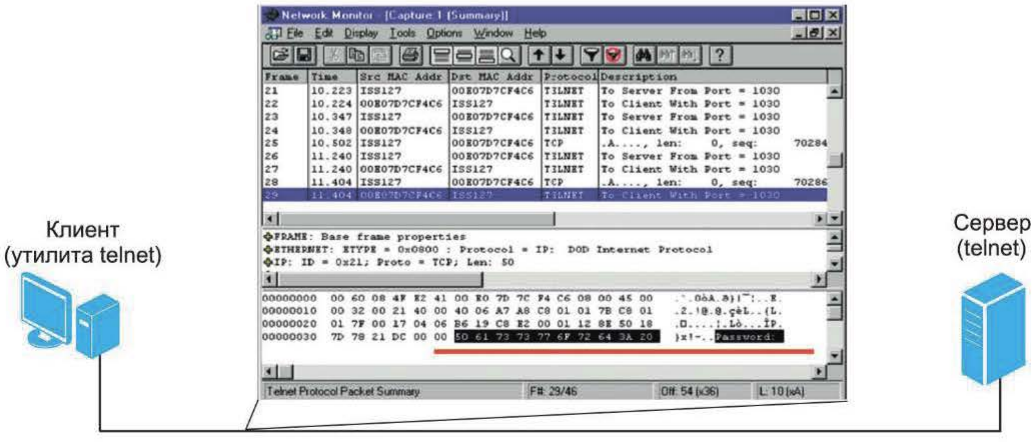


Рис. 1.4. Передача критичной информации в открытом виде

соответствующего протокола, либо применять криптографические защитные механизмы.

Уязвимости реализации. Значительная часть уязвимостей возникает на этапе реализации (программирования). Например, уязвимость CVE-2014-4113, обнаруженная в драйвере win32k.sys всех поддерживаемых версий Windows (2k3 и выше), позволяет несанкционированно (в обход ограничений ОС) выполнить код в режиме ядра и повысить привилегии запускаемого эксплойтом приложения до максимально возможного уровня (SYSTEM) (рис. 1.5).

Рис. 1.5. Описание уязвимости CVE-2014-4113 в каталоге CVE

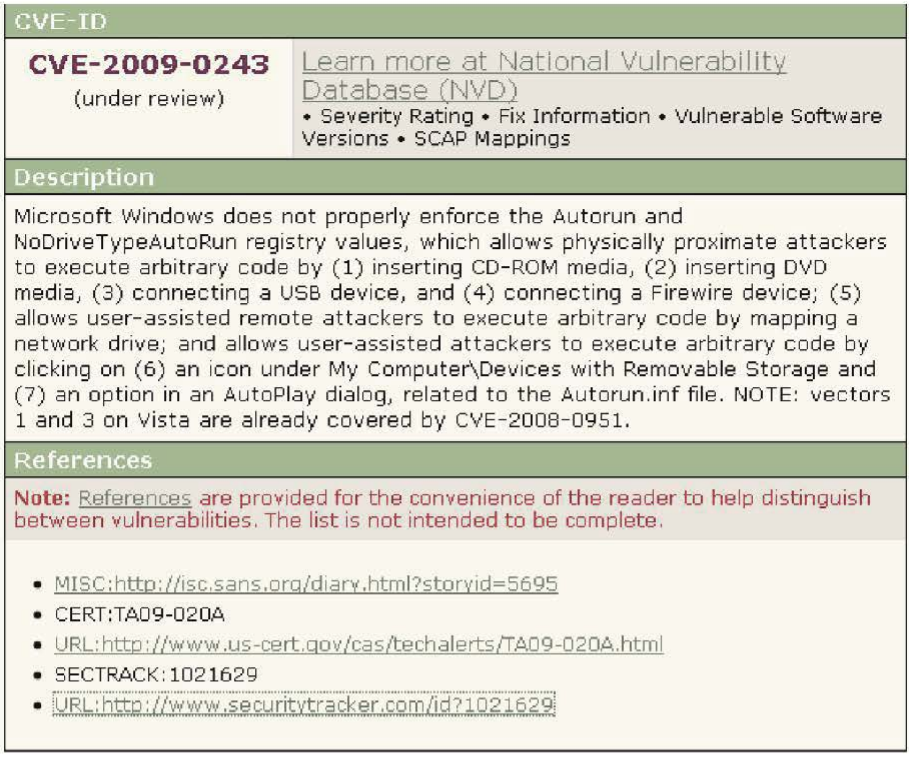
Суть уязвимости заключается в том, что одна из функций драйвера не проверяет возвращаемое ей другой функцией значение (предполагаемый указатель) и передает его дальше. Это в результате приводит к тому, что получатель оперирует неправильным значением.

Уязвимости эксплуатации. Уязвимости могут быть также следствием ошибок, допущенных в процессе эксплуатации информационной системы, в частности, к ним можно отнести:

- использование конфигураций «по умолчанию»;
- некорректно заданные параметры защитных механизмов;
- неиспользуемые сетевые сервисы, доступные удаленно.

Устранение уязвимостей данной группы обычно сводится к внесению соответствующих изменений в конфигурацию системы. В качестве примера можно привести уязвимость CVE-2009-0243, заключающуюся в том, что функционал autorun, часто используемый вредоносным ПО как один из способов распространения, не удается выключить окончательно, даже следуя рекомендациям Microsoft (рис. 1.6).

Подробная и правильная инструкция для устранения данной уязвимости предоставлена координационным центром CERT.


CVE-ID

CVE-2009-0243
(under review)

Learn more at National Vulnerability Database (NVD)
• Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings

Description

Microsoft Windows does not properly enforce the Autorun and NoDriveTypeAutoRun registry values, which allows physically proximate attackers to execute arbitrary code by (1) inserting CD-ROM media, (2) inserting DVD media, (3) connecting a USB device, and (4) connecting a Firewire device; (5) allows user-assisted remote attackers to execute arbitrary code by mapping a network drive; and allows user-assisted attackers to execute arbitrary code by clicking on (6) an icon under My Computer\Devices with Removable Storage and (7) an option in an AutoPlay dialog, related to the Autorun.inf file. NOTE: vectors 1 and 3 on Vista are already covered by CVE-2008-0951.

References

Note: References are provided for the convenience of the reader to help distinguish between vulnerabilities. The list is not intended to be complete.

- MISC:http://isc.sans.org/diary.html?storyid=5695
- CERT:TA09-020A
- URL:http://www.us-cert.gov/cas/techalerts/TA09-020A.html
- SECTRACK:1021629
- URL:http://www.securitytracker.com/id?1021629

Рис. 1.6. Описание уязвимости CVE-2009-0243

1.5. Источники информации по уязвимостям

Попытки систематизации информации об обнаруженных уязвимостях привели к появлению нескольких крупных общедоступных источников, содержащих подобного рода сведения. Ниже представлено несколько примеров такой систематизации.

База уязвимостей (Bugtraq). Один из самых известных ресурсов Security Focus (www.securityfocus.com/bid), содержащих информацию об обнаруженных уязвимостях в ПО (рис. 1.7), существует с 1999 г. В 2002 г. был приобретен компанией Symantec.

Уязвимости, помещаемые в базу Bugtraq, обозначаются уникальным индексом BID (Bugtraq ID), который используется многими программными продуктами для ссылок на уязвимости или атаки (рис. 1.8).

База уязвимостей X-Force. Группа X-Force — это команда исследователей и разработчиков, занимающаяся как анализом ПО на наличие уязвимостей, так и мониторингом информации об уязвимостях, поступающей из различных источников: списков рассылки, сайтов эксплойтов или непосредственно от производителя ПО. Группа X-Force (xforce.iss.net) создана компанией ISS (Internet Security Systems) для обновления баз сигнатур своих

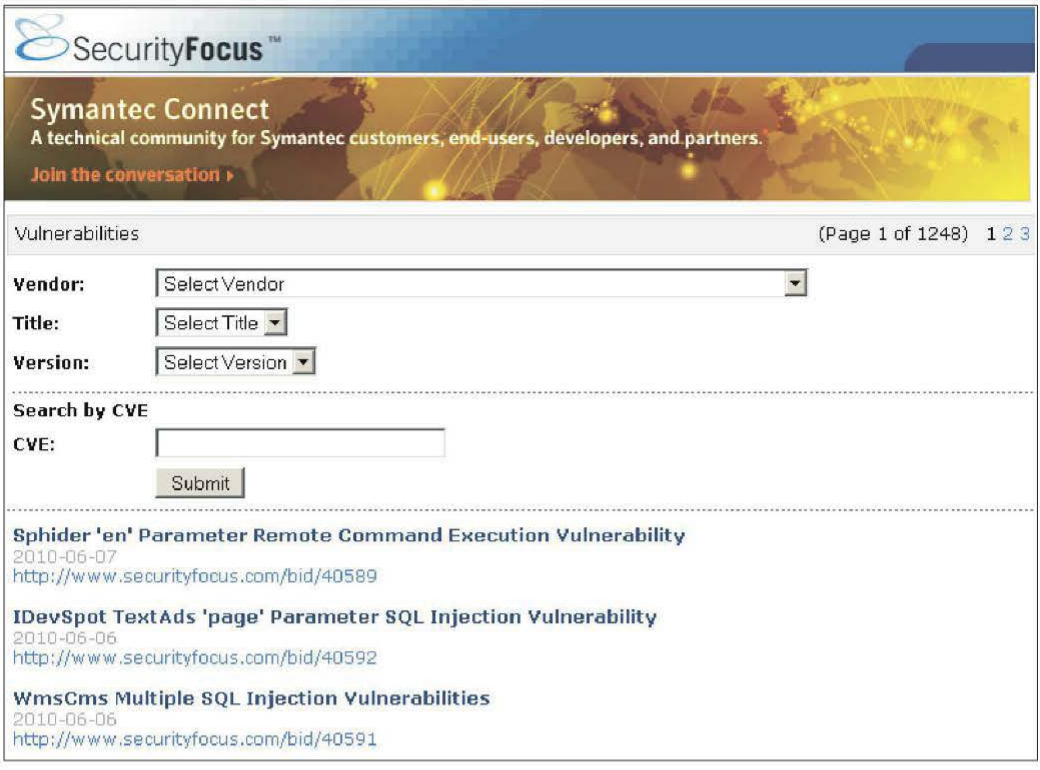


Рис. 1.7. База уязвимостей Bugtraq

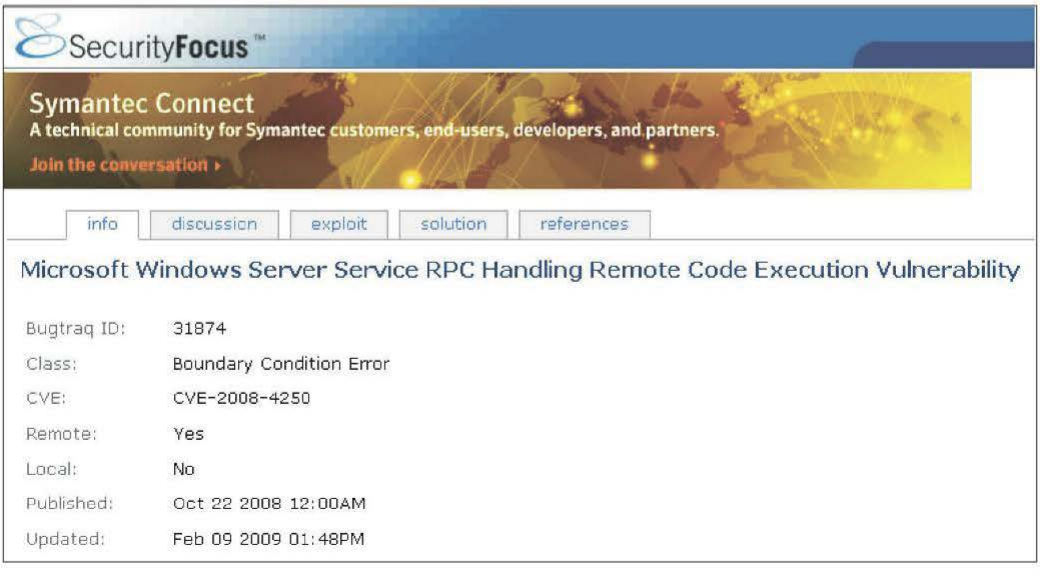


Рис. 1.8. Пример уязвимости из базы Bugtraq

продуктов. В 2006 г. компания ISS была приобретена компанией IBM. Пример описания уязвимости из базы X-Force представлен на рис. 1.9. Видно, что уязвимости присвоены уникальный номер и идентификатор, состоящий из ключевых слов, характеризующих уязвимость.

Рис. 1.9. Пример уязвимости из базы X-Force

База уязвимостей US-CERT Vulnerability Notes Database. Координационный центр CERT (Coordination Center, CERT/CC) создан как команда реагирования на инциденты в области информационной безопасности. Кроме того, центр занимается собственными исследованиями в области выявления уязвимостей, результатом которых стало создание базы US-CERT Vulnerability Notes Database (www.kb.cert.org/vuls/).

Уязвимости обозначаются префиксом VU# и имеют уникальный номер (рис. 1.10).

SecurityTracker. Популярным ресурсом является SecurityTracker (securitytracker.com). Для обозначения уязвимостей в этой базе используется уникальный числовой идентификатор (рис. 1.11).

Secunia. Компания Secunia, занимающаяся исследованиями в области выявления уязвимостей и разработкой соответствующего ПО, также поддерживает собственную базу уязвимостей (рис. 1.12).

Open Source Vulnerability Database. Идея создания базы OSVDB (Open Source Vulnerability Database) предложена в 2002 г. на одной из конференций

Рис. 1.10. Пример уязвимости из базы US-CERT

Рис. 1.11. Пример уязвимости из базы SecurityTracker

Рис. 1.12. Пример описания уязвимости в базе Secunia

Defcon, ресурс osvdb.org открыт для публичного доступа в 2004 г. (рис. 1.13). Его отличительной особенностью можно назвать возможность скачивания базы данных (БД) в виде готового файла для системы управления БД (СУБД).

Банк данных угроз безопасности информации. В начале 2015 г. Федеральная служба по техническому и экспортному контролю (ФСТЭК) России представила свой вариант базы уязвимостей — Банк данных угроз безопасности информации (рис. 1.14).

Уязвимости в этой базе обозначены идентификаторами, содержащими год добавления в базу и уникальный номер (рис. 1.15).

Как видно из описания уязвимости, в данной базе задействовано довольно много критериев классификации уязвимостей.

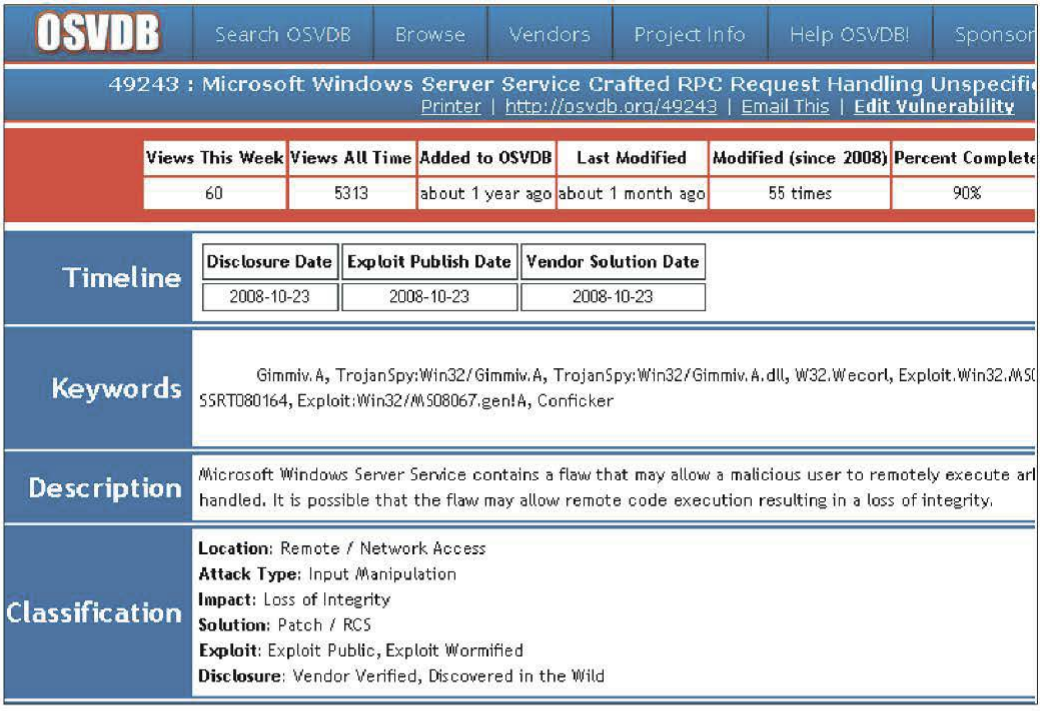


Рис. 1.13. Пример уязвимости из базы OSVDB

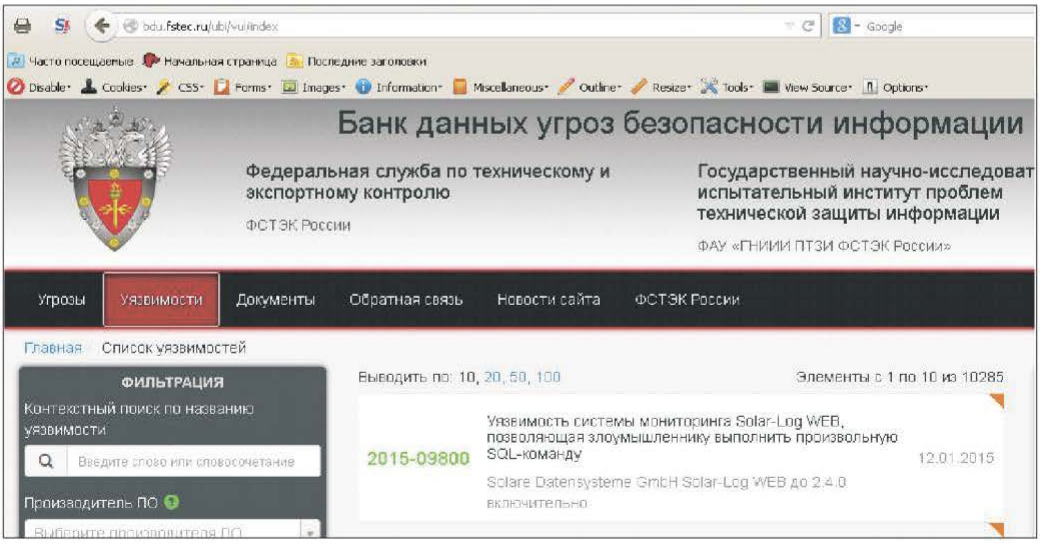


Рис. 1.14. Банк данных угроз безопасности информации

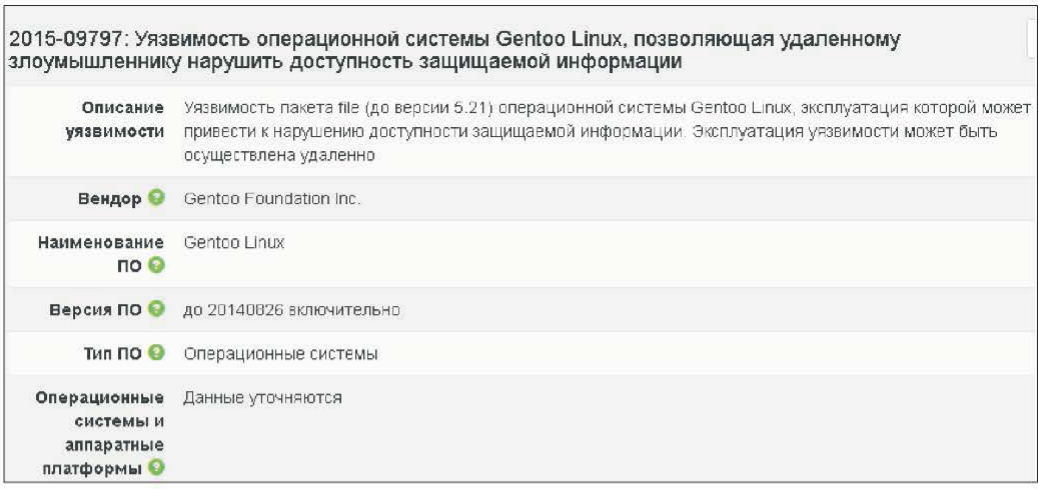


Рис. 1.15. Пример уязвимости из базы ФСТЭК России

1.6. Принятые обозначения уязвимостей

Common Vulnerabilities and Exposures. Выше были приведены источники информации по уязвимостям, в основном поддерживаемые компаниями, занимающимися собственными исследованиями в этой области или разработкой соответствующего ПО. При этом каждый из ресурсов имеет собственную систему обозначений для уязвимостей и формат их описания, в то время соответствующие ссылки, позволяющие быстро найти информацию по определенной уязвимости в разных источниках, не всегда имеются в наличии.

На рис. 1.16 приведены примеры названия уязвимости CVE-2008-4250, которая привела к массовому заражению систем Windows вирусом Kido/Conficker в различных базах.

Рис. 1.16. Примеры названия уязвимости CVE-2008-4250

Обеспечить совместимость между разными источниками поможет каталог уязвимостей CVE (Common Vulnerabilities and Exposures). Проект CVE (http://cve.mitre.org), запущенный в 1999 г., в настоящее время фактически стал промышленным стандартом для обозначения уязвимостей.

Запись об уязвимости в каталоге CVE содержит уникальный индекс, краткое описание уязвимости и ссылки на источники, где можно получить более подробную информацию (рис. 1.17).

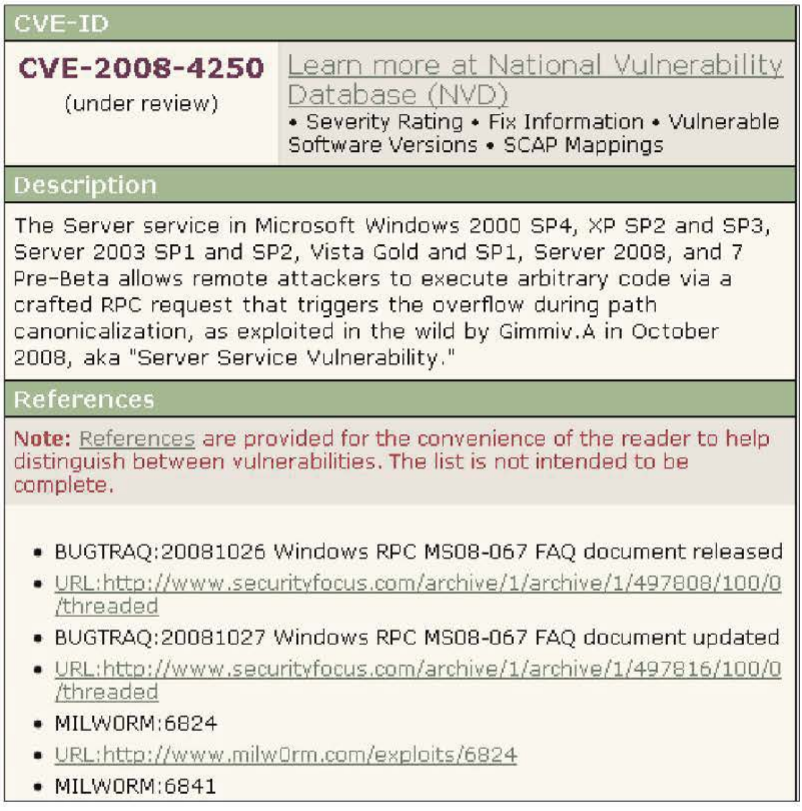


Рис. 1.17. Информация об уязвимости в каталоге CVE

Обозначение уязвимости включает в себя префикс «CVE», год обнаружения и уникальный номер, например CVE-2008-4250. Представленная краткая информация об уязвимости помогает лишь в общих чертах понять, о чем идет речь, для получения более подробной информации можно воспользоваться приведенными ссылками. В свою очередь, представленные выше источники информации об уязвимостях, как правило, содержат соответствующую ссылку на каталог CVE, что и обеспечивает совместимость различных баз уязвимостей.

В дополнение к каталогу CVE можно рекомендовать ресурс http://www.cvedetails.com/ (рис. 1.18), содержащий необходимую статистику по уязвимостям и удобные средства поиска.

Рис. 1.18. Ресурс http://www.cvedetails.com/

1.7. National Vulnerability Database

Еще одним каталогом уязвимостей, основанным на CVE, является NVD (National Vulnerability Database) (http://nvd.nist.gov). В этом каталоге по каждой уязвимости приведена примерно такая же информация, как и в CVE (рис. 1.19).

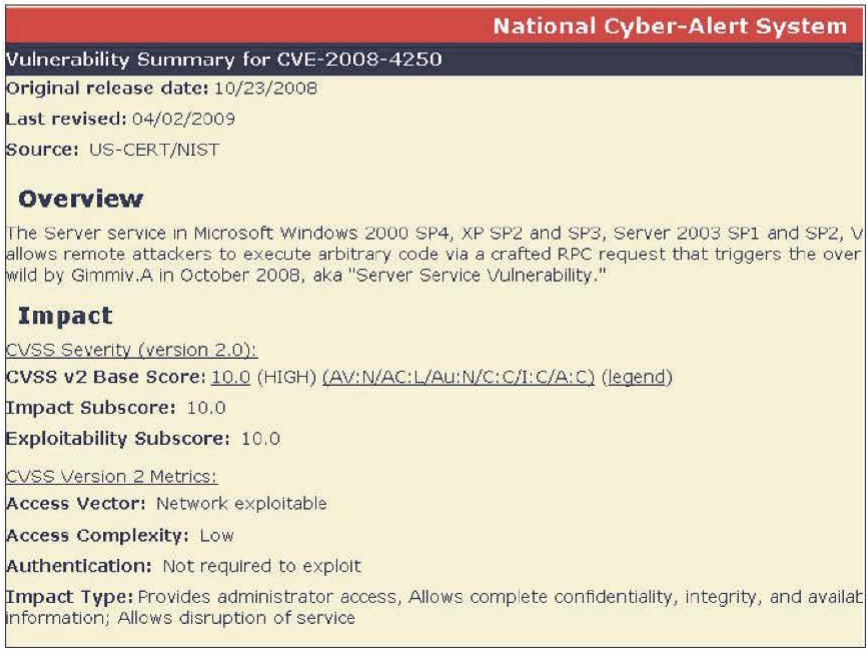


Рис. 1.19. Фрагмент информации об уязвимости в каталоге NVD

В этом каталоге приведено несколько существенных дополнений, в частности, по каждой уязвимости представлены числовые показатели, характеризующие степень ее опасности. Для расчета используется система CVSS (Common Vulnerability Scoring System).

Вторым дополнением является указание типа уязвимости Vulnerability Type (рис. 1.20).

Рис. 1.20. Тип уязвимости в каталоге NVD

Типы уязвимостей, используемые в каталоге NVD, представлены в табл. 1.1.

Таблица 1.1

Типы уязвимостей, используемые в каталоге NVD

Name	CWE-ID	Description
Authentication Issues	CWE-287	Failure to properly authenticate users
Buffer Errors	CWE-119	Buffer overflows and other buffer boundary errors in which a program attempts to put more data in a buffer than the buffer can hold, or when a program attempts to put data in a memory area outside of the boundaries of the buffer
Cross-Site Request Forgery (CSRF)	CWE-352	Failure to verify that the sender of a web request actually intended to do so. CSRF attacks can be launched by sending a formatted request to a victim, then tricking the victim into loading the request (often automatically), which makes it appear that the request came from the victim. CSRF is often associated with XSS, but it is a distinct issue

Продолжение табл. 1.1

Name	CWE-ID	Description
Credentials Management	CWE-255	Failure to properly create, store, transmit, or protect passwords and other credentials
Permissions, Privileges, and Access Control	CWE-264	Failure to enforce permissions or other access restrictions for resources, or a privilege management problem
Cross-Site Scripting (XSS)	CWE-79	Failures of a site to validate, filter, or encode user input before returning it to another user's web client
Cryptographic Issues	CWE-310	An insecure algorithm or the inappropriate use of one; an incorrect implementation of an algorithm that reduces security; the lack of encryption (plaintext); also, weak key or certificate management, key disclosure, random number generator problems
Path Traversal	CWE-22	When user-supplied input can contain ".." or similar characters that are passed through to file access APIs, causing access to files outside of an intended subdirectory
Code Injection	CWE-94	Causing a system to read an attacker-controlled file and execute arbitrary code within that file. Includes PHP remote file inclusion, uploading of files with executable extensions, insertion of code into executable files, and others
Format String Vulnerability	CWE-134	The use of attacker-controlled input as the format string parameter in certain functions
Configuration	CWE-16	A general configuration problem that is not associated with passwords or permissions
Information Leak / Disclosure	CWE-200	Exposure of system information, sensitive or private information, fingerprinting, etc
Input Validation	CWE-20	Failure to ensure that input contains well-formed, valid data that conforms to the application's specifications. Note: this overlaps other categories like XSS, Numeric Errors, and SQL Injection
Numeric Errors	CWE-189	Integer overflow, signedness, truncation, underflow, and other errors that can occur when handling numbers

Окончание табл. 1.1

Name	CWE-ID	Description
OS Command Injections	CWE-78	Allowing user-controlled input to be injected into command lines that are created to invoke other programs, using system() or similar functions
Race Conditions	CWE-362	The state of a resource can change between the time the resource is checked to when it is accessed
Resource Management Errors	CWE-399	The software allows attackers to consume excess resources, such as memory exhaustion from memory leaks, CPU consumption from infinite loops, disk space consumption, etc
SQL Injection	CWE-89	When user input can be embedded into SQL statements without proper filtering or quoting, leading to modification of query logic or execution of SQL commands
Link Following	CWE-59	Failure to protect against the use of symbolic or hard links that can point to files that are not intended to be accessed by the application
Other	No Mapping	NVD is only using a subset of CWE for mapping instead of the entire CWE, and the weakness type is not covered by that subset
Not in CWE	No Mapping	The weakness type is not covered in the version of CWE that was used for mapping
Insufficient Information	No Mapping	There is insufficient information about the issue to classify it; details are unkown or unspecified
Design Error	No Mapping	Vulnerability is characterized as a “Design error” if there exists no errors in the implementation or configuration of a system, but the initial design causes a vulnerability to exist

Понятие «тип уязвимости» указывает на ее характер и имеет определенный практический смысл.

Приведенные в табл. 1.1 типы уязвимостей являются подмножеством системы классификации CWE (Common Weakness Enumeration) (рис. 1.21).

Проект CWE является попыткой создания единой системы классификации уязвимостей по разным критериям.

Для каждого из приведенных выше типов уязвимостей имеется подробное описание. Рассмотрим, например, один из самых распространенных ти-

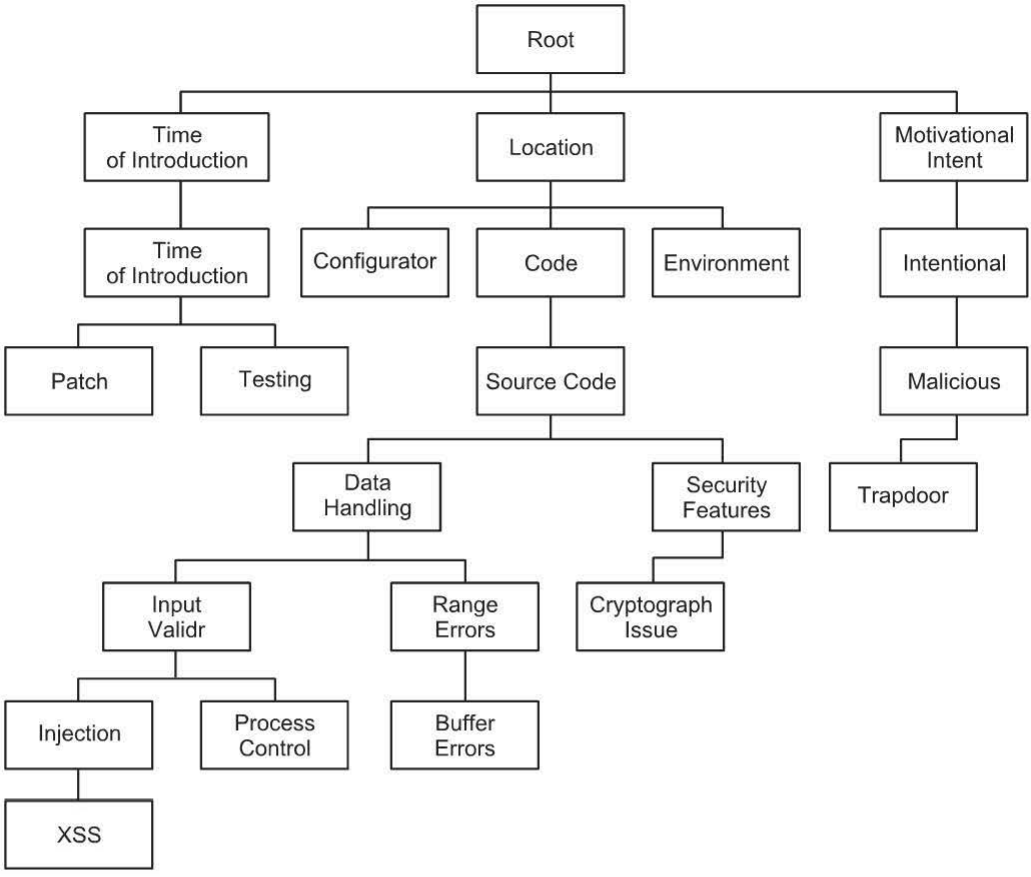


Рис. 1.21. Система классификации уязвимостей CWE

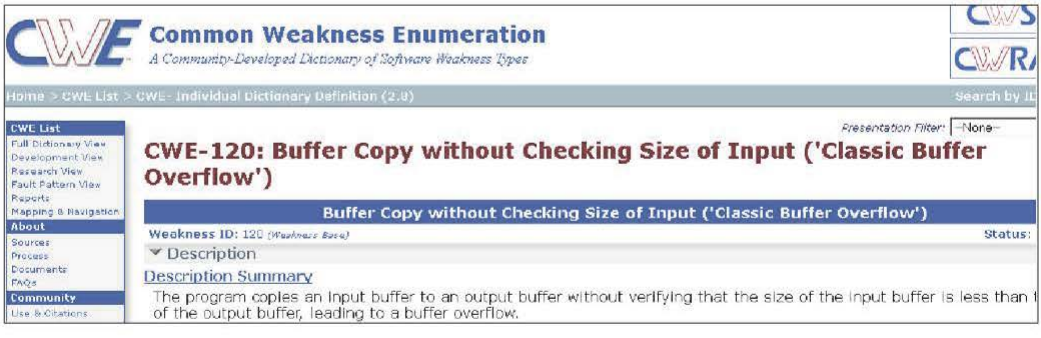


Рис. 1.22. Подробное описание уязвимостей типа CWE-120

пов — «классическое» переполнение буфера (CWE-120, http://cwe.mitre.org/data/definitions/120.html) (рис. 1.22).

Суть уязвимости заключается в том, что программа выполняет копирование «входного» буфера в «выходной», не убедившись в том, что размер

«входного» буфера меньше «выходного». Другими словами, программа пытается скопировать в буфер больше данных, чем он может вместить.

Самый простой пример — использование функции strcpy() языка C:

```
void manipulate_string(char* string){
char buf[24];
strcpy(buf, string);
...
}
```

В приведенном фрагменте кода не предусмотрена проверка соответствия размера переменной string выделенному для нее буферу (buf), что может создать условия для переполнения.

1.8. Уязвимости и безопасность промышленных систем управления

Рассмотрим примеры, относящиеся к уязвимостям систем SCADA, базирующиеся на исследованиях, проведенных экспертами компании Positive Technologies (http://www.ptsecurity.ru/about/news/40989/).

Понятие SCADA. Поясним, что понимается под термином «SCADA». SCADA (Supervisory Control and Data Acquisition) (диспетчерское управление и сбор данных) — программный пакет, предназначенный для разработки или обеспечения работы в реальном времени систем сбора, обработки, отображения и архивирования информации об объекте мониторинга или управления. SCADA может являться частью автоматизированной системы управления технологическим процессом (АСУ ТП), системы экологического мониторинга, научного эксперимента, автоматизации задания и т. д. Системы SCADA используются во всех отраслях хозяйства, где требуется обеспечивать операторский контроль за технологическими процессами в реальном времени. Данное ПО устанавливается на компьютеры и для связи с объектом использует драйверы ввода-вывода или OPC/DDE-серверы. Программный код может быть как написан на языке программирования (например, на C++), так и сгенерирован в среде проектирования.

Исследователями отмечено, что в 2014 г. произошел двукратный рост числа умышленных вредительств, разработанных для атак на системы SCADA.

Промышленные системы управления за последние годы вышли на принципиально новый уровень благодаря развитию информационных технологий и сети Интернет. Однако новый виток автоматизации создал и новые проблемы: некорректное применение технологий защиты и обработки данных приводит к серьезным уязвимостям.

В связи с этим промышленные системы управления все чаще становятся мишенью для злоумышленников. На смену отдельным «червям» Stuxnet

(2010) и Flame (2012) пришли более изощренные схемы многоступенчатых атак. Так, для распространения трояна Havex в 2014 г. хакеры взламывали сайты производителей ПО для управления промышленными предприятиями (SCADA) и заражали официальные дистрибутивы систем SCADA, которые затем устанавливались на предприятиях, что позволило злоумышленникам получить контроль над системами управления в нескольких европейских странах.

Тенденции защищенности АСУ ТП. Среди общих тенденций, которые наблюдаются в процессе работ по анализу защищенности АСУ ТП, можно отметить следующие.

Открытые двери. Многие системы, управляющие производством, транспортом, водоснабжением и энергоресурсами, можно найти в Интернете с помощью общедоступных поисковых систем. В январе 2015 г. исследователи Positive Technologies обнаружили таким образом более 140 000 различных компонентов АСУ ТП. Причем владельцы таких систем не осознают, насколько хорошо их ресурсы «видны снаружи». Это позволяет обнаруживать возможности для атак на АСУ ТП через kiosk mode и облачные сервисы, через сенсоры и физические порты, через индустриальный Wi-Fi и другие виды доступа, которые зачастую вообще не рассматриваются как угрозы.

Один ключ ко многим замкам. Быстрый рост числа организаций, внедряющих АСУ ТП, при ограниченном числе производителей приводит к состоянию, когда одна и та же платформа SCADA используется для управления критически важными объектами в разных отраслях. Например, обнаружены уязвимости в системе, которая управляет Большим адронным коллайдером (БАК), несколькими аэропортами Европы и атомными электростанциями Ирана, крупнейшими трубопроводами и установками водоснабжения в разных странах, поездами и химическими заводами в России. Одна и та же уязвимость, обнаруженная и изученная, позволяет злоумышленникам атаковать множество разных объектов по всему миру.

Угрозы развиваются быстрее, чем защита. Сложная организация АСУ ТП и требование непрерывности технологических процессов, с одной стороны, приводят к тому, что базовые компоненты систем управления (индустриальные протоколы, ОС, СУБД) устаревают, но не обновляются, и их уязвимости не устраняются в течение многих лет. С другой стороны, развитие автоматизированных инструментов значительно увеличивает скорость работы хакеров. В рамках конкурса Critical Infrastructure Attack на форуме PHDays IV в течение двух дней было взломано несколько современных платформ SCADA, которые используются на промышленных предприятиях.

«Безумный дом». Термин «АСУ ТП» появился в 1980-е гг., когда основными объектами автоматизации являлись крупные промышленные предприятия. Однако удешевление и миниатюризация техники привели к тому, что компьютеризированные устройства, управляющие жизнеобеспечением зданий, системами мониторинга и распределения электроэнергии, активно входят в повседневную жизнь. При этом ни производители, ни по-

требители не уделяют должного внимания безопасности этих систем: в данном исследовании показано, как много подобных устройств доступно через Интернет.

Методика исследования. Для сбора информации об уязвимостях использовали базы уязвимостей (ICS-CERT, NVD/CVE, SCADA Strangelove, Siemens Product CERT и др.), сборники эксплойтов (SAINTexploit, Metasploit Framework, Immunity Canvas и др.), уведомления производителей, а также доклады научных конференций и публикации на специализированных сайтах.

Опасность уязвимостей определяли на основе CVSS v.2. Необходимо учитывать, что на статистику влияют такие факторы, как отсутствие типовых описаний уязвимостей или политика разглашения: зачастую производители преуменьшают риск или совсем не разглашают информацию об уязвимостях (более подробно об этих факторах можно прочитать в полной версии отчета). Таким образом, реальная ситуация с безопасностью АСУ ТП может быть даже хуже, чем показывает наша статистика.

Общая система оценки уязвимостей (Common Vulnerability Sevring Sistem, CVSS) является промышленным стандартом и используется для оценки безопасности компьютерной системы (computer system security) и ее слабых мест (vulnerabilities). Эта классификационная система позволяет установить, насколько конкретная уязвимость более критична по отношению к другим известным уязвимостям. Применяя этот подход, можно ранжировать уязвимости и соответственно оперативность и усилия по их устранению.

Сбор данных о доступности АСУ ТП в сети Интернет осуществлялся пассивными методами с использованием общедоступных поисковиков (Shodan, Project Sonar, Google, Bing) и результатов сканирования портов. Анализ данных проводили с использованием БД фингерпринтов, состоящей из 740 записей, которые позволяют на основе баннера сделать заключение о производителе и версии продукта. Большинство фингерпринтов относятся к протоколам SNMP(240) и HTTP(113), примерно треть — к различным промышленным протоколам (Modbus, DNP3, S7 и пр.).

Число уязвимостей. Всего в рамках исследования выявлена 691 уязвимость в компонентах АСУ ТП (рис. 1.23). Отмечается их резкий рост после 2009 г.: за три следующих года (2010–2012) число обнаруженных уязвимостей АСУ ТП выросло в 20 раз (с 9 до 192). После этого среднегодовое число выявляемых уязвимостей стабилизировалось (181 в 2014 г.).

Анализ уязвимостей. Уровень опасности выявленных уязвимостей также сохраняет тенденции 2012 г. Основное число уязвимостей имеет высокую (58 %) и среднюю (39 %) степени опасности.

Если рассмотреть векторы CVSS, то больше половины уязвимостей имеют высокую метрику по такому важному показателю, как доступность. Кроме того, высок показатель удаленной эксплуатации, что в совокупности со слабыми механизмами аутентификации повышает риск атак.

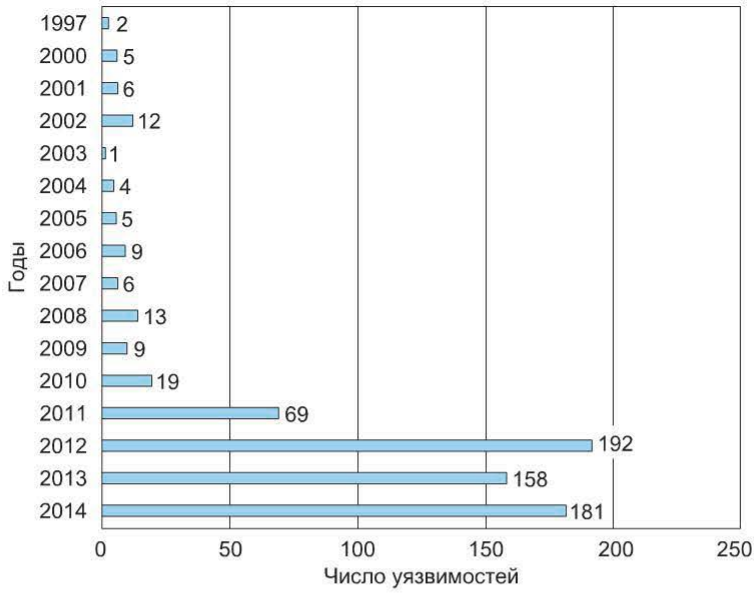


Рис. 1.23. Число уязвимостей в АСУ ТП с 1997 по 2014 гг.

Поскольку в открытом доступе информация о процессе устранения уязвимостей не публикуется, в исследовании использованы данные, полученные экспертами компании Positive Technologies от производителей. Ситуация выглядит более удручающей, чем в 2012 г., когда большинство недостатков безопасности (около 81 %) были оперативно ликвидированы производителями еще до того, как о них становилось широко известно, или в течение 30 дней после нескоординированного разглашения информации. По данным на I квартал 2015 г., лишь 14 % уязвимостей были устранены в течение трех месяцев, 34 % — более трех месяцев, а оставшиеся 52 % — либо не исправлены, либо производитель не сообщает время их устранения (рис. 1.24).

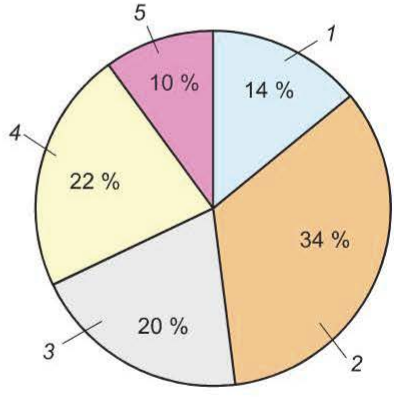


Рис. 1.24. Устранение уязвимостей АСУ ТП:

1 — устранены в течение трех месяцев; *2* — более трех месяцев; *3* — отправлены производителю, в данный момент время устранения не определено; *4* — отправлены производителю, информация по статусу не известна; *5* — не исправлены

Уязвимости производителей. Список производителей, лидирующих по числу уязвимостей в собственных продуктах, практически не изменился: Siemens (124 уязвимости), Schneider Electric вместе с приобретенной ею компанией Invensys (96), Advantech (51), General Electric (31). В то же время общий список производителей с выявленными уязвимостями вырос. На рис. 1.25 представлены

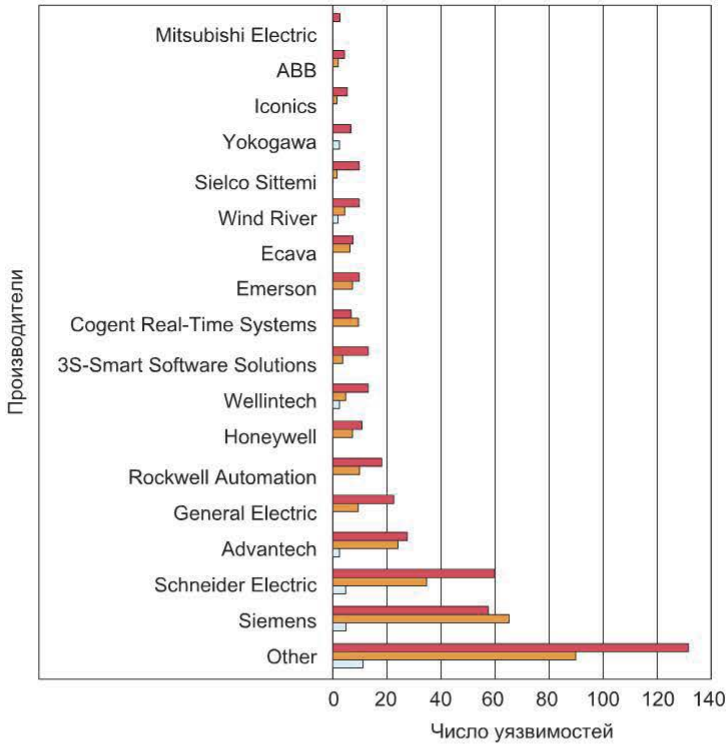


Рис. 1.25. Число уязвимостей в АСУ ТП различных производителей

компании с наибольшим числом уязвимостей по степени риска (отмечены цветом); остальные 88 производителей объединены в строке Other.

География доступности и уязвимости АСУ ТП. Всего в рамках исследования выявлено 146 137 компонентов АСУ ТП, к которым можно получить доступ через Интернет. Самыми распространенными являются системы для автоматизации зданий Tridium (Honeywell), а также системы мониторинга и управления электроэнергией, в том числе на основе технологий солнечных батарей (SMA Solar Technology). Наибольшее число доступных компонентов — PLC/RTU, на втором месте системы мониторинга и управления инверторами. Далее следуют сетевые устройства и HMI/SCADA-компоненты.

Страны, являющиеся технологическими лидерами, имеют высокий уровень автоматизации, поэтому концентрация промышленных систем этих стран в Интернете довольно высока (рис. 1.26). Лидером, как и прежде, остается США (33 %), но на втором месте уже не Италия, а Германия, причем с большим отрывом (19 %). В целом Европейский регион показал заметный рост интернет-доступности промышленных систем. При этом в Азиатском регионе распространены локальные, недостаточно известные на мировом рынке компоненты АСУ ТП, которые не всегда удается идентифицировать.

Путем анализа версий доступных компонентов АСУ ТП было выявлено более 15 000 уязвимых компонентов: наибольшее число в США, затем

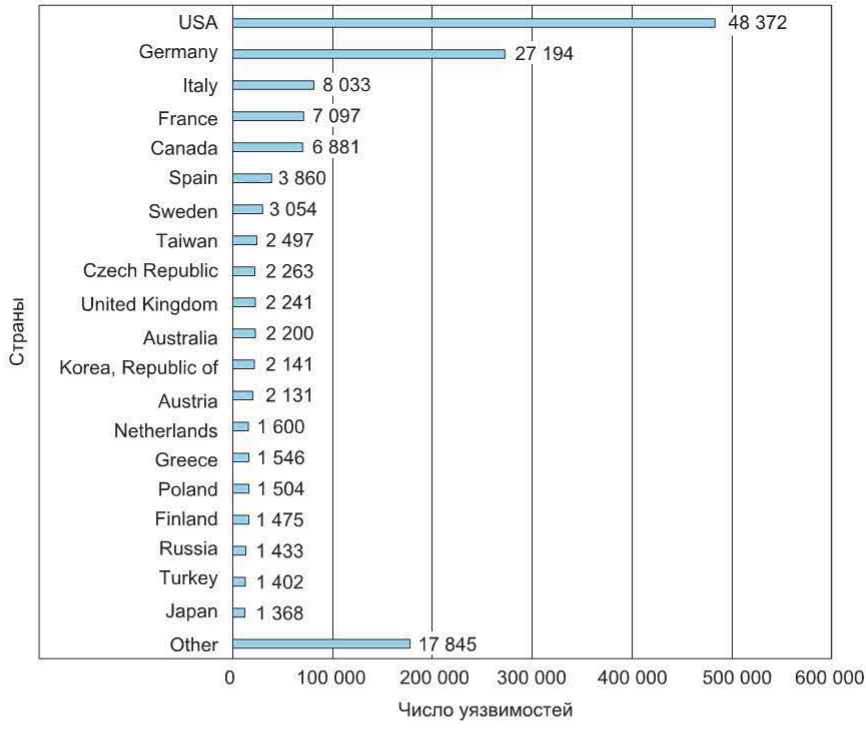


Рис. 1.26. Распространение доступных АСУ ТП

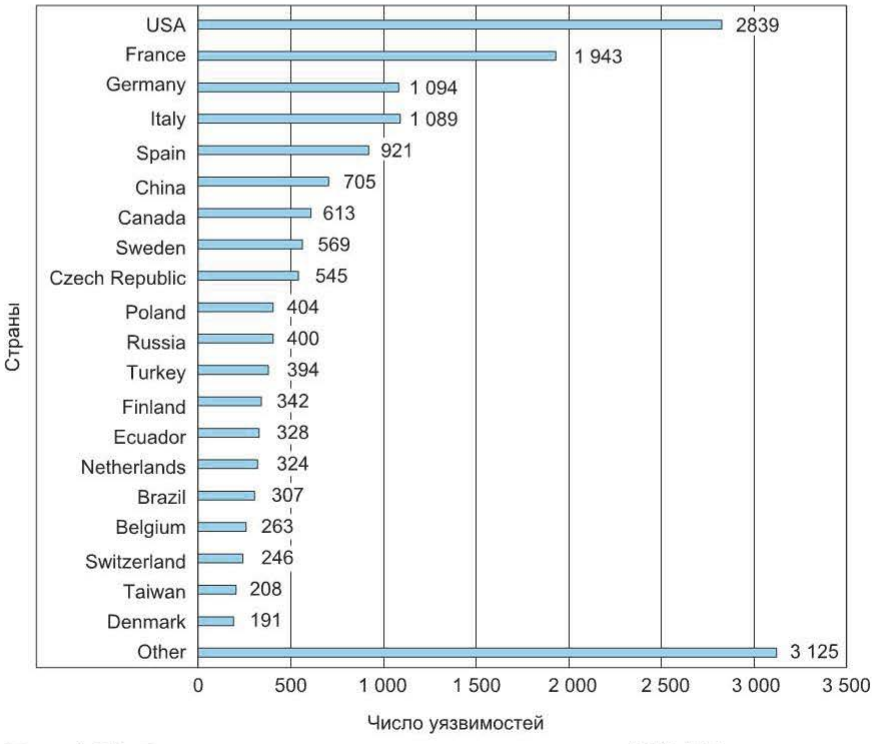


Рис. 1.27. Распределение уязвимых компонентов АСУ ТП по странам

следуют Франция, Италия и Германия, что согласуется с общей картиной распространенности этих систем. Необходимо отметить, что в компонентах, наиболее используемых в сети Интернет, уязвимостей выявлено мало. В целом уязвимыми оказались более 10 % доступных АСУ ТП (см. рис. 1.26, рис. 1.27).

Контрольные вопросы

1. Какова цель защиты корпоративной информационной системы?
2. Назовите защитные механизмы. Как их можно классифицировать?
3. Дайте определения понятиям «угроза информационной безопасности», «уязвимость» и «атака». Опишите взаимосвязь между ними.
4. Перечислите основные классификационные схемы уязвимостей. Каковы их основные достоинства и недостатки?
5. Назовите основные источники информации об уязвимостях. Какая из общедоступных баз данных об уязвимостях более предпочтительна?

Глава 2. МЕТОДЫ ВЫЯВЛЕНИЯ УЯЗВИМОСТЕЙ И СИСТЕМЫ АНАЛИЗА ЗАЩИЩЕННОСТИ

2.1. Основные приемы выявления уязвимостей

Как отмечалось выше, причинами возникновения уязвимостей являются ошибки проектирования, реализации и эксплуатации. Ниже приведены методы выявления таких ошибок.

Анализ алгоритма программно-аппаратного обеспечения. Данный метод используется в основном для поиска уязвимостей проектирования. Примером практической реализации может служить система PVS (Prototype Verification System), разработанная в Computers Science Laboratory института SRI (http://pvs.csl.sri.com/).

На практике часто выполняется поиск ошибок реализации (кода), осуществляемых с помощью следующих методов.

Динамический анализ безопасности приложения. Одним из наиболее простых и распространенных при поиске уязвимостей реализации является DAST (Dynamic Application Security Testing) — динамический (т. е. требующий выполнения) анализ безопасности приложения без доступа к исходному коду и среде исполнения серверной части. Другими словами, анализ приложения методом «черного ящика».

В этом контексте довольно часто используется термин «фаззинг» (fuzz testing, fuzzing). Данный метод предполагает изучение поведения ПО с помощью подачи на вход различных значений переменных. Чаще всего это граничные или маловероятные значения, которые могут создать условия, приводящие к переполнению буфера, выходу за границы массивов, записи в недопустимые области памяти и т. д.

Имеется множество инструментов, позволяющих автоматизировать процесс поиска уязвимостей методом фаззинга, например:

- MiniFuzz (www.microsoft.com);
- FileFuzz (labs.idefense.com/software/fuzzing.php).

Элементы фаззинга присутствуют в популярном инструменте эксплуатации уязвимостей metasploit (рассматривается далее). Механизмы фаззинга в том или ином виде встроены в сетевые сканеры безопасности. Например, в популярном сканере XSpider имеется очень простой «фаззер» для сетевых приложений (FTP, SMTP и т. д.) (рис. 2.1).

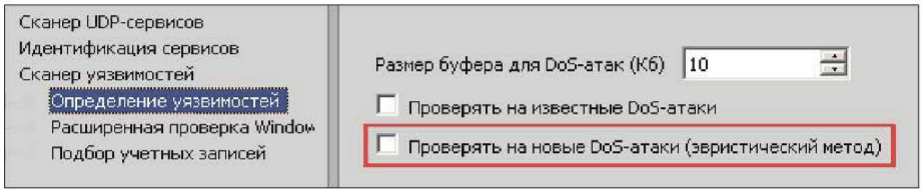


Рис. 2.1. Настройки «фаззера» в профиле сканера XSpider

Статический анализ безопасности приложения. Этот подход подразумевает синтаксический и семантический анализ исходного текста, анализ конструкций, иногда делаются попытки построения алгоритма по исходному тексту. В некоторых случаях может использоваться дизассемблирование с последующим анализом полученного кода. В любом случае данный метод не требует выполнения приложения и предполагает доступ к его исходным кодам.

Часто этот метод называют SAST (Static Application Security Testing) — статический (т. е. не требующий выполнения) анализ безопасности приложения с доступом к исходному коду (или производным, например, байт-коду) приложения. В противоположность методу DAST SAST можно назвать анализом методом «белого ящика».

Наиболее популярный объект проверки методом SAST — это web-приложения, так как получить исходные коды для них обычно не вызывает затруднений.

Несмотря на то что есть ряд инструментов, позволяющих автоматизировать поиск «слабых мест» на основе исходного текста, данный метод используется достаточно редко. Анализ результатов такого поиска предполагает знание языков программирования, а существенное число ложных срабатываний требует дополнительной «верификации» найденных уязвимостей.

В качестве примера инструмента поиска уязвимостей в исходном тексте можно привести продукт Application Inspector компании Positive Technologies (рис. 2.2).

Дополнительно следует отметить, что в приведенном продукте верификация найденных уязвимостей может осуществляться с помощью автоматически сгенерированного эксплойта, что существенно облегчает в дальнейшем взаимодействие с разработчиками уязвимого кода.

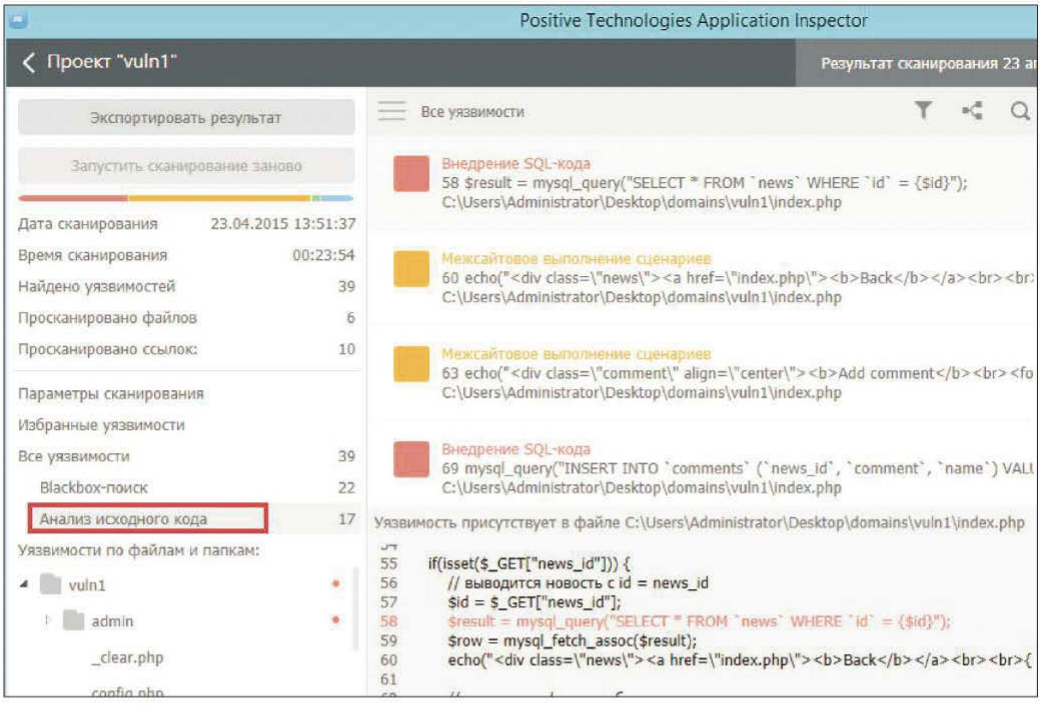


Рис. 2.2. Результаты анализа исходного кода web-приложения

Смешанный анализ безопасности приложения. В общем случае данный подход предполагает запуск приложения (как в DAST) и «наблюдение» за его действиями (вызванными подачей на вход различных данных).

Проверке могут подвергаться:

- корректность выполнения операций с памятью;
- корректность работы с указателями;
- вызовы потенциально «опасных» функций;
- «движение» данных.

В большинстве случаев подход основан на модификации анализируемого приложения: встраивании в определенные участки анализируемой программы специального кода или меток для трассировки. По сути, его можно применять даже на этапе разработки приложения, а механизмы могут встраиваться в отладчики и компиляторы. В качестве примера можно привести утилиту Heap Agent (http://www.microquill.com/heapagent/index.html), встраиваемую в компиляторы Microsoft Visual C++/Visual Studio.

Такой вариант анализа приложения IAST (Interactive Application Security Testing) называют *динамическим анализом безопасности приложения* с доступом к исходному коду и среде исполнения серверной части. Это смешанный подход, предполагающий запуск приложения (как в DAST) и доступ к его исходным текстам (как в SAST).

2.2. Выявление «известных» уязвимостей

Приведенные выше методы предназначены для выявления новых, ранее не известных уязвимостей. Однако чаще всего перед администраторами безопасности возникает задача поиска известных уязвимостей, возможно, имеющихся в используемом ПО.

Обсуждаемые далее тесты и логические выводы направлены прежде всего на выявление известных уязвимостей, внесенных в каталоги.

Из приведенных определений уязвимости и атаки следует, что для выявления уязвимости можно просто выполнить атаку с ее использованием. Если атака окажется успешной, делается вывод, что уязвимость в системе присутствует, в противном случае можно считать, что уязвимости нет. Такой способ выявления уязвимостей иногда называют *тестированием*.

Таким образом, *тест* — это алгоритм определения присутствия уязвимости в тестируемой системе путем имитации атаки, использующей данную уязвимость.

Соответственно процесс тестирования представляет собой серию атак на систему. Такой механизм поиска уязвимостей называют также *активным анализом*.

Второй способ выявления уязвимостей, не предусматривающий проведение атак, осуществляется путем предположений (логических выводов) на основе собранной информации об исследуемой системе. Таким образом, *логический вывод* — это алгоритм определения наличия уязвимости в тестируемой системе без имитации атаки, использующей данную уязвимость, по косвенным признакам (номер версии сетевой службы, присутствие на узле какого-либо файла и т. п.). Такой механизм поиска уязвимостей называют также пассивным анализом, но это не совсем верно, поскольку существует пассивный метод сбора информации о системе (Passive Fingerprinting).

2.3. Системы анализа защищенности

Возможности и архитектура. Автоматизировать процесс выявления известных уязвимостей помогут средства анализа защищенности (системы управления уязвимостями), получившие развитие из так называемых сканеров безопасности или сканеров уязвимостей (Vulnerability Scanners). Использование этих средств поможет определить уязвимости узлов корпоративной сети и устранить их до того, как ими воспользуются злоумышленники.

Обычно функционал типовой системы управления уязвимостями включает в себя:

- управление информационными активами (Asset Management);
- оценку защищенности (выявление уязвимостей);
- контроль соответствия различным наборам требований (Compliance Management);

• автоматизацию отдельных этапов процесса управления уязвимостями (формирование отчетов, выполнение действий по расписанию, управление инцидентами и т. д.).

Как правило, системы управления уязвимостями имеют распределенную архитектуру и состоят из компонентов двух типов: компонентов управления и сканирующих модулей (агентов).

Компоненты управления служат для передачи управляющих воздействий сканирующим модулям, а также обеспечивают накопление, хранение, обработку результатов работы системы (в данном случае это информация о найденных уязвимостях).

Однако главной составляющей таких систем являются сканирующие модули, которые и выполняют проверки.

Проверка — тест или заключение, направленные на выявление в системе уязвимости. По существу, *процедура сканирования* — это проведение набора проверок, состоящего в свою очередь из тестов и заключений.

В рамках изучаемого курса рассматриваются в основном возможности сканирующих модулей.

Сканирующие модули. Применяют следующие три способа проверки систем на наличие уязвимостей:

• дистанционно, путем подключения к объектам сканирования с использованием сетевых технологий;

• пассивный анализ сетевого трафика;

• использование агентов, запускаемых непосредственно на объектах сканирования.

Сканирующие модули (агенты) бывают трех типов (рис. 2.3):

• дистанционные (network-based);

• пассивные (passive);

• локальные (host-based).

В настоящее время более распространены сетевые (дистанционные) агенты сканирования (рис. 2.4).

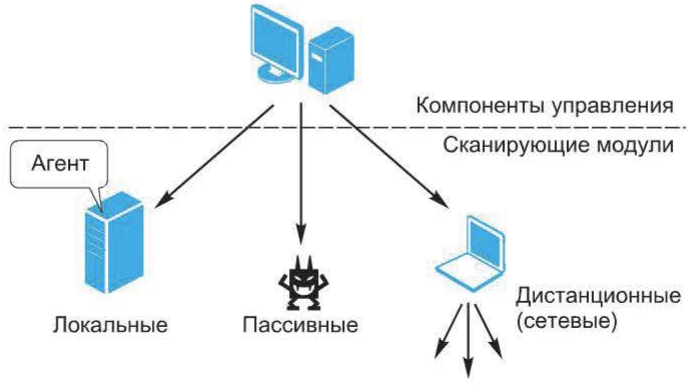


Рис. 2.3. Типы сканирующих модулей

Они имеют следующие особенности:

• выполнение проверок осуществляется дистанционно, т. е. по сети, что влияет как на скорость сканирования (сравните, например, подбор учетных записей по сети и локальный «взлом» хэшей), так и на достоверность результатов;

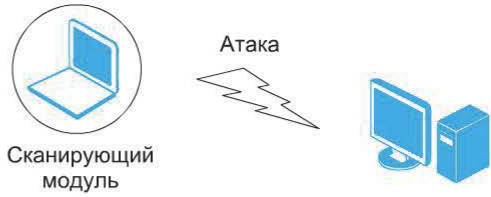


Рис. 2.4. Сетевые (дистанционные) агенты сканирования

• использование разных методов выявления уязвимостей;

• использование различных учетных данных для подключения к службам сканируемого узла.

Пассивные агенты выполняют анализ сетевого трафика для выявления признаков уязвимостей в содержимом перехваченных сетевых пакетов (рис. 2.5).

Такой подход имеет следующие преимущества:

• выявление уязвимостей на узлах, к которым затруднен или невозможен доступ в сети;

• отсутствие влияния на объект проверки;

• непрерывность работы.

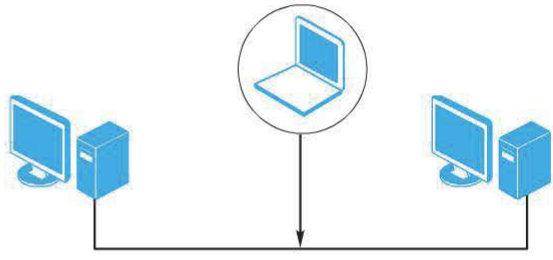


Рис. 2.5. Пассивные агенты

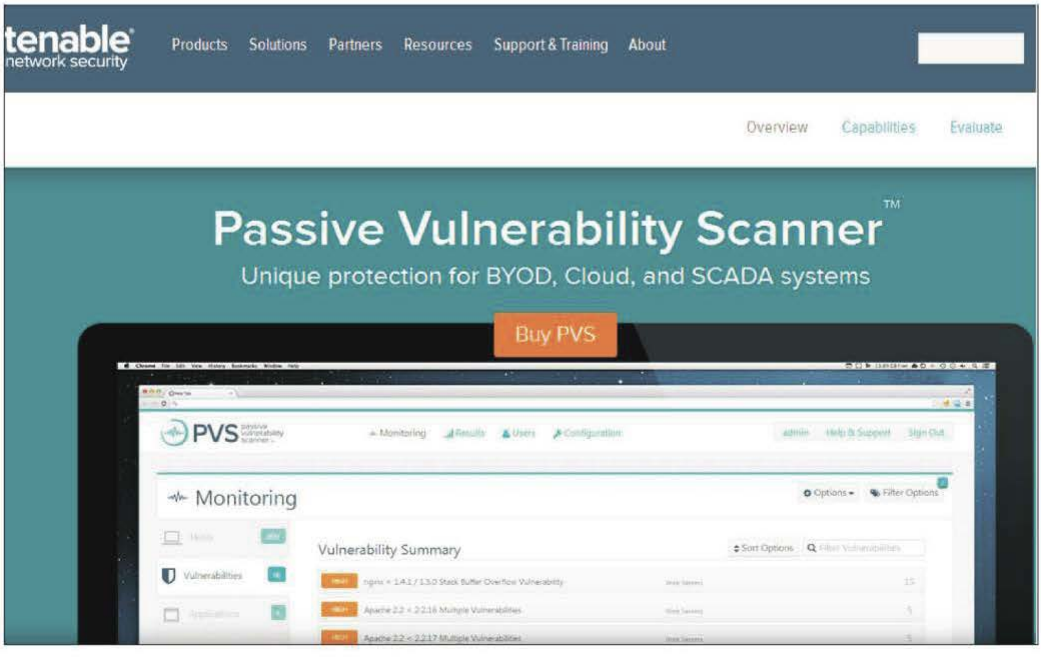


Рис. 2.6. Passive Vulnerability Scanner компании Tenable

В настоящее время такой подход не очень распространен, в качестве наиболее известного инструмента такого типа можно привести Passive Vulnerability Scanner компании Tenable (рис. 2.6) (http://www.tenable.com/products/passive-vulnerability-scanner).

Локальные агенты устанавливаются непосредственно на сканируемом узле, как правило (но не всегда), работают от имени учетной записи с максимальными привилегиями и выполняют поиск уязвимостей с помощью так называемых системных (локальных) проверок.

К преимуществам таких систем можно отнести относительную скорость работы и достоверность результатов.

Как отмечалось выше, в настоящее время наиболее распространены сканеры, выполняющие проверки дистанционно.

2.4. Примеры средств анализа защищенности

В настоящее время ряд вендоров предлагает технические средства для выполнения анализа защищенности сетей, некоторые из них приведены в табл. 2.1.

Таблица 2.1

Средства анализа защищенности (общая информация)

Вендор	Состав решения	Источник
Tenable	SecurityCenter Nessus vulnerability scanner Passive Vulnerability Scanner Nessus Enterprise Cloud	http://www.tenable.com/solutions/vulnerability-management
Positive Technologies	MaxPatrol Request Tracker MP Report Portal XSpider	http://www.ptsecurity.ru/
Qualys	QualysGuard Suite	http://www.qualys.com/enterprises/qualysguard/
Beyondtrust	Retina CS Retina CS for Mobile Retina Network Security Scanner Retina Web Security Scanner BeyondSaaS Cloud-Based Scanning	http://www.beyondtrust.com/Home/AllProducts/#home-vm
Outpost24	OUTSCAN OUTSCAN PCI Web Application Scanner (WAS) HIAB (hacker-in-a-box)	http://www.outpost24.com/products
McAfee	McAfee Vulnerability Manager	http://www.mcafee.com/ru/products/vulnerability-manager.aspx

Окончание табл. 2.1

Вендор	Состав решения	Источник
Symantec	Symantec™ Control Compliance Suite Standards Manager Symantec Control Compliance Suite Vulnerability Manager (CCS VM)	http://www.symantec.com/page.jsp?id=control-compliance-suite
Assuria	Assuria Auditor	http://www.assuria.com
АЛТЭКС-СОФТ	RedCheck	http://www.redcheck.ru/
GFI	LANguard	http://www.gfi.ru/languard
Rapid7	Nexpose Metasploit	http://www.rapid7.com/products/nexpose/

Контрольные вопросы

1. Перечислите методы обнаружения уязвимостей, возникших на этапах проектирования, программирования и эксплуатации.
2. Охарактеризуйте динамический и статический анализ безопасности приложения. В чем их принципиальная разница?
3. Перечислите три способа проверки систем на наличие уязвимостей.
4. Охарактеризуйте достоинства и недостатки сетевых, локальных и пассивных агентов сканирования.

Глава 3. СЕТЕВЫЕ СКАНЕРЫ БЕЗОПАСНОСТИ

В настоящее время наиболее используемыми являются сканеры (сканирующие модули), выполняющие проверки дистанционно. Рассмотрим их возможности более подробно.

Для выполнения проверок агентам данного типа необходимо сетевое взаимодействие с объектами сканирования. Это обусловливает следующие особенности:

- длительность сканирования;
- влияние средств защиты;
- создание нагрузки на сеть.

Для выявления уязвимостей сетевые агенты применяют разные способы. Выше уже упоминались два способа выявления уязвимостей: тесты и логические выводы. Сетевые агенты имеют следующие категории проверок:

- баннерные проверки;
- подбор учетных записей;
- системные (локальные) проверки;
- эксплойты.

Кроме того, сетевые агенты используют различные учетные данные для подключения к сканируемым узлам. С этой точки зрения проверки можно разделить на две категории:

- выполняемые с учетной записью (credential check);
- выполняемые без учетной записи (non-credential check).

Проверки, выполняемые с учетной записью, обычно используют какой-либо механизм взаимодействия с объектом сканирования, например SSH или WMI.

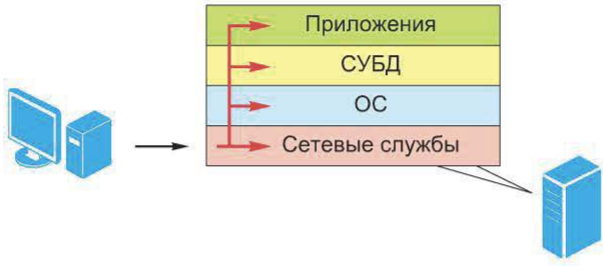


Рис. 3.1. Возможности сетевого сканера

Необходимо отметить, что сетевой сканер имеет возможность осуществлять поиск уязвимостей почти на всех уровнях инфраструктуры корпоративной информационной системы (за исключением уровня пользователя) (рис. 3.1).

3.1. Размещение сетевых агентов сканирования в сети

Расположение сетевых агентов относительно объектов сканирования может быть выбрано не только на основе влияния устройств фильтрации трафика, но и в зависимости от поставленных задач. В связи с этим для рассмотрения возможных вариантов размещения сетевых сканирующих модулей можно выделить три области (рис. 3.2):

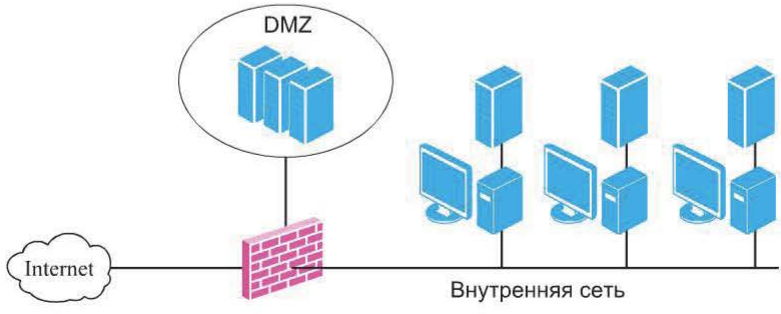


Рис. 3.2. Возможные области размещения сетевых сканирующих модулей

- внутреннюю сеть;
- демилитаризованную зону;
- внешнюю сеть.

Например, довольно часто возникает задача оценки защищенности сетевого периметра. В этом случае расположение сканирующего модуля снаружи позволяет смоделировать действия внешнего нарушителя (рис. 3.3).

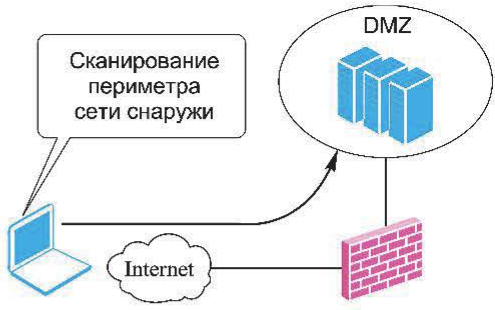


Рис. 3.3. Оценка защищенности сетевого периметра

Часто такое сканирование начинается со сбора информации об объекте сканирования — получения регистрационных данных, изучения публично доступных сведений. Затем выполняется инвентаризационное сканирование узлов, доступных снаружи. Для идентификации сервисов, приложений, делаются предположения о возможных уязвимостях.

Аналогично при сканировании из внутренней сети может проводиться оценка защиты от потенциального внутреннего нарушителя (рис. 3.4).

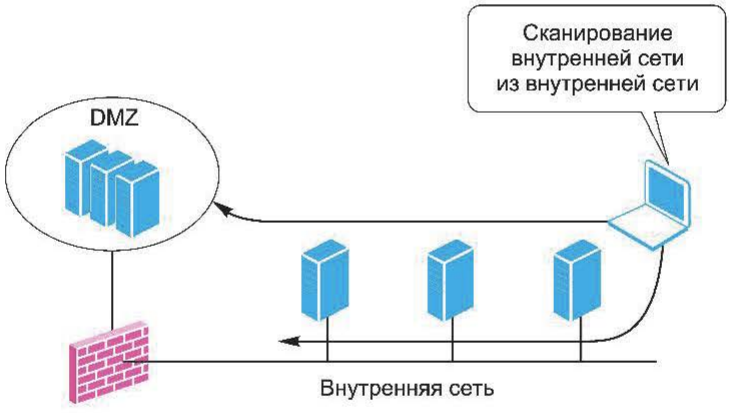


Рис. 3.4. Оценка защиты от потенциального внутреннего нарушителя

Чаще всего такое расположение сканирующих модулей выбирается для того, чтобы обеспечить максимально полный доступ к объектам сканирования.

Еще один возможный вариант размещения сканирующего модуля — сегмент DMZ (рис. 3.5). При таком расположении может, например, проводиться оценка защищенности внутренней сети с точки зрения нарушителя, получившего доступ к узлу в DMZ.

Кроме того, такой вариант размещения сетевого агента обеспечивает удобный доступ к узлам DMZ: без ограничений, накладываемых устройствами фильтрации трафика.

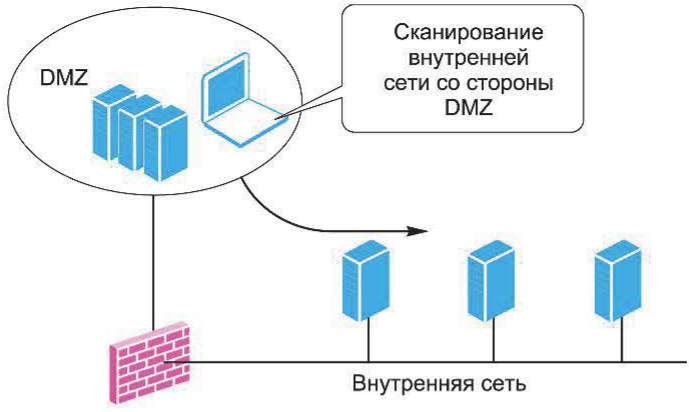


Рис. 3.5. Оценка защищенности внутренней сети нарушителем из DMZ

Таким образом, выбор варианта размещения сетевых агентов проводится, исходя из необходимости моделирования действий потенциального нарушителя или для снятия ограничений, связанных с фильтрацией трафика.

3.2. Сетевые агенты и сбор информации

Сетевые агенты подключаются к сканируемым узлам удаленно, при этом с помощью одного агента проверяется большое число узлов сети. Поэтому важной функциональной составляющей сетевого агента является модуль сбора информации, выполняющий так называемые инвентаризационные проверки.

Инвентаризационная информация необходима для работы некоторых категорий проверок, например баннерных. Кроме того, наличие подробной информации об объектах сканирования может помочь в принятии решения об устранении найденных уязвимостей. В целом сетевой сканер выполняет следующие инвентаризационные проверки:

- идентификацию устройств сети;
- определение топологии, взаимного расположения узлов;
- идентификацию открытых портов;
- идентификацию сервисов;
- идентификацию приложений;
- идентификацию операционных систем.

Контрольные вопросы

1. Какие категории проверок имеют сетевые агенты?
2. Перечислите возможные варианты размещения сетевых сканирующих модулей.
3. С чем связан выбор варианта размещения сетевых агентов?
4. Какие инвентаризационные проверки выполняет сетевой сканер?

Глава 4. СПОСОБЫ СБОРА ИНФОРМАЦИИ О СЕТИ. ПРЕДВАРИТЕЛЬНОЕ ИЗУЧЕНИЕ ЦЕЛИ

4.1. Способы сбора информации о сети

В процессе дистанционного контроля состояния защищенности возникает необходимость сбора информации о сетевых объектах. В зависимости от варианта контроля защищенности, выбранной методологии и других факторов могут быть использованы различные приемы сбора информации. Например, при выполнении аудита внутренней сети на соответствие требованиям политики безопасности обычно не требуется определять топологию сети или собирать информацию регистрационного характера. Все это уже, как правило, известно. Кроме того, оценке защищенности периметра обычно предшествует этап предварительного изучения цели, предполагающий сбор информации «с нуля».

Способы сбора информации о системе можно разделить на две группы (рис. 4.1):

- *активные* (Activefingerprinting), предполагающие использование ключевых воздействий на систему и анализ откликов;
- *пассивные* (Passivefingerprinting), предполагающие использование информации, «добровольно» рассылаемой исследуемой системой.

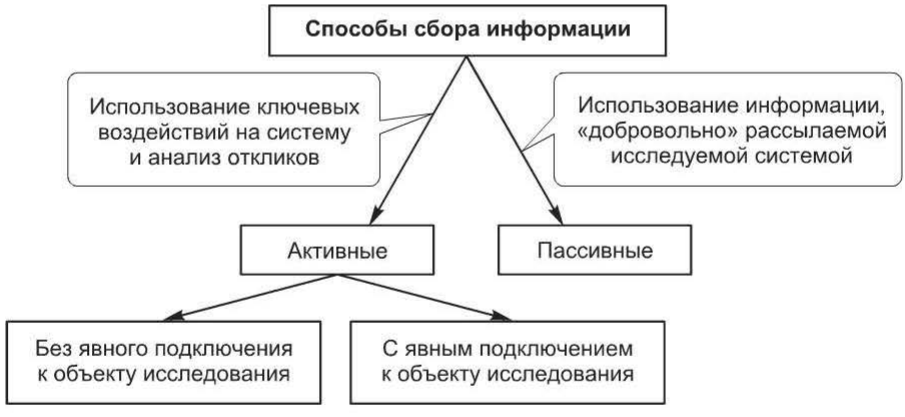


Рис. 4.1. Способы сбора информации о сети

Активные методы сбора информации в свою очередь можно разделить на требующие явного подключения к сетевым службам объекта сканирования (например, идентификация сервисов с помощью посылки различных запросов) и не требующие такого подключения. Последняя группа методов называется также *предварительное изучение цели*.

В ходе сбора информации о системе могут быть получены различные сведения, например:

- информация регистрационного и организационного характера, обычно доступная через Интернет;

- принадлежащие организации домены;
- доступные сетевые объекты;
- открытые порты, поддерживаемые протоколы;
- службы, соответствующие открытым портам;
- приложения, реализующие серверные части служб;
- ОС узлов;
- используемые средства защиты.

Следует отметить, что в этот список включается любая другая дополнительная информация, которая может оказаться полезной в ходе контроля защищенности.

Сведения могут быть получены любым из приведенных выше способов.

4.2. Предварительное изучение цели

Некоторые приемы. Способы получения информации, не требующие явного подключения к объекту исследования, часто называют предварительным изучением цели. К ним можно отнести следующие приемы:

- анализ публично доступных ресурсов;
- использование поисковых систем;
- социальная инженерия;
- сбор информации регистрационного характера.

Анализ публично доступных ресурсов. Отправной точкой для сбора информации часто выбирают web-сайт компании. Информация, собранная на сайте, может послужить основой для дальнейших изысканий. Вследствие большого разнообразия представленной на сайте информации систематизировать ее крайне затруднительно, однако можно выделить следующие объекты исследования:

- информация о компании;
- контакты;
- вакансии;
- решения;
- описание системы защиты;
- исходный код страниц.

На сайте практически любой компании существует раздел «О компании», в котором представлена информация о направлениях деятельности компании, истории ее создания и т. д. Эти, казалось бы, малозначительные, сведения могут помочь нарушителю в планировании атак с использованием социальной инженерии. Например, если в данном разделе указаны часы работы, злоумышленник может позвонить за 5...10 мин до окончания рабочего дня и задать кучу вопросов. Сотрудник в этой ситуации, торопясь домой, может отвечать на них, не особо задумываясь. Или, наоборот, нарушитель может выполнять свои несанкционированные действия в нерабочие часы без боязни быть обнаруженным.

Раздел «Вакансии» может подсказать нарушителю направления развития компании и используемые ею технологии. Например, объявление о поиске специалиста по защите периметра со знанием межсетевого экрана Check Point говорит о том, какие информационные технологии использует или планирует использовать организация (рис. 4.2).

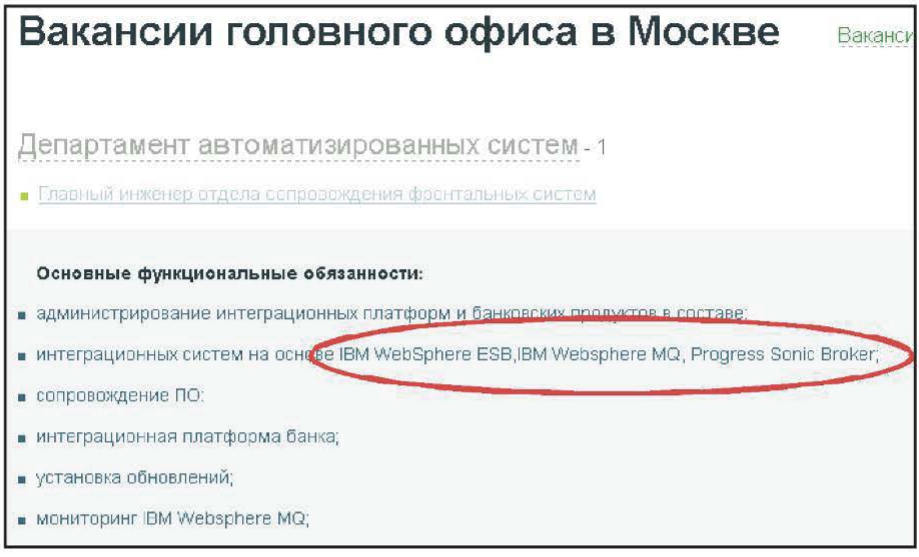


Рис. 4.2. Окно раздела «Вакансии» корпоративного web-сайта

Эти и другие сведения, собранные профессионалом, могут быть использованы для планирования дальнейших действий.

В некоторых случаях при анализе web-сайта может помочь использование инструментов, позволяющих записать его содержимое, а затем искать там нужную информацию. Пример такого инструмента — программа HTTrack (www.httrack.com).

Программа HTTrack — это так называемый оффлайн-браузер (рис. 4.3). Она позволяет скачать содержимое web-сайта из Интернета на локальный диск, сохранив структуру каталогов, ссылок и т. д. Затем можно просматривать содержимое сайта так же, как и в режиме on-line.

Программа работает в средах Windows и UNIX, имеет графический интерфейс и интерфейс командной строки.

Например, команда

```
httrack "http://www.ru/" -O "/tmp/www.ru"
```

предписывает программе HTTrack скачать содержимое сайта www.ru и сохранить его в папке "/tmp/www.ru".

Хотя программа HHTrack позволяет исследовать содержимое web-сайта без явного подключения к нему, сам процесс создания его копии порождает подозрительную активность, которая легко может быть обнаружена. Причем

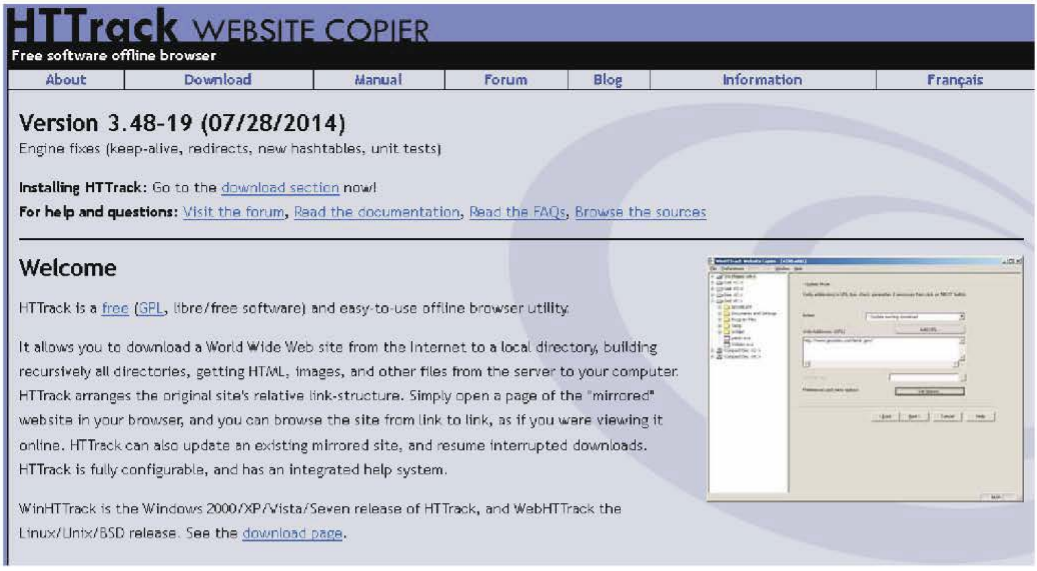


Рис. 4.3. Программа HTTrack

весь процесс может занимать продолжительное время при большом объеме содержимого web-сайта.

Использование поисковых систем. Поисковые системы, например Google (http://www.google.com/), представляют собой множество возможностей для поиска необходимой информации в ходе предварительного изучения цели. При этом поиск может быть значительно оптимизирован, если использовать различные параметры поиска, предоставляемые самой поисковой системой.

Например, Google позволяет искать файлы различных форматов (.xls, .doc, .pdf). Для этого в строке поиска следует ввести следующий текст: filetype:xls (для файлов excel).

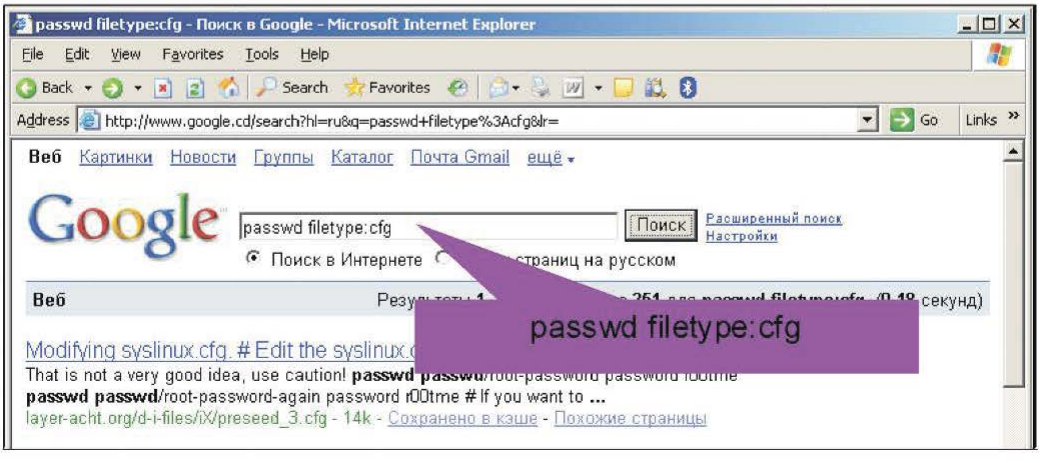


Рис. 4.4. Поиск файлов различных форматов

При этом искать можно любые файлы, например .cfg (рис. 4.4) или .pwd.

Другой полезный параметр поиска — inurl. Он позволяет искать заданный текст в URL (рис. 4.5).

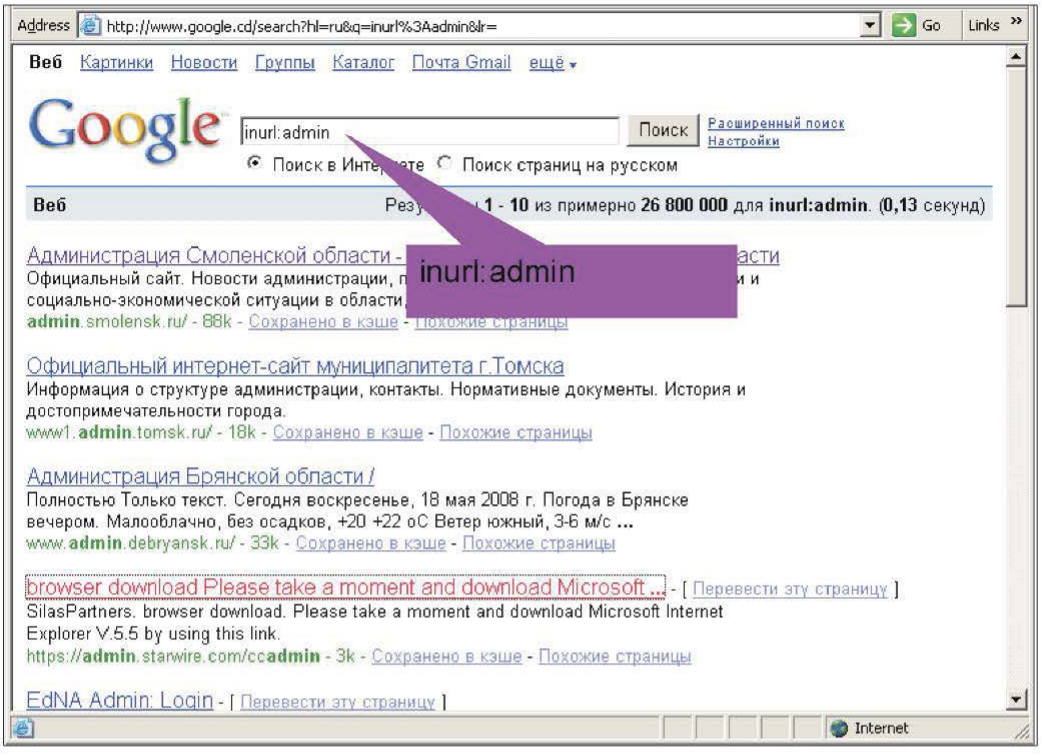


Рис. 4.5. Поиск заданного текста в URL

Например, если ввести в строке поиска inurl:admin, результатом будут ресурсы, которые имеют слово admin в URL.

Поиск уязвимых систем с использованием Google стал настолько популярен, что породил специальные инструменты для поиска, например Goolag Scanner.

Кроме того, существуют базы, содержащие обновляемые результаты такого поиска (рис. 4.6).

Получение информации регистрационного характера. В ходе предварительного изучения цели достаточно популярным приемом является использование сайтов, предоставляющих *информацию регистрационного характера*, в частности:

- www.register.com
- www.geektools.com
- www.ripn.net
- www.leader.ru/secure/who.html

Рис. 4.6. Пример базы, содержащей обновляемые результаты поиска

Кроме того, в процессе поиска могут помочь инструменты Web based, например:

- http://www.kloth.net/services/nslookup.php
- http://www.zoneedit.com/lookup.html
- http://swhois.net/
- http://centralops.net/co/
- http://network-tools.com/nslook/

Этот способ позволяет убедиться, что искомый домен действительно принадлежит указанной организации. Обычно по такому запросу может быть предоставлен список узлов, таких как www-серверы, серверы имен и т. д. Контактная информация может содержать адреса электронной почты (они могут принадлежать другому домену, который тоже включается в список для дальнейшего исследования).

Таким образом, этот метод предполагает некоторое предварительное знакомство с организацией, ее партнерами и т. д. При этом может показаться, что собранная информация окажется лишней, тем не менее этот этап необходим.

Промежуточный результат данного этапа — максимально полный список доменов, имеющих отношение к исследуемой организации.

Использование DNS. Сбор информации регистрационного характера позволяет как минимум определить адрес сервера имен, обеспечивающего работу исследуемого домена (рис. 4.7). Далее обычно следуют уже прямые обращения к этому серверу: запросы на получение информации о зоне и т. д.

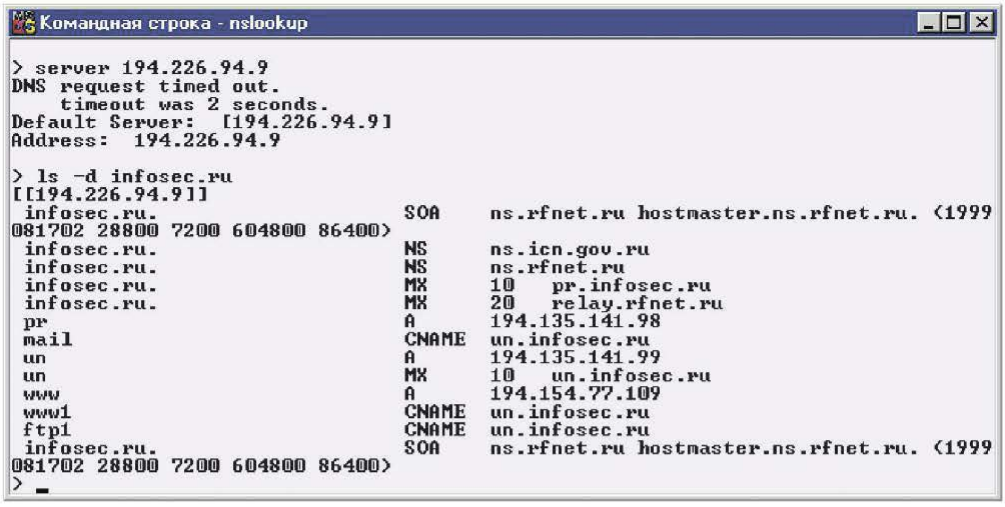


Рис. 4.7. Сбор информации регистрационного характера с помощью утилиты nslookup

Цель данного этапа — получение списка узлов, принадлежащих домену, и получение соответствий «IP-адрес/имя». На основе данного списка может быть получен диапазон адресов, принадлежащих организации. В дополнение к запросам к базе здесь также можно использовать несколько техник:

- некоторые адреса/имена обязаны существовать просто для того, чтобы домен работал. Это, например, сервер(ы) имен (NS) и почтовые серверы (MX). Адрес сервера имен можно получить из информации регистрационного характера, а адрес почтового сервера — из базы сервера имен;
- некоторые адреса/имена с большой степенью вероятности будут присутствовать в домене, например, www, mail, gate, firewall. Необходимо проверить существование узлов с такими именами;
- как правило, узлы принадлежат одной подсети, поэтому, получив один адрес, следует проверить и остальные адреса подсети;
- использование базы данных сервера имен (передача зоны).

Следует учитывать, что прямой DNS-запрос (Имя → IP-адрес) и обратный (IP-адрес → Имя) не всегда дают сопоставимые результаты.

Основными результатами предварительного изучения цели следует считать:

- список доменов, имеющих отношение к исследуемой организации;
- диапазоны адресов для дальнейшего исследования;
- другую информацию, собранную при исследовании цели.

Эта информация может быть использована в качестве исходной для сбора сведений другими методами.

Контрольные вопросы

1. Охарактеризуйте способы сбора информации о системе.
2. Какие сведения могут быть получены входе сбора информации о системе?
3. Перечислите и охарактеризуйте приемы предварительного изучения цели.

Глава 5. ИДЕНТИФИКАЦИЯ СЕТЕВЫХ ОБЪЕКТОВ

Глава 4 была посвящена методам сбора информации без явного подключения к объекту исследования.

Рассмотрим приемы, предполагающие явное подключение при идентификации:

- сетевых объектов;
- статуса порта;
- сервисов;
- приложений;
- операционных систем.

Задача идентификации сетевых устройств заключается в том, чтобы удаленная система отреагировала на какой-либо запрос (рис. 5.1).

Рис. 5.1. Реакция системы на запрос

Под реакцией системы понимается генерация какого-либо ответа или сообщения об ошибке. Это и будет доказательством того, что система присутствует в сети. Причем задача состоит именно в доказательстве присутствия системы, а не в определении каких-либо ее характеристик (работающие службы, ОС и т. п.). Для решения этой задачи можно использовать различные протоколы: IP, ICMP, UDP, TCP.

5.1. Использование протокола ICMP

Общие сведения о протоколе ICMP. Протокол ICMP (RFC792) служит для выявления проблем, связанных с сетевым уровнем (в стеке TCP/IP этот уровень представлен протоколом IP). Сообщения протокола ICMP передаются в виде IP-датаграмм, т. е. к ним добавляется заголовок IP. Формат ICMP-пакета представлен на рис. 5.2.

Type (тип)	Code (код)	Checksum (Контрольная сумма)
Данные...		

Рис. 5.2. Формат ICMP-пакета

Существует несколько типов сообщений ICMP. Каждый тип сообщения имеет свой формат, при этом все они начинаются с приведенных в табл. 5.1 трех полей:

- 8-битного целого числа, обозначающего тип сообщения (TYPE);
- 8-битного поля кода (CODE), который конкретизирует назначение сообщения;
- 16-битного поля контрольной суммы (CHECKSUM).

Все типы сообщений ICMP можно условно разделить на две группы:

- сообщения об ошибках (например, Destination Unreachable);
- запросы и ответы (например, Echo Request и Echo Reply).

Сообщения об ошибках содержат заголовок и первые 64 бит данных пакета IP, при передаче которого возникла ошибка. Это делается для того, чтобы узел-отправитель более точно проанализировал причину ошибки, так как все протоколы прикладного уровня стека TCP/IP содержат наиболее важную информацию для анализа именно в первых 64 бит своих сообщений. Для большинства сообщений об ошибках задействовано поле кода.

В табл. 5.1 представлены возможные значения полей TYPE и CODE.

Таблица 5.1

Возможные значения полей TYPE и CODE

Type	Name	Code
0	EchoReply	0
3	Destination Unreachable	0. Net Unreachable 1. Host Unreachable 2. Protocol Unreachable 3. Port Unreachable 4. Fragmentation Needed and Don't Fragment was Set 5. Source Route Failed 6. Destination Network Unknown 7. Destination Host Unknown 8. Source Host Isolated4 9. Communication with Destination Network is Administratively Prohibited5 10. Communication with Destination Host is Administratively Prohibited6 11. Destination Network Unreachable for Type of Service 12. Destination Host Unreachable for Type of Service 13. Communication Administratively Prohibited 14. Host Precedence Violation 15. Precedence cutoff in effect
4	Source Quench	0
5	Redirect	0. Redirect Datagram for the Network (or subnet) 1. Redirect Datagram for the Host 2. Redirect Datagram for the Type of Service and Network 3. Redirect Datagram for the Type of Service and Host

Окончание табл. 5.1

Type	Name	Code
6	Alternate Host Address	0
8	Echo Request	0
9	Router Advertisement	0
10	Router Selection	0
11	Time Exceeded	0. Time to Live Exceeded in Transit 1. Fragment Reassembly Time Exceeded
12	Parameter Problem	0. Pointer indicates the error 1. Missing a Required Option 2. Bad Length
13	Timestamp	0
14	Timestamp Reply	0
15	Information Request	0
16	Information Reply	0
17	Address Mask Request	0
18	Address Mask Reply	0
19	Reserved (for Security)	0
30	Traceroute	0
31	Datagram Conversion Error	0
32	Mobile Host Redirect	0
33	IPv6 Where-Are-You	0
34	IPv6 I-Am-Here	0
35	Mobile Registration Request	0
36	Mobile Registration Reply	0
39	SKIP	0
40	Photuris	0. Reserved 1. Unknown security parameters index 2. Valid Security Parameters, but Authentication Failed 3. Valid Security Parameters, but Decryption Failed

Идентификация сетевых устройств с помощью протокола ICMP может быть выполнена двумя способами:

- посылка запроса, получение ответа;
- вызов ситуации ошибки, получение сообщения об ошибке.

Использование сообщений ICMP Echo (Type 8) и Echo Reply (Type 0). Самый простой и распространенный способ определения доступности узла — посылка сообщения ICMP Echo (Type 8). Если система доступна и отсутствует фильтрация трафика данного типа, то в ответ придет сообщение ICMP Echo Reply (Type 0) (рис. 5.3).

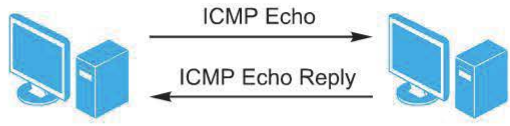


Рис. 5.3. Определение доступности узла

Type (8 или 0)	Code = 0	Checksum
Identifier		Sequence Number
Данные...		

Рис. 5.4. Формат сообщений (ICMP Echo и ICMP Echo Reply)

Узел, отправляющий ICMP-запрос, устанавливает значения полей Identifier (для того, чтобы отличить ответы, пришедшие от различных узлов) и Sequence Number (чтобы отличить несколько ответов, пришедших от одного и того же узла). В поле Code записывается ноль, поле данных произвольно (например, алфавит). Отвечающая сторона должна заменить значение поля Type на 0 и отправить датаграмму обратно (рис. 5.4). Для выполнения данной операции обычно используется утилита ping, входящая в состав большинства ОС.

Опрос множества устройств. Опрос сразу нескольких узлов (диапазона) с использованием ICMP-запросов (Echo) называется ICMP Sweep или Ping Sweep. Использование утилиты ping в этом случае нерационально, поскольку узлы опрашиваются последовательно, и в случае большой сети требуется значительное время. Для исследования большой сети потребуются утилиты, посылающие ICMP-запросы параллельно, например:

- fping для UNIX-систем;
- Pinger для Windows-систем (http://visualsoft.newmail.ru);
- FPinger для Windows-систем (http://fpinger.mastak.com).

Следует отметить утилиту nmap.

Утилита fping. Утилита fping (http://www.fping.com) позволяет проводить исследование сети с помощью протокола ICMP, но в отличие от утилиты ping возможен параллельный опрос сразу нескольких узлов, список которых может быть задан непосредственно или получен из файла.

Синтаксис:

```
fping [ опции ] [ узлы... ]
```

Список наиболее используемых опций:
-c — число отправляемых пакетов к каждому из узлов;
-bn — количество байт в отправляемом пакете;
-g — указание списка сканируемых узлов;
-f — указание файла со списком сканируемых узлов.
Ниже приведены примеры использования утилиты fping.
Сканирование сети класса C:

```
fping –g 200.2.2.0/24
```

или

```
fping –g 200.2.2.1 200.2.2.254
```

Сканирование с посылкой одного пакета:

```
fping –g 200.2.2.1 200.2.2.254 –c 1
```

Утилита nmap. Данная утилита (www.insecure.org/nmap, более подробно рассматривается далее) также может быть использована для опроса сетевых устройств, например:

```
nmap -sP -PI 200.2.2.1-254
```

Здесь ключ sP — это указание проводить посылку запросов ICMP Echo, ключ PI — отключение ACK-сканирования (включенного по умолчанию).

Примечание. Ping Sweep (вследствие того, что параллельно посылается множество ICMP-запросов) с бóльшей вероятностью фиксируется системами обнаружения атак, чем одиночные пакеты ICMP Echo.

Использование Broadcast ICMP. Один из способов определения доступности множества узлов — посылка запроса ICMP Echo по широковещательному адресу или адресу сети, например:

```
ping 200.1.1.255
```

или

```
ping 200.1.1.0
```

Такой запрос будет получен всеми узлами сети и теоретически ответ от каждого из узлов должен прийти узлу, пославшему запрос (рис. 5.5).

Такой способ обнаружения устройств работает только для ОС семейства UNIX. ОС Windows (исключая Windows NT ниже SP4) не отвечают на запросы, где в качестве адреса получателя указан широковещательный адрес или адрес сети.

Следует отметить, что системы обнаружения атак обычно фиксируют такие запросы ICMP Echo, как атаку Smurf. Действительно, подобный запрос порождает большое число ответов, направленных на один узел, что может создать ситуацию «отказ в обслуживании».

Рис. 5.5. Широковещательный запрос ICMP Echo

Использование других типов ICMP-сообщений. Кроме рассмотренных выше сообщений ICMP (Echo и Echo Reply) для идентификации сетевых устройств можно использовать и другие ICMP-сообщения (табл. 5.2).

Таблица 5.2

ICMP-сообщения

Запрос	Ответ	Примечание
Time Stamp Request (Type 13)	Time Stamp Reply (Type 14)	Запрос отметки времени и ответ
Information Request (Type 15)	Information Reply (Type 16)	Определение адреса сети для бездисковых станций
Address Mask Request (Type 17)	Address Mask Reply (Type 18)	Получение маски подсети
Router Solicitation (Type 10)	Router Advertisement (Type 9)	—

Запрос отметки времени. Этот тип ICMP-запроса используется для получения текущего значения времени на исследуемом узле. Его формат представлен на рис. 5.6.

В ответ на такой запрос должно прийти ICMP-сообщение Time Stamp Reply.

Type (13 или 14)	Code=0	Checksum
Identifier		Sequence number
Originate time stamp		
Receive time stamp		
Transmit time stamp		

Рис. 5.6. Тип ICMP-запроса для получения текущего значения времени

Однако не все ОС отвечают на подобные запросы:

- Windows NT — не отвечает;
- Windows 2000 — отвечает.

Для генерации запроса может быть использована утилита sing (www.sourceforge.net/projects/sing), позволяющая конструировать любые ICMP-сообщения.

Например, для посылки одного сообщения Time Stamp Request используется следующий синтаксис:

```
sing -c 1 -tstamp <узел>
```

Сообщение Information Request. Данный тип сообщения служит для определения адреса сети для бездисковых станций. В настоящее время используются протоколы BOOTP и DHCP, и в большинстве случаев обработка таких сообщений не поддерживается. Однако некоторые ОС все же отвечают на подобные запросы, это, в частности, HP-UX, AIX и CiscoIOS.

Формат пакета представлен на рис. 5.7.

Type (15 или 16)	Code=0	Checksum
Identifier		Sequence number

Рис. 5.7. Формат пакета для определения адреса сети для бездисковых станций

И в этом случае для генерации запроса может быть использована утилита sing:

```
sing -info <узел>
```

Запрос маски. Цель запроса — получение маски подсети, которую необходимо использовать в данной локальной сети. Формат пакета представлен на рис. 5.8.

Type (17 или 18)	Code=0	Checksum
Identifier		Sequence number
Subnetad dress mask		

Рис. 5.8. Формат пакета для получения маски подсети

Различные ОС по-разному реагируют на данный запрос: отвечают Windows 9x/Me, Solaris; игнорируют Windows 2000/NT SP6, Cisco IOS, Linux, FreeBSD.

Пример запроса с использованием утилиты sing:

```
sing -mask <узел>
```

ICMP-пакеты, сообщающие об ошибках. Сообщения об ошибках, передаваемые с помощью протокола ICMP, часто бывают более информативны,

чем просто ответ (Reply). Например, сообщение ICMP Destination Unreachable (port unreachable), полученное в ответ на UDP-пакет, указывает на то, что требуемый порт на узле закрыт, но узел доступен (поскольку ответ пришел).

Использование ICMP-сообщений об ошибках для обнаружения устройств сводится к вызову ситуации ошибки на тестируемом узле. Поскольку вызов ситуации ошибки осуществляется, как правило, с помощью других протоколов (не ICMP), рассмотрим эти методы, в частности, характерный пример —способы обнаружения узлов с помощью IP-протокола.

5.2. Идентификация узлов с помощью протокола ARP

В локальной сети эффективным способом обнаружения узлов является посылка ARP-запросов. При этом узел ответит в любом случае, даже если блокируется весь трафик.

ARP-запросы и ответы используют один и тот же формат пакета. На рис. 5.9 приведен формат пакета протокола ARP для передачи по сети Интернет.

Рис. 5.9. Формат пакета протокола ARP для передачи по сети Интернет

В поле типа сети для сетей Интернет указывается значение 1. Поле типа протокола позволяет использовать пакеты ARP не только для IP-протокола, но и для других сетевых протоколов. Для IP-протокола значение этого поля равно 0x800.

Длина MAC-адреса для протокола Интернет равна 6 байт, а длина IP-адреса — 4 байт. В поле операции для ARP-запросов указывается значение 1 для запроса и 2 — для ответа.

Узел, отправляющий ARP-запрос, заполняет в пакете все поля, кроме поля искомого MAC-адреса. Значение этого поля заполняется узлом, опознавшим свой IP-адрес.

В качестве примера программы, использующей такую технику выявления доступных узлов, можно привести утилиту ettercap.

Контрольные вопросы

1. В чем заключается задача идентификации сетевых устройств?
2. Как с помощью посылки ARP-запросов можно обнаружить узлы сети?
3. Как используются ICMP-сообщения об ошибках для обнаружения устройств?
4. Какими способами может быть выполнена идентификация сетевых устройств с помощью протокола ICMP?

Глава 6. ОПРЕДЕЛЕНИЕ ТОПОЛОГИИ СЕТИ

С задачей идентификации сетевых объектов связана задача построения топологии (карты) сканируемой сети. Часто (например, при сканировании внутренней сети) топология известна заранее, но в некоторых случаях топологию необходимо специально исследовать.

6.1. Отслеживание маршрутов

Общие сведения. При определении топологии сети (на этапе начального сбора сведений) распространенным приемом является отслеживание маршрутов. Для этого используется утилита traceroute, входящая в состав систем UNIX (в Windows она называется tracert). Цель такого исследования — получить точный маршрут движения IP-пакета от одного узла сети до другого.

Рассмотрим работу утилиты tracert из состава Windows на следующем примере.

Пусть команда tracert выполняется в отношении узла 200.2.2.222:

```
>tracert 200.2.2.222 -d:
```

1. На первом шаге посылается ICMP-запрос (Echo) со следующими параметрами:

адрес отправителя — 200.0.0.161;
адрес получателя — 200.2.2.222;
TTL=1.

2. Маршрутизатор уменьшает значение этого поля на 1 (оно становится равным 0), что вызывает генерацию сообщения ICMP_TIME_EXCEEDED.

3. Затем поле TTL в отправляемых ICMP-пакетах последовательно наращивается на 1, т. е. на данном шаге оно будет равно 2 и пакет пройдет через маршрутизатор TTL=2

4. Поскольку в данном случае пакет достиг требуемого узла, от него будет получен обычный ICMP-ответ (Echo-Reply).

Схема работы утилиты traceroute с использованием протокола UDP приведена на рис. 6.1.

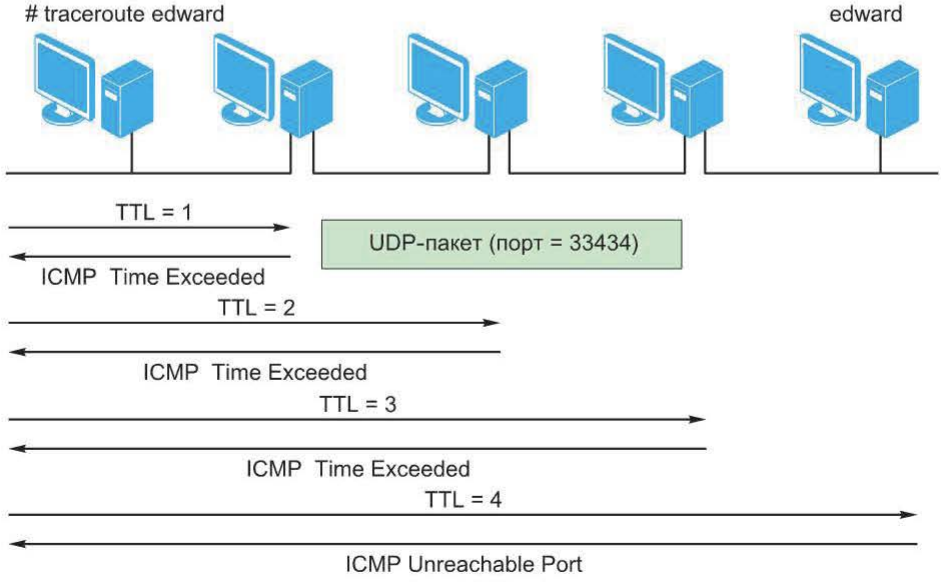


Рис. 6.1. Схема работы утилиты traceroute с использованием протокола UDP

В этом случае порт получателя растет с каждым последующим запросом (начальное значение по умолчанию равно 33 434).

Поскольку попытка отслеживания маршрута к узлу может являться предварительным действием перед атакой, большинство систем обнаружения атак могут обнаруживать такое событие. Обычно они реагируют на появление в сети пакета ICMP_TIME_EXCEEDED. По содержимому пакета можно определить цель атаки.

6.2. Отслеживание маршрутов и фильтрация

Используя рассмотренные выше утилиты, можно проследить путь IP-датаграммы до любого узла IP-сети. Задача отслеживания маршрутов усложняется, если на пути IP-датаграммы встречаются устройства, осуществляющие фильтрацию трафика. Например, пакетный фильтр (рис. 6.2) может иметь такую конфигурацию, что он блокирует весь трафик UDP, кроме тех случаев, когда порт отправителя или получателя равен 53 (служба DNS).

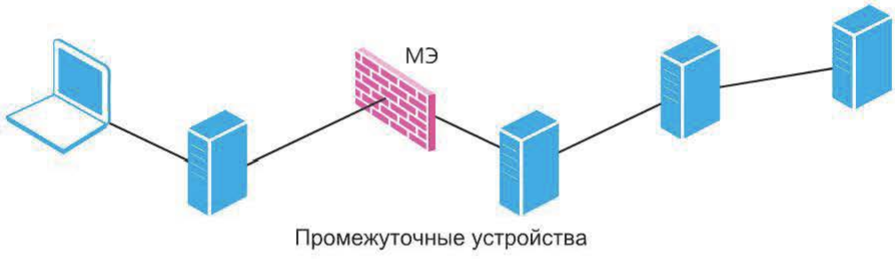


Рис. 6.2. Пакетный фильтр на МЭ

В этом случае правильным будет использование протокола ICMP:

`traceroute 200.0.0.110` — неправильно;
`traceroute -I 200.0.0.110` — правильно.

Как быть, если в дополнение к этому фильтруются и пакеты ICMP Echo? В этом случае важно значение порта получателя в момент прохождения UDP-датаграммы через пакетный фильтр. Значение порта получателя в любой момент зависит от следующих параметров:

- начальное значение порта получателя в момент начала работы утилиты traceroute start_destination_port
- число пакетов, посылаемых с одним и тем же значением TTL (по умолчанию 3) num_of_probes
- число маршрутизаторов на пути до узла осуществляющего фильтрацию num_of_hopes

Таким образом, несложно вычислить начальное значение порта получателя, требуемое для обхода фильтрации:

```
start_destination_port = (53 – (num_of_hopes*num_of_probes)) - 1
```

где 53 — порт разрешенного протокола DNS.

Однако следующий пакет будет иметь уже другое значение порта получателя и будет блокирован. Поэтому желательно, чтобы значение порта получателя было постоянным. Это может быть достигнуто установкой исправления к утилите traceroute (к версии 1.4a5). Ее синтаксис после применения исправления имеет вид

```
traceroute -S –p53 200.0.0.222
```

6.3. Утилита traceproto

Делая вывод по применению утилиты traceroute, можно сказать, что она работает на сетевом уровне (IP), однако в условиях фильтрации трафика ее применение ограничено транспортным уровнем (тем, что разрешено из вышележащих протоколов — UDP, ICMP, TCP). Поэтому с помощью утилиты traceroute можно определить последний шлюз, от которого пришел ответ. С точки зрения отслеживания маршрутов важным является значение поля TTL из заголовка IP, но не более того. Протоколы UDP и ICMP лишь транспортируют данные, поэтому без ущерба могут быть заменены любым другим протоколом транспортного уровня, в частности TCP.

Эта идея реализована в утилите traceproto (http://traceproto.sourceforge.net/index.php).

Утилита traceproto аналогична утилите traceroute, но позволяет использовать также и протокол TCP для отслеживания маршрутов.

Синтаксис утилиты traceproto:

```
traceproto [-cCfhv] [-p protocol] [-d dst_port] [-D
max_dst_port] [-s src_port] [-S max_src_port] [-m min_ttl]
[-M max_ttl] [-w response_timout ] [-W send_delay] [-a
account_level] [-P payload_size] [-k skips] [-H packets_
per_hop] [-i incr_pattern] [-o output_style]
```

Видно, что протокол выбирается с помощью опции –p (по умолчанию TCP), порт получателя задается опцией –d (по умолчанию 80), разумеется, можно задать и порт источника. О назначении остальных опций можно узнать из руководства к утилите.

Контрольные вопросы

1. Опишите прием «отслеживание маршрутов» при определении топологии сети.
2. В чем заключается специфика задачи отслеживания маршрутов при наличии устройств, осуществляющих фильтрацию трафика?
3. Каково назначение утилиты traceproto?

Глава 7. ИДЕНТИФИКАЦИЯ СТАТУСА ПОРТА

7.1. Сканирование портов

Взаимодействие узлов по протоколам TCP и UDP предполагает использование портов для идентификации приложений. Следовательно, определив номера открытых портов на удаленном узле, можно в дальнейшем узнать о работающих на нем приложениях, сделать вывод о роли этого узла в корпоративной сети, узнать версию ОС. Определить состояние портов на удаленном узле можно, например, последовательным перебором. Этот процесс обычно называют сканированием портов (рис. 7.1).

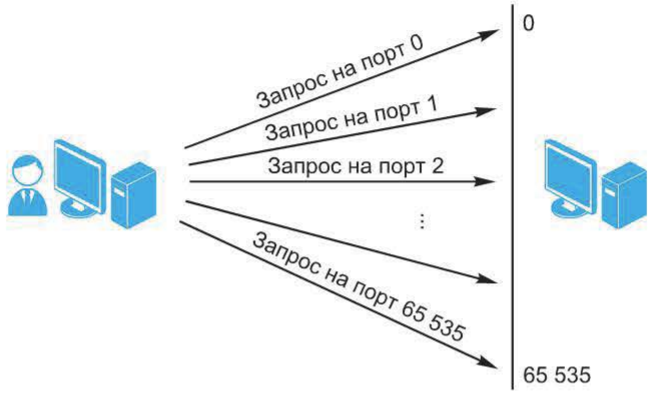


Рис. 7.1. Процесс сканирования портов

Фактически порт на удаленном узле может находиться в одном из двух состояний: открыт, закрыт.

Но при удаленном подключении вследствие влияния межсетевых экранов не всегда можно точно узнать статус порта. В этом случае обычно указывается, что порт фильтруется (рис. 7.2).

Задачу идентификации статуса порта можно решать несколькими способами. Рассмотрим некоторые из них.

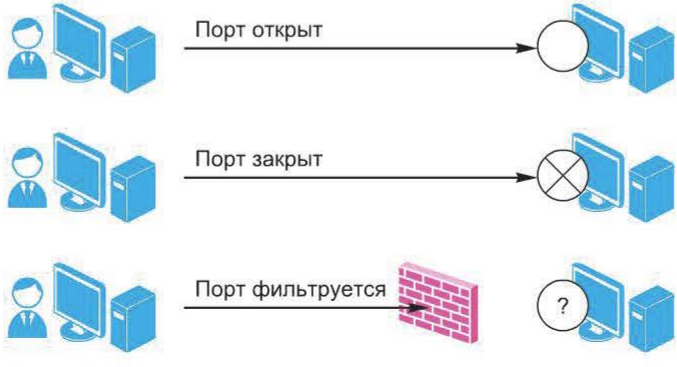


Рис. 7.2. Определение статуса порта при удаленном подключении

7.2. Сканирование портов TCP

Сканирование с установлением соединения. Чтобы убедиться в том, что порт TCP открыт, достаточно попытаться установить с ним соединение. Обычно для этой цели используются возможности ОС. В ОС реализация TCP, как правило, представляет собой отдельный драйвер, а интерфейс между прикладным процессом и TCP — набор системных вызовов, с помощью которых можно открыть или закрыть соединение, отправить или принять данные.

В частности, для установления TCP-соединения может быть использован интерфейс сокетов (функция connect()). После установления соединения его можно тут же разорвать штатной процедурой. Обмен пакетами в этом случае представлен на рис. 7.3.

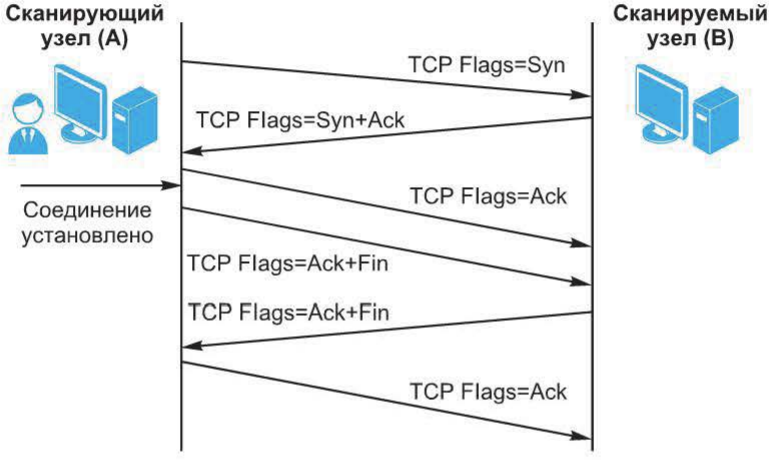


Рис. 7.3. Использование интерфейса сокетов для установления TCP-соединения

Если сканируемый порт закрыт, для установления соединения используют интерфейс, приведенный на рис. 7.4.

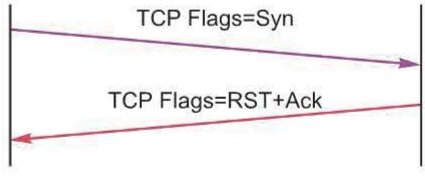


Рис. 7.4. Использование интерфейса сокетов для установления TCP-соединения при закрытом порте

Для ускорения процесса сканирования вместо штатной процедуры завершения соединения можно использовать его аварийное завершение. В этом случае обмен пакетами имеет вид, представленный на рис. 7.5.

SYN-сканирование. Сканирование с установлением соединения имеет следующие достоинства: позволяет использовать штатные возможности ОС и отличается высокой достоверностью.

К недостаткам этого метода обычно относят: недостаточную производительность и невозможность определения фильтрации порта.

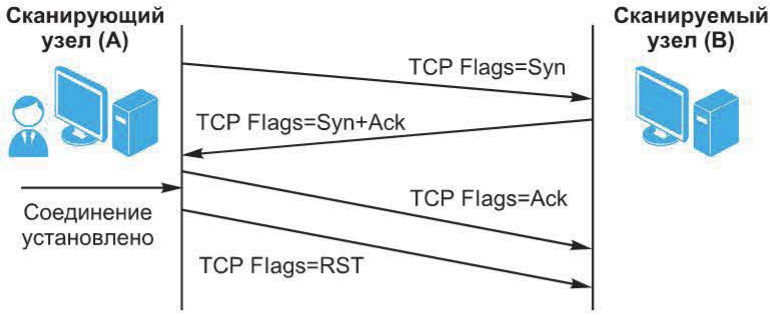


Рис. 7.5. Аварийное завершение соединения

В отличие от сканирования с установлением соединения SYN-сканирование обладает несколько большей производительностью, поскольку тестируемым портом не устанавливается полноценное TCP-соединение. Сканирующий узел (A) отправляет SYN-пакет, как бы намереваясь установить соединение, и ожидает ответа. Наличие флагов SYN|ACK в ответе, пришедшем от узла B, указывает на то, что порт открыт, а флаги RST|ACK в ответе означают обратное (рис. 7.6).

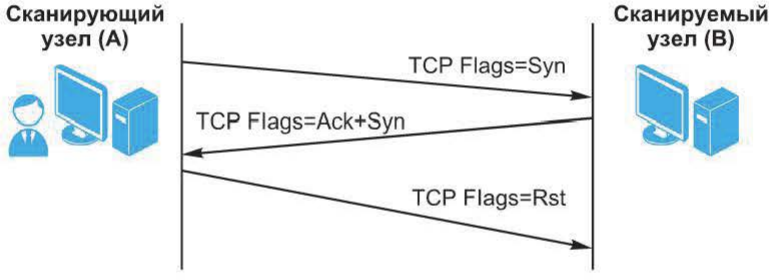


Рис. 7.6. SYN-сканирование

Если же ответ не пришел (но при этом известно, что узел включен), это означает, что порт фильтруется (рис. 7.7).

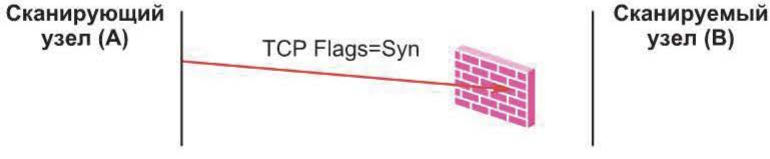


Рис. 7.7. SYN-сканирование при фильтрации порта

Разумеется, для проведения такого сканирования необходимо использовать механизм генерации сетевых пакетов и анализа приходящих ответов. Этим можно объяснить тот факт, что после получения пакета SYN|ACK в ответ отправляется RST-пакет для сброса еще не установленного соединения (эту операцию выполняет ОС).

Фильтрация на прикладном уровне. Представленная выше методика определения факта фильтрации TCP-трафика работает при предположении, что блокировка TCP-пакетов осуществляется на основе критериев транспортного уровня, в данном случае это номер порта.

Однако в ряде случаев фильтрация работает на прикладном уровне, т. е. после того, как TCP-соединение будет установлено. Обмен пакетами в этом случае представлен на рис. 7.8.

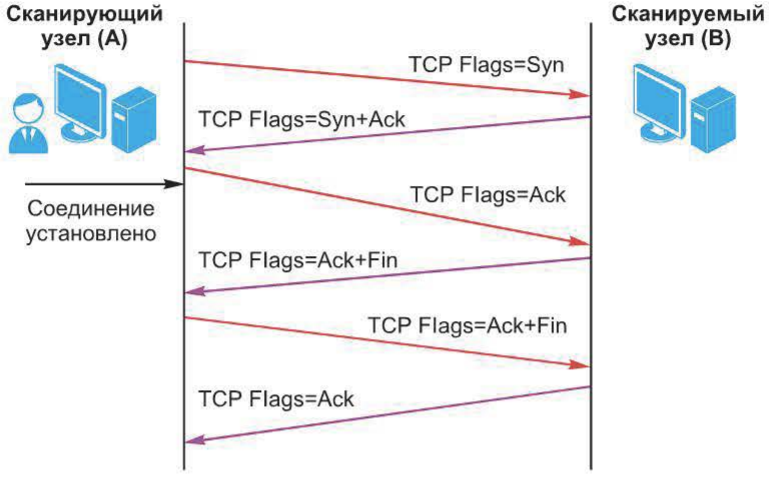


Рис. 7.8. Обмен пакетами при фильтрации на прикладном уровне

Таким образом, приложение разрешает установление соединения TCP, но тут же завершает его штатной операцией. В некоторых приложениях имеется встроенный функционал, ограничивающий сетевое взаимодействии с ними путем задания перечня «разрешенных» IP-адресов. В качестве примера можно привести Microsoft IIS, настройки которого позволяют разграничить доступ к нему на основе IP-адресов (рис. 7.9).

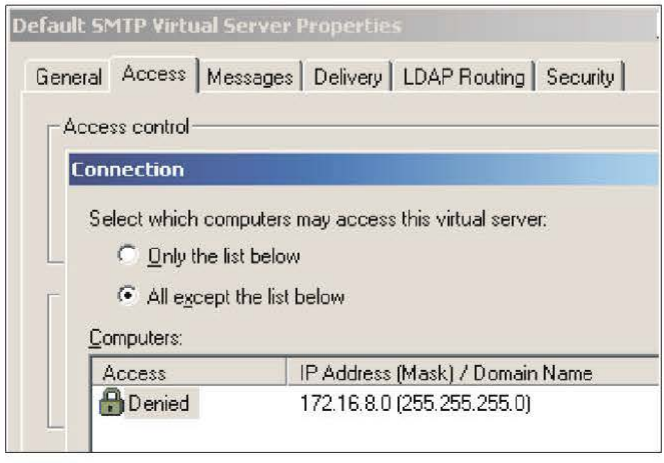


Рис. 7.9. Фильтр IP-адресов Microsoft IIS

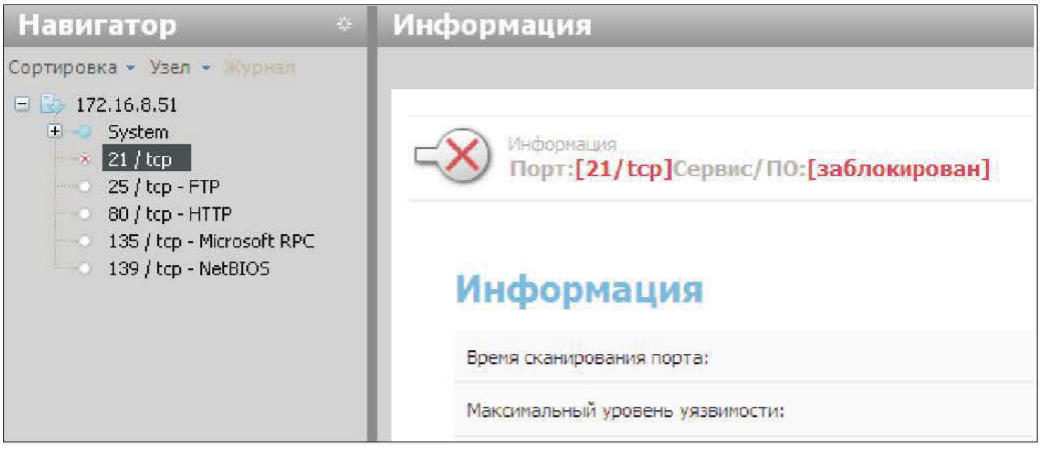


Рис. 7.10. Обнаружение XSpider факта фильтрации на прикладном уровне

Некоторые сканеры умеют определять факт фильтрации на прикладном уровне, например XSpider (при этом статус порта обозначается «заблокирован») (рис. 7.10).

7.3. Сканирование портов UDP

Этот метод используется для определения, какие UDP-порты на сканируемом узле являются открытыми. На требуемый порт сканируемой машины отправляется UDP-пакет (обычно пустой). Если в ответ было получено ICMP-сообщение «Destination Unreachable», это означает, что порт закрыт (рис. 7.11).

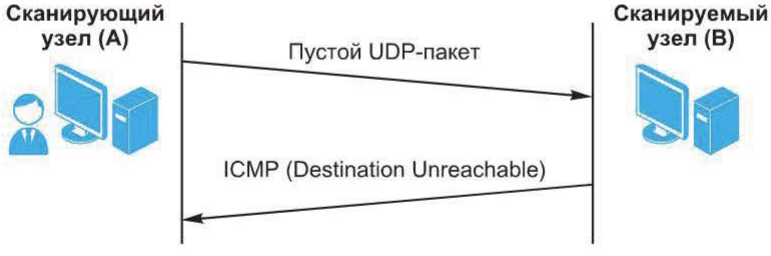


Рис. 7.11. Определение открытых UDP-портов на сканируемом узле

В противном случае (нет ответа) считается, что сканируемый порт открыт.

При UDP-сканировании возникают следующие проблемы:

• возможная потеря UDP-пакетов. В этом случае ответ также не будет получен, и порту может быть ошибочно присвоен статус «открыт»;

• высокая степень вероятности фильтрации UDP- или (и) ICMP-трафика. Результат тот же, что и в предыдущем случае — порт может быть ошибочно считаться открытым.

Это приводит к тому, что в случае неполучения ответа от узла нельзя быть уверенным в том, что порт открыт. Первая проблема решается введением двух параметров, которыми можно регулировать достоверность UDP-сканирования:

• число посылаемых UDP-пакетов;

• время ожидания ответа.

Вторая проблема гораздо сложнее. Для ее решения разработчики сканеров используют различные усовершенствования, в частности одно из них используется в известном сканере уязвимостей Internet Scanner.

Перед сканированием заданных пользователем портов UDP Internet Scanner проводит UDP-сканирование портов из начала диапазона 1-65535 (230-240), из середины диапазона (2050-2060) и из конца диапазона (45270-45280) (рис. 7.12). Тогда выбранные порты с большой долей вероятности окажутся закрытыми.

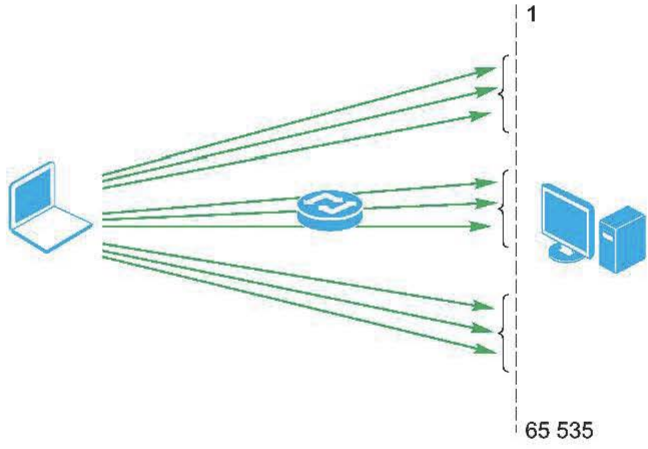


Рис. 7.12. UDP-сканирование портов сканером Internet Scanner

Следовательно, ICMP-сообщение «Destination Unreachable» должно быть получено с большой долей вероятности. Если не будет получено ни одного такого сообщения, Internet Scanner не предпринимает попыток UDP-сканирования и идентификации уязвимостей, основанных на UDP.

Одним из способов повышения достоверности сканирования портов UDP является использование «осмысленных» запросов к сервисам вместо пустых UDP-пакетов. В этом случае ответ от сервиса означает, что порт открыт (рис. 7.13).

Такой способ сканирования UDP-портов используется, например, в сканерах XSpider, MaxPatrol, Nessus.

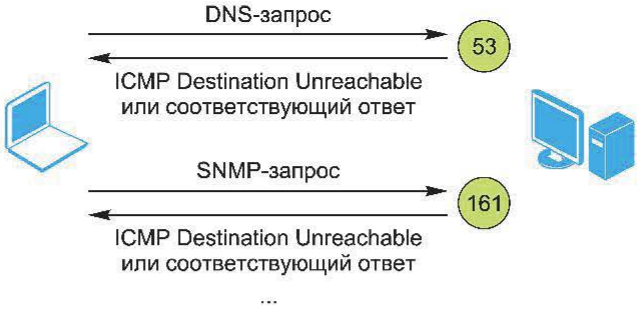


Рис. 7.13. «Осмысленные» запросы к сервисам

Контрольные вопросы

1. В чем заключается задача идентификации статуса порта?
2. Опишите особенности сканирования TCP-портов.
3. В чем отличие TCP-сканирования от сканирования UDP-портов?

Глава 8. ИДЕНТИФИКАЦИЯ СЕРВИСОВ И ПРИЛОЖЕНИЙ

8.1. Идентификация TCP-служб

Задача идентификации служб (приложений) — самая важная в контексте анализа защищенности. Значительная часть уязвимостей относится к уровню приложений. На основе информации, собранной на данном этапе, строятся методы выявления уязвимостей по косвенным признакам.

Использование баннеров. «Классический» метод сбора информации о запущенной на сканируемом узле службе — анализ баннеров. Этот метод заключается в анализе приветствий, выводимых службами при подключении на заданный порт. Часто баннеры содержат информацию об используемой службе, вплоть до номера версии. Поскольку не все службы являются абсолютно переносимыми, это дает возможность делать предположения об используемой ОС, например:

```
telnet ftp.dmn1.ru 21
220 telnetftp.dmn1.ruFTPserver (Versionwu-2.4(37) MonFeb
15 16:48:38 MSK 1999) ready.

telnet smtp.dmn1.ru 25
220 smtp.dmn1.ru ESMTP Sendmail 8.11.2/8.11.2; Thu, 21
Jun 2001 18:34:19 +0400
...
```

При этом следует отметить следующие недостатки:

• многие службы позволяют администратору произвольно редактировать свои приветствия, т. е. существует вероятность (хотя и довольно малая), что служба не та, за кого себя выдает;

• есть риск, что ОС сканируемого узла работает в какой-нибудь среде эмуляции (например, VMWare). Это может оказать влияние на проверки, основанные на особенностях реализации стека TCP/IP.

Эти недостатки не позволяют использовать такой метод как единственный для идентификации служб.

Учет особенностей работы протоколов. Более достоверными являются методы, основанные на анализе особенностей работы служб. Суть этих методов состоит в посылке запросов, которые незначительно отличаются от стандарта, в использовании редких (малоизвестных) команд или опций.

SMTP-сервер. Несколько стандартов — RFC 821, RFC 1425, RFC 1985 — определяют поведение SMTP-сервера: команды, которые SMTP-клиент может выполнить, подключившись к серверу, обязательные возможности сервера, допустимые аргументы и данные. Однако, как обычно, не все реализации SMTP-серверов удовлетворяют этим требованиям. Кроме того, анализу подлежат и сообщения об ошибках, выдаваемые сервером, хотя эти сообщения могут быть изменены администратором сервера, что снижает достоверность данного метода. Как правило, достаточно кода ошибки. Ниже приведено несколько приемов, позволяющих отличить один SMTP-сервер от другого:

• корректно заданная команда MAILFROM без предварительно переданной команды HELO. Некоторые серверы это позволяют (возвращая код ошибки 220), другие — запрещают (501 или 503);

• команда HELO без указания имени домена. Стандарт этого не разрешает, но некоторые серверы позволяют выполнить команду таким образом;

• использование команды MAILFROM<имя> без указания символа «:» после FROM. Некоторые серверы, например, qmail, это позволяют, хотя стандарт явно запрещает;

• использование команды MAILFROM: <> с пустым адресом отправителя. Все серверы должны это разрешать, но возможны исключения;

• некорректное задание адреса отправителя в команде MAILFROM. Некоторые серверы это запрещают, т. е. проверяют существование указанного домена.

Одним из распространенных методов идентификации SMTP-сервера является проверка поддержки некоторых команд: HELP; VRFY; EXPN; TURN; SOML; SAML; NOOP; EHLO.

Следует отметить технику mail-bouncing, которая недостаточно распространена из-за большой сложности и малой скорости работы. Смысл техники заключается в анализе заголовков электронных писем, специально составленных и посланных в исследуемую сеть. Так, интерес представляют письма для несуществующих пользователей, поскольку они возвращают уведомления о невозможности доставки (не всегда). В этих уведомлениях содержится некоторая информация о почтовых серверах, участвующих в процессе доставки письма. На основе нескольких таких «писем-бумерангов» можно узнать некоторое число узлов внутренней сети (не имея к ней непосредственного доступа) и топологию почтовых пересылок. Кроме того, почтовый протокол

позволяет отправлять письма с явным указанием нескольких промежуточных пунктов пересылки. Это дает возможность создать письмо, которое, проделав заданный маршрут внутри исследуемой сети, вернется к отправителю (все это, конечно, существенно зависит от настроек почтовых серверов).

Web-сервер. Важная служба прикладного уровня — HTTP. Протокол HTTP версии 1.1 описан в стандарте RFC 2068. В нем предусмотрен метод OPTIONS, согласно которому HTTP-сервер возвращает развернутую информацию о себе, например:

```
OPTIONS * HTTP\1.1
HTTP/1.1 200 OK
Date Wed 20 Jun 2001 17:41:42 GMT
Server: Apache/1.3.19 (Unix) PHP/4.0.5 mod_jk rus/PL30.4
Content-Length: 0
Allow: GET, HEAD, OPTIONS, TRACE
Connection: close
```

Это лишь один из способов. Для получения дополнительной информации можно использовать характеристики стандартных конфигураций различных www-серверов под различные платформы.

Инструменты сканирования. Известная утилита nmap наряду с возможностями по сканированию портов и идентификации ОС имеет возможность идентификации служб.

Сканер приложений amap (http://packetstormsecurity.org/groups/thc/amap-2.5.tar.gz) позволяет идентифицировать приложения, работающие на нестандартных портах (в том числе и так называемые non-ascii based applications). Утилита использует простой метод: посылку на выбранный порт специальным образом построенного пакета и анализ ответа. БД запросов представляет собой текстовый файл appdefs.trig Полученный от узла ответ сравнивается с типовыми ответами из файла appdefs.resp

Простейший вариант использования утилиты:

```
amap <адрес><порт>
```

Сканер amap может использовать результаты утилиты nmap, в этом случае формат запуска следующий:

```
amap -i results.nmap -o results.amap –m
```

Имеются также специализированные инструменты анализа служб, например, программа SMTP Scan.

Тесты, выполняемые программой в отношении SMTP-сервера, определены в файле /usr/local/share/smtpscan/tests Почти все тесты представляют собой запросы, ответы на которые точно не определены в соответствующих документах RFC. В ответ на каждый запрос SMTP-сервер возвращает код ошибки, на основе которого и делается вывод о версии и модели сервера.

8.2. Идентификация UDP-служб

Сканирование на прикладном уровне. Работающий без установления соединения протокол UDP затрудняет не только сканирование UDP-портов (об этом уже говорилось выше), но и идентификацию служб, использующих эти порты.

Большинство прикладных служб сети Интернет используют протокол TCP в качестве транспорта, только некоторые из них пользуются протоколом UDP. В большинстве случаев используются стандартные номера портов. Эти два факта лежат в основе метода, используемого в программе XSpider: сканирование UDP-портов сразу на прикладном уровне. Идея очень простая: вместо пустого UDP-пакета посылать запрос, содержащий данные прикладного уровня, характерные для этой службы. Например, на порт 53 посылается DNS-запрос, на порт 161 — SNMP-запрос и т. д. Это уменьшает число ложных срабатываний, но сужает диапазон сканирования, делая его состоящим только из «хорошо известных» портов.

Анализ параметров повторной передачи. Поскольку UDP ненадежный протокол, обеспечение надежности — задача приложения. Обычно для обеспечения надежности используется метод повторной передачи. Если ответ от узла не пришел в течение определенного времени, проводится повторная передача пакета. При этом используются следующие параметры:

- время ожидания перед повторной передачей пакета;
- изменение времени ожидания с каждой последующей передачей пакета;
- число передаваемых пакетов.

Как правило, стратегия повторной передачи не определена стандартом, и разработчик ПО выбирает ее сам. Следовательно, можно идентифицировать базирующуюся на UDP службу на основе определения стратегии повторной передачи, которая варьируется от приложения к приложению. Например, этот метод реализован в программе IKEScan.

8.3. Сканирование протоколов

Выше были рассмотрены методы сканирования портов и идентификации служб. Но вся эта информация относилась к транспортному и прикладному уровням. Однако на сетевом уровне иногда возникает задача определения используемых в сети IP-протоколов. Например, на рис. 8.1 приведен модуль слежения системы обнаружения атак, установленный между маршрутизатором и межсетевым экраном.

Обычная ситуация в этом случае — регулярно появляющиеся события Unknown protocol. Для настройки системы обнаружения атак потребуется определить используемые в сети протоколы. Иногда в такой ситуации используют сканирование протоколов, т. е. сканирование сетевых устройств

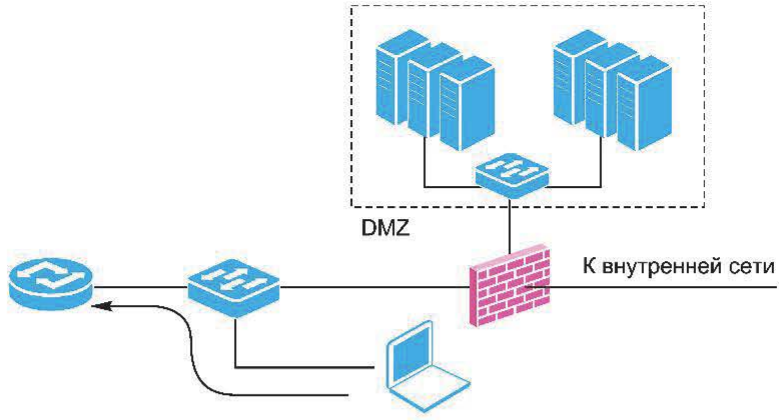


Рис. 8.1. Модуль слежения IDS между маршрутизатором и межсетевым экраном

с целью определения поддерживаемых ими типов IP. Полученная информация может быть использована для предварительной настройки модуля слежения системы обнаружения атак.

Принцип сканирования заключается в следующем (рис. 8.2).

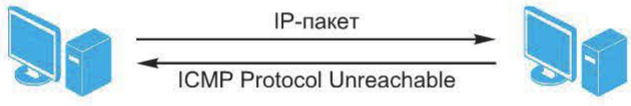


Рис. 8.2. Сканирование протоколов

На узел посылается IP-пакет с требуемым значением поля «тип протокола» в заголовке.

Если в ответ пришло сообщение ICMP Protocol Unreachable, протокол не поддерживается. Если ответ не пришел, протокол поддерживается. Этот тип сканирования очень напоминает UDP-сканирование, поэтому ему присущи те же проблемы. Например, при сканировании межсетевых экранов пакет ICMP Protocol Unreachable может быть заблокирован, и тогда возможно большое число ложных срабатываний.

Для проведения сканирования можно использовать сканер nmap.

Синтаксис:

```
nmap –sO <сканируемый узел>
```

Контрольные вопросы

1. В чем заключается суть метода анализа баннеров?
2. Перечислите методы, основанные на анализе особенностей работы служб. Каковы их преимущества по сравнению с методом анализа баннеров?
3. Дайте характеристику и приведите особенности работы утилиты nmap.
4. Как проводится идентификация UDP-служб?
5. В чем заключается метод сканирования протоколов?

Глава 9. ИДЕНТИФИКАЦИЯ ОПЕРАЦИОННЫХ СИСТЕМ

Один из этапов сбора информации о сетевых ресурсах — определение типа и версии ОС удаленного узла.

Все известные методы определения ОС можно сгруппировать следующим образом:

- простейшие методы;
- TCP/IP Fingerprinting;
- основанные на использовании протокола ICMP;
- малоизвестные, редко используемые.

Рассмотрим более подробно приведенные методы.

9.1. Простейшие методы определения ОС

К данной группе относятся:

- анализ наборов открытых портов;
- использование сервисов прикладного уровня;
- анализ баннеров сервисов прикладного уровня;
- использование команд протоколов прикладного уровня;
- анализ результатов идентификации сервисов и приложений.

Анализ наборов открытых портов. Наиболее простой метод основан на очевидном факте, что ряд сервисов прикладного уровня жестко «связан» с платформой. Например, открытый порт 22 (обычно используемый сервисом SSH) почти однозначно указывает на ОС UNIX, а порты 135, 139 — на ОС Windows (рис. 9.1).

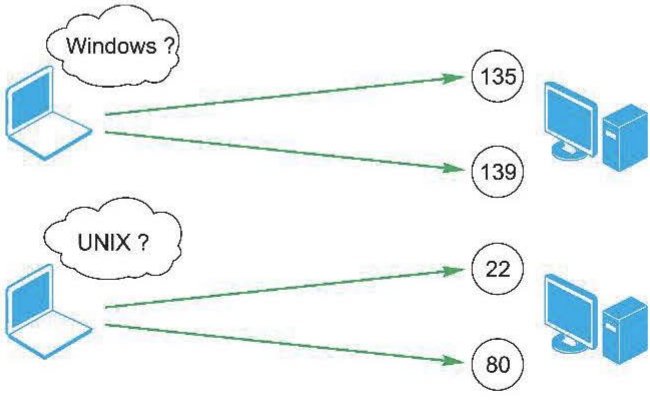


Рис. 9.1. Анализ наборов открытых портов

Использование сервисов прикладного уровня. Один из самых простых методов определения ОС удаленного узла — подключение на открытые порты и анализ отклика работающих на них служб (рис. 9.2).

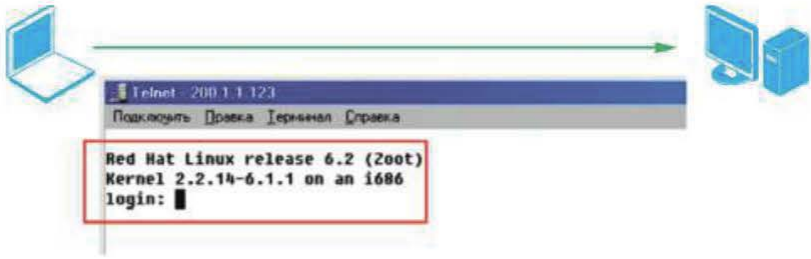


Рис. 9.2. Подключение на открытые порты и анализ отклика работающих на них служб

Часто приветствия (баннеры) прикладных сервисов содержат также информацию об ОС.

Отметим еще один метод — использование команд служб прикладного уровня, например команда SYST протокола FTP (рис. 9.3).

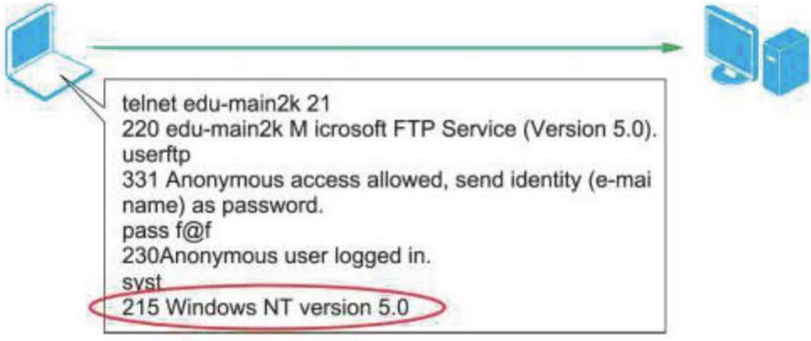


Рис. 9.3. Использование команд служб прикладного уровня

Возможно также использование протокола SMB (рис. 9.4).

Наконец, вывод о типе ОС может быть сделан на основе результатов идентификации сервисов и приложений (рис. 9.5).

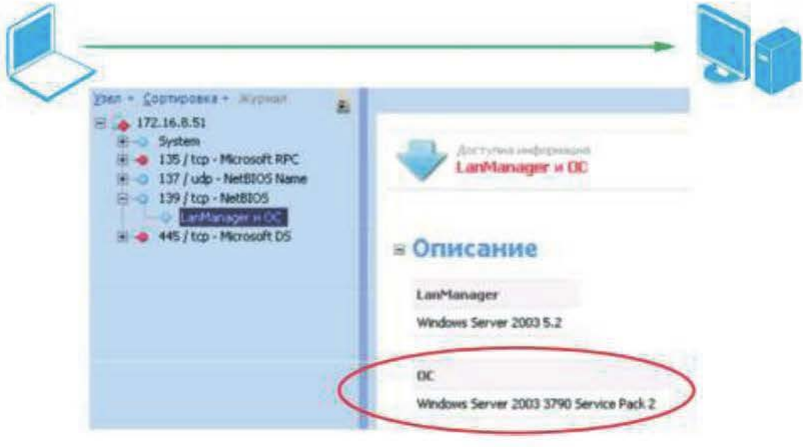


Рис. 9.4. Использование протокола SMB

Рис. 9.5. Идентификация сервисов и приложений

Например, если на удаленном узле открыт порт 220, а результаты идентификации сервисов указывают на SSH, с большой долей вероятности можно утверждать, что это ОС UNIX.

9.2. Опрос стека TCP/IP

Различать ОС можно на основе различий в реализации стека TCP/IP (рис. 9.6).

Из-за этих различий реакция ОС на определенные сетевые пакеты будет разной. Метод, основанный на данном наблюдении, называется TCP/IP Stack Fingerprinting.

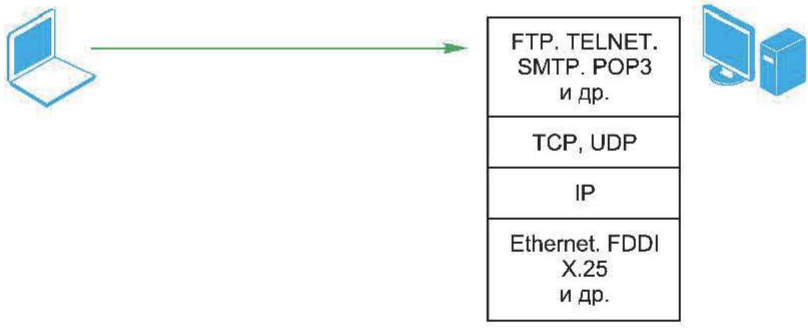


Рис. 9.6. Различия в реализации стека TCP/IP

Исследование значения ISN. Запрос на установление TCP-соединения содержит флаг SYN в заголовке пакета и начальное значение в поле Sequence Number (рис. 9.7). Это значение называется ISN (Initial Sequence Number).

Узел, получив запрос на соединение, увеличивает значение ISN на единицу и записывает полученное значение в поле Acknowledgement Number. При этом в поле Sequence Number записывается значение ISN сканируемого узла. Это значение и является предметом исследования в данном случае. Запомнив его, сканирующий узел повторяет запрос на установление соединения и опять получает значение ISN. Операция повторяется до тех пор, пока не будет выявлен закон изменения ISN сканируемого узла (рис. 9.8).

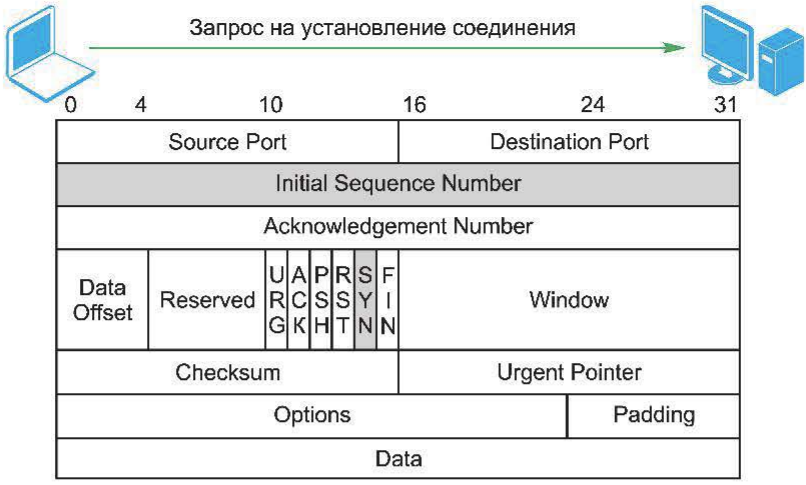


Рис. 9.7. Запрос на установление TCP-соединения

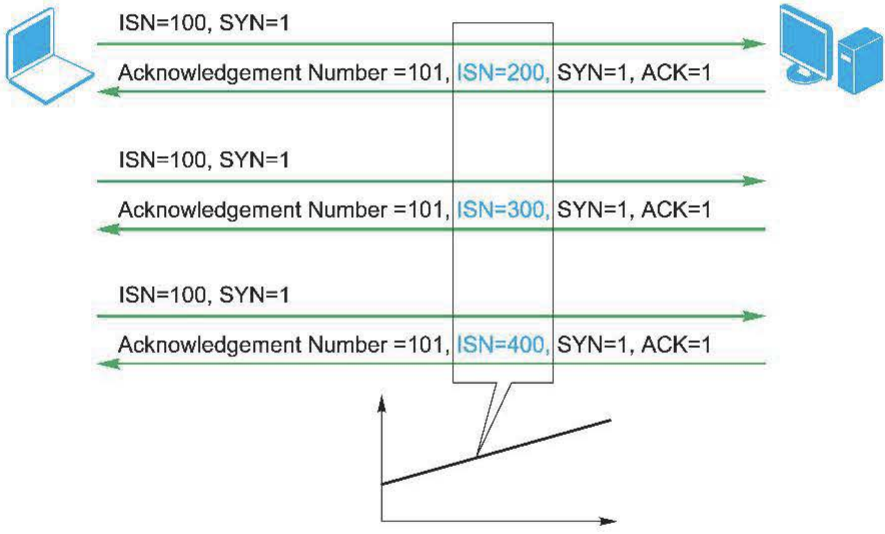


Рис. 9.8. Выявление закона изменения ISN сканируемого узла

При этом возможны следующие ситуации:

• «постоянное приращение» (традиционный закон «+64» характерен для старых версий UNIX). Значение ISN-сервера, записываемого в поле Sequence Number ответа на запрос на установление соединения, увеличивается каждый раз на постоянную величину (например, на 64);

• «случайное приращение» (новые версии Solaris, IRIX, FreeBSD, DigitalUNIX, Cray): приращение ISN носит случайный характер;

• «истинно случайные значения» (Linux 2.0.x, OpenVMS, новые AIX): значение ISN является случайной величиной;

• закон «время зависимых приращений» (Windows): значение ISN периодически во времени увеличивается на некоторую небольшую величину;

• постоянный (концентраторы 3Com [ISN=0x803], принтеры Apple LaserWriter [0xC7001]): значение ISN остается постоянным.

Исследование поля Window TCP-пакета. При анализе принятых от сканируемого узла TCP-сегментов можно обратить внимание на поле Window в их заголовках, поскольку значение этого поля является своеобразной константой, характеризующей ОС (табл. 9.1). В некоторых случаях этого поля достаточно для однозначного определения типа ОС.

Таблица 9.1

Примеры полей Window в их заголовках

№ п/п	ОС	Значение поля Window в ответе на SYN-запрос на открытый порт (десятичное значение)
1	LinuxRedHat 9 (2.4.20-8)	5 840
2	Windows 98	8 576
3	Windows 2003 Server	16 616
4	Windows 2000 Server SP4	65 535
5	Windows NT SP6	8 576
6	Windows 2000 Prof SP4	65 535

Использование поля Reserved (Bogus-flag). Посылка запроса на установление соединения (установлен флаг SYN) с установленным в TCP-заголовке неиспользуемым флагом BOGUS. Флаг BOGUS не является настоящим. Этот термин подразумевает установку в поле Reserved заголовка TCP-пакета 100 000 бит (вместо всех нулей в соответствии с RFC 793) (рис. 9.9).

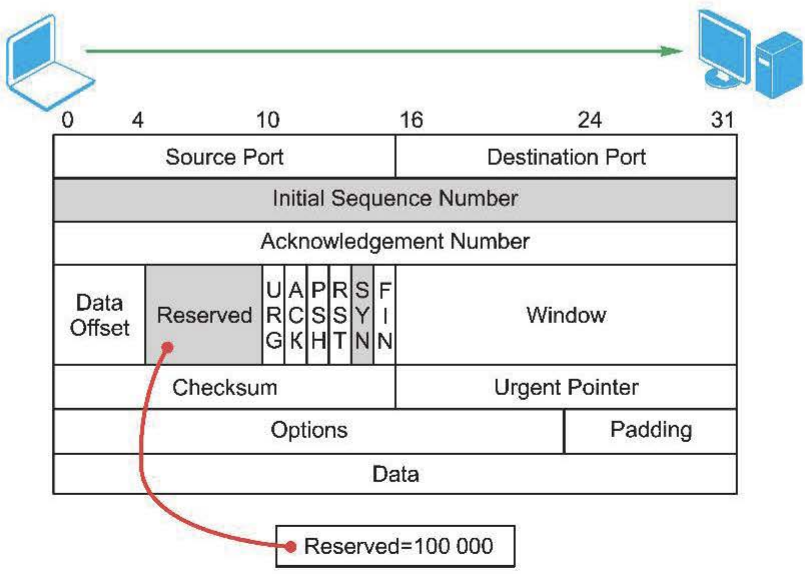


Рис. 9.9. Установка 100 000 в поле Reserved

Например, ОС Linux до 2.0.35 сохраняет в ответе этот флаг, а некоторые ОС разрывают соединение при получении такого пакета (отвечают пакетом с флагом RST). Большая же часть ОС отвечает на данный запрос обычным образом.

Исследование поля ACK. В RFC 793 определено стандартное изменение поля ACK в TCP-пакетах при установлении соединения, передаче данных и закрытии соединения. Однако в нестандартных ситуациях различные ОС по-разному устанавливают значение этого поля.

Исследование проводится следующим образом. На закрытый TCP-порт отправляется пакет с известным значением ISN и установленными флагами FIN|PSH|URG. Большинство современных версий ОС поместят в поле ACK-ответа ISN+1. Некоторые скопируют значение ISN в поле ACK-ответа.

Отметим, что, если отправить пакет с установленными флагами SYN|FIN|PSH|URG, большая часть ОС отвечает значением ISN+2.

9.3. Инструменты

Наиболее известным инструментом, в котором реализованы приведенные выше методы (а также многие другие), является утилита nmap. Менее известна и практически не используется в настоящее время утилита QueSO.

В утилите nmap для определения ОС используются 16 пакетов (рис. 9.10).

Рис. 9.10. Определение ОС с помощью утилиты nmap

9.4. SinFP

SinFP — утилита для идентификации ОС, в основу которой положена методика, схожая с TCP/IP Stack Fingerprinting. Работа утилиты строится на предположении, что в отношении сканируемого узла выполняются следующие условия:

- открыт как минимум один порт TCP;
- все остальные порты TCP и UDP могут быть отфильтрованы;
- фильтрация осуществляется с использованием технологии State Fulin Section на транспортном и прикладном уровнях.

Исходя из приведенных условий, отправляемые сетевые пакеты должны быть корректны с точки зрения пакетного фильтра.

В отношении узла отправляется три TCP-пакета:

- SYN-запрос на открытый порт без опций;
- SYN-запрос на открытый порт с различными опциями;
- пакет с флагами SYN+ACK.

Полученные ответы сопоставляются с БД.

9.5. Использование протокола ICMP

Рассмотренные выше утилиты nmap и queso при определении ОС используют TCP- и UDP-протоколы. Далее рассматривается методика идентификации ОС на основе ICMP-протокола и соответствующей утилиты xprobe2.

Исследования в области использования протокола ICMP для идентификации ОС были проведены О. Аркином и Ф. Ярошкиным. Использование ICMP-протокола дает следующие преимущества:

- небольшое число посылаемых IP-датаграмм. Утилита xprobe использует четыре пакета;
- сложность обнаружения IDS; в процессе идентификации не требуется посылки некорректных пакетов.

Примером инструмента, использующего рассматриваемые ниже методы, является утилита xprobe2. Суть методов заключается в следующем: необходимо, чтобы удаленный узел отправил ICMP-сообщение об ошибке; затем это сообщение тщательно анализируется.

Размер ICMP-сообщения. Любое ICMP-сообщение об ошибке включает в себя заголовок IP и по крайней мере первые 8 байт датаграммы, вызвавшей ситуацию ошибки (рис. 9.11).

Согласно RFC 1122, сообщение об ошибке может содержать и более 8 байт. Большинство ОС, однако, пересылают именно 8 байт, некоторые ОС пересылают более 8 байт, тем самым позволяя себя идентифицировать.

Рис. 9.11. Структура IP-пакета с ICMP-сообщением об ошибке

К ним относятся: Linux 2.0.x/2.2.x/2.4.x; Sun Solaris 2.x; HPUX 11.x; MacOS 7.x—9.x; Nokia boxes; Foundry Switches.

Изменения в исходной датаграмме. Реализации стека TCP/IP в некоторых ОС при генерации сообщения об ошибке вносят изменения в заголовок датаграммы, вызвавшей появление этого сообщения и в поле данных этой датаграммы. Меняться должны только два поля заголовка IP:

- Timetolive (при прохождении датаграммы до получателя это поле может изменяться);
- контрольная сумма (пересчитывается при изменении поля Timetolive).

Идентификация ОС в данном случае осуществляется путем поиска изменений в остальных полях заголовка IP и в поле данных.

Поле Total Length. Некоторые ОС увеличивают значение поля Total Length на 20 байт, другие уменьшают на 20 байт. Наконец, некоторые ОС оставляют значение этого поля без изменений.

Поле Identification. Некоторые ОС изменяют порядок следования бит в данном поле. Например, Linux (ядро 2.4.0.-2.4.4) устанавливает значение этого поля 0 (это справедливо только для сообщений ICMP Echo Request и Reply).

Поля Flags и Fragment Offset. Аналогично предыдущему случаю некоторые ОС меняют порядок следования бит в данных полях.

Контрольная сумма заголовка IP. Некоторые ОС неверно пересчитывают контрольную сумму в заголовке IP пакета, передаваемого в ICMP-сообщении об ошибке, другие ОС обнуляют это поле.

Контрольная сумма заголовка UDP. В этом случае действия ОС аналогичны действиям при пересчете контрольной суммы пакета, передаваемого в ICMP-сообщении об ошибке.

Поле TTL в запросах и ответах ICMP. Как правило, поле TTL в запросе ICMP может не совпадать с полем TTL в его ответе. Это поле позволяет по ответу на запрос идентифицировать ОС или их группу ОС. Его можно использовать как дополнительный критерий к приведенным выше. Поле TTL используется и при пассивных методах идентификации ОС (например, на основе анализа трафика).

Поле «Код» в запросах и ответах ICMP. Поле «Код» используется в сообщениях ICMP об ошибках, в запросах и ответах оно обычно равно 0. Однако если в запросе ICMP Echo (type 8) в поле «Код» задать какое-либо значение, то некоторые ОС (например, Windows) в ответе обнуляют это поле, другие — сохраняют значение поля «Код».

Поле Type of Service. Правила использования этого поля в ICMP-сообщениях определены в RFC 1349. Они отличаются для сообщений об ошибках (Destination Unreachable, Source Quench, Redirect, Time Exceeded, Parameter Problem), запросов (Echo, Router Solicitation, Timestamp, Information Request, Address Mask Request) и ответов (Echo Reply, Router Advertisement, Timestamp Reply, Information Reply, Address Mask Reply). Сообщения об ошибках должны содержать 0 в поле Type of Service (это значение по умолчанию). Запросы

ICMP могут иметь любое значение в этом поле. Ответы на запросы имеют то же значение в этом поле, что и запросы. Некоторые ОС игнорируют требования RFC 1349 и не устанавливают в ответах на запросы правильное значение поля Type of Service.

Утилита xprobe2. Практической реализацией приведенных методов является утилита xprobe2 http://www.sys-security.com/archive/tools/xprobe2/xprobe2-0.2.tar.gz. Ее архитектура и последовательность работы модулей приведены на рис. 9.12.

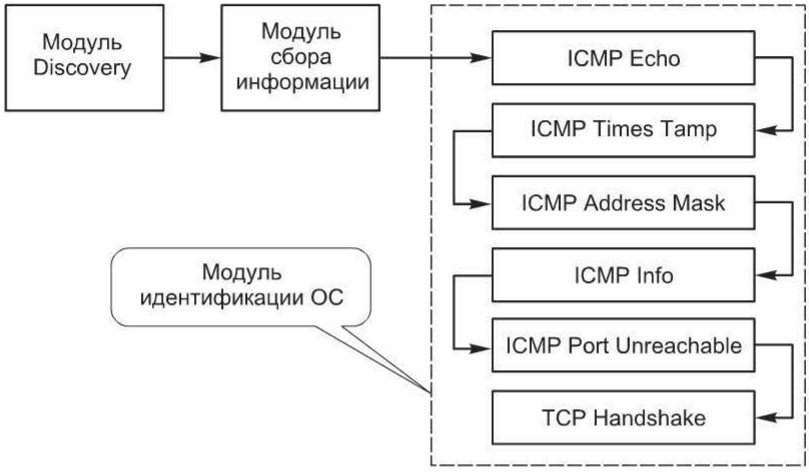


Рис. 9.12. Архитектура утилиты xprobe2

Модуль Discovery служит для идентификации сетевых объектов, он использует приведенные выше методы ICMP Echo, TCP ping, UDP ping. Модуль сбора информации выполняет отслеживание маршрутов и сканирование портов. Модуль идентификации ОС собственно и выполняет определение ОС сканируемого узла. Для этого он вначале посылает различные ICMP-запросы и анализирует пришедшие на них ответы. Затем на закрытый порт посылается UDP-датаграмма с целью получения ICMP-сообщения Port Unreachable. Это сообщение анализируется и делается вывод об ОС сканируемого узла.

9.6. Retransmission Timeout

Выше рассматривалась схема установления соединения по протоколу TCP, согласно которой в ответ на SYN-запрос отправлялся пакет с флагами SYN|ACK. Что произойдет, если третий, завершающий процедуру установления соединения пакет не будет отправлен? Поскольку TCP-протокол является надежным, через некоторое время RTO (Retransmission Timeout) пакет SYN|ACK будет отправлен повторно. Еще через некоторое (возможно, чуть большее) время пакет SYN|ACK будет передан в третий раз и т. д. (рис. 9.13).

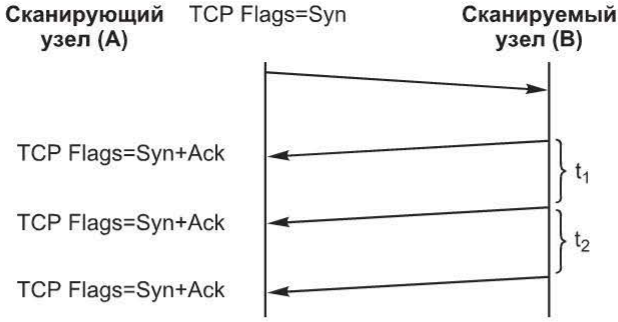


Рис. 9.13. Retransmission Timeout

Если после определенного числа пересылок соединение так и не будет установлено, оно сбрасывается.

Таким образом, в этом случае используются три параметра:

- число пересылок;
- время ожидания (Retransmission Timeout);
- закон, по которому увеличивается время ожидания с каждой следующей пересылкой.

Поскольку RFCs не содержат четких рекомендаций, разные реализации стека TCP/IP используют различные значения. Это позволяет различать ОС. Практическая реализация этого метода — утилита RING (http://www.intranode.com/pdf/techno/ring-0.0.1.tar.gz). Утилита отправляет SYN-запрос на открытый порт и анализирует приходящие от исследуемого узла пакеты SYN|ACK.

9.7. Port 0 OS Fingerprinting

Рассмотрим еще один интересный метод идентификации ОС — исследование ответа от порта с номером 0. Согласно RFC 1700, порт 0 зарезервирован и не может быть использован какой-либо службой. Как известно, если порт источника равен 0, то при установлении соединения он динамически назначается ОС.

Однако многие ОС отвечают на запросы в отношении порта 0. Поскольку реакции разных ОС могут быть различны, их можно использовать для идентификации ОС.

Рекомендуемые тесты:

P1:	send	tcp	packet	from	source	port	0	to	port	0
P2:	send	tcp	packet	from	source	port	X	to	port	0
P3:	send	tcp	packet	from	source	port	0	to	open	port
P4:	send	tcp	packet	from	source	port	0	to	closed	port
P5:	send	udp	packet	from	source	port	0	to	port	0
P6:	send	udp	packet	from	source	port	53	to	port	0
P7:	send	udp	packet	from	source	port	0	to	closed	port

9.8. Активная идентификация ОС — перспективы

На правильность результатов идентификации ОС значительное влияние оказывает взаимное расположение сканирующего и сканируемого узлов. Межсетевые экраны затрудняют определение ОС сканируемого узла.

Для повышения точности идентификации ОС должно быть использовано как можно большее число различных тестов. Часть из них может окончиться неудачей, если приняты меры по защите сканируемого узла.

Довольно часто сложно различать ОС, если они относятся к одной группе (например, Windows). В этом случае один из путей решения проблемы — идентификация служб сканируемого узла. Это косвенно может помочь при идентификации ОС.

Контрольные вопросы

1. Перечислите все известные методы идентификации операционных систем.
2. Перечислите и дайте характеристику инструментарию для идентификации операционных систем.

Глава 10. ИДЕНТИФИКАЦИЯ УЯЗВИМОСТЕЙ ПО КОСВЕННЫМ ПРИЗНАКАМ

10.1. Методы идентификации уязвимостей по косвенным признакам

Выше были рассмотрены методы сбора информации о сканируемом объекте. В системах анализа защищенности значительная часть этой информации используется для того, чтобы сделать вывод о наличии уязвимости. Такой способ называют *идентификацией уязвимостей по косвенным признакам*. В целом проверки, встроенные в сетевые системы анализа защищенности, можно классифицировать следующим образом (рис. 10.1).

Ниже рассмотрены баннерные и локальные проверки.

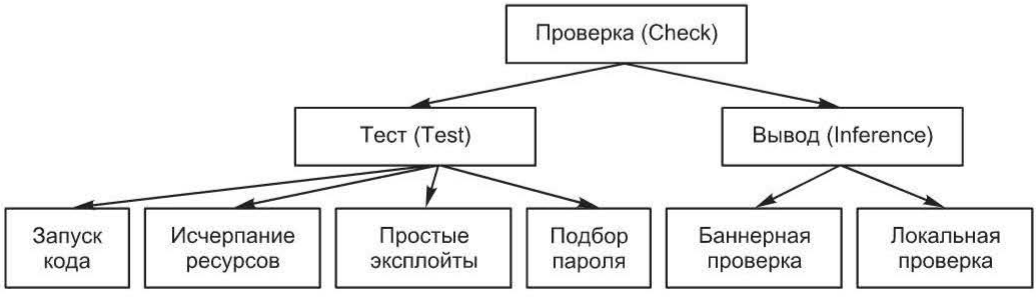


Рис. 10.1. Проверки, встроенные в сетевые системы анализа защищенности

10.2. Баннерные проверки

По результатам идентификации сервисов и приложений вывод о наличии той или иной уязвимости обычно делается на основе версии. Например, в Internet Scanner имеется проверка Bindnxtbo, которая выявляет наличие уязвимости CVE-1999-0833 в сервере BIND. Работа этой проверки заключается в следующем:

- Internet Scanner отправляет запрос серверу BIND для определения номера версии;
- если номера версий 8.2, 8.2 p1 или 8.2.1, делается вывод о наличии уязвимости.

Пример описания проверки в сканере xSpider, работающей аналогично.

Название: переполнение буфера (Lotus Domino).

Краткое описание: уязвимость сервера Lotus Domino, являющаяся следствием переполнения буфера, возникающего при обработке сервером запроса HTTPPOST и приводящая к выполнению произвольного кода или созданию ситуации «отказа в обслуживании».

.....

Ложные срабатывания: вывод о наличии уязвимости сделан на основе версии по результатам идентификации сервисов и приложений на сканируемом узле. Если при установке обновления номер версии не менялся, возможно ложное обнаружение уязвимости.

Ссылки: CVE (CAN-2005-1101): http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2005-1101

Bugtraq (Bid 13130): http://www.securityfocus.com/bid/13130

XForce (lotus-timedate-bo, 20042): http://xforce.iss.net/xforce/xfdb/20042

Securitylab: http://www.securitylab.ru/54005.html

Сайт производителя: http://www-1.ibm.com/support/docview.wss?rs=463&uid=swg21202431

По описанию уязвимости видно, что ее результат во многом зависит от корректности инвентаризационной информации.

10.3. Сетевые сервисы как объект сканирования

Очевидно, что сетевые сервисы можно назвать основным объектом анализа защищенности, выполняемого сетевым сканером. После того как в ходе инвентаризации были определены открытые порты, соответствующие им сервисы, реализующие эти сервисы приложения, начинается этап идентификации уязвимостей. Значительная часть проверок, направленных на выявление уязвимостей сетевых сервисов, таких как DNS, HTTP, SSH, FTP, — это выше баннерные проверки. Рассмотрим на нескольких примерах проверки некоторых сетевых сервисов более подробно.

Сканирование DNS. Проверки в отношении сервера DNS выполняются двумя способами: путем посылки запросов и анализа версии.

Первым способом выполняются следующие две проверки: поддержка рекурсивных запросов и возможность получения файла «зоны».

Вторая группа проверок выполняется на основе анализа информации, полученной путем запроса version.bind (рис. 10.2).

```
C:\>nslookup
Default Server:  srv-d[redacted].ru
Address:  10.10.0.2

> set class=chaos
> set type=txt
> version.bind
Server:  srv-d[redacted].ru
Address:  10.10.0.2

version.bind.[redacted].ru text =

        "Microsoft DNS 6.0.6001 (17714726)"
```

Рис. 10.2. Запрос version.bind

Запись в том же классе chaos — это authors.bind, она тоже может быть использована для получения информации о сервере DNS.

Сканирование SSH. Протокол SSH (Secure Shell) служит в основном для защиты удаленного управления различными системами (в подавляющем большинстве случаев это UNIX-системы), но возможно его использование и для защиты других сервисов. Архитектурно он состоит из трех частей (уровней):

- Transport Layer Protocol;
- User Authentication Protocol;
- Connection Protocol.

Подробное рассмотрение данного протокола выходит за рамки курса, тем более что сканер безопасности, выполняя баннерные проверки сервиса SSH, взаимодействует только с Transport Layer Protocol.

SSH Transport Layer Protocol обычно работает поверх протокола TCP и обеспечивает:

- конфиденциальность (шифрование трафика);
- аутентификацию сервера;
- контроль целостности;
- сжатие (необязательно).

SSH Transport Layer Protocol (как и большинство других прикладных сервисов) предполагает наличие клиентской и серверной частей. Работая поверх протокола TCP, серверная часть обычно ожидает подключений клиентов на порт 22.

Соединение всегда инициируется клиентом. После установления соединения стороны должны обменяться строками идентификации (Identification String), которые имеют следующий вид:

```
SSH-protoversion-softwareversion SP comments CR LF
```

Строка, переданная сервером, может быть проанализирована, например, с помощью telnet-клиента (рис. 10.3).

Рис. 10.3. Анализ строки идентификации с помощью telnet-клиента

Строка должна начинаться с текста SSH-, затем следует версия протокола, далее версия приложения, реализующего сервис. Поле comments не является обязательным, но если используется, оно отделено пробелом. Содержимое этого поля может быть задано в произвольном файле, например /etc/ssh/banner. Указание выводить содержимое данного файла в качестве приветствия задается в файле /etc/ssh/sshd_config (директива Banner).

Поскольку строка идентификации сервера регламентируется стандартом, ее умышленное изменение администратором маловероятно. Поэтому баннерные проверки сервиса SSH полностью основаны на тщательном анализе строки идентификации сервера.

Иногда строка идентификации может быть изменена умышленно. Обычно это осуществляется путем редактирования файла version.h:

строка

```
#define SSH_VERSION "OpenSSH_4.3"
```

меняется на

```
#define SSH_VERSION "Undisclosed_Version"
```

Это приводит к изменению строки идентификации на SSH-2.0-Undisclosed_Version.

В этом случае баннерная проверка даст неправильный результат.

Методика анализа результатов баннерных проверок. Поскольку результат баннерных проверок зависит от многих факторов, при верификации найденных уязвимостей рекомендуется использовать следующие приемы:

- ручную проверку сервиса (подключение на заданный порт, анализ баннера, использование команд соответствующего сервиса);
- поиск информации об уязвимости в различных базах;
- локальную проверку (версия, конфигурационные файлы);
- проверку действительного существования уязвимости.

10.4. Локальные проверки

В некоторых случаях вывод о наличии уязвимости может быть сделан на основе анализа атрибутов файла. Например, в программе Internet Scanner проверка WinMs03043Patch (CVECAN-2003-0717) проверяет, установлено ли соответствующее обновление. Работает она следующим образом:

- выполняется подключение к ресурсу ADMINS;
- проверяется дата файла \system32\msgsvc.dll. Если получена дата ранее 2 октября 2003, 18:17:32.0, то считается, что уязвимость на узле присутствует.

При анализе защищенности систем Windows источником информации для последующих выводов о наличии уязвимостей служит также реестр. Например, проверка LM_Security в сканере Internet Scanner (проверка поддержки аутентификации по схеме LM) работает следующим образом:

- выполняется подключение к реестру;
- просматриваются значимые элементы ключа реестра System\CurrentControlSet\Control\LSA.

Анализ файловой системы и реестра предполагает подключение к сканируемому узлу с учетной записью, имеющей достаточный уровень привилегий. Если речь идет о системах UNIX , то сканер выполняет подключение по протоколу SSH, используя учетную запись root, и выполняет так называемые локальные проверки (Local Security Check).

В случае систем Windows для сбора подобной информации требуется доступ с привилегиями администратора к реестру или файловой системе. Кроме того, в некоторых случаях может играть роль политика в отношении так называемого «нулевого» сеанса.

10.5. Механизмы взаимодействия с системами Windows

Механизмы сбора информации о системах Windows можно разделить на несколько групп:

- удаленный доступ к реестру;
- доступ к административным общим ресурсам (C$, ADMINS и т. д.);
- WMI (Windows Management Interface);
- удаленный вызов процедур RPC (Remote Procedure Call).

Удаленный доступ к реестру требует административных привилегий. Кроме того, должна быть запущена соответствующая служба (рис. 10.4).

Этим способом может быть получена информация о конфигурации узла.

Использование административных общих ресурсов позволяет получить доступ к файловой системе узла и, например, определить наличие обновлений. Для того чтобы этот способ дал результаты, должны быть включены соответствующие общие ресурсы (рис. 10.5).

Что касается удаленного вызова процедур, то здесь имеется следующая особенность: возможность использования анонимного подключения, которое иногда называют «нулевым» сеансом.

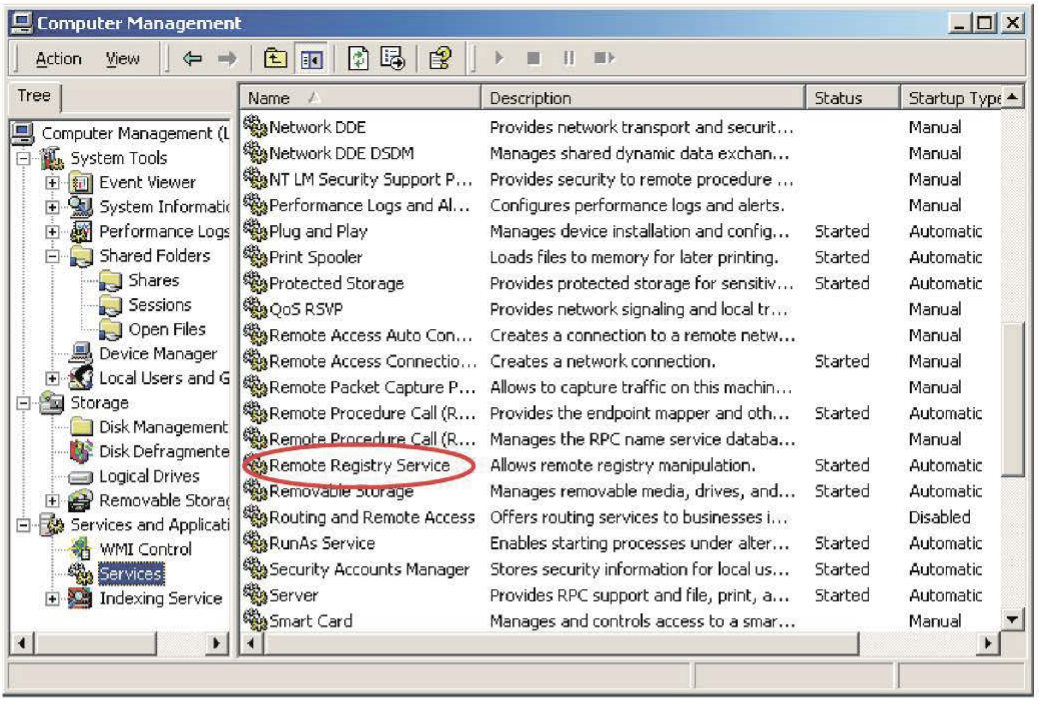


Рис. 10.4. Настройка службы удаленного доступа к реестру

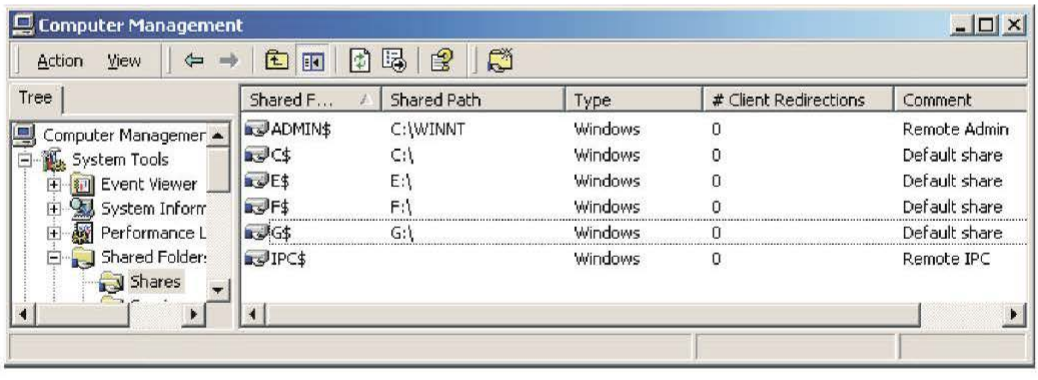


Рис. 10.5. Административные общие ресурсы

Получение информации через «нулевой» сеанс содержит следующие шаги:

1) установление TCP-соединения (порт 445 или 139);
2) установление SMB-соединения (аутентификация с пустыми именем и паролем);
3) подключение к общему ресурсу IPCS;
4) открытие какого-либо именованного канала;
5) подключение к интерфейсу DCE-RPC;
6) выполнение операций через RPC.

При этом информация может быть получена через следующие именованные каналы:

- \pipe\samr: SAM (Security Account Manager) RPC server;
- \pipe\lsarpc: LSA (Local Security Authority) RPC server;
- \pipe\netlogon: Netlogon RPC server;
- \pipe\svcctl: SCM (Service Control Manager) RPC server;
- \pipe\eventlog: Eventlog service RPC server;
- \pipe\srvsvc: Server service RPC server;
- \pipe\wkssvc: Workstation service RPC server.

Контрольные вопросы

1. Каким образом можно классифицировать проверки, встроенные в сетевые системы анализа защищенности?
2. Какими способами выполняются проверки в отношении сервера DNS? Охарактеризуйте эти способы.
3. Как можно идентифицировать уязвимости на основе анализа атрибутов файла?
4. Каким способом можно собрать информацию о системах Windows?

Глава 11. PASSIVE FINGERPRINTING

Пассивная идентификация (Passive Fingerprinting) узлов, ОС, служб и т. д. использует те же методы анализа информации, что и активная, но реализована иначе. Применяются различные способы получения информации, подлежащей анализу. Пассивный метод использует информацию, «добровольно» рассылаемую исследуемой системой. Он основан (см. выше) на следующих приемах:

- анализ сетевого трафика;
- анализ запросов от сканируемого узла.

Таким образом, суть пассивной идентификации заключается в анализе информации, доступной без непосредственного воздействия на исследуемую систему. В сканерах безопасности эти методы либо не реализованы, либо реализованы не в полной мере. Однако эти методы широко используются в сканерах безопасности для беспроводных сетей.

11.1. Анализ сетевого трафика

Использование протокола ARP. Утилита arpscan (http://ish.cx/~jason/arpscan/) после запуска прослушивает трафик и анализирует проходящие по сети arp-запросы. На их основе собирается информация об используемых IP-адресах в данном сегменте.

Используемые поля заголовков. Анализ проходящего трафика может дать информацию об ОС узлов сети, ее топологии и т. д. Для этого обычно проводится анализ отдельных полей заголовков проходящих пакетов.

Например, на рис. 11.1 изображен IP-пакет с используемыми для Passive Fingerprinting полями.

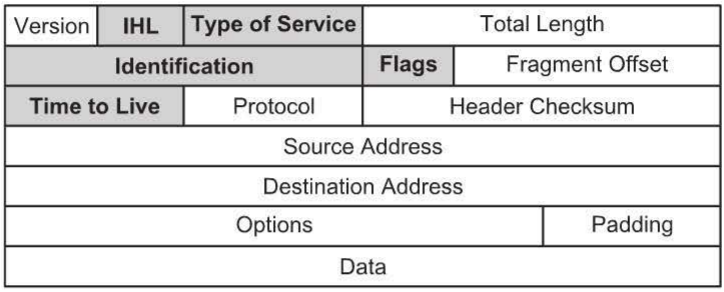


Рис. 11.1. IP-пакет с используемыми для Passive Fingerprinting полями

На рис. 11.2 приведены используемые поля TCP-сегмента.

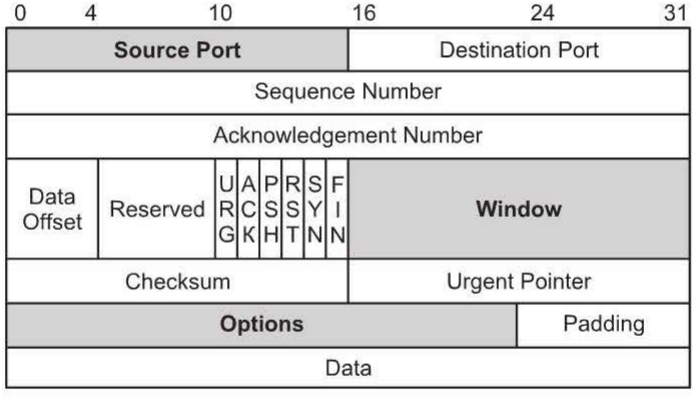


Рис. 11.2. TCP-сегмент с используемыми для Passive Fingerprinting полями

Поля, используемые для Passive Fingerprinting, приведены в таб. 11.1.

Таблица 11.1

Поля, используемые для Passive Fingerprinting

Поле	Заголовок	Назначение
TTL	IPHeader	ОС, топология
Fragmentation Flags	IPHeader	ОС
IHL	IPHeader	ОС
TOS	IPHeader	ОС

Окончание табл. 11.1

Поле	Заголовок	Назначение
IP Identification	IP Header	ОС, служба
Source Port	TCP Header	ОС, служба
Window	TCP Header / Options	ОС
Max Segment Size	TCP Header / Options	ОС
Sack OK	TCP Header / Options	ОС
Nop Flag	TCP Header / Options	ОС

Идентификация ОС. Анализируя приведенные в табл. 11.1 поля заголовка пакета, можно определить тип ОС. Ни одна из этих характеристик отдельно не определяет точно ОС. Точность определения возрастает при комплексном анализе этих полей.

Поле TTL. Обычно ОС используют следующие начальные значения TTL: 32, 64, 128, 255. При этом значение уменьшается на единицу на каждом маршрутизаторе. В большинстве случаев несложно, имея текущее значение TTL, вычислить начальное. Кроме информации о сетевой топологии таким образом может быть получена информация и об ОС узла. Пусть, например, TTL=45. Вероятно, пакет преодолел 19 маршрутизаторов, а начальное значение TTL составляло 64. Следовательно, можно предположить, что узел имеет ОС Linux или FreeBSD. Информация может быть подтверждена отслеживанием маршрута до узла с помощью утилиты traceroute (но это уже активный метод).

Поле Window. Следующий параметр — размер окна передачи (Window Size). Оно также характеризует ОС. Пусть, например, значение этого параметра равно 0x7D78. Такое значение используется по умолчанию ОС Linux.

Флаг Don't Fragment (DF) и поле TOS. Большинство систем устанавливает бит DF в пакете, поэтому применение этого значения для идентификации ограничено. Однако это облегчает идентификацию небольшого числа ОС, не устанавливающих этот бит, таких как SCO или OpenBSD. Поле TOS также малопригодно для идентификации ОС. Его значение больше зависит от используемого протокола, чем от ОС.

Инструменты. Очевидно, что необходимым инструментом для проведения такого анализа является сетевой анализатор (tcpdump, Network Monitor, Ethereal). Однако пытаться искать нужную информацию, просматривая весь перехваченный трафик, неэффективно.

В среде UNIX можно использовать комбинацию инструментов tcpdump, sort, awk, uniq для поиска в трафике необходимой информации и вывода ее в удобной для просмотра форме. Дополнительно можно воспользоваться мощным механизмом создания фильтров в tcpdump на основе типа протокола, направления и т. п.

Существуют также инструменты, автоматизирующие анализ перехваченного трафика:

- Siphon (http://siphon.datanerds.net);
- p0f (http://www.stearns.org/p0f).

Siphon использует главным образом информацию из заголовка IP (TTL, DF и т. п.). Утилита p0f использует информацию из заголовка IP и TCP-сегмента (SYN-запроса). Поскольку утилита Siphon практически не развивается и в настоящее время не является актуальной, далее рассматривается утилита p0f.

Утилита p0f работает на многих платформах, в том числе и в среде Windows. Для работы в среде UNIX требуется libpcap, для работы в среде Windows — WinPcap и Cygwin. Определение ОС осуществляется на основе информации из файла p0f.fp.

Параметры запуска утилиты (табл. 11.2):

```
p0f [-f file] [-i device] [-o file] [-s file] [-vKUtq]
['filter rule']
```

Таблица 11.2

Формат утилиты p0f

Опция	Описание	Комментарий (пример использования)
-ffile	Указание файла с информацией об операционных системах	p0f -f c:\my_oses\p0f.fp
-i	Сетевой адаптер для сбора трафика	—
-sfile	Указание файла с перехваченным трафиком для анализа	—
-ofile	Указание файла для размещения результатов работы	Лучше всего использовать с опциями -vt
-v	Отладочный режим	—
-t	Добавление времени обнаружения	—

Кроме того, можно просматривать не весь трафик, а только его часть. Это достигается с помощью фильтров, например:

```
p0f 'src host 1.2.3.4'
```

Просмотр содержимого пакетов. Для получения информации, большей, чем ОС и используемые службы (например, о возможных уязвимостях), требуется анализ содержимого пакетов (дополнительно к заголовкам). Хотя эта задача решается системами обнаружения атак, простейший анализ можно провести, пользуясь утилитой ngrep (http://ngrep.sourceforge.net). Ее синтаксис напоминает синтаксис tcpdump.

11.2. Анализ запросов от сканируемого узла

Другой способ для пассивного сбора информации — анализ информации, получаемой серверными приложениями исследующей системы от различных удаленных клиентов.

ICMP Echo-request. Пакет ICMP Echo-request, посылаемый, например, с помощью утилиты ping, содержит произвольные данные, на которые узел отвечает пакетом ICMP Echo Reply, содержащим те же самые данные. С точки зрения пассивного сбора информации интерес представляет способ формирования данных, заполняющих пакет. Разные ОС используют различное наполнение. Так, например, ОС Windows заполняет содержимое пакета строчными символами латинского алфавита («abcde...xyzabcd...»), а ОС RedHat Linux заполняет запрос цифрами и специальными символами. Это различие может быть использовано для идентификации ОС узла, от которого пришел запрос.

HTTP-запрос. Протокол HTTP позволяет серверу получить некоторую информацию о клиенте, основываясь, главным образом, на составе и порядке заголовков в запросе (несущих вспомогательную информацию). Например, заголовок User-Agent содержит информацию об используемом браузере. Иногда этот заголовок содержит информацию и об ОС клиента.

Более подробную информацию о клиенте сервер может получить, отослав клиенту html-документ, содержащий специальный javascript-код, определяющий необходимые серверу параметры и возвращающий их серверу, используя, например, механизм CGI. Однако подобная техника не считается полностью пассивной, так как может быть обнаружена на стороне клиента.

FTP-клиент. Протокол FTP также позволяет серверу достаточно точно определить клиентское ПО. При успешном соединении клиент в начале ftp-сессии использует некоторые из следующих команд: AUTH, USER, PASS, PWD, PORT, SYST, EPSV, PASV, LIST, CWD. Состав команд и их порядок позволяют различать многих клиентов. В табл. 11.3 представлены некоторые из них.

Таблица 11.3

Состав команд и их порядок

№ п/п	Клиент	Команды
1	Linux-клиент	AUTH, USER, PASS, SYST, PORT
2	Windows-клиент	USER, PASS, PORT
3	Клиент, встроенный в Far	USER, PASS, PWD
4	FreeBSD-клиент	USER, PASS, SYST, EPSV
5	MSIE	USER anonymous, PASS IEUser@, TYPE I, PASV, CWD
6	Go!Zilla	USER anonymous, PASS gozilla@anon.com, PASV, LIST
7	ReGet	USER anonymous, PASS User@x-x-x-xxx.ReGet.Com, SYST

Протокол Telnet. При установлении соединения по протоколу Telnet происходит согласование определенных параметров между серверной и клиентской сторонами. Различные реализации имеют разные наборы параметров и их порядок при согласовании, что позволяет идентифицировать клиентское ПО.

Электронная почта (SMTP и POP3). Служебные заголовки сообщений электронной почты содержат подробную информацию об отправителе и процессе пересылки письма. В заголовках всегда имеется IP-адрес или имя узла — отправителя письма. Рассмотрение таких полей, как Message-ID, X-Mailer, User-Agent, дает возможность определить клиентское ПО, использованное при написании и отсылке письма (вплоть до номера версии), и часто ОС клиента, например:

- Message-ID: (это Linux, Pine v4.10);
- X-Mailer: QUALCOMM Windows Eudora Version 4.3.2;
- X-Mailer: Microsoft Outlook Express 5.00.3018.1300.

Таким образом, механизм Passive Fingerprinting может быть использован в следующих случаях:

- для сбора информации о сети при проведении анализа защищенности внутренней сети с использованием методологии Penetrationtesting;
- обнаружения неизвестных устройств в сети;
- инвентаризации ресурсов сети (узлов, ОС, служб) без влияния на производительность.

Контрольные вопросы

1. В чем заключается суть механизма Passive Fingerprinting?
2. Как происходит анализ сетевого трафика с использованием пассивных методов?
3. Каким образом осуществляется пассивный сбор информации на основе анализа данных различных удаленных клиентов?

Глава 12. ВЫЯВЛЕНИЕ УЯЗВИМОСТЕЙ С ПОМОЩЬЮ ТЕСТОВ

Наиболее понятный и очевидный способ поиска какой-либо уязвимости — попытаться использовать ее, т. е. имитация атаки, ее использующей. Согласно приведенному выше определению, этот способ называется *тестированием*. Применение данного способа имеет определенные сложности, в частности при оценке:

- результатов тестирования;
- влияния тестирования на исследуемую систему.

12.1. Эксплойты и их разновидности

При проведении тестирования в отношении узла (службы, работающей на узле) запускаются реальные атаки. Они называются также exploit check и выполняются с помощью так называемых эксплойтов — программ (утилит), использующих уязвимость.

Эксплойт (от англ. exploit) — это документированный метод или программа (сценарий), использующие уязвимость. Многие общедоступные базы уязвимостей (например, www.securityfocus.com) содержат инструкции или код для использования большинства обнаруженных уязвимостей.

Существуют три основных разновидности программ эксплойтов:

- программы, использующие технику запуска произвольного кода на узле — объекте атаки;
- простые эксплойты;
- инструменты, выполняющие удаленный подбор пароля (bruteforce-tools).

12.2. Использование техники запуска кода

Общие сведения. Причина переполнения буфера — ошибки программирования (реализации). Обычно такие ошибки могут быть использованы нарушителем двумя способами:

- запуск кода, выполняющего какие-либо действия, например, предоставление удаленного «шелла» с правами суперпользователя;
- выведение узла из строя. Обычно используется в отношении систем Windows, поскольку для них сложнее написание кода, выполняющего определенные действия.

Пример эксплойта первого типа — программа kaht2 (см. практическую работу 1). Эта программа использует уязвимость CAN-2003-0352. Запущенная в отношении узла, имеющего данную уязвимость, она предоставляет удаленный шелл с правами администратора.

Пример эксплойта второго типа — программа SMBdie.exe, использующая уязвимость CAN-2002-0724. Описание данной программы в каталоге CVE:

CAN-2002-0724
Phase: Proposed (20020830)
Reference: BUGTRAQ:20020822 CORE-20020618: Vulnerabilities in Windows SMB (DoS)
Reference: URL:http://marc.theaimsgroup.com/?l=bugtraq&m=103011556323184&w=2
Reference: MS:MS02-045
Reference: URL:http://www.microsoft.com/technet/security/bulletin/ms02-045.asp
Reference: CERT-VN:VU#311619
Reference: URL:http://www.kb.cert.org/vuls/id/311619
Reference: CERT-VN:VU#342243

Reference: URL:http://www.kb.cert.org/vuls/id/342243
Reference: CERT-VN:VU#250635
Reference: URL:http://www.kb.cert.org/vuls/id/250635
Description:
Buffer overflow in SMB (Server Message Block) protocol in Microsoft Windows NT, Windows 2000, and Windows XP allows attackers to cause a denial of service (crash) via a SMB_COM_TRANSACTION packet with a request for the (1) NetShareEnum, (2) NetServerEnum2, or (3) NetServerEnum3, aka "Unchecked Buffer in Network Share Provider Can Lead to Denial of Service".
Votes:
ACCEPT(5) Baker, Wall, Foat, Cole, Armstrong
MODIFY(1) Frech
NOOP(2) Christey, Cox
Voter Comments:
Christey> XF:win-smb-packet-bo(9933)
URL:http://www.iss.net/security_center/static/9933.php
BID:5556
URL:http://www.securityfocus.com/bid/5556
Frech> XF:win-smb-packet-bo(9933)

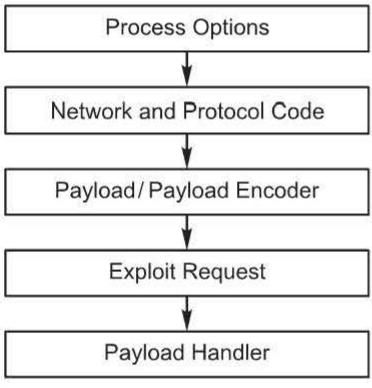


Рис. 12.1. Структура программы-эксплойта, предоставляющего нарушителю доступ к узлу

Структура эксплойта. Структура программы-эксплойта, предоставляющего нарушителю доступ к узлу, приведена на рис. 12.1.

Exploit Options — компонент, отвечающий за обработку введенных пользователем данных, например адреса цели и других опций.

Network and Protocol Code — компонент, отвечающий за сетевое соединение с узлом, разрешение имени, создание сокета и т. п.

Самая важная часть эксплойта — полезная нагрузка, которая и является кодом, который должен быть выполнен. Чаще всего полезная нагрузка представляет собой так называемый шелл-код. Таким образом, после создания ситуации переполнения буфера адрес возврата должен быть таким, чтобы он указывал на данный шелл-код для передачи ему управления.

Срабатывание эксплойта обеспечивает Exploit Request. Собственно, здесь формируется запрос, вызывающий ситуацию переполнения буфера.

Payload Handler — обработчик полезной нагрузки, выполняющий такие операции, как связывание оболочки с сокетом и т. п.

Следует отметить, что эксплойты имеют схожие структуры.

Metasploit Framework. Это законченная среда для написания, тестирования и использования кода эксплойтов (http://www.metasploit.com/). Такая среда позволяет загружать различную полезную нагрузку для использования с эксплойтами.

12.3. Простые эксплойты

К этой группе относят эксплойты, использующие уязвимости, имеющиеся преимущественно в Web-based системах. Передача с помощью обычного браузера специальным образом подобранной строки и получение, например, содержимого диска web-сервера — пример такого эксплойта:

CVE: CAN-2000-0886

Запрос, построенный подобным образом:

```
http://site/scripts/test.bat"+&+dir+c:/+.com
```

позволяет просмотреть содержимое диска С сервера HTTP под управлением Windows.

12.4. Удаленный подбор пароля

Удаленный подбор пароля к сетевым службам может быть эффективен в достаточно быстрой сети при наличии достаточного времени. Например, при выполнении теста внутренней сети на устойчивость к взлому распространенным приемом является подбор пароля к общим ресурсам Windows. Известные инструменты для этой цели: ADMsmb, NAT, Legion.

Протокол SNMP использует для аутентификации строку community, которая также может служить мишенью для подбора по сети. Инструментом является ADMsnmp.

Один из самых известных и популярных инструментов для удаленного подбора пароля — программа Brutus (www.hoobie.net/brutus). С ее помощью можно осуществлять удаленный подбор пароля к следующим службам:

- HTTP;
- FTP;
- SMB;
- POP3;
- Telnet.

Имеется возможность добавления других служб. Для подбора пароля можно использовать следующие методы:

- по словарю;
- полным перебором;
- комбинированный.

12.5. Оценка стойкости паролей

Оценка стойкости паролей — важная составляющая сканирования как на уровне сети, так и на уровне узла. При выполнении тестирования на устойчивость к взлому часто выполняются попытки взлома паролей, при выполнении аудита внутренней сети — проверка достаточной стойкости паролей у пользователей.

Пароли обычно не хранятся в открытом виде, вместо этого хранятся их хэши. Когда пользователь вводит пароль при входе в систему, генерируется хэш и сравнивается с тем, который хранится. Задача оценки стойкости паролей сводится к тому, чтобы по хэшу восстановить пароль. Хэши паролей могут быть получены следующим образом:

• перехвачены в момент передачи по сети. Пароли некоторых сетевых служб (например, POP3) передаются в открытом виде;

• получены из места их хранения в сканируемой системе. Это, например, может быть файл, содержащий пароли (/etc/passwd, sam и т. д.). Обычно в этом случае требуется доступ к узлу с правами администратора.

После получения хэшей паролей следует процесс восстановления паролей различными методами:

• атака по словарю (dictionaryattack). Это наиболее быстрый способ, при котором используются наиболее распространенные слова из словаря (текстового файла). В сети Интернет можно найти различные словари для разных языков;

• гибридная атака (hybridattack). В этом случае к словам из словаря добавляются подстановки последовательностей букв или цифр (password1, password2), иногда буквы слова из словаря заменяются цифрами или специальными символами (micro$oft, 40in);

• атака последовательным перебором (bruteforce). Это наиболее надежный способ получения паролей, поскольку он предполагает перебор всех вариантов. Теоретически любой пароль может быть получен таким методом, но на практике для этого может потребоваться значительное время. Однако часто это время меньше того, которое установлено политикой безопасности для смены пароля. Кроме того, задача восстановления пароля может быть распределена по нескольким узлам.

Оценка стойкости паролей обычно выполняется сканерами уровня узла, поскольку в этом случае можно легко получить доступ к хранилищу хэшей паролей.

Для сканеров сетевого уровня эта задача превращается в попытки удаленного подбора паролей (по сети). Этот способ имеет следующие недостатки:

• низкую скорость перебора (так как требуется сетевое подключение);

• возможность блокировки учетных записей пользователей.

Довольно обширные возможности по удаленному подбору паролей имеет программа ShadowScan (http://www1.rsh.kiev.ua/downe.htm). Как видно на рис. 12.2, программа осуществляет подбор паролей по словарю для служб FTP, POP3 и др.

Следует отметить, что в сканерах сетевого уровня подбор паролей обычно ограничивается именами и паролями по умолчанию (наиболее распространенными комбинациями).

Рис. 12.2. Подбор паролей по словарю для служб FTP, POP3 в программе ShadowScan

12.6. Тестирование

Обычно тестирование отличается от запуска «настоящего» эксплойта тем, что в качестве результата возвращается какой-либо код (например, «система уязвима») вместо, например, удаленного шелла. Часто разработчики сканеров предоставляют простые утилиты (фактически, являющиеся эксплойтами) для тестирования узла на наличие той или иной уязвимости. Например, на сайте компании ISS можно найти утилиты для проверки узлов на наличие уязвимостей ms03-039 и ms03-043:

http://www.iss.net/support/product_utilities/Xfrpcss.php

http://www.iss.net/support/product_utilities/ms03-043/

Если результатом запуска эксплойта является так называемый удаленный шелл, то сделать заключение о наличии уязвимости можно сразу же. Другая категория эксплойтов оставляет систему в уязвимом состоянии (например, добавляет пользователя в группу Administrators или дописывает знак «+» в файл rhosts). В этом случае необходимо проделать отдельную операцию для того, чтобы убедиться в наличии уязвимости (например, выполнить подключение к узлу). Следовательно, тесты можно разделить на две группы:

- тесты, результаты которых видны сразу (directly observed exploitation);
- тесты, просмотр результатов которых требует отдельного действия (indirectly observed exploitation).

Для тестов первой группы, чтобы сделать вывод о наличии уязвимости, достаточно того же подключения к узлу, которое использовалось для проведения тестирования. Вторая группа тестов требует отдельного соединения или серверной части на сканирующем узле для подключения со стороны сканируемого узла.

Иногда довольно сложно отнести тест к одному из двух указанных типов, например, проверки с помощью протоколов ICMP или UDP.

12.7. Анализ результатов

В некоторых случаях результаты тестирования доступны, но автоматизировать их анализ затруднительно, например при поиске уязвимостей CGI-сценариев.

В других случаях результаты тестирования доступны сразу, но в то же время их можно получить и с помощью отдельной операции. Характерный пример — обнаружение уязвимости Sun Telnet DoS Attack (посылка потока символов ^D на узел с операционной системой Solaris на Telnet-порт). Определить результат можно, подключившись к узлу еще раз (отдельная операция) или с помощью анализа задержек пакетов с флагом ACK, относящихся к тому же соединению (рис. 12.3).

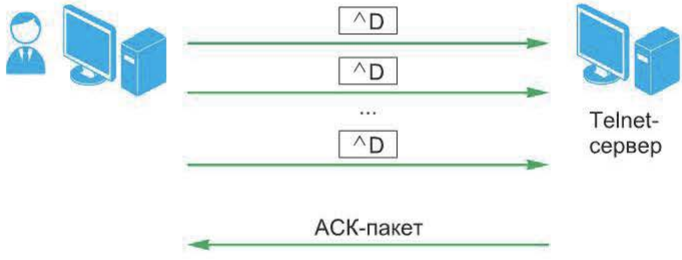


Рис. 12.3. Обнаружение уязвимости Sun Telnet DoS Attack

Если достоверность результатов, полученных в результате тестирования (после выполнения отдельного подключения), невысока, можно выполнить проверку несколько раз.

Наконец, многие тесты требуют анализа задержек. Например, DoS-атаки, направленные на службы NT/2000, приводят к повышенному расходу ресурсов процессора. Результаты таких тестов могут быть получены только с помощью анализа временных характеристик.

Большое число различных реализаций служб делает тестирование довольно сложной задачей. Например, некоторые службы по-разному отвечают на стандартные запросы. Некоторые тесты могут вызвать выведение системы из строя. Например, некоторые сетевые принтеры HP могли быть выведены из строя при обычном сканировании портов.

12.8. Отказ в обслуживании

Отдельного рассмотрения требует задача тестирования узлов на устойчивость к «отказу в обслуживании». В общем случае задача сводится к тому, чтобы после проведения тестирования попытаться подключиться на требуемый порт и убедиться в том, что подключение невозможно. После этого делается вывод о наличии уязвимости. При этом возникает ряд затруднений, например влияние межсетевых экранов и систем обнаружения атак. Довольно часто системы обнаружения атак настраиваются таким образом, что при обнаружении DoS-атаки проводится реконфигурация межсетевого экрана, так что последующие подключения со стороны сканирующего узла становятся невозможны. В этом случае определить причину недоступности системы сложно. Эта проблема влияет на тесты, просмотр результатов которых требует выполнения отдельной операции.

Некоторые тесты приводят к выведению из строя всей системы вместо отдельной службы. В этом случае возникает вопрос, какова реальная причина выведения системы из строя и связана ли она с тестированием.

Контрольные вопросы

1. Каковы особенности выявления уязвимостей с помощью тестов?
2. Что такое exploit check?
3. Как проводится оценка стойкости паролей?
4. В чем отличие тестирования от запуска «настоящего» эксплойта?
5. В чем заключается особенность тестирования узлов на устойчивость к «отказу в обслуживании»?

Глава 13. СЕТЕВОЙ СКАНЕР NESSUS

Выше были приведены основные принципы анализа защищенности, методы тестирования различных служб, примеры инструментов для проведения тестов. Сканеры безопасности, рассматриваемые далее, сочетают в себе возможности отдельных инструментов и реализуют различные методы сканирования.

13.1. Обзор возможностей сканера

Nessus — сканер уязвимостей, который может быть использован для сканирования одного или нескольких узлов сети. Это свободно распространяемый инструмент сканирования с регулярно обновляемой базой проверок. Рассмотрим его основные характеристики и возможности.

Модульная архитектура. Каждая проверка, выполняемая сканером, представляет собой внешний модуль (plugin). Это позволяет легко добавлять

новые проверки. Полный список имеющихся проверок может быть найден по адресу http://cgi.nessus.org/plugins

Язык NASL. Сканер имеет встроенный язык описания проверок — NASL (Nessus Attack Scripting Language), позволяющий создавать пользовательские проверки. Проверки могут быть написаны и на языке C.

Регулярно обновляемая база проверок. Проверки новых уязвимостей могут добавляться каждый день.

Распределенная архитектура (клиент/сервер). Сканер состоит из двух частей: сервера, выполняющего проверки, и клиента, предоставляющего пользовательский интерфейс. Эти части могут быть распределены по нескольким узлам. Таким образом, можно выполнять сканирование большой сети, управляя процессом с одного рабочего места.

Параллельное сканирование нескольких узлов сети. Возможности по одновременному сканированию ограничены лишь производительностью узла, на котором запущена серверная часть сканера.

Идентификация служб. Сканер учитывает, что службы могут использовать нестандартные порты.

Сканирование нескольких одинаковых служб, находящихся на одном узле. Если, например, на узле имеются два web-сервера (использующие разные номера портов), сканер будет выполнять проверки в отношении каждого из них.

Система генерации и экспорта отчетов. Отчеты содержат подробное описание уязвимостей и рекомендации по их устранению. Клиентская часть сканера имеет возможности экспорта отчетов в различные форматы (ASCIItext, LaTeX, HTML, HTML с графикой).

Динамическое подключение/выключение проверок. Сканер использует результаты уже осуществленных проверок для выполнения следующих. В зависимости от результатов этапа сбора информации ненужные проверки могут быть выключены.

13.2. Архитектура сканера

Сканер состоит из двух частей: Nessus-сервер и Nessus-клиент (рис. 13.1). Это сканер сетевого уровня, выполняющий дистанционные проверки. Серверная часть может работать на любой UNIX-платформе (FreeBSD, Linux, BSDI, Solaris и др.). Клиентская часть работает на различных платформах (имеются клиенты для X11 и Win32).

В зависимости от задач и сетевого окружения можно предложить различные варианты расположения серверных и клиентских частей сканера Nessus. Например, удобно переносной компьютер с клиентской и серверной частями сканера подключать к различным участкам сети и проводить сканирование. Для сканирования различных участков сети с разных точек зрения целесообразно установить несколько серверных частей сканера и управлять ими с помощью одного клиента (рис. 13.2).

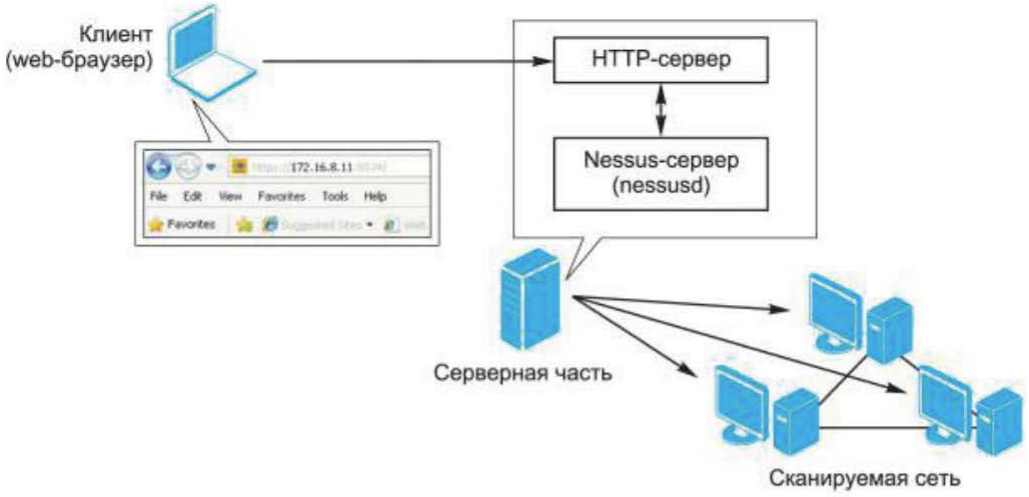


Рис. 13.1. Архитектура сканера Nessus

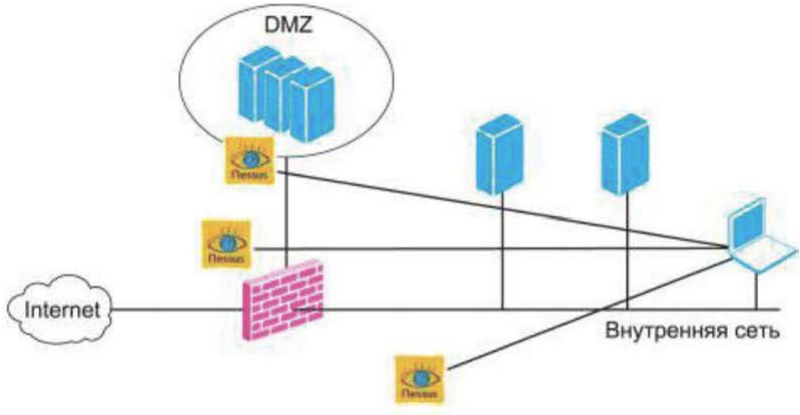


Рис. 13.2. Схема расположения нескольких серверных частей сканера Nessus с одним клиентом

При проведении сканирования несколькими пользователями можно использовать одну серверную часть сканера, подключаясь к ней с различных клиентов.

Для защиты взаимодействия между клиентской и серверной частями сканера используется SSL.

13.3. Получение и установка сканера

Получение дистрибутива. Дистрибутив сканера Nessus может быть получен по адресу: http://www.nessus.org в виде сценария nessus-installer.sh. Дополнительно потребуется скачать файл MD5.txt, содержащий контрольную сумму. Эти два файла необходимо поместить в какой-либо каталог

(например, /root/soft/nessus) на том узле, где планируется развернуть серверную часть. По этому же адресу может быть получен и клиент Win32.

Установка сканера. Далее необходимо перейти в каталог с полученным дистрибутивом сканера и запустить установку командой: sh nessus-installer.sh. В процессе установки потребуется ответить на ряд вопросов, и через некоторое время серверная часть (а также клиент для X11) будет установлена. После установки к переменной $PATH будут добавлены записи: /usr/local/bin и /usr/local/sbin.

Для проверки можно воспользоваться командой echo $PATH.

Кроме того, в файл /etc/ld.so.conf необходимо добавить строку /usr/local/lib, а затем запустить ldconfig (также в процессе установки).

После установки необходимо выполнить ряд действий по подготовке серверной части:

• создать сертификат сервера командой nessus-mkcert. Это необходимо для защиты взаимодействия между клиентом и сервером;

• добавить пользователя для подключения со стороны клиента (локального или удаленного) командой nessus-adduser.

Удаление сканера. Для удаления сканера необходимо выполнить команду uninstall-nessus

Затем, возможно, потребуется вручную удалить некоторые каталоги и файлы (можно воспользоваться поиском по ключевому слову nessus).

Запуск сканера. Для запуска сканера необходимо запустить серверную и клиентскую части. Серверная часть сканера может быть запущена командой #nessusd-D При этом параметры задаются в файле /usr/local/etc/nessus/nessusd.conf

Для запуска интерфейса сканера (должна быть загружена графическая оболочка) необходимо в командной строке набрать #nessus

Обновление базы проверок можно проводить двумя способами.

Обновление вручную. Для подключения новых внешних модулей необходимо:

• скопировать их с web-сервера http://www.nessus.org в каталог .../nessus/lib/nessus/plugins;

• завершить выполнение процесса nessusd: kill -9 <pid>, где pid — идентификатор процесса nessusd;

• вновь запустить сервер nessusd с помощью команды nessusd —D.

После запуска сервера новые модули будут доступны для проведения проверок с их использованием.

Обновление автоматически через Интернет. Для этого на узле должны быть установлены: Lynx, tar, gzip.

Обновление осуществляется с помощью команды nessus-update-plugins. При этом выполняется подключение к сайту www.nessus.org и скачивание новых проверок.

Если подключение осуществляется через proxy-сервер, потребуется создать файл в домашнем каталоге пользователя с именем .nessus-update-

pluginsrc, где и указать необходимые параметры. Например, файл может содержать следующие строки:

```
proxy_user=root
proxy_passwd=qwerty
proxy=200.4.4.254:8080
```

13.4. Работа со сканером

Сбор информации о сканируемой сети. Вне зависимости от выбранного варианта сканирования (тестирование на устойчивость к взлому снаружи или аудит внутренней сети) необходим сбор начальной информации о сканируемой сети. Для этого можно использовать рассмотренные выше способы и средства, но, как правило, сканеры уязвимостей имеют встроенные возможности инвентаризации сетевых ресурсов. Рассмотрим возможности сканера Nessus по идентификации узлов сети и служб.

Идентификация узлов. Сканер Nessus поддерживает четыре способа идентификации узлов (рис. 13.3):

- ARP Ping;
- ICMP Ping;
- TCP Ping;
- UDP Discovery.

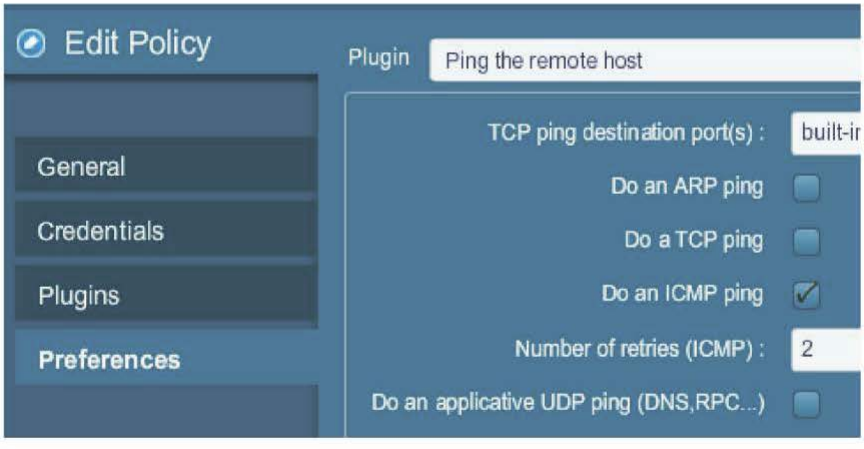


Рис. 13.3. Способы идентификации узлов сканером Nessus

Сканирование портов. Следующий шаг — поиск работающих на узле сетевых служб. Это типичный шаг для большинства сканеров уязвимостей. Следует понимать, что переход к этому шагу сканер осуществляет лишь в том случае, если узел по результатам предыдущего шага был признан доступным.

Сканер Nessus использует два метода сканирования портов:

- TCP Scan — сканирование с установлением соединения;
- SYN Scan — сканирование без установления соединения.

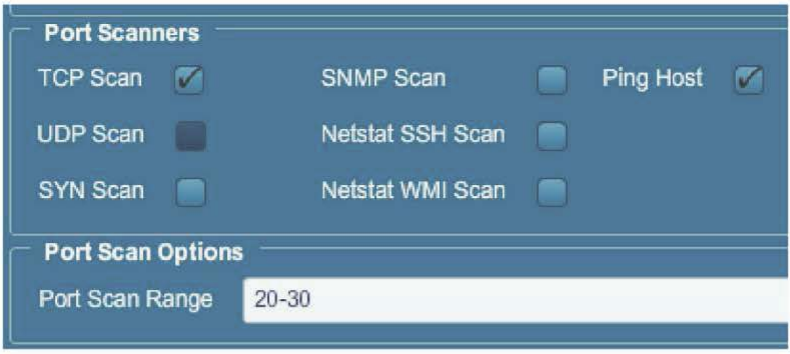


Рис. 13.4. Закладка Port Scanners

Оба эти метода рассматривались выше. Их можно включить через закладку Port Scanners (рис. 13.4).

Для проведения сканирования портов UDP необходимо использовать опцию UDP Scan.

Результат данного этапа — список открытых портов на сканируемом узле (найденных на основе указанных настроек).

Идентификация сервисов и приложений. Идентификация сервисов и приложений (рис. 13.5) осуществляется плагинами Service Detection:

- Service Detection;
- Service Detection (2nd Pass);
- Service Detection (3 ASCII Digit Code Responses).

В этой же группе представлены проверки для идентификации приложений, например для FTP-сервера (рис. 13.6).

Рис. 13.5. Идентификация сервисов и приложений

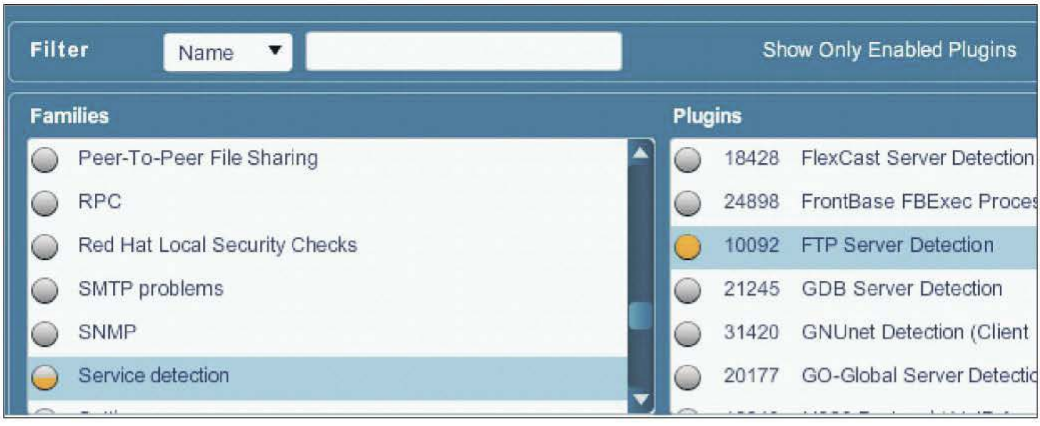


Рис. 13.6. Проверки для идентификации приложений

Идентификация ОС. Проверки для идентификации ОС находятся в группе General (рис. 13.7).

Проверка с помощью OS Identification позволяет сделать окончательный вывод об ОС узла на основе результатов других проверок. Методы, используемые другими проверками, понятны по их названиям. В основном это методы, базирующиеся на использовании сервисов прикладного уровня.

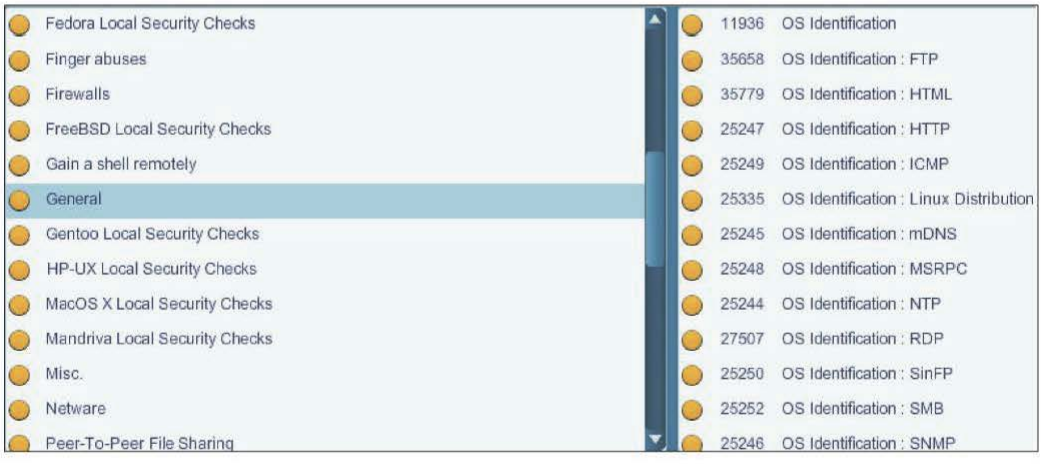


Рис. 13.7. Проверки для идентификации ОС

Выбор проверок. Последний шаг — выбор проверок для сканирования узла. Их можно включить (выключить) в секции Plugins (рис. 13.8).

Управление типами проверок. Выбрав необходимые проверки и выполнив сканирование, можно получить список выявленных уязвимостей. Однако при этом пользователь сталкивается со следующими проблемами:

• Falsepositives (присутствие в списке уязвимостей, реально отсутствующих на сканируемом узле);

Рис. 13.8. Проверки для сканирования узла

• Falsenegatives (пропущенные, невыявленные уязвимости, присутствующие в системе).

Для проведения анализа результатов работы сканера следует учитывать, что проверки делятся на две категории: определяющие уязвимость по косвенным признакам и предпринимающие попытки атак в процессе сканирования. Последние считаются потенциально опасными и могут привести к выведению сканируемого узла из строя. В сканере Nessus для выключения механизма активного анализа предусмотрена опция Safe Checks (рис. 13.9).

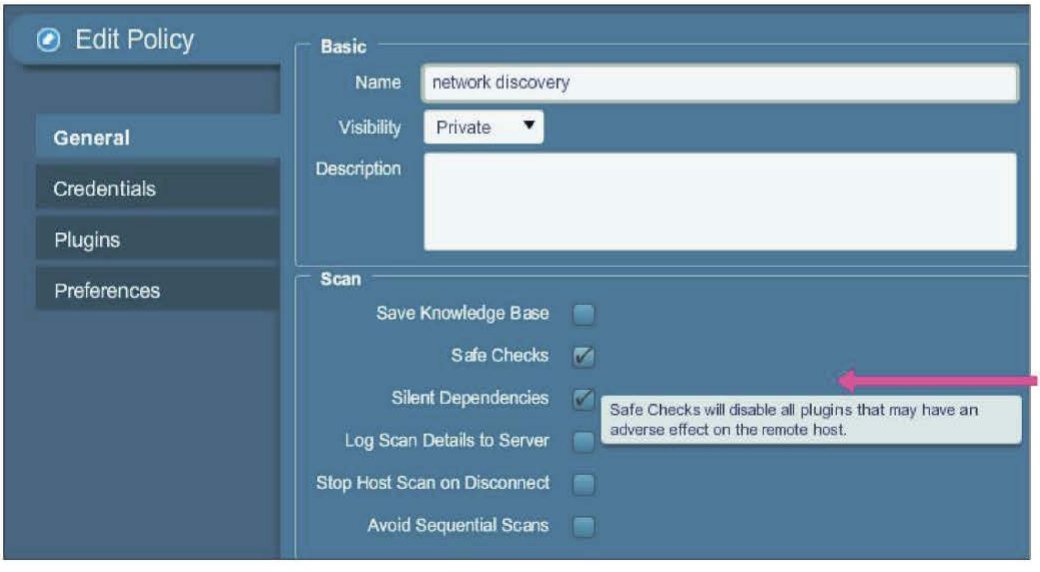


Рис. 13.9. Выключение механизма активного анализа

При включенном режиме Safe Checks сканер делает вывод об отсутствии или наличии уязвимости только на основе собранной информации (номера версий, баннеры и т. п.). В таком режиме могут появляться ложные срабатывания.

Контрольные вопросы

1. Каковы возможности сканера Nessus?
2. Опишите архитектуру сканера Nessus.
3. Каков порядок получения и установки сканера Nessus?
4. Опишите порядок работы сканера Nessus.

Глава 14. ЯЗЫК ОПИСАНИЯ АТАК NASL

Язык написания сценариев атак NASL (Nessus Attack Scripting Language) разработан специально для сетевого сканера Nessus. Он позволяет быстро создавать сценарии (скрипты) с целью выявления уязвимостей узлов сети. Для версии сканера 2.0 язык NASL был полностью переписан и получил название NASL2. Тесты для сканера Nessus могут быть также написаны и на языке C.

Отметим преимущества использования языка NASL:

- оптимизация для сканера Nessus;
- сходство с языком C;
- безопасность;
- простота модификации;
- переносимость.

К недостаткам языка NASL можно отнести отсутствие поддержки структур и средства отладки (существует выделенный интерпретатор NASL).

14.1. Структура сценария

Общие сведения. После установки сценарии находятся в каталоге /usr/local/lib/nessus/plugins. Файлы сценариев имеют расширение .nasl, например, account_lp.nasl. Структура требует наличия двух секций: регистрации (registersection) и описания атаки (attacksection).

Пример заготовки для сценария:

```
#
# Сценарий Nasl
#

if(description)
{
#
#
# Секция регистрации
#
#

exit(0);
}
```

```
#
# Секция описания атаки
#
#
```

Переменная description — это глобальная переменная (флаг регистрации), принимающая значения TRUE или FALSE.

Секция регистрации. Эта секция должна содержать вызовы следующих процедур:

- **script_name(language1:<name>, [...])** — задает имя сценария, которое будет отображаться в соответствующем окне графического интерфейса клиентской части;
- **script_description(language1:<desc>, [...])** — задает описание сценария, отображающееся в соответствующем окне (Description) графического интерфейса клиента;
- **script_summary(language1:< summary>, [...])** — задает краткую аннотацию о сценарии, отображающуюся в контекстных подсказках (tooltips). Аннотация должна умещаться на одной строке;
- **script_category(<category>)** — задает категорию сценария из списка, приведенного в табл. 14.1;

Таблица 14.1

Категории и описание сценариев

№ п/п	Категория	Описание
1	ACT_ATTACK	Сценарий пытается получить доступ к тестируемому узлу
2	ACT_MIXED_ATTACK	Сценарий пытается выполнить атаку с возможным выведением из строя службы на тестируемом узле
3	ACT_DESTRUCTIVE_ ATTACK	Сценарий может вызвать разрушение данных на тестируемом узле (например, при переполнении буфера)
4	ACT_GATHER_INFO	Сценарий запускается одним из первых и направлен на сбор информации. В частности, он не должен оказывать серьезного влияния на тестируемый узел, например, приводить к выведению из строя
5	ACT_DENIAL	Сценарий явно направлен на выведение из строя какого-либо сервиса тестируемого узла
6	ACT_SCANNER	Этот сценарий сканирует определенный порт
7	ACT_INIT	Сценарий создает несколько записей в базе знаний (в виде глобальных переменных) для использования их другими сценариями

Окончание табл. 14.1

№ п/п	Категория	Описание
8	ACT_SETTINGS	Похожа на категорию ACT_INIT, но сценарии данной категории запускаются только после сценариев категории ACT_SCANNER, т. е. после подтверждения доступности узла
9	ACT_KILL_HOST	Сценарий может привести к выведению узла из строя

• **script_copyright(language1:<copyright> , [...])** — задает информацию о защите авторских прав на сценарий (имя автора или что-либо другое);

• **script_family(language1:<family>, [...])** — задает группу (семейство), к которой принадлежит сценарий. Несмотря на то что можно зарегистрировать сценарий в собственной группе, более предпочтительно отнести его к одной из существующих групп:

- Backdoors;
- CGI abuses;
- CISCO;
- Denial of Service;
- Finger abuses;
- Firewalls;
- FTP;
- Gain a shell remotely;
- Gain root remotely;
- General;
- Misc;
- Netware;
- NIS;
- Ports scanners;
- Remote file access;
- RPC;
- Settings;
- SMTP problems;
- SNMP;
- Untested;
- Useless services;
- Windows;
- Windows : User management.

Большинство из приведенных выше функций требует указания аргумента language1. При этом точный синтаксис имеет следующий вид:

script_function(english:english_text, [francais:french_text, deutsch:german_text, ...]).

Кроме приведенных функций может быть использована функция script_dependencies (), позволяющая запускать данный сценарий после какого-либо другого. Это эффективно в том случае, когда возникает необходимость использования одним сценарием результатов другого сценария, помещенных в базу знаний. Функция имеет следующий вид: script_dependencies(filename1 [,filename2, ..., filenameN]), где filename — имя запускаемого сценария. Запуск осуществляется в порядке, соответствующем порядку аргументов функции script_dependencies().

Секция описания атаки. Эта секция содержит операторы, необходимые для реализации атаки. После завершения атаки оповещение о найденных уязвимостях в системе безопасности может быть осуществлено с помощью функций:

- security_info();
- security_warning();
- security_hole().

Первая и вторая функции могут быть использованы в тех случаях, когда обнаруженная уязвимость не имеет критически важного значения. Для того чтобы подчеркнуть опасность обнаруженной уязвимости, используется третья функция. Синтаксис функций следующий:

- security_warning(<port> [, protocol:<proto>]);
- security_hole(<port> [, protocol:<proto>]);

или

- security_warning(port:<port>, data:<data> [, protocol:<proto>]);
- security_hole(port:<port>, data:<data> [, protocol:<proto>]).

В первом случае информация, отображаемая на клиентской стороне, содержит описание сценария, созданное с помощью функции script_description().

Во втором случае клиент отображает аргумент data. Это удобно при необходимости оперативно отображать полученную при анализе защищенности информацию.

Пример сценария, выполняющего проверку наличия SSH на тестируемом узле:

```
#
# Check for ssh
# Секция регистрации
if(description)
{

script_name(english:"Ensure the presence of ssh");
script_description(english:"This script makes sure that
ssh is running");
script_summary(english:"connects on remote tcp port
22");
script_category(ACT_GATHER_INFO);
```

```
script_family(english:"Administration toolbox");
script_copyright(english:"This script was written by
Joe U.");
script_dependencies("find_service.nes");
exit(0);
}

# Секция описания атаки
# First, ssh may run on another port.
# That's why we rely on the plugin 'find_service'
#

port = get_kb_item("Services/ssh");
if(!port)port = 22;
# declare that ssh is not installed yet
ok = 0;
if(get_port_state(port))
{
soc = open_sock_tcp(port);
if(soc)
{
# Check that ssh is not tcpwrapped. And that it's
really
# SSH
data = recv(socket:soc, length:200);
if("SSH" >< data)ok = 1;
}
close(soc);
}

#
# Only warn the user that SSH is NOT installed
#
if(!ok)
{
report = "SSH is not running on this host !";
security_warning(port:22, data:report);
}
```

14.2. Синтаксис языка и подключаемые библиотеки

Рассмотрим синтаксис языка. Более подробные сведения приведены в документах The Nessus Attack Scripting Language Reference Guide и The NASL2 reference manual.

Комментарии. Для комментариев используется символ "#". Комментарий распространяется на часть строки справа от этого символа.

Переменные. NASL2 поддерживает работу с переменными следующих типов:

- **integer** (целый) — любая последовательность цифр со знаком. Поддерживаются восьмиричная, десятичная и шестнадцатиричная системы счисления. Восьмиричные числа должны начинаться с нуля, шестнадцатиричные — с 0x (0x10=020=16);
- **string** (строковый) — строка символов;
- **array** (массивы) — массив элементов целого или строкового типов (элементы нумеруются с 0);
- **boolean** (логический) — может принимать значения TRUE или FALSE. Считается, что неопределенное значение переменной или NULL — это FALSE. Для переменной целого типа 0 — это FALSE.

NULL — значение неинициализированной переменной или возвращаемое в случае ошибки. Для проверки значения переменной используется функция isnull:

```
v = NULL;
# isnull(v)=TRUE and typeof(v)="undef"
x = v[2];
# isnull(x)=TRUE and typeof(x)="undef"
# But isnull(v)=FALSE and typeof(v)="array"
```

Сетевые функции. Поскольку сканер Nessus является сканером сетевого уровня, наибольший интерес представляют его возможности по работе с сетью. NASL содержит множество функций, учитывающих особенности тестируемых служб. Рассмотрим операции с сокетами.

Открытие сокета. Функции open_sock_tcp() и open_sock_udp() предназначены для открытия TCP- и UDP-сокетов соответственно. Эти функции используют анонимные аргументы:

```
# OpenasocketonTCPport 80 :
soc1 = open_sock_tcp(80);
# Open a socket on UDP port 123 :
soc2 = open_sock_udp(123);
```

Функции open_sock при неудачной попытке установления соединения возвращают значение 0. Обычно использование open_sock_udp() заканчивается успешно, поскольку определить, открыт или закрыт UDP-порт на тестируемом узле, невозможно. Если TCP-порт тестируемого узла закрыт, то функция open_sock_tcp() возвращает значение 0.

Простейший пример сканера TCP-портов имеет следующий вид:

```
start = prompt("First port to scan ? ");
end = prompt("Last port to scan ? ");
```

```
for(i=start;i<end;i=i+1)
{
soc = open_sock_tcp(i);
if(soc) {
display("Port ", i, " is open\n");
close(soc);
}
}
```

Закрытие сокета. Для закрытия сокета используется функция close(). Прежде чем фактически закрыть сокет, она осуществляет вызов функции shutdown().

Запись и чтение с сокета. Запись и чтение выполняются с помощью одной из следующих функций:

recv(socket:<socketname>, length:<length> [,timeout : <timeout>)

Осуществляет чтение <length> байт из сокета с именем <имя сокета>. Данная функция может быть использована как для TCP, так и для UDP-сокетов. Необязательный параметр timeout задает значение таймаута в секундах:

recv_line(socket:<socketname>, length:<length> [, timeout: <timeout>])

Работает аналогично recv(), но recv_line завершает чтение данных, как только встречается символ \n. Эта функция работает только с сокетами TCP:

send(socket:<socket>, data:<data> [, length:<length>])

Посылает данные <data> сокету <socket>. Необязательный аргумент length указывает длину в байтах пересылаемого сокету блока данных. Если он не указан, посылка данных будет проводиться до тех пор, пока не встретится символ NULL.

Функции, используемые для чтения данных с сокета, имеют внутреннее значение таймаута, равное 5 с. Если таймаут исчерпан, функции возвращают значение FALSE.

Пример использования функции чтения данных с сокета:

```
# Пример иллюстрирует чтение FTP-баннера тестируемого
узла

soc = open_sock_tcp(21);
if(soc)
{
data = recv_line(socket:soc, length:1024);
if(data)
{
display("The remote FTP banner is : \n", data, "\n");
}
else
```

```
{
display("The remote FTP server seems to be tcp-
wrapped\n");
}
close(soc);
}
```

Подключаемые библиотеки. Содержат дополнительные функции и реализованы в виде файлов с расширением .inc. Это в основном специфичные функции для работы со службами прикладного уровня.

Контрольные вопросы

1. Опишите назначение и возможности языка NASL.
2. Перечислите функции NASL, которые учитывают особенности тестируемых служб.

Глава 15. СКАНЕРЫ БЕЗОПАСНОСТИ КОМПАНИИ POSITIVE TECHNOLOGIES

На современном рынке средств анализа защищенности наблюдается преобладание программных комплексов, позиционируемых как системы управления уязвимостями, которые обычно включают в себя компоненты управления и сканирующие модули. Помимо собственно выявления уязвимостей такие программные продукты имеют возможности масштабирования, формирования отчетов, интеграции с другими системами, адаптации под конкретную информационную систему, управления информационными активами.

Фактически можно считать, что сканер безопасности в такой системе представлен как отдельный сканирующий модуль. Таким образом, в настоящее время сканер безопасности может быть реализован как отдельный автономный программный продукт или в виде модуля сканирования в составе системы управления уязвимостями.

15.1. Краткая историческая справка

XSpider — сканер сетевого уровня (network-based), выполняющий дистанционные проверки узлов сети и не имеющий распределенной архитектуры (рис. 15.1).

Сканер безопасности XSpider появился 2 декабря 1998 г. Первая версия этого сканера называлась Spider, но вскоре сканер был переименован в XSpider. В 2000 г. программа XSpider была выложена в сети Интернет для свободного скачивания.

Коммерческая версия сканера XSpider 7.0 появилась в 2002 г., в этом же году была создана компания Positive Technologies. Первоначально основным направлением деятельности компании были услуги в области защиты информации: аудит внешних и внутренних сетей и др.

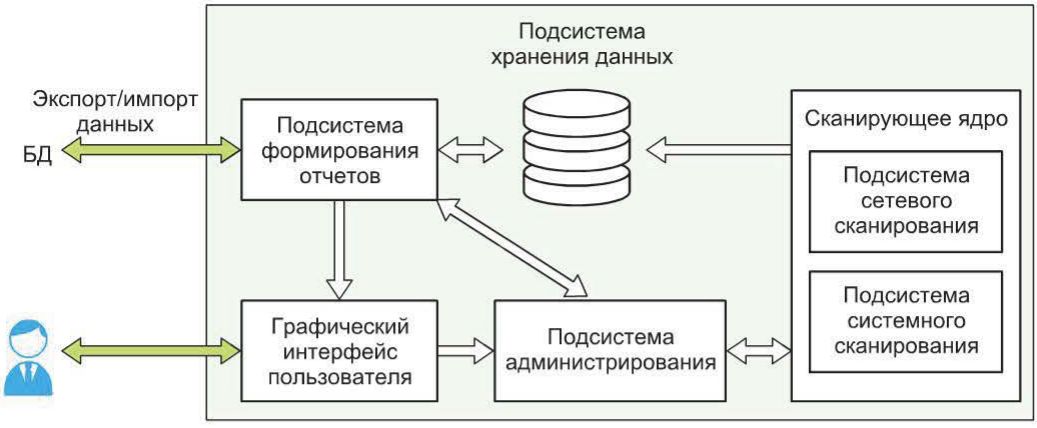


Рис. 15.1. Архитектура сканера XSpider

В 2006 г. выпущена версия 7.5, в 2008 г. — версия 7.7. Фактически версия 7.7 была последним крупным обновлением сканера.

Параллельно развитию сканера XSpider формировалась концепция нового флагманского продукта компании Positive Technologies — системы контроля защищенности и соответствия стандартам MaxPatrol, которая появилась в 2009 г.

В 2011 г. было принято решение продолжить развитие сканера XSpider на новом «движке» — появилась версия 7.8. Основные возможности этой версии рассмотрены ниже.

15.2. Архитектура и основные возможности сканера XSpider

Характерными особенностями сканера являются:

- простой и удобный интерфейс пользователя, отражающий типичные этапы работы сетевого сканера безопасности;
- расширенные возможности по идентификации служб и приложений сканируемого узла;
- подбор паролей к большинству используемых сетевых служб (FTP, SMTP, POP3, Telnet, SSH, RDP, MySQL, MSSQL, SMB, Oracle[1], OracleSID/servicename, SNMP, VNC, Radmin);
- расширенные возможности анализа защищенности web-приложений;
- наличие специальных механизмов, уменьшающих число ложных срабатываний;
- расширенные проверки систем Windows, направленные на инвентаризацию ПО, установленных лицензий, выявление уязвимостей, недоступных при сканировании в режиме черного ящика[2];

[1] Для подбора учетных записей и SID/servicename СУБД Oracle необходимо установить дополнительное ПО Oracle client.

[2] Список ПО, которое анализируется при расширенных проверках, ограничен.

• поддержка системы расчета степени риска уязвимостей CommonVulnerabilityScoringSystem (CVSS);
• наличие встроенного профиля PCI DSS ASV;
• гибкая политика лицензирования по числу сканируемых хостов.

15.3. Этапы работы сканера XSpider

Для XSpider, как и для любого сетевого сканера, характерны следующие типичные этапы работы:

1) идентификация узлов из заданного диапазона;
2) сканирование портов TCP;
3) сканирование портов UDP;
4) идентификация сервисов, приложений, ОС;
5) выявление уязвимостей.

Результат первого этапа — перечень «живых» узлов из диапазона, заданного для сканирования. После этапов 2) и 3) становится известным перечень открытых портов TCP и UDP. Идентификация сервисов включает в себя идентификацию служб (протоколов прикладного уровня), соответствующих найденным открытым портам, и идентификацию приложений, реализующих эти службы. На последнем этапе проводится поиск уязвимостей найденных сетевых служб.

15.4. Сбор информации о сети

Идентификация узлов. Для идентификации узлов в сканере XSpider предусмотрены два метода (рис. 15.2): ICMP Ping и TCP Ping.

Если узел не отвечает на запрос ICMP Echo и опции TCP Ping и «Сканировать неотвечающие узлы» не задействованы, его дальнейшее сканирование проводиться не будет.

Если задействован метод TCP Ping, после неудачной идентификации сканируемых узлов методом ICMP Ping сканер переходит к использованию метода TCP Ping.

Если же необходимо перейти к следующему этапу (сканированию портов) в любом случае, даже если узел не отвечает, следует задействовать опцию «Сканировать неотвечающие узлы».

Сканирование портов и идентификация сетевых служб. Если по результатам этапа идентификации сетевых объектов узел считается доступным, сканер переходит к следующему этапу — идентификации открытых портов. Данная задача решается в два этапа: сканирование TCP-портов и сканирование UDP-портов.

Для этого в сканер XSpider встроены два метода сканирования портов:

• tcp connect() scan — сканирование с установлением соединения;
• UDP Scan — сканирование UDP-сервисов.

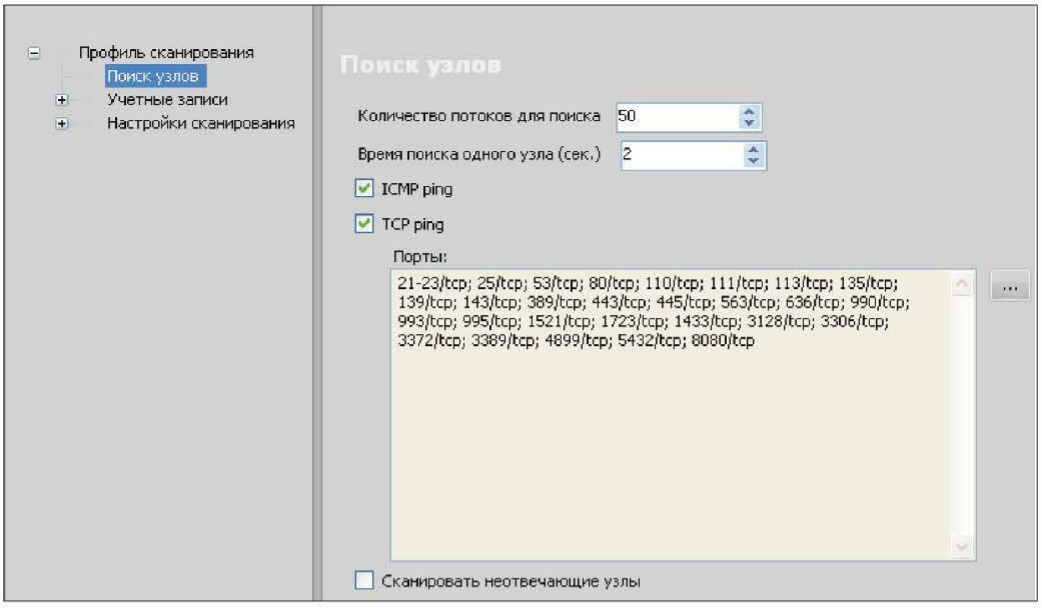


Рис. 15.2. Идентификация узлов в XSpider

Для проверки статуса произвольного TCP-порта на сканируемом узле используется стандартная функция ОС — tcp connect(). При этом с портом на сканируемом узле устанавливается полноценное TCP-соединение, которое сразу же «аварийно» разрывается (рис. 15.3).

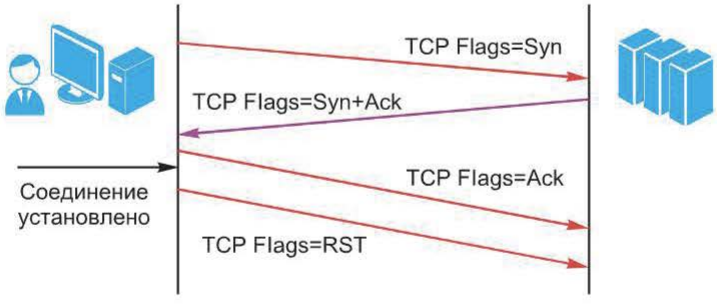


Рис. 15.3. Проверка статуса произвольного TCP-порта

Сканирование UDP-порта выполняется путем отправки «осмысленных» запросов UDP-сервисам: на заданный UDP-порт посылается не пустой UDP-пакет, а «осмысленный» запрос соответствующей службе (ожидаемой на данном порту). Это позволит сделать вывод о том, что порт открыт на основе получения ответа (рис. 15.4).

При этом, разумеется, сканируется не весь диапазон UDP-портов, а только основные порты (например, 53, 161, 500 и т. п.).

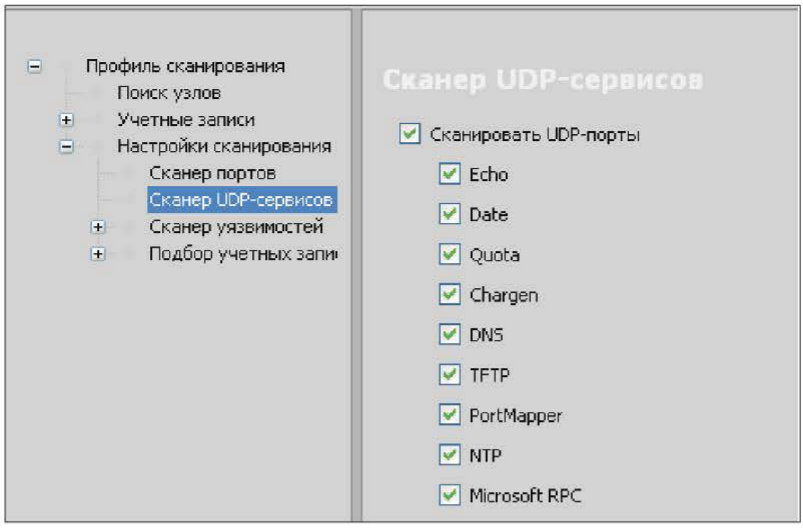


Рис. 15.4. Сканирование UDP-портов

Перед тем, как переходить к следующему шагу — поиску уязвимостей, необходимо провести идентификацию служб на найденных открытых портах.

Таким образом, идентификация приложений выполняется разными способами. Причем используемые приемы могут быть основаны на собственном опыте разработчиков сканеров, а следовательно, могут быть уникальными. Поскольку окончательное решение принимается на основе нескольких проверок, имеет смысл назвать такие методы эвристическими. С одной стороны, чем больше способов задействовано, тем больше времени сканер будет тратить на определение приложений, но, с другой стороны, приложение будет идентифицировано более достоверно. Отключить использование эвристических методов можно путем редактирования профиля сканирования (рис. 15.5).

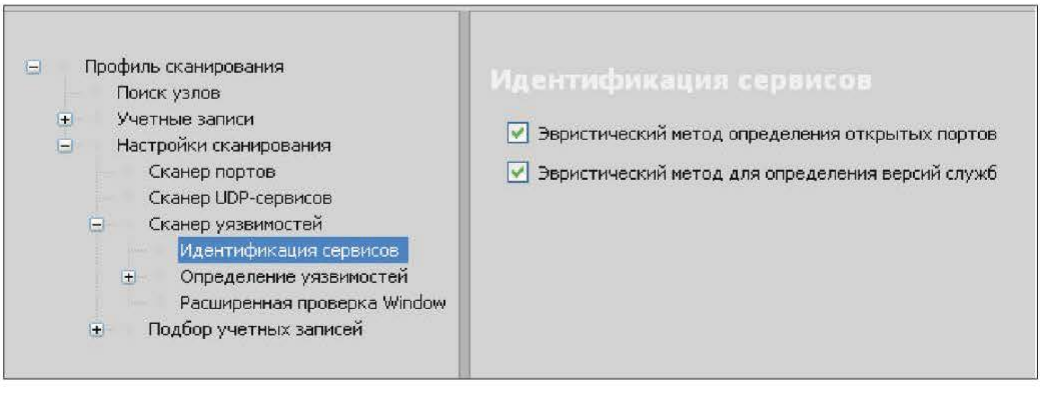


Рис. 15.5. Редактирование профиля сканирования

Сканер XSpider выполняет полную идентификацию служб на всех портах с высокой степенью достоверности.

Результат данного шага — найденные (идентифицированные) сетевые службы.

Если служба не идентифицирована, выводится название службы, использующей данный порт по умолчанию.

15.5. Идентификация уязвимостей

Типы проверок. Проверки в сканере XSpider принципиально ничем не отличаются от других средств анализа защищенности. Их структура приведена на рис. 15.6.

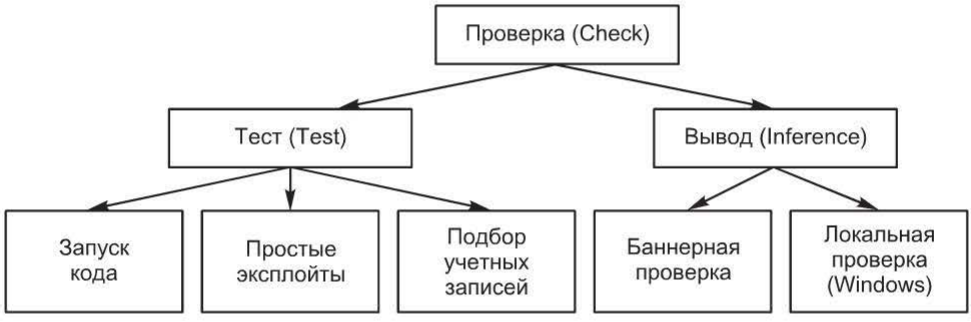


Рис. 15.6. Типы проверок, встроенных в сканер XSpider

Большая часть встроенных в сканер проверок относится к так называемым выводам, которые делаются на основе собранной информации.

Как видно на рис. 15.6, проверки подразделяют на две категории: баннерные и локальные.

Баннерные проверки. Баннерные проверки работают на основе информации, собранной в ходе инвентаризации. Чаще всего такой информацией являются результаты идентификации служб и приложений. Несмотря на название, вывод о наличии уязвимости не всегда делается только по баннеру сервиса, возможен вывод по результатам сбора информации в целом. По сути, на основе собранной информации проводится поиск в базе уязвимостей, а затем делаются выводы. При поиске учитывается информация о версии сервиса, версии приложения, иногда учитывается ОС. Таким образом, точность результатов зависит от двух факторов:

- качественной идентификации сервисов и приложений;
- качественного анализа версий приложений с учетом ОС, дистрибутивов и различных «ответвлений».

Пример описания проверки, работающей таким образом, приведен на рис. 15.7.

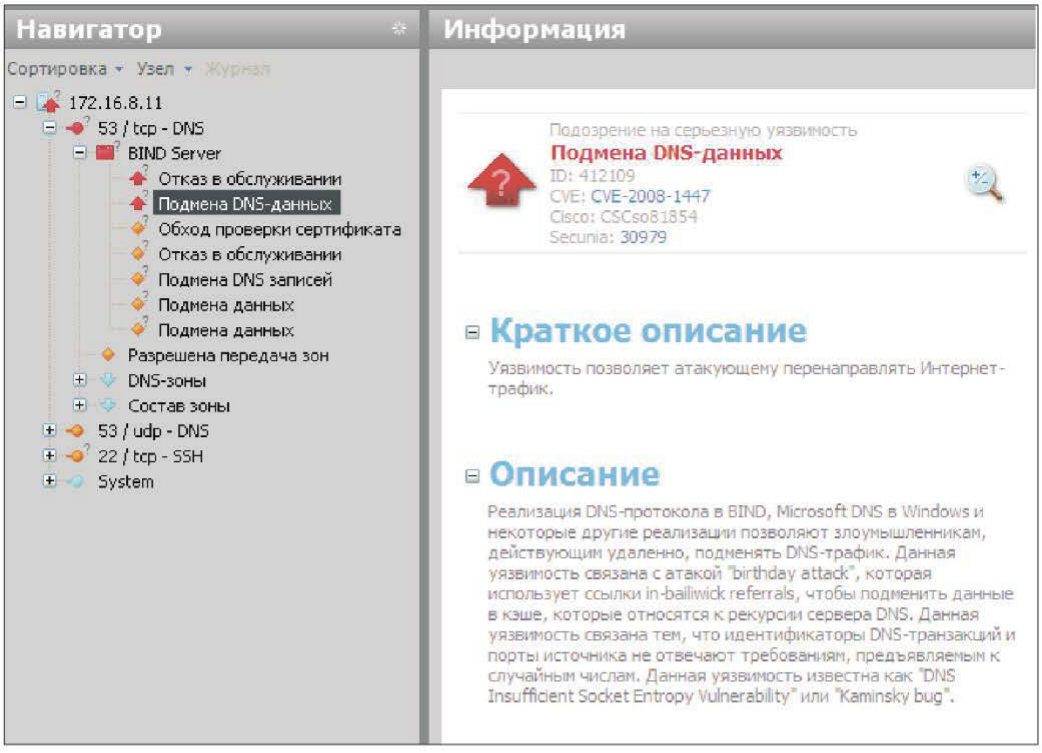


Рис. 15.7. Баннерные проверки

Такие проверки в базе XSpider представлены в значительном количестве. Большая часть проверок, направленных на выявление уязвимостей сетевых сервисов, таких как DNS, HTTP, SSH, FTP, — это именно баннерные проверки.

Благодаря качественной идентификации сервисов и приложений число ложных срабатываний при выполнении таких проверок минимально. В сравнительных тестах баннерных проверок сканер XSpider практически всегда показывал лучшие результаты.

Подбор учетных записей. Достаточно полно в сканере XSpider представлен подбор учетных записей (рис. 15.8).

В ходе выполнения проверок по подбору пароля используется следующая последовательность действий:

- обнаружение сетевой службы (для которой задействован подбор паролей);
- построение списка учетных записей;
- выбор механизма аутентификации (из числа поддерживаемых объектом сканирования);
- подбор пароля методом интерактивного перебора.

На первом этапе при идентификации сервисов и приложений XSpider обнаруживает сетевую службу, для которой в профиле задействован под-

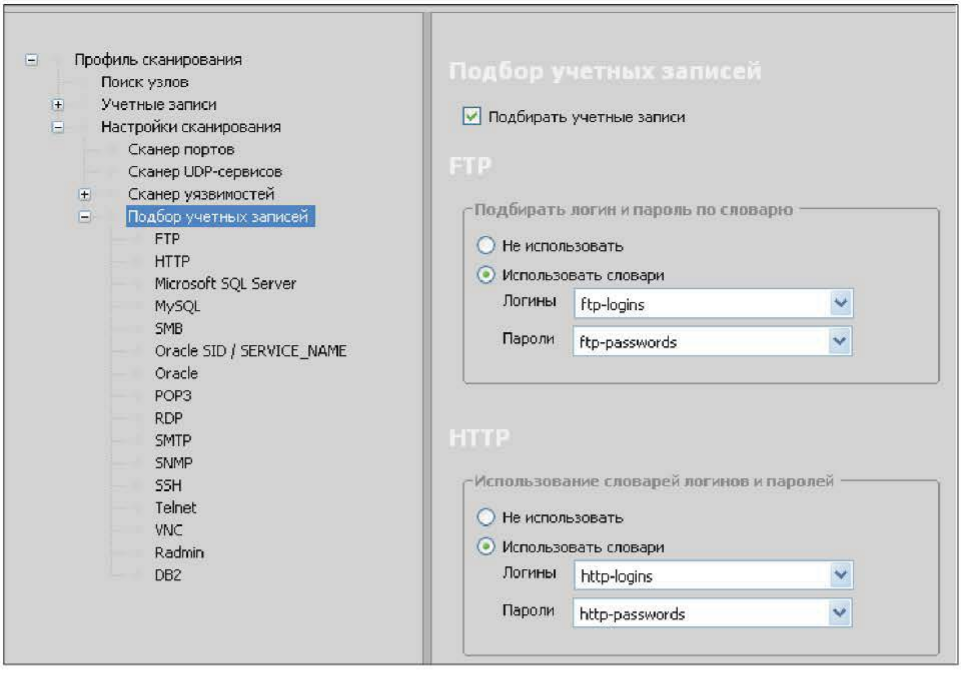


Рис. 15.8. Подбор учетных записей в сканере XSpider

бор паролей. Затем строится список учетных записей, для которых будет проводиться подбор паролей. Этот список формируется на основе встроенных данных, словарей логинов (если эта опция задействована) и ранее обнаруженных «логинов». Для сбора учетных записей пользователей могут использоваться различные механизмы, такие как «нулевой сеанс» в ОС Windows.

Затем определяется поддерживаемый объектом сканирования механизм аутентификации. Если поддерживается несколько методов, выбирается наиболее эффективный с точки зрения подбора.

15.6. Локальные проверки систем Windows

Одна из категорий проверок, встроенных в сканер XSpider, — локальные, или системные, проверки Windows, например:

- контроль обновлений ОС Windows;
- инвентаризация установленного ПО;
- анализ настроек системы и приложений;
- проверки учетных записей (групп).

Локальные проверки включаются в профиле секции «Расширенная проверка Windows» (рис. 15.9).

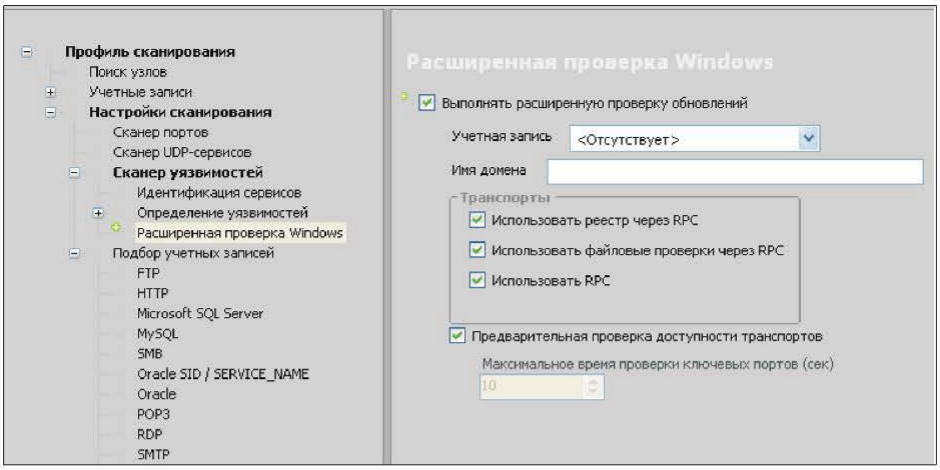


Рис. 15.9. Локальные проверки Windows

15.7. Выявление уязвимостей web-приложений

Хотя сканер XSpider и не является специализированным инструментом анализа защищенности web-приложений, рассмотрим его возможности в этой области.

В настоящее время существует несколько классификаций уязвимостей web-приложений. Наиболее структурированные из них — классификации OWASP и Web Application Security Consortium.

Система классификации Web Application Security Consortium предполагает использование различных вариантов представления (Data Views) перечня угроз в зависимости от цели.

Базовым (основным) вариантом является представление, содержащее перечень атак и уязвимостей (weaknesses), наличие которых может привести к компрометации web-приложения, его данных или пользователей.

Ниже представлен перечень атак:

- злоупотребление функциональными возможностями (Abuse of Functionality);
- подбор (Brute Force);
- переполнение буфера (Buffer Overflow);
- подмена содержимого (Content Spoofing);
- предсказуемое значение идентификатора сессии (Credential/Session Prediction);
- межсайтовое выполнение сценариев (Cross-Site Scripting, XSS);
- межсайтовая подделка запроса (Cross-Site Request Forgery);
- отказ в обслуживании (Denial of Service);
- идентификация приложений (Fingerprinting);
- атака на функции форматирования строк (Format String);
- контрабанда HTTP-ответа (HTTP Response Smuggling);

- расщепление HTTP-ответа (HTTP Response Splitting);
- контрабанда HTTP-запроса (HTTP Request Smuggling);
- расщепление HTTP-запроса (HTTP Request Splitting);
- целочисленное переполнение (Integer Overflows);
- внедрение операторов LDAP (LDAP Injection);
- E-mail инъекция (Mail Command Injection);
- инъекция нулевого байта (Null Byte Injection);
- выполнение команд ОС (OS Commanding);
- обратный путь в директориях (Path Traversal);
- предсказуемое расположение ресурсов (Predictable Resource Location);
- выполнение удаленного файла (Remote File Inclusion (RFI));
- обход маршрутизации (Routing Detour);
- фиксация сессии (Session Fixation);
- злоупотребление SOAP (SOAP Array Abuse);
- внедрение серверных расширений (SSI Injection);
- внедрение операторов SQL (SQL Injection);
- злоупотребление перенаправлениями (URL Redirector Abuse);
- внедрение операторов XPath (XPath Injection);
- переполнение XML-атрибутов (XML Attribute Blowup);
- внедрение внешних XML-атрибутов (XML External Entities);
- расширение XML-сущностей (XML Entity Expansion);
- XML-инъекция (XML Injection);
- XQuery-инъекция (XQuery Injection).

Перечень уязвимостей (weaknesses):

- некорректная настройка приложения (Application Misconfiguration);
- индексирование директорий (Directory Indexing);
- некорректная установка разрешений файловой системы (Improper Filesystem Permissions);
- некорректное управление вводом данных (Improper Input Handling);
- некорректное управление выводом данных (Improper Output Handling);
- утечка информации (Information Leakage);
- небезопасная индексация (Insecure Indexing);
- недостаточное противодействие автоматизации (Insufficient Anti-automation);
- недостаточная аутентификация (Insufficient Authentication);
- недостаточная авторизация (Insufficient Authorization);
- небезопасное восстановление паролей (Insufficient Password Recovery);
- недостаточная проверка процесса (Insufficient Process Validation);
- отсутствие таймаута сессии (Insufficient Session Expiration);
- недостаточная защита транспортного уровня (Insufficient Transport Layer Protection);
- неверная конфигурация сервера (Server Misconfiguration).

Если в ходе сканирования портов и идентификации служб был найден web-сервер, проводится поиск уязвимостей, соответствующих типу сервера

(Internet Information Server, Apache и т. д.), а также установленных расширений (FrontPage, OpenSSL и т. п.).

Следующим этапом является авторизация и проверка хорошо известных уязвимостей web-приложений.

После этого включается механизм поиска скрытых директорий и индексации содержимого. В ходе сбора содержимого сканирующее ядро XSpider использует не только содержимое web-страниц. Различные служебные и информационные файлы, содержащиеся на сервере (например, robots или readme.txt), также анализируются на предмет наличия гиперссылок. В XSpider входит базовый анализатор JavaScript, позволяющий работать с AJAX-приложениями.

После построения карты сайта сканер переходит к режиму поиска уязвимостей, которые отображаются в консоли программы по мере обнаружения.

Контрольные вопросы

1. Опишите архитектуру и основные возможности сканера XSpider.
2. Перечислите этапы работы сканера XSpider.
3. Каким образом осуществляется идентификация уязвимостей?
4. Как проводятся локальные проверки систем Windows?
5. Каким образом происходит выявление уязвимостей web-приложений?

Глава 16. АНАЛИЗ ЗАЩИЩЕННОСТИ НА УРОВНЕ УЗЛА

Ранее были рассмотрены сканеры сетевого уровня, выполняющие дистанционные проверки. Сетевой сканер идентифицирует уязвимости самой высокой степени риска, которые требуют немедленного реагирования. Данная глава посвящена сканерам уровня узла (host-based). Такие сканеры, установленные непосредственно на сканируемый узел, выполняют проверки локально. Рассмотрим их особенности, принципы работы, решаемые задачи, а также оценку стойкости паролей.

16.1. Задачи локального сканирования

Сканеры уровня узла выполняют поиск уязвимостей более тщательно и достоверно, поскольку установлены на сканируемом узле и работают от имени учетной записи с максимальными привилегиями (root, SYSTEM). Сканеры выполняют те же проверки, что и сетевые сканеры. Например, они могут осуществлять поиск работающих на узле устройств, таких как модемы, а также обнаруживать установленные на узле приложения или контролировать режим работы сетевого адаптера (селективный либо неселективный). С помощью сканеров уровня узла целесообразно выполнять те проверки, которые невозможны или трудновыполнимы для сетевых сканеров или занимают много времени.

Сканеры уровня узла обычно реализуют следующие механизмы:

- оценка стойкости паролей;
- контроль целостности;
- анализ журналов ОС и приложений, например, для поиска следов, оставленных нарушителем.

Средства сканирования на уровне узла применяются для защиты наиболее важных серверов: почтовых, web, удаленного доступа, управления БД. Эти узлы часто содержат наиболее критичные данные для ведения бизнеса, и сканирование на системном уровне поможет найти уязвимости высокой степени риска и предоставить администратору информацию для устранения найденных проблем. Сканирование на системном уровне применяется и при защите межсетевых экранов, которые обычно запускаются под управлением ОС Unix и Windows 2000 и часто содержат хорошо известные уязвимости и конфигурации, установленные по умолчанию.

16.2. Архитектура

Локальные агенты, запускаемые непосредственно на объекте проверки, обеспечивают, как правило, высокую достоверность результатов, поскольку имеют полный доступ к файловой системе, реестру и другим необходимым компонентам. Их можно подразделить на два типа: постоянные и временные.

Постоянные агенты представляют собой «полноценное», установленное на объекте проверки ПО. Обычно сканирование проводится периодически по расписанию или по команде от компонентов управления. Такой подход был популярен в прошлом, в настоящее время такое решение в чистом виде используется крайне редко. Из используемых сегодня систем такого типа можно привести Assuria Auditor (http://www.assuria.com/products-new/assuria-auditor.html) (рис. 16.1).

Рис. 16.1. Интерфейс Assuria Auditor

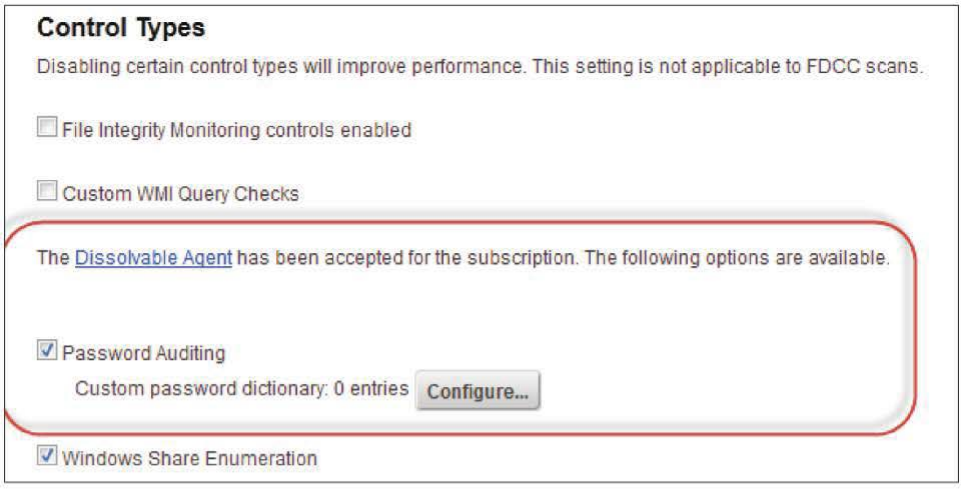


Рис. 16.2. «Растворяющийся» агент

Временные — «растворяющиеся» — (dissolvable) *агенты* не устанавливают на объект проверки, а копируют туда временно, например, по сети или посредством использования съемных носителей. Соответственно после сканирования и сохранения результатов их удаляют. В качестве примера такого агента можно привести Dissolvable Agent компании Qualys (рис. 16.2).

16.3. Сбор информации и идентификация уязвимостей

Сканеры уровня узла используют только пассивные методы идентификации уязвимостей. Это очевидно, поскольку их, как правило, устанавливают на важном сервере, и они должны оказывать на него минимальное влияние. Источники данных для сканеров уровня узла:

- файловая система узла. Часто признаком наличия уязвимости считается номер версии того или иного файла. Кроме того, изменения некоторых важных файлов могут служить признаками следов нарушителя;
- журналы регистрации;
- конфигурация, параметры, влияющие на безопасность. Например, для ОС Windows основной источник таких данных — реестр. К этой же категории относится информация о пользователях, работающих на узле служб, установленных в приложениях.

Проверки, выполняемые сканером уровня узла, осуществляют поиск уязвимостей следующими способами:

- путем сравнения версий файлов или их атрибутов с имеющимися значениями в базе данных проверок;
- поиска файлов или ключей реестра, свидетельствующих о наличии на узле того или иного приложения (вируса и т. п.);

• сравнения текущих значений параметров конфигурации с требуемыми значениями. В этом случае пользователь должен задать эти значения, которые обычно являются частью политики безопасности.

16.4. Сканер Assuria Auditor

Наглядным примером сканера уровня узла служит Assuria Auditor, разработанный компанией Assuria (рис. 16.3). Он имеет распределенную архитектуру (агенты для различных ОС и управляющая консоль).

Агент представляет собой службу, работающую от имени учетной записи «local system» (рис. 16.4). Она запускается при старте ОС, находится в слушающем режиме и по команде с консоли запускает процесс сканирования

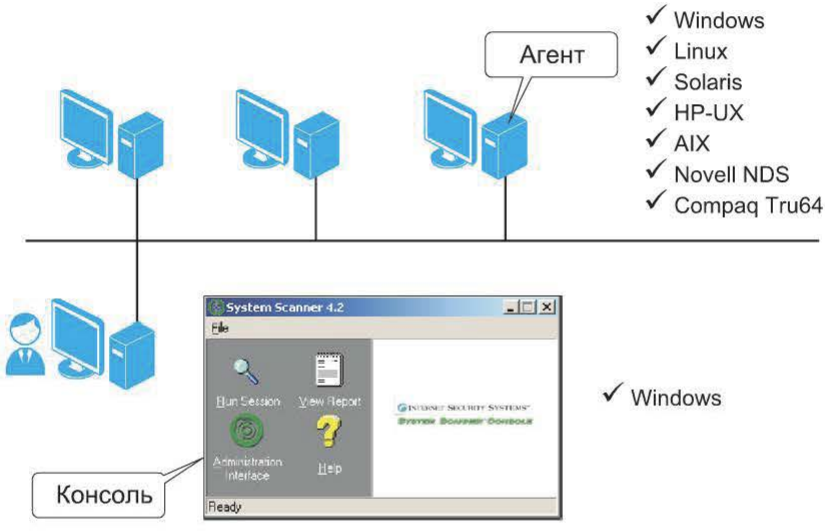


Рис. 16.3. Распределенная архитектура сканера Assuria Auditor

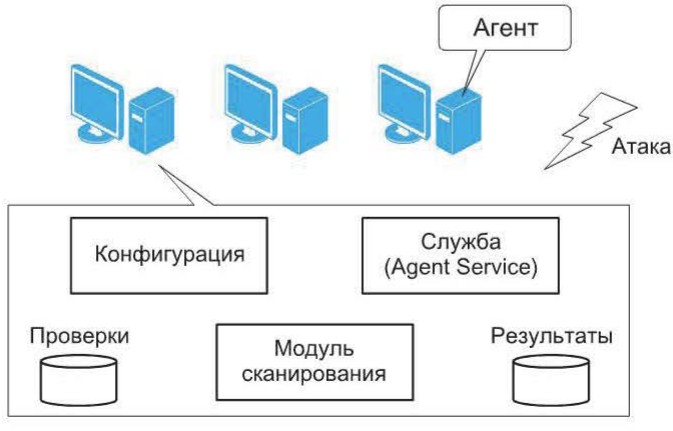


Рис. 16.4. Архитектура агента Assuria Auditor

узла (модуль сканирования). Результаты сохраняются локально, а затем передаются на консоль.

В контексте Assuria Auditor сессия сканирования определяется множеством агентов, к которым применяется какая-либо политика сканирования. Сканирование с данной политикой запускается одновременно на всех агентах, входящих в это множество.

Политика сканирования — это набор групп проверок, выполняющихся одновременно.

У сканера Assuria Auditor имеется особенность, согласно которой проверки объединяются в группы (по типу), а затем группы объединяются в политику сканирования.

Контрольные вопросы

1. Перечислите задачи локального сканирования. Дайте характеристику каждой задаче.
2. Каковы особенности архитектуры сканеров уровня узла?
3. Каким образом осуществляется идентификация уязвимостей?
4. Перечислите источники данных для сканеров уровня узла.
5. Что такое сканер Assuria Auditor?

Глава 17. СПЕЦИАЛИЗИРОВАННЫЕ СРЕДСТВА АНАЛИЗА ЗАЩИЩЕННОСТИ

17.1. Классификация сканеров безопасности по назначению

Применяется также классификация сканеров безопасности по их назначению. При этом выделяют две категории: сканеры общего характера и специализированные сканеры.

Пояснить этот способ классификации на примере сетевых сканеров можно следующим образом. Проверки, выполняемые сетевыми сканерами безопасности, направлены прежде всего на сетевые службы. Конечно, при этом осуществляется поиск уязвимостей не только сетевых служб, но и ОС, а также некоторых приложений, установленных на сканируемом узле. Но следует признать, что проверки, встроенные в сетевые сканеры, носят общий характер, а если и направлены в отношении приложений, то это наиболее распространенные приложения и наиболее известные уязвимости. Та же ситуация и со сканерами уровня узла. Их проверки, возможно, несколько более направлены на ОС узла, где установлен агент, а также могут быть направлены и на конкретные приложения, но ни одно из них не выделяется. Таким образом работают сканеры общего характера. Другими словами, в них «всего понемногу». Часть проверок, например, направлена на поиск уязвимостей

в почтовой службе Sendmail, другая часть — в HTTP-сервере Apache и т. д. В некоторых случаях возможно выделение какого-либо приложения. Например, сканер XSpider имеет довольно много проверок, направленных в отношении web-приложений, а сканер NeXpose компании Rapid7 имеет много проверок для Lotus Domino. Но в целом сканеры общего характера содержат «универсальный» набор проверок.

Разумеется, разработчикам сетевых сканеров безопасности нет смысла встраивать детальные проверки для одного (двух) приложений. Зачем потребителю платить за неиспользуемый функционал? Однако если перед администратором или аудитором стоит задача оценки защищенности определенного приложения, можно использовать специализированные сканеры безопасности. Таким образом, необходимость в применении специализированных сканеров безопасности обусловлена тем, что для проверки используемого в корпоративной сети крупного приложения возможностей обычных сетевых сканеров может быть недостаточно (рис. 17.1).

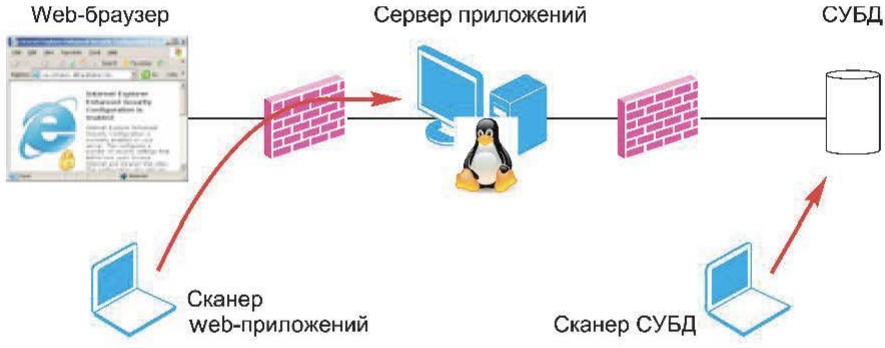


Рис. 17.1. Использование специализированных сканеров в корпоративной сети

Приведем перечень приложений, при анализе защищенности которых могут потребоваться специализированные сканеры:

- web-приложения;
- СУБД и приложения, их использующие;
- системы электронного документооборота (например, Lotus Domino);
- системы управления предприятием, так называемые ERP-системы (например, SAP R/3, OracleApplications).

Следует отметить, что приложения такого рода содержат типовые компоненты, и в принципе для оценки их защищенности можно использовать принятую методологию и несколько обычных сканеров безопасности общего характера. Однако если приложение широко распространено, методология анализа его безопасности сформировалась, для оценки его защищенности специализированные инструменты можно не использовать.

Несколько примеров специализированных сканеров web-приложений приведены ниже.

WebInspect	http://www.spidynamics.com/products/webinspect/index.html
Acunetix	http://www.acunetix.com
AppScan	http://www.watchfire.com
Nikto	http://www.cirt.net/code/nikto.shtml

17.2. Угрозы и уязвимости СУБД

Сервер баз данных — достаточно сложное сетевое приложение, поддерживающее различные сетевые протоколы, широко использующее ресурсы ОС и имеющее специфические методы хранения и обработки данных, а также инструменты администрирования. В его работе задействованы как сетевые технологии, так и ОС узла, на котором он установлен. Исходя из этого, угрозы безопасности СУБД целесообразно рассматривать по следующим уровням: физический, сетевой, уровни ОС, СУБД, приложения и пользователя.

На *физическом уровне* серверу БД требуется защита, нейтрализующая угрозу обхода защитных механизмов при получении физического доступа к узлу, например к его жесткому диску.

Сетевой уровень обеспечивает передачу данных между СУБД и взаимодействующим с ней приложением, поэтому возникает угроза перехвата информации, если сетевое взаимодействие должным образом не защищено.

Для работы сервера БД используется какая-либо ОС, при этом, разумеется, в работе сервера задействованы и ее компоненты: службы, файлы на диске, ключи реестра, конфигурационные файлы. Для этих компонентов на уровне ОС могут и должны быть задействованы такие механизмы защиты, как разграничение доступа и аудит. Кроме того, некоторые действия уровня ОС (доступ к файловой системе, реестру, запуск командной строки) могут быть выполнены средствами СУБД, следовательно, существует угроза компрометации ОС через СУБД.

Уровень СУБД следует выделить отдельно, поскольку современные СУБД имеют собственные механизмы защиты. Ряд угроз, относящихся к уровню СУБД:

- получение паролей пользователей;
- повышение привилегий;
- уязвимости реализации (переполнение буфера и т. п.);
- отказ в обслуживании;
- ошибки конфигурирования.

Уровень *приложений* привносит свои, свойственные ему угрозы, например, возможность внедрения SQL-кода в данные, передаваемые сервером приложений серверу БД.

Наконец, *пользователи*, имеющие доступ к конфиденциальным данным, которые хранит СУБД, особенно те, которые имеют административные привилегии, также представляют собой серьезную угрозу. Последние аналитические отчеты консалтинговых компаний показали, что неавторизованный доступ к данным и кража конфиденциальной информации являются главными

составляющими потерь предприятий. Межсетевые экраны, системы обнаружения атак и анализа защищенности могут оказаться неэффективными перед утечкой информации по вине персонала и злонамеренными действиями «всесильных» администраторов БД.

17.3. Особенности анализа защищенности СУБД

Учетные данные для подключения к серверу БД. В зависимости от выбранной методологии сканирования могут потребоваться учетные записи для подключения к сканируемому серверу БД. Если выполняется тестирование сервера на устойчивость к взлому (Penetration Test), учетная запись для подключения к серверу не требуется. Но если выполняется аудит сервера БД на соответствие требованиям политики безопасности, необходима учетная запись с определенными привилегиями. Учетная запись задается в параметрах сканера и используется в процессе сканирования. Пример задания учетной записи с привилегиями в программе Database Scanner компании Safety Lab приведен на рис. 17.2.

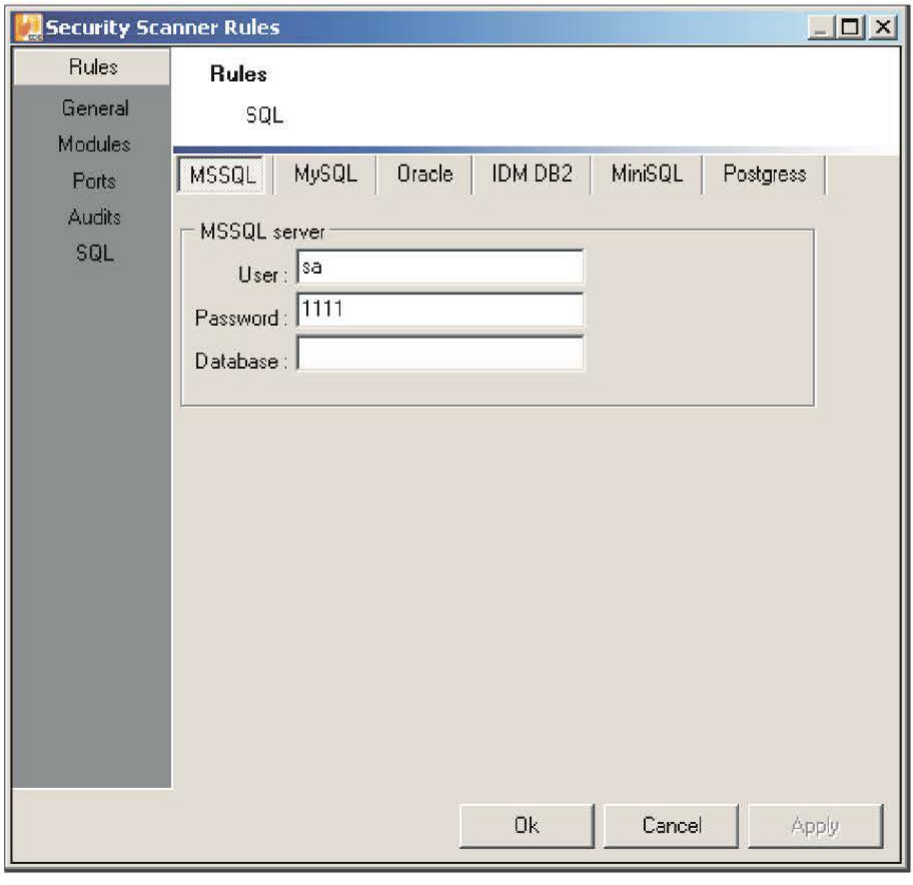


Рис. 17.2. Указание учетной записи с привилегиями в программе Database Scanner

Возникает вопрос: какой уровень привилегий требуется для выполнения проверок? Разумеется, это не должна быть учетная запись sa (для MS SQL Server) или администратора СУБД. Для целей сканирования создается отдельная учетная запись именно с необходимыми привилегиями. Например, в документации к программе Database Scanner компании ISS приведен список привилегий, необходимых учетной записи для проведения сканирования. Кроме того, в политике сканирования по каждой проверке представлена информация о требуемых для ее успешного выполнения допусках и привилегиях. Так, на рис. 17.3 приведено описание проверки наличия пользователя «Guest», где видно, какие допуски требуются для ее выполнения.

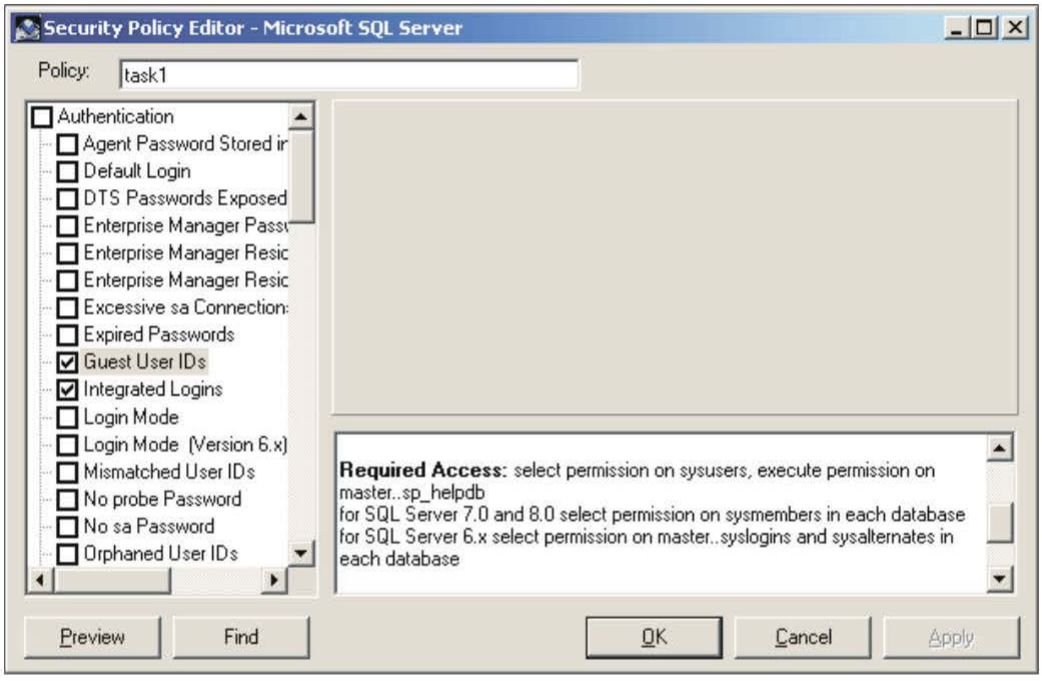


Рис. 17.3. Описание проверки наличия пользователя «Guest»

Учетные данные для подключения к узлу. При сканировании СУБД следует учитывать выбор учетных данных уровня ОС (для подключения к узлу, на котором развернут сервер БД). Чаще всего это необходимое условие сканирования серверов БД на платформе Windows, поскольку отдельные проверки требуют удаленного доступа к реестру. Сканер может выполнять подключение с полномочиями текущего пользователя или, как это предусмотрено, например, в про-

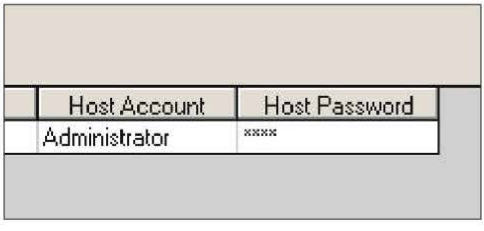


Рис. 17.4. Задание имени и пароля для сканирования

грамме Database Scanner компании ISS, использовать имя и пароль, заданные для целей сканирования (рис. 17.4).

Особенностью сервера Microsoft SQL Server является то, что для обращения к реестру могут быть использованы не только функции Win32 API, требующие указания учетной записи, но и хранимые процедуры сервера БД, в частности, процедуры:

xp_regdeletevalue
xp_regwrite
xp_regread

могут быть использованы для чтения и изменения информации, хранящейся в реестре Windows. Выбор метода обращения к реестру показан на рис. 17.5.

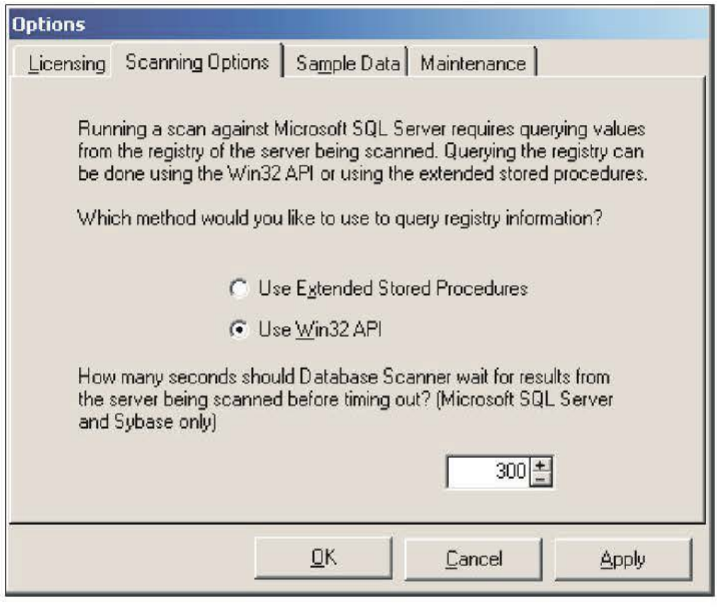


Рис. 17.5. Выбор метода обращения к реестру

Особенности работы проверок. Подключившись к серверу БД, сканер выполняет серию SQL-запросов для получения необходимой информации. Все проверки работают по принципу анализа собранной информации. Единственной особенностью выполнения проверок можно назвать требование наличия на узле клиентского ПО для подключения к СУБД, например, для сканирования СУБД Oracle требуются:

- SQL*NET driver;
- NET* driver;
- Oracle 8 driver.

Для сканирования СУБД Sybase необходимы ПО:

- Sybase Adaptive Server ODBC;
- Driver Open Client Library.

17.4. Примеры программ-сканеров уязвимостей СУБД

Средства анализа защищенности СУБД позволяют проводить и локальный, и дистанционный анализ серверов БД. В качестве примеров можно привести следующие системы:

• AppSentry компании Integrity (http://www.integrigy.com/products/appsentry);

• AppDetectivePro компании Trustwave (https://www.trustwave.com/Products/Database-Security/);

• продукты компании NGSSecure (http://www.ngssecure.com/services/information-security-software.aspx);

• McAfee Security Scanner for Databases (http://www.mcafee.com/us/products/security-scanner-for-databases.aspx);

• Shadow Database Scanner компании Safety Lab (http://www.safety-lab.com/en/products/6.htm /);

• SecureSphere Discovery and Assessment Server (http://www.imperva.com/products/dsc_database-discovery-and-assessment-server.html);

• Scuba (http://www.imperva.com/products/dsc_database-discovery-and-assessment-server.html).

Контрольные вопросы

1. Как классифицировать сканеры безопасности по назначению?
2. Дайте характеристику угрозам и уязвимостям СУБД.
3. В чем заключаются особенности анализа защищенности СУБД?
4. Приведите примеры программ-сканеров уязвимостей СУБД. Как их использовать?

Глава 18. МЕТОДОЛОГИЯ АНАЛИЗА ЗАЩИЩЕННОСТИ ETHICAL HACKING

18.1. Необходимость методологии анализа защищенности

Выше были рассмотрены возможности сканеров уязвимостей и отмечена необходимость методологии при анализе защищенности. Однако полагаться только на результаты работы сканера уязвимостей нельзя. На основе этих результатов можно предпринять следующие действия:

• установить обновления в соответствии с найденными уязвимостями;

• провести мероприятия по снижению вероятности использования уязвимости, если она не может быть устранена немедленно (например, вследствие проблем совместимости);

• проверить правильность (корректность) устранения уязвимостей;

• внести изменения в политику безопасности, архитектуру системы в целях учета последних изменений.

Методология представляет собой четкую последовательность шагов, которые необходимо предпринять для выполнения определенной задачи. Применение методологии при анализе защищенности обеспечивает следующие преимущества:

- возможность придерживаться определенного плана действий;
- уверенность в том, что предприняты все необходимые действия и ничего не было упущено;
- упрощение процедуры согласования действий с заказчиком, возможность предоставления отчета по каждому этапу работ.

Конечно, следует понимать, что методология может быть изменена и адаптирована для каждого конкретного случая, сетевого окружения и т. д.

Существует несколько методик анализа защищенности, некоторые из них уже имеют свои устоявшиеся названия, например, Penetration Testing — тестирование системы на устойчивость к взлому.

Методологии анализа защищенности можно разделить на две большие группы.

1. *Аудит* (Audit) — проверка соответствия используемых механизмов защиты заданным требованиям. Обычно эти требования изложены в политике безопасности (например, длина пароля должна быть не менее восьми символов, пароли необходимо менять один раз в месяц и т. п.).

2. *Оценка защищенности* (SecurityAssessment) — проверка устойчивости систем к атакам, поиск уязвимостей в системе безопасности и т. п. Применяемая методология анализа защищенности может использовать методы, свойственные обеим группам.

Рассмотрим проверку сети на устойчивость к взлому с помощью методологии Penetration Testing или Ethical Hacking.

18.2. Penetration Testing — общие сведения

Penetration Testing (тест на проникновение) — оценка защищенности, в процессе которой субъект, выполняющий оценку, опирается на собственное понимание того, как реализована тестируемая система. Цель такого теста — поиск способов получения доступа к системе с помощью инструментов и приемов, используемых нарушителями. Тестирование с помощью данной методологии можно рекомендовать для особо критичных систем.

Однако следует учитывать и такие особенности данной методологии, как трудоемкость, значительная нагрузка на сеть, риск выведения из строя тестируемых узлов. Кроме того, поскольку в процессе тестирования могут быть использованы инструменты и методы, запрещенные политикой безопасности организации или законодательством, необходимо получить письменное соглашение на проведение тестирования. Такое письменное соглашение может включать в себя:

- диапазон IP-адресов для тестирования;
- список узлов, не подлежащих тестированию;

• список допустимых техник (социальная инженерия, выведение из строя и т. д.) и инструментов (сетевые анализаторы, сканеры уязвимостей и т. д.);

• время проведения тестирования (выходные, рабочие часы и т. п.);

• IP-адреса узлов, с которых предпринимается тестирование (например, для того чтобы администраторы могли отличить тестирование от реальной атаки).

Текст соглашения может варьироваться в зависимости от варианта проведения теста на проникновение.

18.3. Разновидности Penetration Testing

Уведомление IT-персонала. Тест на проникновение может быть открытым — явным (overt) — и скрытым (covert). В соответствии с первым вариантом предполагается, что IT-персонал организации поставлен в известность о проведении тестирования и в процессе проведения тестирования будет принимать в нем требуемое участие.

Во втором варианте тест проводится без уведомления IT-персонала, но с разрешения высшего руководства организации. Этот вариант тестирования направлен на проверку не только используемых средств защиты, но и на проверку реакции IT-персонала на выполняемые атаки, а также на проверку адекватности и работоспособности принятой в организации политики безопасности.

Первый вариант (открытый) требует меньших затрат и, соответственно, больше используется. Второй вариант, поскольку он скрытый, требует больше времени и, соответственно, дороже. В этом случае сканирование и другие действия выполняются медленнее с целью «обмана» межсетевых экранов и систем обнаружения атак. Однако именно этот вариант дает более точную картину состояния системы безопасности организации.

Тестирование изнутри и снаружи. Тестирование может быть выполнено как снаружи, так и изнутри сети. Если проводится тестирование снаружи и изнутри, тестирование снаружи обычно выполняется в первую очередь.

Тестирование снаружи имеет следующие особенности. Наличие межсетевого экрана ограничивает количество и разновидности трафика, достигающего внутренней сети. Поэтому на начальном этапе тестирования, как правило, используется разрешенный к прохождению трафик (обычно это HTTP, FTP, SMTP и т. п.). В целом это увеличивает время, сложность и стоимость тестирования.

Для повышения точности тестирования обычно предоставляется минимальная информация об объекте тестирования, как правило, это диапазон IP-адресов. Поэтому требуется тщательный предварительный сбор информации с использованием всех методов, рассмотренных выше, начиная с информации, доступной через Интернет, и заканчивая идентификацией уязвимостей.

Тестирование изнутри сети напоминает тестирование снаружи за исключением следующих особенностей. Как правило, предоставляется определен-

ный уровень привилегий (например, на уровне пользователя). В этом случае может быть поставлена задача оценки возможности повышения привилегий. Вполне возможно, что не будет предоставлено какого-либо уровня привилегий. В этом случае тестирующему просто предоставляется подключение к сети (сетевой кабель). Поэтому при тестировании изнутри часто используются пассивные методы сбора информации (рассмотренный выше Passive fingerprinting). Чаще всего тестирование изнутри сети представляет собой «взгляд» на сеть с точки зрения рядового пользователя или администратора (т. е. пользователя с определенным уровнем привилегий).

В целом Penetration Testing — это итеративный процесс. Как правило, после получения доступа к объекту тестирования с минимальными привилегиями делается попытка повысить уровень привилегий. Или, например, при получении контроля над одним из узлов может быть предпринята попытка использовать его в качестве платформы для получения доступа к другим узлам сети.

18.4. Структура Penetration Testing

Планирование. Структура Penetration Testing приведена на рис. 18.1.

На этапе планирования определяются:

• правила тестирования, например, может быть оговорено время тестирования, возможность использования проверок, способных вывести систему из строя и т. п.;

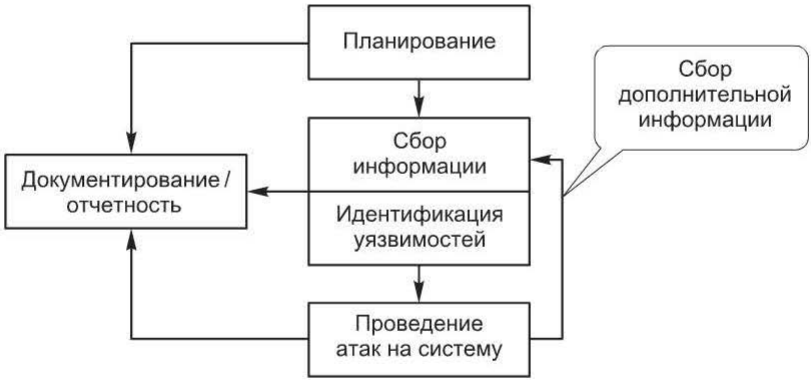


Рис. 18.1. Структура Penetration Testing

• цели тестирования, например, оценить возможность пользователей внутренней сети повысить уровень привилегий.

На этапе планирования никакие попытки тестирования сети не предпринимаются.

Сбор информации. Следующий этап — сбор информации. На этом этапе используются все рассмотренные выше методы сбора информации о сети (табл. 18.1).

Таблица 18.1

Методы сбора информации

№ п/п	Метод	Способ сбора
1	Информация, доступная через Интернет (информация регистрационного характера). Служба Whois	Снаружи
2	Получение диапазона адресов, соответствующего домену (доменам), так называемый Foot Printing	Снаружи
3	Идентификация доступных сетевых устройств	
4	Определение топологии сети	
5	Идентификация операционных систем	Изнутри/снаружи
6	Идентификация открытых портов	
7	Идентификация служб	
8	Пассивный вариант методов № 3–7 (анализ сетевого трафика)	Изнутри

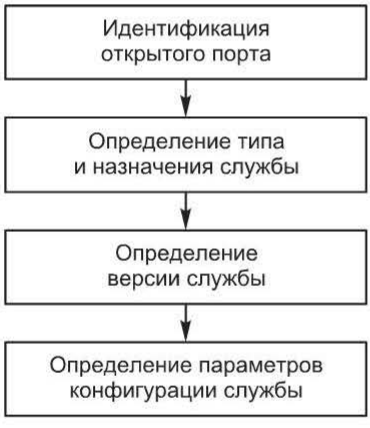


Рис. 18.2. Получение информации об активных сетевых службах

Основной итог данного этапа — точная информация об активных сетевых службах. Последовательность действий приведена на рис. 18.2.

На этом этапе вместе со сканерами безопасности используются различные средства идентификации узлов, сканирования портов, служб и т. д.

Идентификация уязвимостей. Следующий этап (связанный с предыдущим) — идентификация уязвимостей. На этом этапе используется информация о найденных узлах, установленных на них ОС и службах. Главным образом используется информация о службах (см. рис. 18.2). Эта информация сопоставляется с информацией об известных уязвимостях, т. е. с базой уязвимостей. Сканер безопасности выполняет этот процесс автоматически. Это можно делать и вручную, сопоставляя результаты сбора информации со своей базой уязвимостей. Ручной вариант надежнее и достовернее, но медленнее, чем автоматизированный.

На данном этапе можно предпринять следующие шаги:

• подобрать необходимые инструменты для поиска и использования предполагаемых уязвимостей в найденных службах;

• сделать предположение о возможных уязвимостях на основе ОС и служб;

• примерно сопоставить уязвимости имеющимся службам;

• провести сканирование узла (узлов) с помощью двух разных сканеров безопасности;

• идентифицировать уязвимости уровня ОС;

• идентифицировать уязвимости уровня приложений (служб).

Основной итог данного этапа — перечень уязвимостей, имеющихся на каждом протестированном узле.

Проведение атак на систему. Цель этого этапа — подтверждение (верификация) уязвимостей, выявленных на предыдущем этапе. Этот этап можно назвать основным в рассматриваемой методологии. Если атака проведена успешно, уязвимость считается подтвержденной.

Последовательность проведения атак с использованием найденных уязвимостей приведена на рис. 18.3.

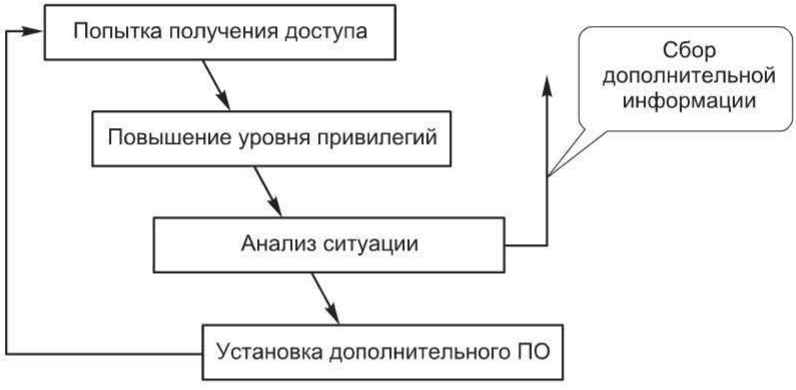


Рис. 18.3. Проведение атак с использованием найденных уязвимостей

Первый шаг данного этапа — выбор цели (целей) и попытка получения доступа (возможно, с минимальными привилегиями) с использованием различных инструментов (как правило, рассмотренных выше эксплойтов).

Если на предыдущем шаге был получен доступ только на уровне пользователя, предпринимается попытка повышения уровня привилегий (в случае систем UNIX это получение доступа с правами пользователя «root», для систем Windows — с правами администратора).

Далее следует еще один шаг — анализ ситуации, оценка того, что было получено. Это может потребовать дополнительного сбора информации (см. рис. 18.3, обозначено обратной стрелкой). Кроме того, может возникнуть необходимость в установке дополнительного ПО для получения доступа или дополнительной информации, что также отмечено обратной стрелкой.

В зависимости от целей тестирования могут быть предприняты различные действия по дальнейшему развитию сценария на основе анализа ситуации после получения доступа к одному из узлов:

• установка серверных частей «троянцев». Для получения доступа к узлу в будущем устанавливаются «троянцы» или наборы rootkit;

• получение доступа к доверенным узлам, попытка обхода механизмов защиты. После получения доступа к узлу локально может появиться возможность просмотра сети с другой точки зрения и сбора дополнительной информации. Здесь могут быть использованы следующие методы:

– анализ локальной файловой системы с целью поиска «полезной» информации;

– анализ трафика сетевого сегмента;

– использование механизмов «человек посередине» на различных уровнях в локальном сегменте;

– сканирование узлов сегмента с подменой IP-адреса;

– создание ситуации «отказ в обслуживании». После получения доступа к локальному сетевому сегменту могут быть выполнены различные DoS-атаки в отношении остальных узлов сегмента.

И наконец, важное место в процедуре Penetration Testing занимает подбор паролей: удаленный или локальный на основе полученных файлов с паролями или перехваченных паролей. Эти методы также были рассмотрены выше.

Формирование отчета. Отчет формируется по каждому этапу тестирования. На этапе планирования разрабатываются:

• правила взаимодействия, обязательства и т. п.;

• план тестирования;

• письменное соглашение на проведение тестирования.

На этапе сбора информации формируется отчет, включающий в себя:

• информацию инвентаризационного характера, а также иную информацию, собранную в процессе тестирования;

• перечень обнаруженных уязвимостей, упорядоченных по степени риска.

При проведении атак на систему документируются сценарии успешного использования найденных уязвимостей.

Вся эта информация должна быть размещена в итоговом отчете, который завершается рекомендациями по повышению защищенности, устранению найденных уязвимостей и т. д.

Восстановление систем. В зависимости от ситуации этот этап может отсутствовать. Его цель — «наведение порядка» после проведения тестирования. Следовательно, залог успеха на этом этапе — наличие подробной документации по тем действиям, которые были совершены в процессе проведения тестирования. Возможные действия, выполняемые на данном этапе:

• удаление пользовательских учетных записей, созданных в процессе тестирования;

• восстановление систем, к которым был получен доступ;

• восстановление систем, выведенных из строя.

Следует отметить, что этот этап может выполняться и силами организации, заказавшей проведение тестирования. Но в любом случае должен быть предоставлен подробный перечень внесенных в процессе тестирования изменений.

Ограничения методологии Penetration Testing. Результаты тестирования на устойчивость к взлому (Penetration Testing) не следует рассматривать как окончательное заключение о степени защищенности объекта тестирования. Рассматриваемая методология имеет следующие ограничения:

• тестируемый объект (сеть, отдельный узел и т. п.) рассматривается как «черный ящик». Следовательно, тестирующий изначально обладает минимумом информации об объекте тестирования, поэтому многие уязвимости могут быть не выявлены. Для их выявления необходимо иметь определенную информацию о тестируемой системе (которой, например, может обладать внутренний пользователь);

• тестирование происходит в какой-то ограниченный период времени, следовательно, результаты определяются известными в данный момент уязвимостями и текущей конфигурацией сети. Ситуация может измениться на следующий же день. Проверка сети на устойчивость к взлому вследствие ее высокой стоимости и возможного негативного влияния на объект тестирования проводится нечасто (например, один раз в год).

Таким образом, к результатам тестирования с использованием рассмотренной методологии следует отнестись серьезно. Как можно быстрее данные должны быть предоставлены руководству. На основе результатов тестирования могут быть предприняты следующие действия:

• устранение обнаруженных и подтвержденных уязвимостей;

• пересмотр политики безопасности и внесение изменений;

• повышение квалификации персонала и другие мероприятия, направленные на повышение защищенности систем.

Следует понимать, что основное ограничение данной методологии заключается в том, что проводится идентификация не всех уязвимостей. Это лишь взгляд на объект тестирования с точки зрения «реального» нарушителя, которому, например, для получения доступа к узлу достаточно обнаружить одну-две серьезные уязвимости.

Поэтому необходимо проводить оценку защищенности сети силами самой организации и с большей периодичностью. Но это уже другая методология: Vulnerability assessment или Network Security Assessment. Фактически это предполагает внедрение анализа защищенности в корпоративной сети как части политики безопасности, рассматриваемой в гл. 19.

Контрольные вопросы

1. Обоснуйте необходимость методологии анализа защищенности.
2. Что входит в понятие «Penetration Testing»?
3. Перечислите особенности Penetration Testing изнутри и снаружи.
4. Перечислите этапы Penetration Testing. Дайте характеристику каждому этапу.
5. По каким причинам применение методологии Penetration Testing может быть ограничено?

Глава 19. ЦЕНТРАЛИЗОВАННОЕ УПРАВЛЕНИЕ УЯЗВИМОСТЯМИ

19.1. Необходимость централизованного управления уязвимостями

Сканеры безопасности как один из видов ПО известны более 10 лет. Но до сих пор их применение в качестве средств обеспечения безопасности вызывает множество дискуссий. И не только потому, что сетевой сканер безопасности — это продукт «двойного» назначения и возможны негативные последствия его использования, а из-за механизма защиты, который в нем реализован.

Сам по себе сканер безопасности как инструмент, позволяющий на выходе получить перечень уязвимостей проверенной им системы, нужен немногим. Он может, например, помочь специалисту при проведении тестирования на проникновение автоматизировать часть рутинной работы или оказаться полезным для злоумышленника, который ищет слабости в системе для получения к ней несанкционированного доступа. Для эффективного применения сканеров безопасности в корпоративной сети необходима их интеграция в существующую инфраструктуру обеспечения безопасности. Поэтому на смену сканерам безопасности в корпоративном секторе постепенно приходят системы управления уязвимостями. Сканер безопасности в такой системе — всего лишь один из модулей, предоставляющий информацию для других модулей или компонентов, а также для других систем.

Управление уязвимостями — это процесс, а не готовый продукт, но этот процесс можно автоматизировать. Собственно, для этого и нужны системы управления уязвимостями.

В самом общем понимании, управление уязвимостями (Vulnerability Management) — процесс, направленный на предотвращение использования известных уязвимостей, потенциально существующих в защищаемой системе или сети. Основной ожидаемый результат — значительное затруднение или полное исключение возможностей для нарушителей использования этих уязвимостей и, соответственно, снижение затрат на ликвидацию последствий атак.

Создать абсолютно защищенную систему принципиально невозможно. К тому же новые уязвимости в компьютерных системах, в используемом ПО обнаруживаются достаточно регулярно. Согласно статистическим данным, которые легко можно получить на основе известного каталога уязвимостей (http://web.nvd.nist.gov), число уязвимостей, обнаруживаемых ежегодно, составляет 5...6 тыс. и более.

И это только уязвимости реализации, а есть еще ошибки проектирования и эксплуатации. Ведь в процессе эксплуатации система может изменяться. Разумеется, размер потенциального ущерба от использования конкретной уязвимости в конкретной защищаемой системе может быть довольно различ-

ным (от нулевого до значительного). Поэтому процесс управления уязвимостями должен обеспечивать не только своевременное выявление очередной уязвимости, но и адекватную оценку степени ее опасности для защищаемой системы, а также выбор соответствующего варианта реагирования. Фактически управление уязвимостями позволяет контролировать и поддерживать на определенном уровне степень защищенности системы, обеспечивая своевременное выявление ее слабостей. Этот процесс можно разложить на отдельные составляющие, и прежде всего инвентаризацию информационных активов.

19.2. Инвентаризация информационных активов

Инвентаризация информационных активов подразумевает создание и поддержание в актуальном состоянии единой базы IT-ресурсов CMDB (Configuration Management Database). Для ее создания необходимо определить:

- характер размещаемой в ней информации;
- механизмы накопления информации.

Характеризующая актив информация может быть различной, например, аппаратное обеспечение, ОС, ПО.

Кроме приведенных выше объективных показателей должна быть обеспечена возможность использования субъективных характеристик ресурса, например:

- категория;
- критичность;
- роль;
- владелец.

Накопление информации в такой базе может поддерживаться путем использования следующих механизмов:

- использование агентов сканирования, входящих в состав системы управления уязвимостями;
- использование локальных агентов инвентаризации;
- импорт из разных источников;
- ручной ввод информации.

Для системы управления уязвимостями эта информация крайне необходима, она может обеспечить принятие необходимого решения по устранению обнаруженных уязвимостей, поскольку многое зависит от роли узла и степени критичности.

19.3. Мониторинг состояния защищенности

Итак, система построена, информация о ней накапливается и обновляется. Следующий этап — мониторинг состояния ее защищенности, направленный на своевременное обнаружение любой слабости в системе. Следует

заметить, что иногда строится система, уже удовлетворяющая некоторому начальному уровню защищенности, и даже, при необходимости, ее состояние может быть как-то зафиксировано. В этом случае одной из задач мониторинга будет обнаружение отклонений от этого состояния. Причем не важно, что обнаружено в ходе мониторинга: очередная уязвимость реализации в используемом ПО или факт несанкционированного размещения и использования точки беспроводного доступа, что запрещено корпоративной политикой.

Таким образом, мониторинг включает в себя отслеживание информации:

- об уязвимостях, которые могут быть следствием ошибок проектирования, реализации или эксплуатации;
- отклонениях от требований, сформулированных на этапе построения системы;
- новых угрозах, например о начале эпидемии очередного сетевого червя.

При этом могут быть использованы следующие механизмы получения информации:

- уведомления вендоров;
- использование сканеров безопасности;
- мониторинг известных баз уязвимостей;
- использование систем управления обновлениями;
- «независимые» источники.

Как может помочь система управления уязвимостями? Фактически она должна содержать приведенные выше механизмы получения информации, а ее пользователи узнают об очередной уязвимости, просто анализируя результаты сканирования, ведь разработчик системы уже позаботился об обновлении сканирующих модулей.

Что касается поиска отклонений, то интерфейс системы может обеспечивать возможность удобного формулирования различных требований, отклонения от которых, собственно, и будут выявляться в ходе проверок. И здесь возникает еще один аспект анализа защищенности — *контроль соответствия* (Compliance Management). При желании задача контроля защищенности вообще может быть сведена к контролю соответствия. Например, необходимость устранения уязвимостей реализации используемого ПО может быть изложена одним требованием: «отсутствие устаревшего или уязвимого ПО». Проблема заключается только в формулировке этих требований или в выборе готового набора (стандарта), которому необходимо соответствовать. При этом система управления уязвимостями может обеспечить удобный интерфейс настройки под конкретную информационную систему, а также иметь готовые наборы требований.

Результат данного этапа — перечень актуальных уязвимостей или нарушений (отклонений), требующих реагирования.

19.4. Устранение уязвимостей и контроль

Устранение уязвимостей. Это наиболее сложный и трудоемкий этап. Сложность состоит в том, что именно на этом этапе приходится вносить изменения в корпоративную информационную систему. Следовательно, по каждой уязвимости или отклонению из полученного на предыдущем этапе списка нужно принимать решение (устранить, оставить как есть, разобраться и т. д.).

После принятия решения об устранении уязвимости следует обоснованный выбор варианта устранения, в случае необходимости можно прибегнуть к тестированию, поскольку внесение в систему изменений может привести к ее неработоспособности.

На практике обычно приходится выбирать один из следующих вариантов:

- обновление системы;
- установка «патча»;
- переход на новую версию;
- изменение конфигурации (workaround);
- отказ от использования уязвимого ПО.

После выбора варианта устранения проводится собственно устранение уязвимости, которое может происходить автоматически (в редких случаях) или вручную. В последнем случае может потребоваться разработка рекомендаций для лиц, задействованных в этом процессе. Обычной практикой при этом является процесс формирования заявок для систем типа Help Desk, Service Desk или других систем управления заявками. Соответствующий функционал может входить и в саму систему управления уязвимостями.

Таким образом, на этом этапе система должна обеспечить соответствующий workflow, начиная от принятия решения по уязвимости и заканчивая формированием заявки и назначением ответственного.

Контроль. Последняя составляющая процесса управления уязвимостями — контроль правильности устранения уязвимостей. Контроль может быть выполнен разными способами, например:

- путем использования сканирующих модулей;
- анализа журналов соответствующих систем.

При этом могут быть эффективны сравнительные отчеты, функционал отслеживания изменений или даже возможность отслеживания динамики изменения состояния защищенности системы.

Контрольные вопросы

1. Чем вызвана необходимость централизованного управления уязвимостями?
2. Раскройте суть инвентаризации информационных активов.
3. Перечислите задачи и способы мониторинга состояния защищенности.
4. К чему сводится устранение уязвимостей?
5. Какими способами осуществляется контроль правильности устранения уязвимостей?

Глава 20. КОНТРОЛЬ ЗАЩИЩЕННОСТИ БЕСПРОВОДНЫХ СЕТЕЙ

20.1. Особенности сканирования беспроводных сетей

В предыдущих главах были рассмотрены две задачи, решаемые сканерами: сбор информации и идентификация уязвимостей. Сбор информации о системе может выполняться двумя способами:

1) Active Fingerprinting — использование ключевых воздействий на систему и анализ откликов;

2) Passive Fingerprinting — использование информации, «добровольно» рассылаемой исследуемой системой.

Если для сканирования обычной сети обычно используются активные методы сбора информации, то для беспроводной сети часть проверок основывается на пассивном анализе трафика. Таким образом, главная особенность сканирования беспроводных сетей заключается в сочетании активных методов сбора информации и пассивного прослушивания эфира. При этом пассивные методы явно преобладают. Поэтому сканеры для беспроводных сетей не получили широкого распространения.

20.2. Сканеры для беспроводных сетей

Задачи сканирования беспроводных сетей. На сетевом уровне и выше анализ защищенности беспроводных сетей принципиально не отличается от анализа защищенности узлов обычной сети. Следует отметить лишь процедуру сканирования точки доступа как объекта, имеющего IP-адрес и открытые порты.

Уязвимости, характерные для беспроводных сетей. Как отмечалось выше, анализ защищенности беспроводной сети состоит из двух частей:

1) явное подключение к точке доступа и выполнение проверок;

2) прослушивание трафика и обнаружение различных проблем.

К первой части можно отнести следующие проверки:

• возможность подключения к точке доступа (без аутентификации, без знания ключа и т. п.);

• возможность получения IP-адреса у сервера DHCP, встроенного в точку доступа;

• возможность конфигурирования точки доступа через беспроводный интерфейс.

Последнюю проверку следует рассмотреть более подробно. Поскольку точка доступа имеет IP-адрес (для ее конфигурирования), он одинаково используется как с проводным интерфейсом, так и с беспроводным. Это означает, что существует угроза подключения к точке доступа через беспроводной интерфейс для изменения настроек. Однако этот адрес используется исключительно для конфигурирования точки доступа и не влияет на работу

узлов, использующих данную точку доступа. Следовательно, IP-адрес точки доступа может быть заменен другим, например 127.0.0.1. Это сделает невозможным подключение к точке доступа напрямую без предварительного сброса параметров в положение «по умолчанию», что уже предполагает наличие физического доступа. В целом можно дать следующие рекомендации по защите точки доступа от реконфигурации.

1. Настроить требуемые параметры точки доступа, подключившись к ней через web-интерфейс.

2. Сохранить конфигурацию в файле. Пример сохранения конфигурации для точки доступа D-Link DWL-2000AP приведен на рис. 20.1.

3. Присвоить точке доступа IP-адрес 127.0.0.1 (рис. 20.2).

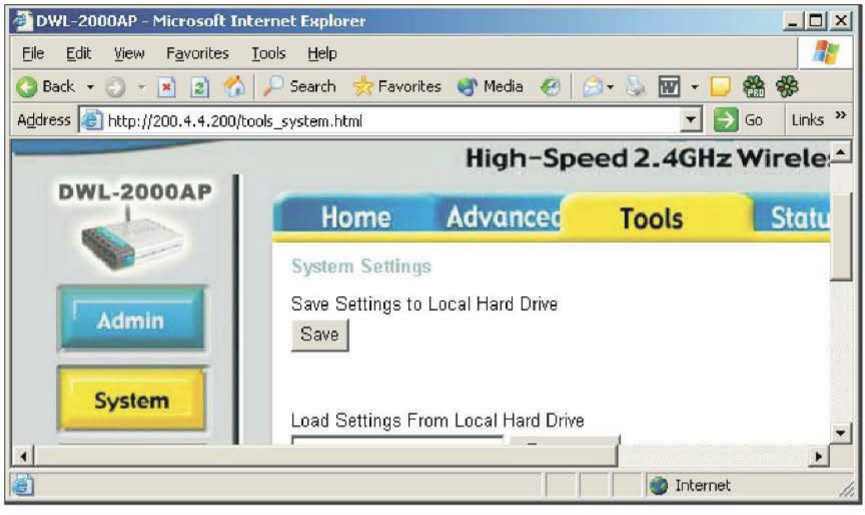


Рис. 20.1. Сохранение конфигурации точки доступа D-Link DWL-2000AP

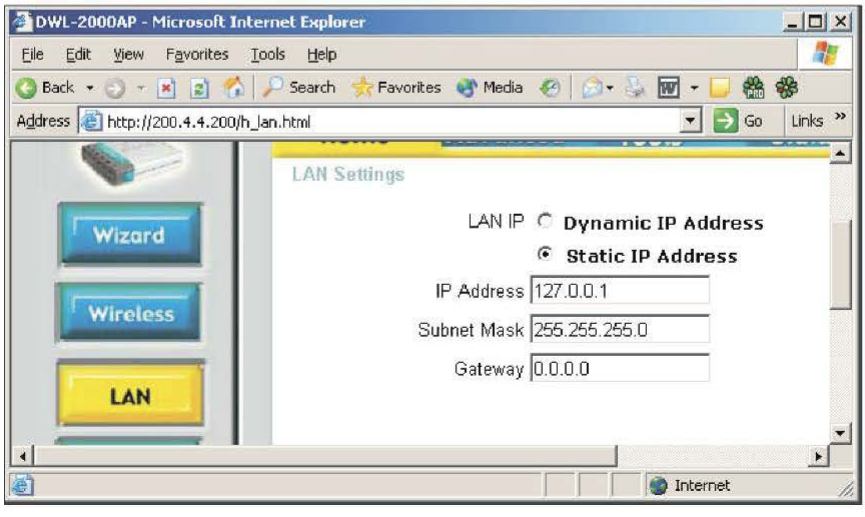


Рис. 20.2. Присвоение IP-адреса точке доступа D-Link DWL-2000AP

Если потребуется изменить конфигурацию, необходимы следующие действия:

- сбросить параметры точки доступа в положение «по умолчанию»;
- подключиться к точке доступа через web-интерфейс;
- загрузить параметры, ранее сохраненные в файле;
- внести изменения;
- сохранить параметры;
- присвоить точке доступа IP-адрес 127.0.0.1.

Ко второй части анализа защищенности (прослушивание трафика) относятся следующие проверки:

- задействование WPA (WPA2) в беспроводной сети;
- включение широковещательной рассылки фреймов «Beacon»;
- задание по умолчанию передаваемой точки доступа SSID.

Поскольку часть проверок основана на прослушивании эфира и анализе перехваченных фреймов, сканеры для беспроводных сетей имеют некоторое сходство с системами обнаружения атак. Например, следующие проверки, выполняемые программой Wireless Scanner, сходны с действиями системы обнаружения атак:

- неудачная попытка аутентификации клиента;
- неверный WPA (WPA2)-ключ.

20.3. Сканирование точки доступа на сетевом уровне

Точка доступа обычно имеет интерфейс для подключения к обычной сети, используемый для конфигурирования. На сетевом уровне точка доступа представляется как объект, имеющий IP-адрес, отвечающий на ICMP-запросы и имеющий открытые порты. Следовательно, можно выполнить ее сканирование сканером сетевого уровня, например Internet Scanner или XSpider.

Рассмотрим пример использования Internet Scanner. Первый этап работы сканера — идентификация узлов. Точка доступа может быть идентифицирована методом ICMP Ping. Второй этап — идентификация открытых портов. Как правило, точка доступа может поддерживать две службы: TCP — HTTP и SNMP.

На рис. 20.3 представлен результат работы сканера Internet Scanner.

Service Name	Port #	Protocol
httpd	80	TCP
snmp	161	TCP

Рис. 20.3. Результат работы сканера Internet Scanner

Сканирование портов UDP также может дать результаты. Обычно это две службы: TFTP и SNMP. На рис. 20.4 представлен результат работы сканера nmap.

Рис. 20.4. Результат работы сканера nmap

Далее следуют этапы идентификации служб (уточнения прикладной службы, обнаруженной на открытом порту) и определения ОС.

Последний этап — идентификация уязвимостей. Как правило, это уязвимости HTTP и SNMP. На рис. 20.5 представлены результаты работы сканера Internet Scanner. Видно, что большая часть найденных уязвимостей относится именно к SNMP.

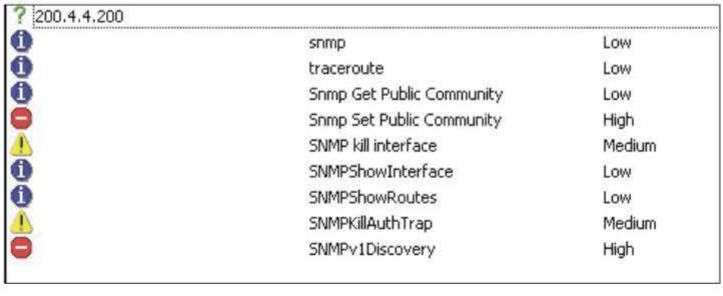


Рис. 20.5. Фрагмент отчета работы сканера Internet Scanner

Кроме того, значительную опасность представляет служба TFTP. Как известно, протокол TFTP не предусматривает аутентификации и применяется, например, для несанкционированной записи файла на сервер TFTP. Для точки доступа он может быть использован в целях обновления «прошивки» (Firmware). Следует еще раз отметить необходимость смены IP-адреса точки доступа.

20.4. Методология аудита

При анализе защищенности любой сети, в том числе и беспроводной, необходимо использовать методологию. При оценке защищенности беспроводной сети с учетом рассмотренных уязвимостей, инструментов и т. п. возможна следующая последовательность действий:

1) планирование, определение целей анализа защищенности;
2) исследование беспроводной сети, сбор информации;
3) анализ собранной информации, планирование атак с использованием найденных уязвимостей;
4) реализация атак;
5) восстановление систем.

Планирование, определение целей анализа защищенности. На этапе планирования определяются задачи, которые необходимо решить при анализе защищенности беспроводной сети. Как правило, анализ защищенности беспроводной сети выполняется в рамках анализа защищенности корпоративной сети в целом.

На выходе этого этапа получается документ (рис. 20.6).

Цель анализа защищенности	Повышение безопасности
Сетевой администратор	
Наличие политики безопасности беспроводной сети	

Рис. 20.6. Вид документа на этапе планирования

Исследование беспроводной сети, сбор информации. Второй этап — наиболее длительный и трудоемкий. Его можно разделить на несколько частей. Для начала необходимо определить зону охвата беспроводного участка. Для этого потребуется переносной компьютер, направленная антенна и программа для обнаружения беспроводной сети (например, Network Stumbler). Если точка доступа сконфигурирована таким образом, что она не рассылает широковещательно фреймы «beacon», для обнаружения беспроводной сети потребуется сетевой анализатор.

При этом должны получиться следующие результаты: зона охвата беспроводной сети и информация о беспроводной сети (SSID, используется ли WPA (WPA2)), модель точки доступа.

Следующая часть — запуск сетевого анализатора (например, AiroPeek) и прослушивание всего трафика сети, а не только широковещательного (как на предыдущем шаге). Цель — получение следующих результатов:

• информация о клиентах (в частности, MAC-адреса для последующего обхода аутентификации на основе MAC-адреса);

• информация об адресации сети для подключения к ней и последующего сканирования с целью поиска уязвимостей сетевого уровня и выше;

• информация из сессий прикладного уровня (имена, пароли, передаваемые в открытом виде, и т. п.).

Это будет возможным, если в беспроводной сети не используется WEP. Если же WEP задействован, выполняется еще одна часть процесса сбора информации — попытка получения WEP-ключа либо перехватом достаточного количества трафика, если не осуществляется фильтрация «weak IV», либо путем подбора ключа по словарю.

Еще одна составляющая этапа сбора информации — определение топологии сети, организации подключения к обычной сети, возможности подключения к точке доступа для ее конфигурирования. Результат — схема подключения беспроводного сегмента к корпоративной сети.

Все дальнейшие действия по сбору информации, в частности, получение списка активных узлов (например, с помощью методов Pingsweep и TCPSweep), идентификация открытых портов и сетевых служб, проводятся так же, как и в обычной сети. Результат — перечень обнаруженных уязвимостей.

На выходе данного этапа должны получиться документы, примеры которых приведены ниже.

Типы используемых беспроводных сетевых технологий

Тип	Используется
802.11b	
802.11a	
802.11g	
802.16	
802.15 (Bluetooth)	
HomeRF	
Другие	

Архитектуры беспроводных сетей

Архитектура	Используется
С точкой доступа (Infrastructure/Managed)	
Одноранговая (Independent/Ad-Hoc)	

Количество точек доступа _____

Количество беспроводных клиентов _____

Используемые каналы:

ESSID _____ Channel _____

ESSID _____ Channel _____

ESSID _____ Channel _____

ESSID _____ Channel _____

ESSID _____ Channel _____

Примерное описание охватываемой территории: _____

_____

_____

Используемые механизмы защиты:

Отключение широковещательной рассылки SSID во фреймах «beacon» _____

Фильтрация на основе MAC-адресов _____

Аутентификация (базовая 802.11):

Open System _____

Shared Key _____

Шифрование WEP _____

Длина ключа WAP (WAP2) _____

Аутентификация 802.11x _____

Наличие сервера RADIUS _____

Тип сервера — хранилища учетных записей пользователей (с указанием параметров) _____

_____

Наличие системы обнаружения атак для беспроводной сети _____

Использование «Honeypots» _____

Анализ собранной информации, планирование и реализация атак. Цель этих двух этапов — подтверждение (верификация) уязвимостей, выявленных на предыдущем этапе. Если атака проведена успешно, уязвимость считается подтвержденной.

Восстановление систем. В зависимости от ситуации этот этап может отсутствовать. Поскольку его цель — «наведение порядка» после проведения тестирования, необходимо наличие подробной документации по тем действиям, которые были совершены в процессе проведения тестирования.

Примеры шагов, выполняемых на данном этапе:

- удаление пользовательских учетных записей, созданных в процессе тестирования;
- восстановление систем, к которым был получен доступ;
- восстановление систем, выведенных из строя.

Этот этап может выполняться и силами организации, заказавшей проведение тестирования. Следует отметить, что в любом случае должен быть предоставлен подробный перечень внесенных в процессе тестирования изменений.

Контрольные вопросы

1. В чем заключаются особенности сканирования беспроводных сетей?
2. Перечислите задачи сканирования беспроводных сетей.
3. Каковы уязвимости, характерные для беспроводных сетей?
4. Опишите методологию сканирования точки доступа на сетевом уровне.
5. В чем заключается методология аудита беспроводных сетей?

Глава 21. ИСТОЧНИКИ ДАННЫХ ДЛЯ СИСТЕМ ОБНАРУЖЕНИЯ АТАК

21.1. Составляющие технологии обнаружения атак

В общем случае система обнаружения атак состоит из компонентов двух типов: компоненты управления и агенты (сенсоры, модули слежения).

При этом обычно в состав компонентов управления входят клиентские (управляющая консоль) и серверные компоненты различного назначения (рис. 21.1).

Например, в состав решения по обнаружению атак IBM Security входят сенсоры для защиты сегментов и отдельных узлов, в качестве системы управления используется система Site Protector, подключиться к ней пользователи могут либо с помощью консоли, либо через web-интерфейс (рис. 21.2).

Рис. 21.1. Компоненты системы обнаружения атак

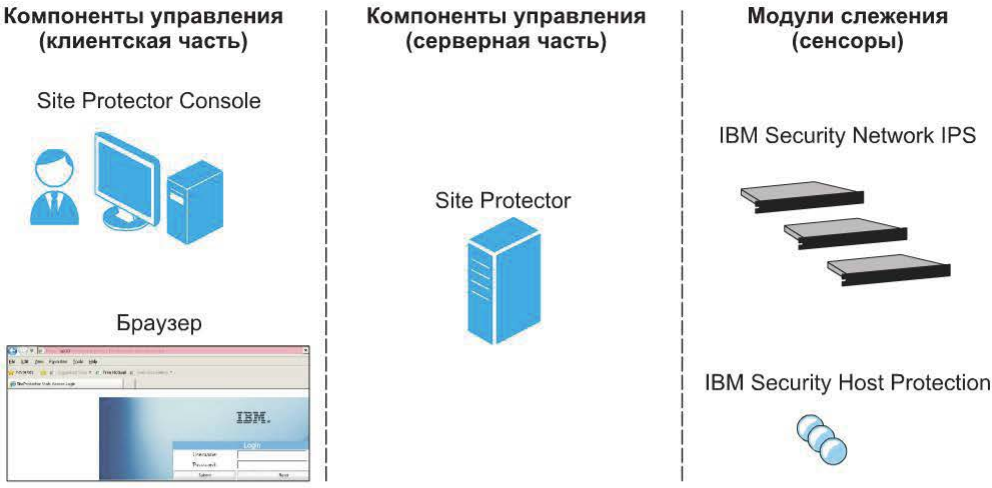


Рис. 21.2. Решение по обнаружению атак IBM Security

Аналогичный вид имеет решение по защите от атак Stonesoft Security Platform. В качестве компонентов управления здесь используются SMC (Stonesoft Management Center) и клиентские компоненты (Management Client и Web Start Java GUI). Модуль слежения может играть разные роли, в том числе технологий IDS/IPS (рис. 21.3).

Рис. 21.3. Решение по защите от атак Stonesoft Security Platform

В данном учебном пособии подробно рассмотрены возможности агентов (сенсоров), а не компонентов управления. Именно в модулях слежения «сосредоточена» технология (алгоритм) обнаружения атак, которая основана на следующих составляющих:

- признаки атак (что обнаруживать?);
- источники информации об атаках (где обнаруживать?);
- методы анализа информации об атаках (как обнаруживать?).

Рис. 21.4. Архитектура модуля слежения

В соответствии с этим модуль слежения (датчик, сенсор) имеет архитектуру, представленную на рис. 21.4.

В качестве источников данных могут быть использованы:

- сетевой трафик;
- журналы;
- действия субъектов системы.

Алгоритм обнаружения может строиться на основе понимания ожидаемого поведения контролируемого объекта или на основе знания всех возможных атак и их модификаций.

Механизмы реагирования могут быть довольно разнообразными, например, оповещение, блокировка, вызов внешней программы и др.

21.2. Сетевой трафик как источник данных

Подключение и схема работы сетевой IDS. Наиболее распространенный источник данных — сетевой трафик, поэтому первые системы обнаружения атак были именно сетевыми (рис. 21.5).

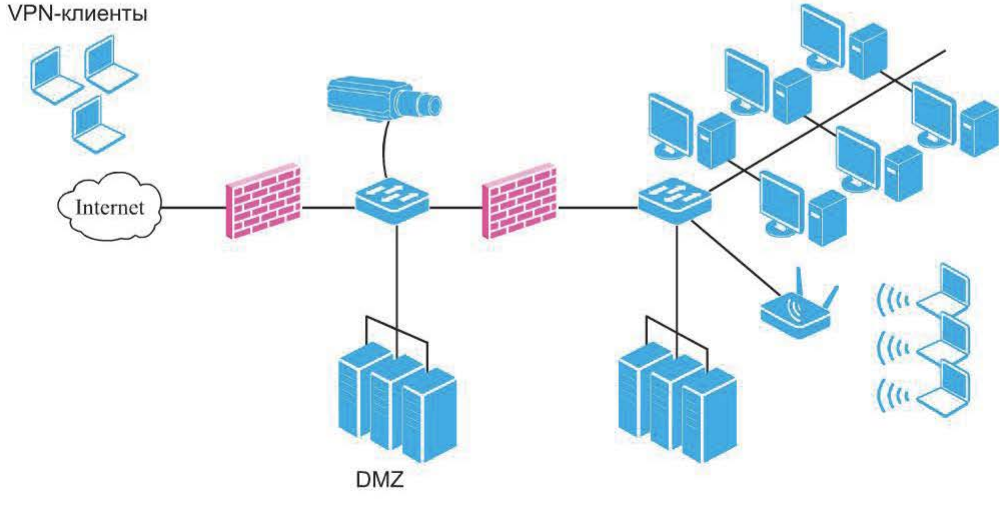


Рис. 21.5. Сетевая IDS

Данная система обычно подключается к SPAN-порту коммутатора и контролирует трафик какого-либо важного участка. Сетевой адаптер такой системы работает в неселективном режиме, захваченный трафик передается системе обнаружения атак (рис. 21.6).

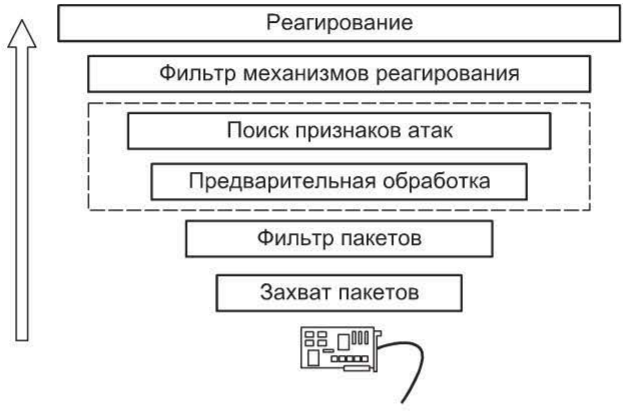


Рис. 21.6. Подключение и схема работы сетевой IDS

Варианты подключения сетевой IDS. Помимо использования SPAN-порта возможны и другие варианты подключения сетевых систем обнаружения атак с использованием концентраторов и разветвителей.

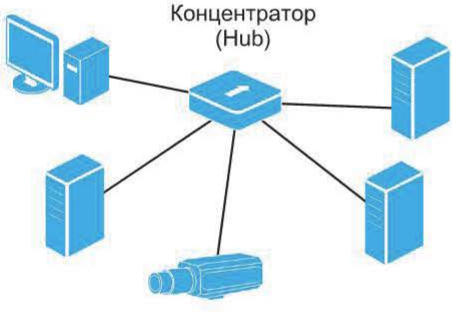


Рис. 21.7. Подключение IDS к любому порту концентратора

Рис. 21.8. Разрыв контролируемого участка сети

При использовании концентратора IDS может быть подключена к любому его порту (рис. 21.7).

Поскольку трафик копируется на все порты концентратора, его можно легко проанализировать на наличие признаков атак. В некоторых случаях концентратор используется как временное решение для разрыва контролируемого участка сети (рис. 21.8).

Использование концентраторов имеет свои преимущества и недостатки. К преимуществам относятся:

- простота;
- в большинстве случаев не требуется менять конфигурацию сети;
- удобно использовать для демонстрации возможностей NIDS;
- поддержка любых механизмов реагирования.

Однако имеются и существенные недостатки таких решений:

• концентраторы в настоящее время практически не используются;

• возможно снижение производительности.

В целом, использование концентратора — не рекомендуемое решение.

В настоящее время наиболее популярным является использование SPAN-порта (рис. 21.9).

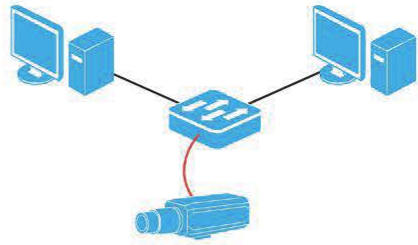


Рис. 21.9. Использование SPAN-порта

SPAN (*Switch Port Analyzer*) или *Mirror Port, Manage Port, Monitor Port, Analyzer Port* — это порт, на который копируется трафик с одного или нескольких портов коммутатора.

Источниками данных для SPAN-порта (рис. 21.10) служат трафик VLAN и трафик нескольких портов коммутатора.

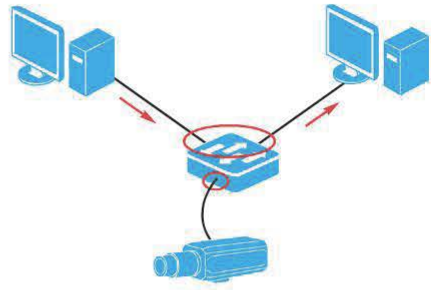


Рис. 21.10. Источники данных для SPAN-порта

На SPAN-порт могут копироваться как трафики Rx, так и Tx или оба трафика. В зависимости от этого SPAN-порт может иметь следующие конфигурации:

• **Receive SPAN (Rx)** — копируется входящий трафик контролируемых портов или VLAN-сетей до модификации и дальнейшей обработки;

• **Transmit SPAN (Tx)** — копируется трафик, покидающий контролируемые порты или VLAN-сети после модификации и обработки;

• **Transmit and Receive SPAN** — копируются оба трафика (в этом случае возможно получение двух копий одного и того же пакета).

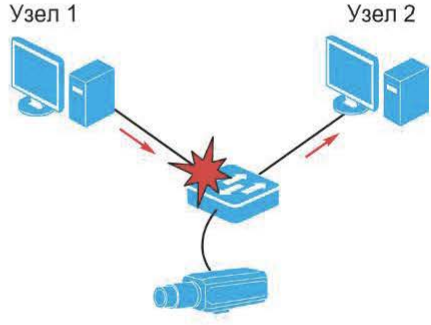


Рис. 21.11. Настройка конфигурации Receive SPAN

На рис. 21.11 приведен пример настройки конфигурации Receive SPAN.

В этом случае трафик обрабатывается в следующей последовательности:

1) пакет от узла 1 к узлу 2 достигает коммутатора;
2) пакет копируется на SPAN-порт коммутатора;
3) сенсор получает копию пакета;
4) пакет покидает коммутатор;
5) пакет достигает узла назначения.

К преимуществам использования SPAN-порта относятся:

• удобное решение, не требующее дополнительного оборудования;
• SPAN-порты поддерживаются большинством коммутаторов.

При использовании SPAN-порта необходимо учитывать следующее:

- не все коммутаторы поддерживают SPAN-порты;
- SPAN-порт может быть уже задействован анализатором протоколов и другими решениями по мониторингу трафика;
- возможны затруднения контроля нескольких VLAN-сетей;
- SPAN-порт может быть одно- или двунаправленным;
- в моменты пиковой загрузки коммутатора SPAN-порт может быть перегружен.

Следующим вариантом подключения системы обнаружения атак является использование разветвителей (рис. 21.12). Разветвители (Network Taps) подключаются между двумя сетевыми устройствами, соединенными через порты A и B.

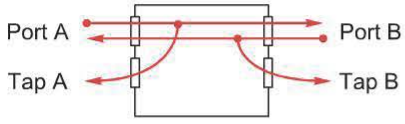


Рис. 21.12. Схема использования разветвителей

При этом данные Rx для порта A копируются в порт доступа Tap Port A, данные Rx для порта B — в порт Tap Port B. При этом трафик не должен передаваться в порты на разветвителе. Обычно разветвители подключают к критичному участку сети (или нескольким участкам), на котором требуется контролировать трафик (рис. 21.13).

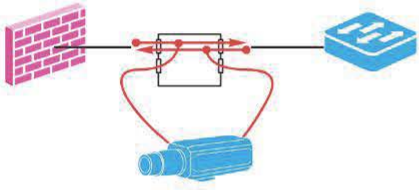


Рис. 21.13. Подключение разветвителей

Для подачи трафика на вход IDS порты Tap A и Tap B разветвителей могут быть объединены устройством балансировки нагрузки УБН (рис. 21.14).

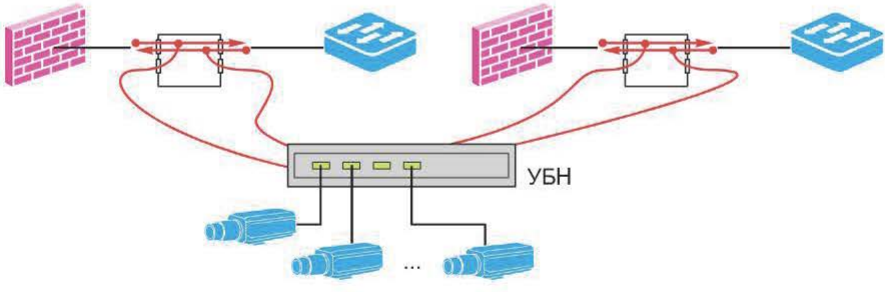


Рис. 21.14. Объединение разветвителей с помощью УБН (IDS Load Balancer)

При этом под балансировкой нагрузки может подразумеваться:

- анализ трафика сразу несколькими сенсорами;
- распределение нагрузки между несколькими сенсорами;
- распределение трафика между сенсорами на основе IP-адресов, протоколов и других характеристик.

Использование разветвителей имеет ряд преимуществ, в частности, минимальное влияние на сеть. Следует отметить и некоторые недостатки:

- ограничения в работе механизмов реагирования;

• необходимость решения для объединения трафика на сенсоре;

• высокая стоимость.

Обычно сетевая IDS работает в скрытом режиме (рис. 21.15).

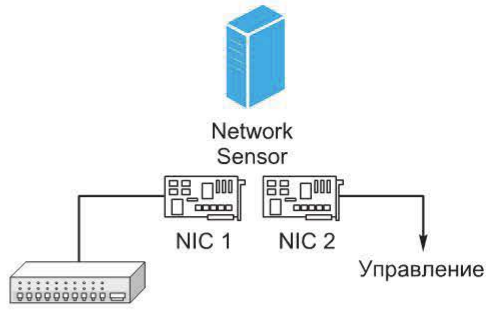


Рис. 21.15. Работа сетевой IDS в скрытом режиме

Такая конфигурация требует присутствия как минимум двух сетевых адаптеров на узле с сетевым сенсором:

• NIC 1 — подключен к контролируемому сегменту, к нему не привязан стек протоколов, т. е. интерфейс не имеет IP-адреса;

• NIC 2 — подключен к внутреннему защищенному сегменту, для него настроен стек TCP/IP и все необходимые службы.

Для ОС Windows настройка скрытого режима осуществляется путем отключения всех параметров сетевого интерфейса (рис. 21.16).

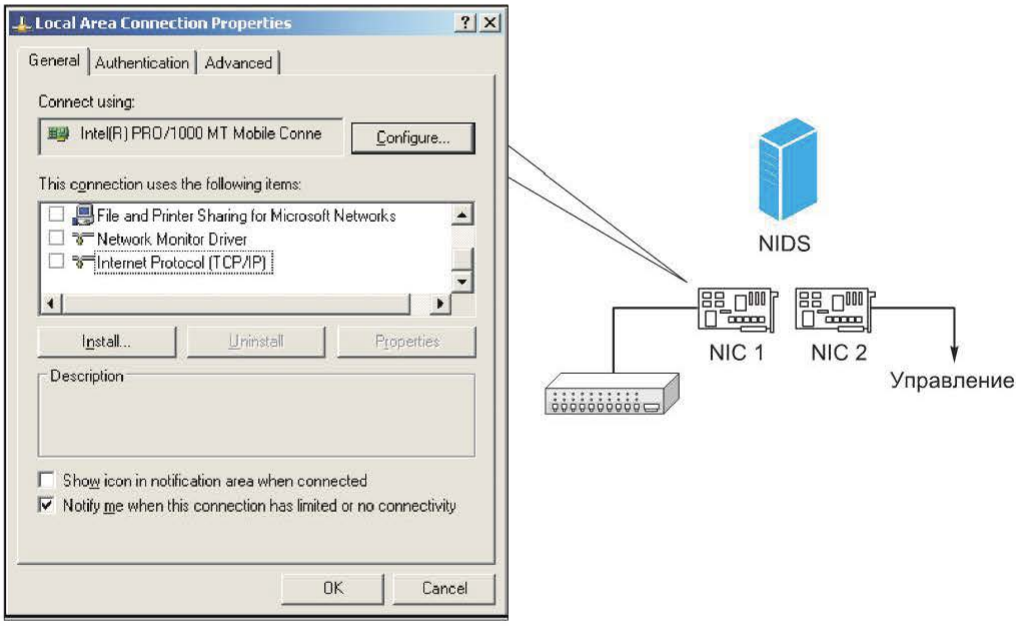


Рис. 21.16. Настройка скрытого режима для ОС Windows

Скрытый режим имеет следующие преимущества:

• невозможность обнаружения злоумышленником сетевого сенсора;

• недоступность для атак, требующих IP-адреса сетевого сенсора;

• в контролируемом сегменте не передаются управляющий и служебный трафики, связанные с работой системы обнаружения атак, а также собранные данные.

Единственный вид атак, к которым сенсор остается уязвимым даже в скрытом режиме, — это атаки с использованием ошибок парсинга. Обычно результат такой атаки — выведение из строя ПО сетевого сенсора (рис. 21.17).

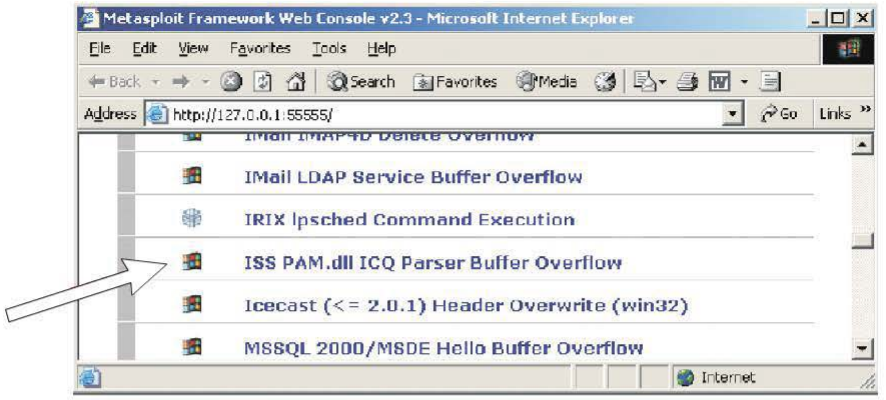


Рис. 21.17. Атака с использованием ошибок парсинга

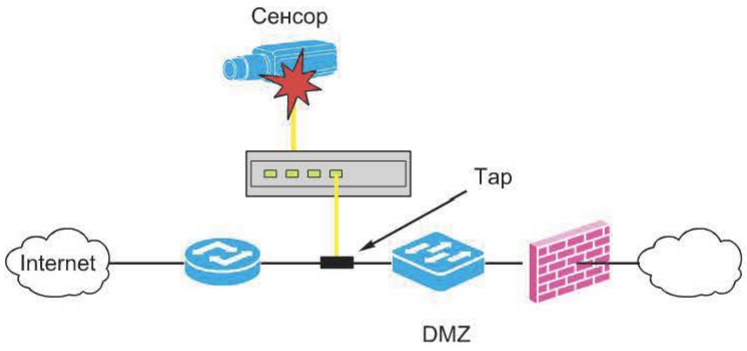


Рис. 21.18. Механизм проведения атаки

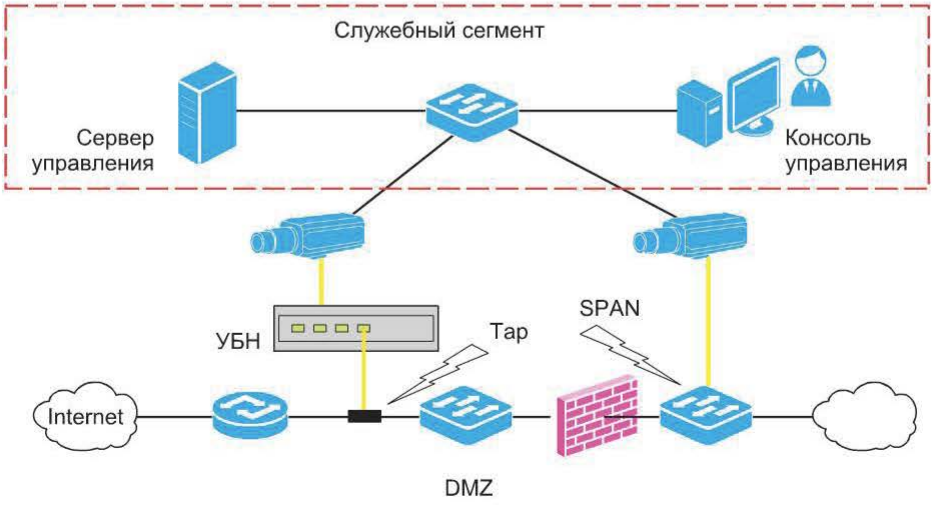


Рис. 21.19. Типовая схема подключения сетевой системы обнаружения атак

Механизм проведения атаки основан на посылке в сеть специальным образом построенного пакета, при перехвате и анализе которого сенсор выходит из строя (рис. 21.18).

Типовая схема подключения сетевой системы обнаружения атак приведена на рис. 21.19.

21.3. Обнаружение атак на уровне узла

Ниже приведены случаи, в которых может быть недостаточно сетевой системы обнаружения атак:

- шифрование трафика;
- наличие высокоскоростных участков;

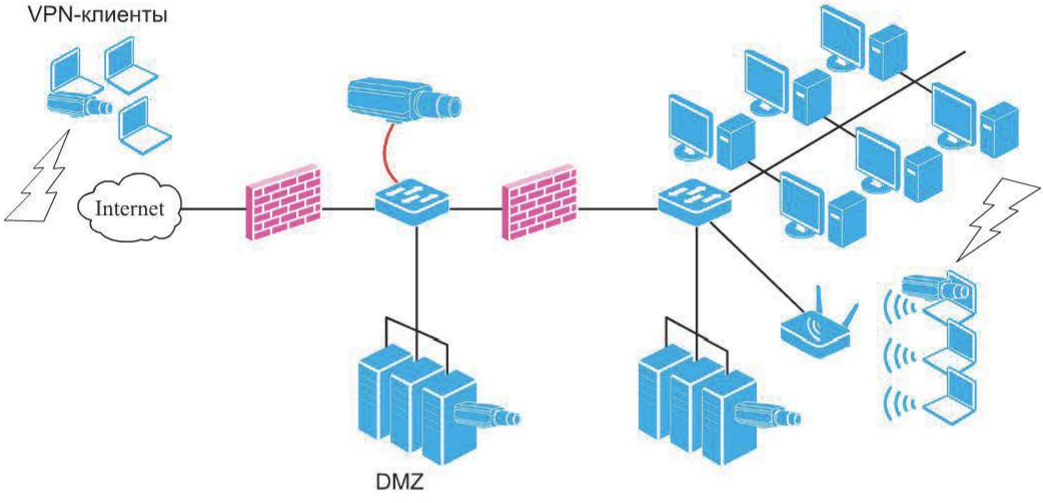


Рис. 21.20. Система обнаружения атак уровня узла

- наличие иных источников данных (кроме сетевого трафика);
- специфика расположения объекта.

В приведенных случаях более эффективно поместить систему обнаружения атак непосредственно на защищаемом узле (рис. 21.20).

Источниками данных для таких систем являются:

- сетевой трафик данного узла;
- журналы ОС и приложений;
- действия субъектов системы.

В части контроля сетевого трафика система предотвращения вторжений HIPS обычно имеет следующий функционал (рис. 21.21):

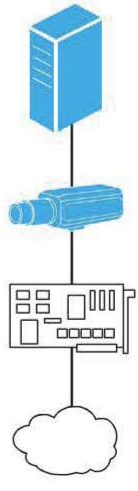


Рис. 21.21. Контроль сетевого трафика HIPS

- анализ трафика на наличие признаков атак (аналогично NIDS);
- фильтрация трафика.

Рассмотрим второй источник данных — журналы событий.

Журнал событий (*лог*, англ. log) — объект (например, файл), содержащий перечень событий, произошедших с различными активами организации (с системами или сетями).

Обычно журнал событий представляет собой совокупность записей (entries), каждая из которых содержит информацию, относящуюся к отдельному событию с системой или сетью.

Обычно для HIDS интерес представляют журналы, содержащие события безопасности, например:

- журналы средств защиты;
- журналы ОС, приложений и СУБД.

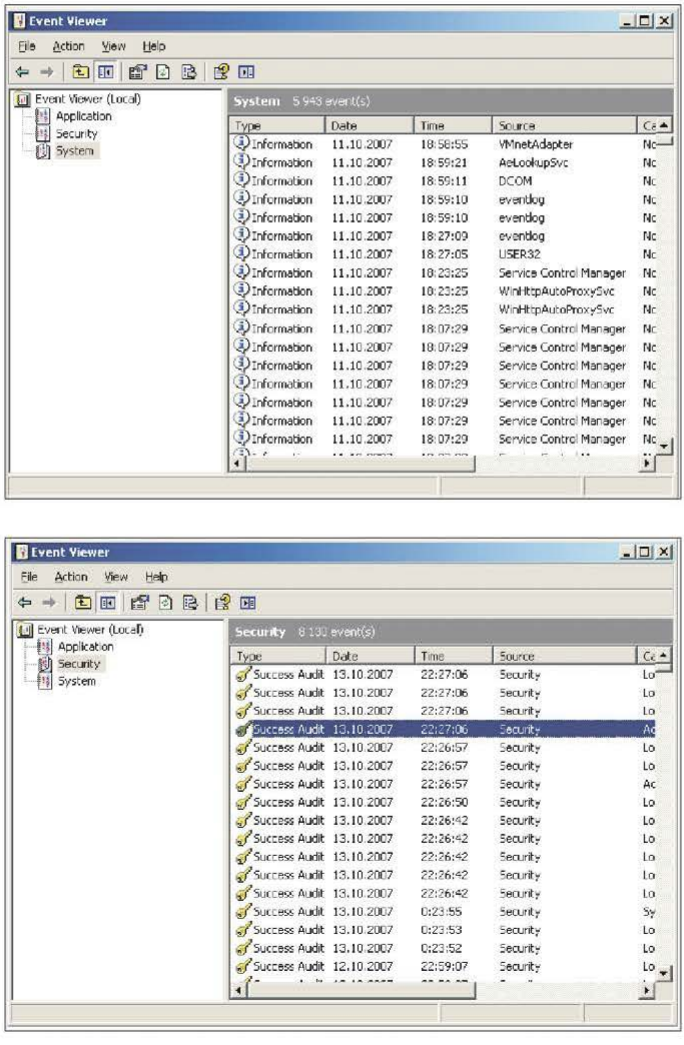


Рис. 21.22. Системный журнал и журнал аудита Windows

Журналы средств защиты, записи в которых могут быть использованы HIDS:

- межсетевые экраны;
- средства противодействия вредоносному коду (системы управления уязвимостями, серверы аутентификации, серверы контроля доступа к сети).

На уровне ОС Windows интерес представляют системные журналы и журналы аудита (рис. 21.22).

Наконец, на защищаемом узле часто контролируются журналы приложений, в частности почтовый, web- и файловый серверы.

21.4. Host IDS — контроль действий субъектов системы

Еще один источник данных для HIDS — действия субъектов системы. Под *субъектом* понимается активная сущность, например процесс. Такой контроль можно разбить на две части:

1) анализ выполняемого кода, к нему относятся обнаружение и предотвращение ситуаций переполнения буфера, а также так называемый анализ поведения;

2) мониторинг файловой системы, под ним следует понимать контроль целостности, контроль попыток обращения к критичным файлам, включая контроль запускаемых приложений, а также антивирусный контроль.

Предотвращение переполнения буфера строится на известной структуре эксплойта, предполагающей наличие так называемой «полезной» нагрузки (Payload) (рис. 21.23).

Один из методов реализации этой техники — «перехват вызовов API-функций» — представлен на рис. 21.24.

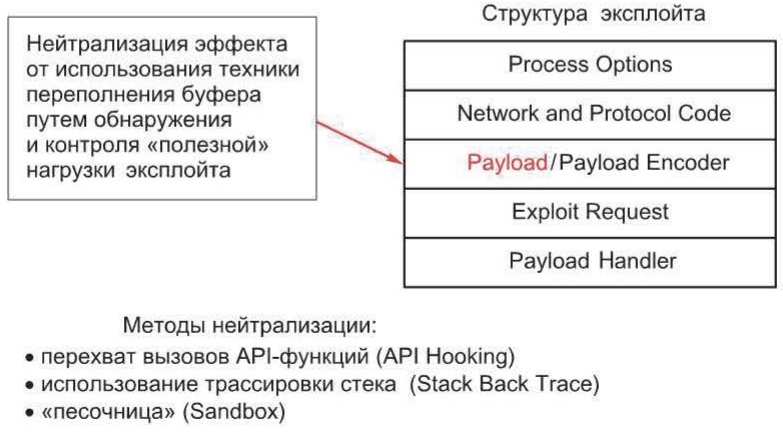


Рис. 21.23. «Полезная» нагрузка (Payload)

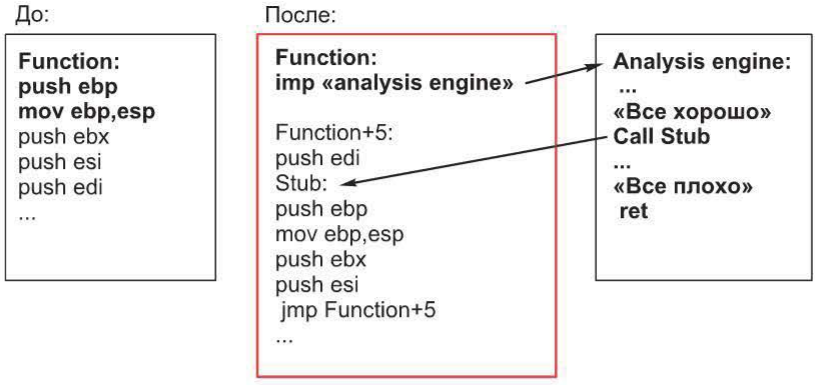


Рис. 21.24. Перехват вызовов API-функций

Рис. 21.25. Анализ «поведения» в виртуальной среде

Анализ «поведения» предполагает запуск тестируемого кода в виртуальной среде, сбор, классификацию, а также изучение его действий и принятие решения о разрешении или запрете исполнения данного кода (рис. 21.25).

21.5. Составляющие обнаружения атак уровня узла

Для систем обнаружения атак уровня узла существует соответствующая специфика, которая предполагает две составляющие защиты на сетевом и на прикладном уровнях.

На сетевом уровне можно выделить следующие составляющие (рис. 21.26):

- блокировка попыток использования переполнения буфера (поз. *1*);
- обнаружение и блокировка атак (поз. *2*);
- фильтрация трафика (поз. *3*).

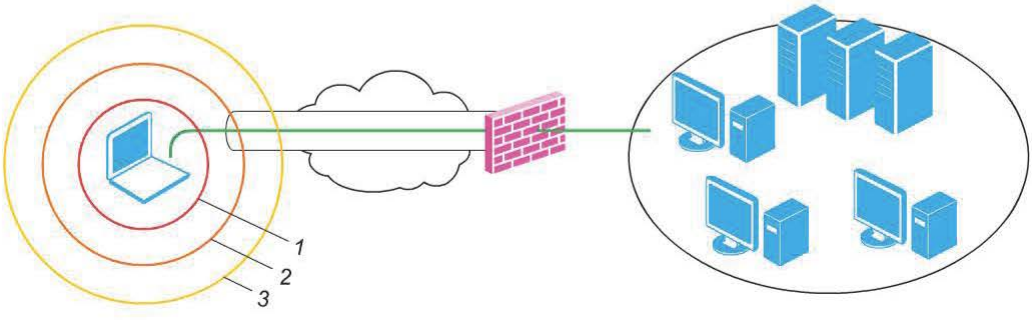


Рис. 21.26. Защита на сетевом уровне

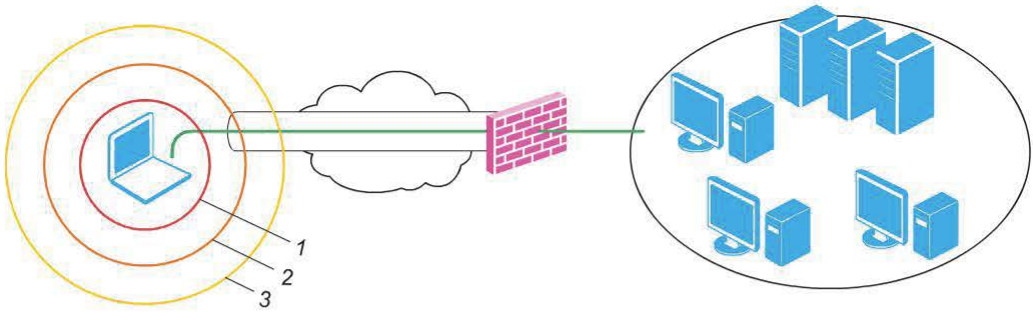


Рис. 21.27. Защита на прикладном уровне

На прикладном уровне также имеются три составляющие (рис. 21.27):
- анализ «поведения» (поз. *1*);
- контроль файловых операций и приложений (поз. *2*);
- антивирусные системы (на основе сигнатур) (поз. *3*).

21.6. Анализ данных о потоке

Говоря об источниках данных, следует рассмотреть специфику анализируемого трафика (рис. 21.28) и характеристику его потока (табл. 21.1).

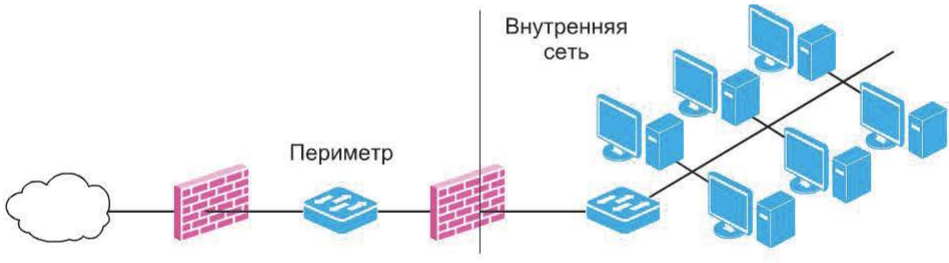


Рис. 21.28. Специфика анализируемого трафика

Таблица 21.1

Характеристика анализируемого трафика потока

Сеть	Десятки узлов (Мегабиты трафика)	Тысячи узлов (Гигабиты трафика)
Приложения	Десятки приложений	Сотни приложений
Политика	Запрет по умолчанию	Разрешение по умолчанию

Во внутренней сети анализируются не отдельные сетевые пакеты, а агрегированные данные. В связи с этим возникает еще один источник данных — Network Flow Data.

Поток (Flow) — «однонаправленная» последовательность сетевых пакетов между двумя узлами сети. Однозначно определяется следующими атрибутами:

- source IP address;
- destination IP address;
- source port number;
- destination port number;
- protocol type;
- type of services;
- router input interface.

Данные о потоке могут включать в себя:

- количество переданных данных;
- время начала (окончания) соединения.

Архитектура Network Flow приведена на рис. 21.29.

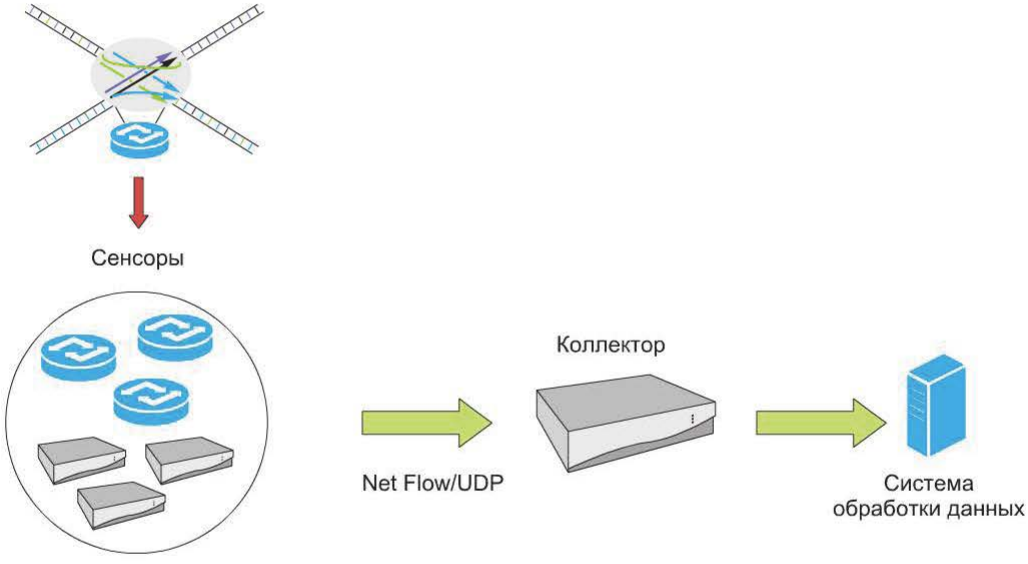


Рис. 21.29. Архитектура Network Flow

Первый стандарт в этой области — протокол sFlow (RFC3176) — предназначен для мониторинга трафика в коммутируемых и сегментированных сетях. В настоящее время на статус стандарта претендует протокол IPFIX (IP Flow Information eXport).

Контрольные вопросы

1. Перечислите составляющие технологии обнаружения атак.
2. Опишите архитектуру сетевой IDS, ее достоинства и недостатки.
3. В чем заключается специфика обнаружения атак на уровне узла? Опишите ее достоинства и недостатки.
4. Как использовать особенности архитектуры Network Flow для обнаружения атак?

Глава 22. ПРИЗНАКИ АТАК

При обнаружении атаки практически всегда можно назвать характерные признаки, на основе которых был сделан вывод о наличии атаки. Например, события, приведенные в фрагменте журнала системы обнаружения атак snort на рис. 22.1, были зафиксированы вследствие использования при подключении к серверу FTP «характерных» имен (в команде USER).

```
[**] [1:144:10] FTP ADMw0rm ftp login attempt [**]
[Classification: An attempted login using a suspicious username was
riority: 2]
12/18/08-12:44:45.557224 192.168.108.224:34398 -> 192.168.104.252:21
TCP TTL:64 TOS:0x10 ID:20814 IpLen:20 DgmLen:63 DF
***AP*** Seq: 0x110354AF  Ack: 0xA8EC2314  Win: 0xB7  TcpLen: 32
TCP Options (3) => NOP NOP TS: 4008388 248409
[Xref => http://www.whitehats.com/info/IDS01]

[**] [1:354:6] FTP iss scan [**]
[Classification: An attempted login using a suspicious username was
riority: 2]
12/18/08-12:44:56.969168 192.168.108.224:34398 -> 192.168.104.252:21
TCP TTL:64 TOS:0x10 ID:20816 IpLen:20 DgmLen:67 DF
***AP*** Seq: 0x110354BA  Ack: 0xA8EC2335  Win: 0xB7  TcpLen: 32
TCP Options (3) => NOP NOP TS: 4019800 248501
[Xref => http://www.whitehats.com/info/IDS331]
```

Рис. 22.1. Фрагмент журнала системы обнаружения атак snort

Признаки, на основе которых можно сделать вывод о наличии атаки, могут быть самыми разнообразными. Кроме того, они могут зависеть от конкретного окружения, например, в одном случае признаком атаки является передача по сети файла с определенным содержимым, в другом — подключение к серверу с определенного адреса.

Понимание признаков атак важно при анализе событий, так как позволяет точно указать причину срабатывания той или иной сигнатуры.

Предложить полный перечень признаков атак достаточно проблематично, укажем некоторые из них:

- использование уязвимостей;
- отклонения от пороговых значений;
- использование известных техник и инструментов для проведения атак;
- отклонения от известных моделей поведения сетевых протоколов.

22.1. Использование уязвимостей как признак атаки

Обычно для атаки используется какая-либо уязвимость. Поэтому практически любой атаке можно поставить в соответствие используемую при ее проведении уязвимость. В связи с этим многие признаки атак строятся на

основе уязвимостей, при этом чаще всего причиной этих уязвимостей являются ошибки реализации.

Например, уязвимость CVE-2000-0738, обнаруженная в антивирусном ПО WebShield SMTP, может быть использована для выведения его из строя путем отправки электронного письма с точкой в конце адреса e-mail (рис. 22.2).

Соответствующая сигнатура для ее обнаружения (Email_Recipient_Dot) срабатывает при использовании точки в конце адреса e-mail (рис. 22.3).

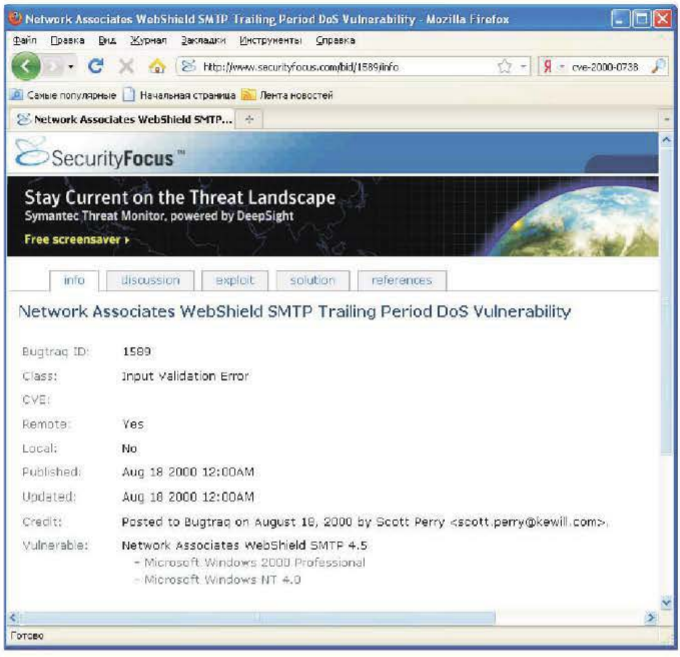


Рис. 22.2. Уязвимость CVE-2000-0738

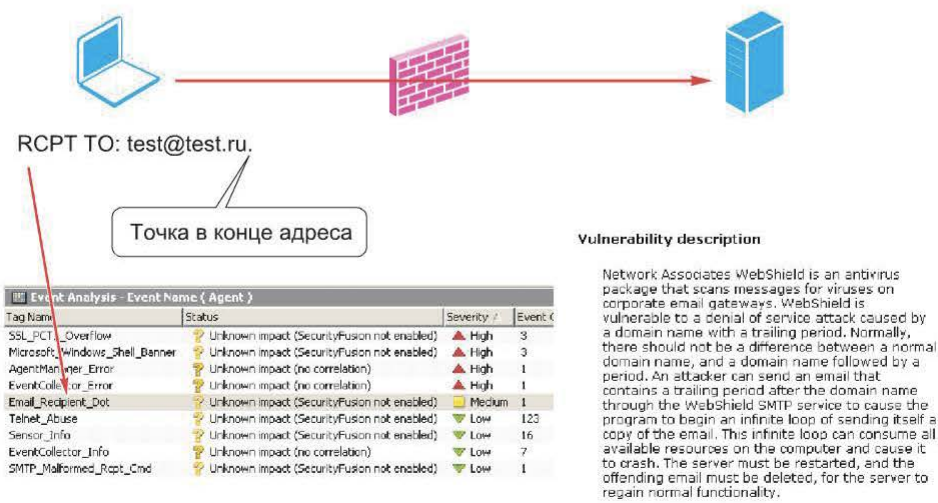


Рис. 22.3. Сигнатура для обнаружения уязвимости CVE-2000-0738

По аналогичному принципу выбран признак атаки для сигнатуры HTTP_Apache_SlashSlash. Как видно из ее описания (рис. 22.4), в качестве признака атаки используется «двойной слэш» в конце запроса HTTP_Get. Описание таких сигнатур обычно содержит указание на уязвимость, например приводятся ее название и номер в одном из известных каталогов.

HTTP_Apache_SlashSlash

Description

This signature detects an HTTP GET followed by a double slash.

Type	Attack
Priority	medium
Protocol	URL - Uniform Resource Locator
Algorithm Id	2105083
Bugtraq References	BID-8898
CVE References	CVE-2003-1138

Vulnerabilities Exploited

- Apache GET request directory traversal

Рис. 22.4. Признак атаки для сигнатуры HTTP_Apache_SlashSlash

Разумеется, срабатывание таких сигнатур не указывает однозначно на наличие атаки, поскольку может быть обусловлено и обычной сетевой активностью. Точка в конце адреса e-mail может быть добавлена пользователем случайно, так же, как и «двойной слэш» в конце URL.

При появлении подобных событий необходимо обратить внимание на объект атаки и выяснить, действительно ли используется уязвимое приложение. Кроме того, следует проанализировать события, произошедшие в близкие моменты времени и направленные с адреса — источника атаки. Если зафиксировано много подобных событий, это может быть признаком сканирования объекта атаки на наличие уязвимостей.

22.2. Отклонения от пороговых значений

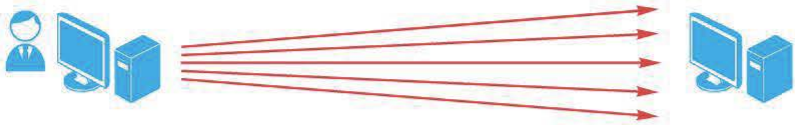
Довольно часто используемым признаком атаки является нарушение допустимых пороговых значений. К таким значениям относятся:

- число однотипных событий в единицу времени;
- допустимые числовые значения определенных параметров.

Повтор событий. В первом случае при обнаружении повтора определенных событий используются следующие пороговые значения:

• число событий;
• интервал времени, в течение которого произошли эти события.

На рис. 22.5 приведено описание сигнатуры для обнаружения слишком большого числа соединений с сервером SMTP, направленной с одного и того же IP-адреса.



Большое число подключений к почтовому серверу
с одного и того же адреса

SMTP service has too many concurrent connected users
(SMTP_Too_Many_Connects)

About this signature or vulnerability

RealSecure Network Sensor, RealSecure Server Sensor:

This signature triggers when it detects too many SMTP connections from the same IP. The thresholds are adjustable using the smtp.connection.count and smtp.connection.interval tuning parameters.

Рис. 22.5. Сигнатуры большого числа соединений с сервером SMTP

Из описания следует, что работа данной сигнатуры регулируется двумя параметрами:

• числом подключений (smtp.connection.count);
• интервалом времени (smtp.connection.interval).

По этому же принципу обнаруживается, например, сканирование портов. В этом случае в качестве параметров используются число запросов к различным портам и временной интервал, в течение которого эти запросы произошли.

Пороговым значением может служить, например, число неудачных попыток входа в систему.

Email_Mime_Name_Overflow

Description

This signature detects if the name following the Content-Type: field in a MIME header exceeds 300 characters, which would indicate an attacker's attempt to overflow a buffer in the recipient's mail client.

Type	Attack
Priority	▲ high
Protocol	MIME - Multipurpose Internet Mail Extensions
Algorithm Id	2102021
Bugtraq References	BID-761
Recommended Responses	• Rewrite

Рис. 22.6. Допустимые размеры определенных числовых параметров

Отклонения от допустимых значений. Второй тип пороговых значений — допустимые размеры определенных числовых значений, например, параметров, передаваемых какому-либо приложению. Использование такого признака позволяет обнаружить попытки создания ситуации переполнения буфера путем передачи приложению параметров недопустимой длины. Пример сигнатуры, основанной на таком принципе, приведен на рис. 22.6. Эта сигнатура срабатывает, если длина значения одного из полей заголовка MIME превысит 300 символов.

22.3. Использование известных техник и инструментов для проведения атак

Иногда признаком атаки может быть не известная уязвимость, а особенности инструмента, используемого нарушителем. В качестве примера можно привести сигнатуру Nmap_OS_Fingerprint, срабатывающую при обнаружении использования программы nmap для сбора информации об узлах сети (рис. 22.7).

Кроме того, в качестве признака атаки могут использоваться определенные техники, например, довольно распространен запуск какой-нибудь «полезной» нагрузки. Очень часто такой «полезной» нагрузкой оказывается так

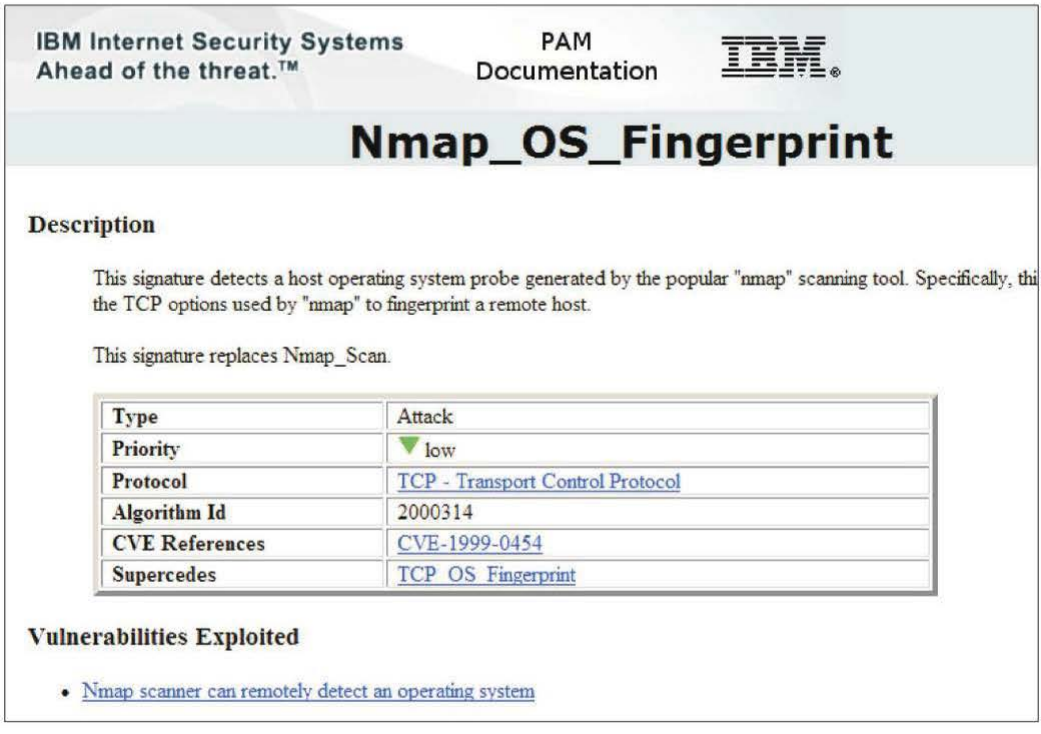


Рис. 22.7. Сигнатура Nmap_OS_Fingerprint

```
(msg:"SHELLCODE HP-UX NOOP"; content:"|08|!|02 80 08|!|02 80 08|!|02 80 08|!|02 80|";
(msg:"SHELLCODE HP-UX NOOP"; content:"|0B|9|02 80 0B|9|02 80 0B|9|02 80 0B|9|02 80|";
(msg:"SHELLCODE sparc NOOP"; content:"|13 C0 1C A6 13 C0 1C A6 13 C0 1C A6 13 C0 1C A
(msg:"SHELLCODE sparc NOOP"; content:"|80 1C|@|11 80 1C|@|11 80 1C|@|11 80 1C|@|11|";
(msg:"SHELLCODE sparc NOOP"; content:"|A6 1C C0 13 A6 1C C0 13 A6 1C C0 13 A6 1C C0 1
(msg:"SHELLCODE x86 NOOP"; content:"|90 90 90 90 90 90 90 90 90 90 90 90 90 90|"; ref
(msg:"SHELLCODE x86 stealth NOOP"; content:"|EB 02 EB 02 EB 02|"; reference:arachnids
(msg:"SHELLCODE x86 0x90 unicode NOOP"; content:"|90 00 90 00 90 00 90 00 90 00|"; cl
(msg:"SHELLCODE Linux shellcode"; content:"|90 90 90 E8 C0 FF FF FF|/bin/sh"; referen
(msg:"SHELLCODE x86 inc ebx NOOP"; content:"CCCCCCCCCCCCCCCCCCCCCCCC"; classtype:shel
(msg:"SHELLCODE x86 NOOP"; content:"AAAAAAAAAAAAAAAAAAAAAAAA"; classtype:shellcode-d
(msg:"SHELLCODE x86 0xEB0C NOOP"; content:"|EB 0C EB 0C EB 0C EB 0C EB 0C EB 0C EB 0C
(msg:"SHELLCODE x86 0x71FB7BAB NOOP"; content:"q|FB|{|AB|q|FB|{|AB|q|FB|{|AB|q|FB|{|A
(msg:"SHELLCODE x86 0x71FB7BAB NOOP unicode"; content:"q|00 FB 00|{|00 AB 00|q|00 FB
```

Рис. 22.8. Примеры сигнатур

называемый шелл-код. В свою очередь, характерным признаком шелл-кода является наличие идущих подряд ассемблерных инструкций NOOP, что может являться признаком атаки. На рис. 22.8 приведено несколько сигнатур такого типа из базы системы обнаружения атак Snort.

22.4. Система обнаружения атак Snort

Snort — свободно распространяемая система обнаружения атак сетевого уровня.

Программа может работать в четырех режимах:

- **анализ трафика** (Sniffer Mode) — простой вывод на экран содержимого сетевых пакетов;
- **запись трафика** (Packet Logger) — запись содержимого пакетов в файл на диске;
- **обнаружение атак** (Intrusion Detection System) — обнаружение событий на основе заданных правил конфигурации;
- **предотвращение атак** (Inline) — обнаружение атак с возможностью блокировки.

В режиме анализа трафика содержимое пакетов отображается на экране. При этом возможны следующие варианты запуска системы:

`/snort -v` — вывод содержимого заголовков IP/ICMP/UDP/TCP;

`/snort -vd` — вывод содержимого заголовков IP/ICMP/UDP/TCP и поля данных;

`/snort -vde` — вывод содержимого заголовков канального уровня, заголовков IP/ICMP/UDP/TCP и поля данных.

В режиме обнаружения атак Snort реагирует на определенные события в соответствии с правилами из файла snort.conf:

```
/snort -vde -l /log -c snort.conf
```

Для захвата сетевых пакетов snort использует библиотеку libpcap. При старте snort считывает содержимое файла snort.conf и загружает в память все указанные в нем сигнатуры. Каждый полученный пакет обрабатывается

декодером (Packet Decoder). Декодер разделяет пакеты IP, UDP, ICMP, TCP для дальнейшего анализа, просматривая заголовки сетевого и транспортного уровней. Если сигнатуры основаны только на сочетании определенных значений полей этих заголовков, они срабатывают уже после обработки трафика декодером. После этого пакеты передаются соответствующим препроцессорам. В составе системы Snort имеется несколько препроцессоров для некоторых протоколов, например HTTP (рис. 22.9).

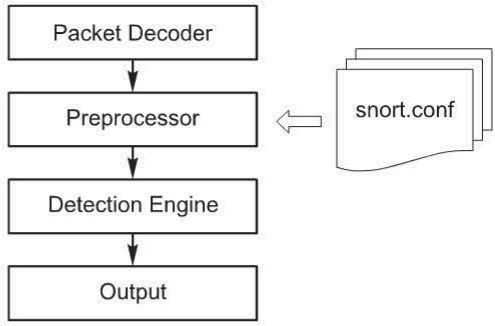


Рис. 22.9. Препроцессоры Snort

Популярность системы Snort во многом обусловлена удобством и простотой языка описания сигнатур (правил). Пример правила системы Snort (сигнатуры) приведен на рис. 22.10.

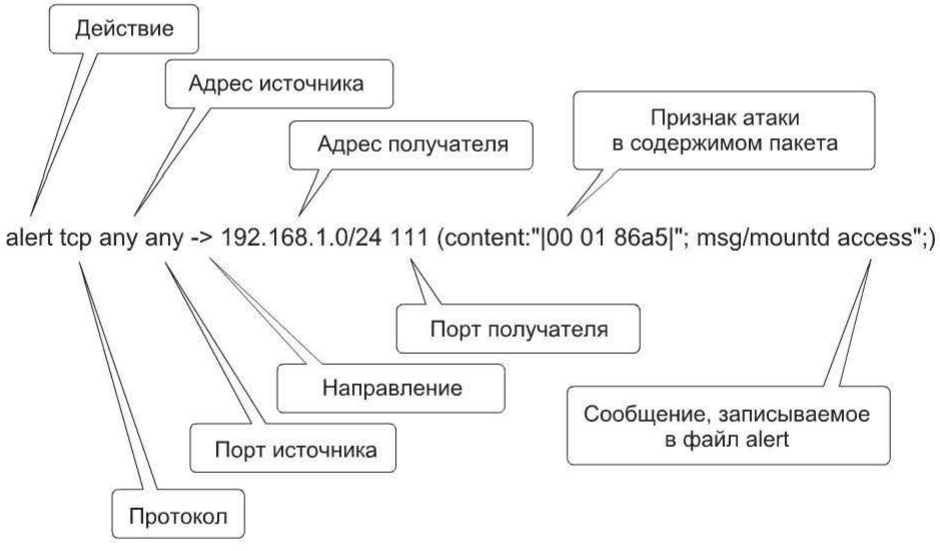


Рис. 22.10. Пример правила системы Snort

Система обнаружения атак Snort поставляется без сигнатур. Возможны следующие варианты получения сигнатур и обновлений к ним:

- по мере выхода (для этого необходима платная подписка);
- через 30 дней с момента появления обновления (при бесплатной регистрации);
- набор сигнатур на момент выхода последней версии Snort.

Отклонения от известных моделей поведения. Данный признак обычно применяется к трафику сервисов прикладного уровня. Признаками атак в этом случае являются «неправильные» или «подозрительные» команды.

В частности, к ним относятся:

- неправильная последовательность команд;
- ошибки в синтаксисе команд;
- «подозрительные» аргументы.

В качестве примера использования неправильной последовательности команд можно привести сигнатуру 5748/0 Non-SMTP Session Start, встроенную в Cisco IPS (рис. 22.11). Она срабатывает, если сессия SMTP не начинается с команд HELO или EHLO.

Рис. 22.11. Сигнатура 5748/0 Non-SMTP Session Start

Примером использования подозрительного аргумента в качестве признака атаки может служить сигнатура HTTP_BAT_Execute, срабатывающая при обращении по протоколу HTTP к файлу с расширением .bat (рис. 22.12).

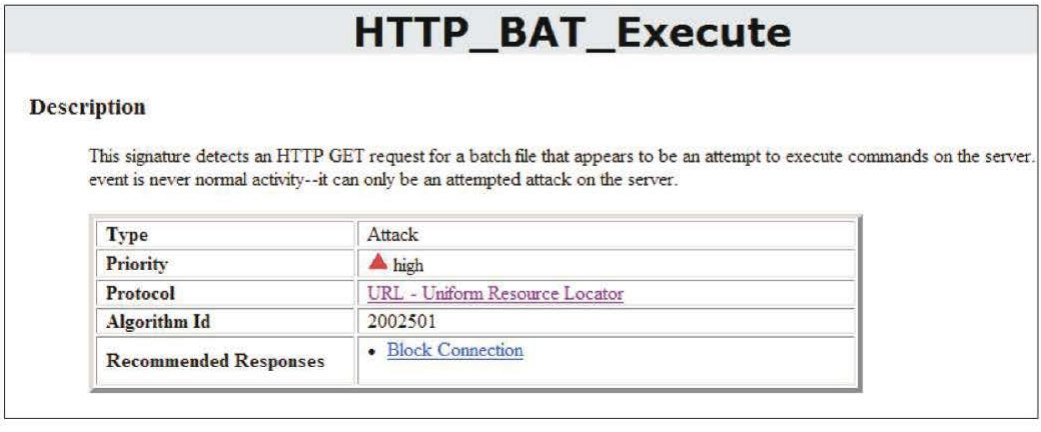


Рис. 22.12. Сигнатура HTTP_BAT_Execute

Более сложный пример — сигнатура SMB_System32_FileWritten, срабатывающая при удаленной попытке записи файлов в каталог \Windows\System32 (рис. 22.13). В частности, эта сигнатура срабатывала при распространении сетевого червя Conficker.

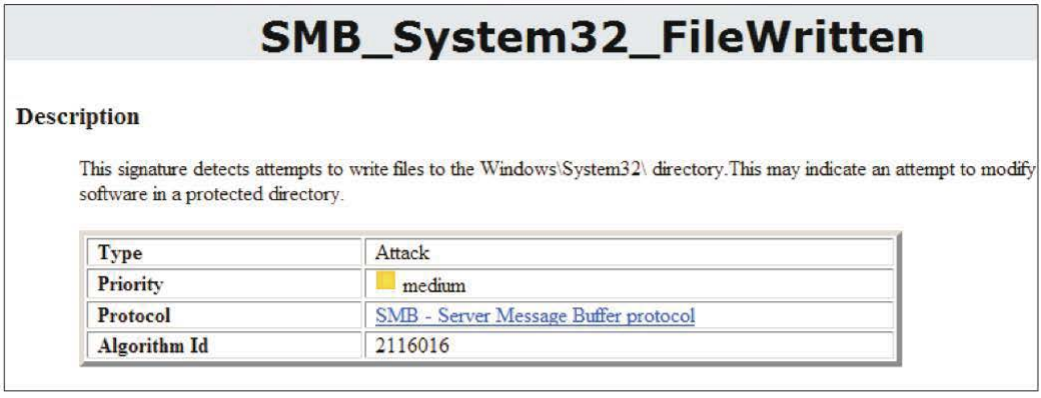


Рис. 22.13. Сигнатура SMB_System32_FileWritten

Применение системы Snort 3.0. Версия 3.0 значительно отличается от предыдущих версий и фактически представляет собой новое поколение данной системы. Функционально Snort 3.0 состоит из двух частей:

- программной среды (framework) Snort Security Platform (SnortSP) 3.0;
- модулей анализа трафика (Engines)[1].

Архитектура Snort 3.0 представлена на рис. 22.14.

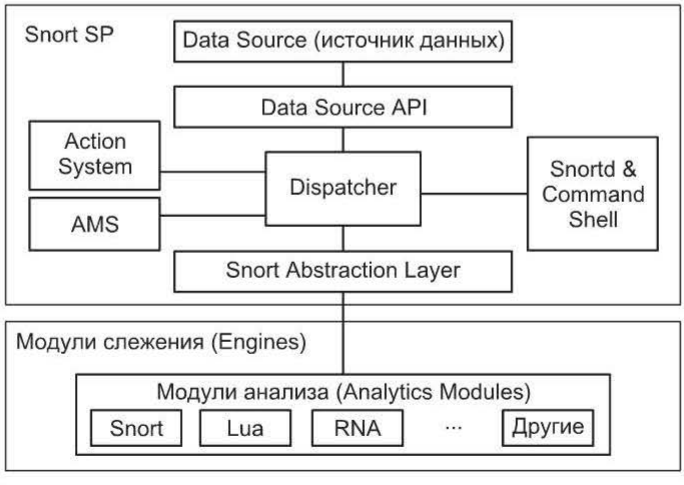


Рис. 22.14. Архитектура Snort 3.0

[1] В состав бета-версии входит модуль анализа трафика «Snort 2.8.2 detection engine».

Модуль Data Source отвечает за получение данных из различных источников и их предварительную обработку для дальнейшего анализа. В состав этого модуля входят следующие компоненты:

- захвата данных DAQ (Data Acquisition);
- контроля взаимодействий (Flow Manager);
- дефрагментации трафика IP (IP Defragmenter);
- препроцессинга трафика TCP (TCP Stream Reassembler);
- декодер (Decoder);
- набор функций для взаимодействия с другими модулями (Data Source API).

Процедура конфигурирования модуля source обычно заключается в создании одного или нескольких объектов, описывающих источники данных для последующего их использования модулями анализа (рис. 22.15).

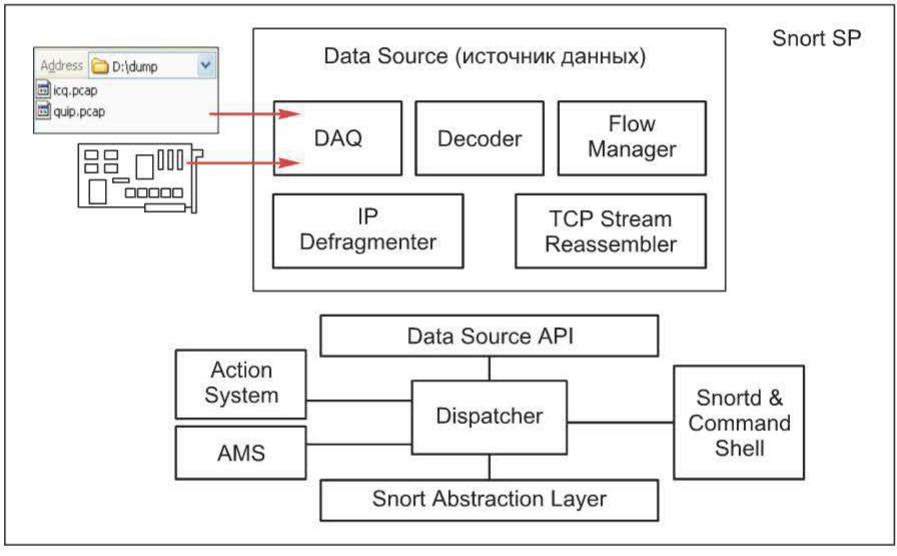


Рис. 22.15. Процедура конфигурирования модуля source

Модуль анализа занимается поиском признаков атак в источниках данных и генерацией соответствующих событий.

Модуль реагирования output отвечает за поддержку оповещений при наступлении событий. Поддерживаются следующие варианты реагирования (рис. 22.16):

- вывод оповещения на консоль;
- запись в журнал ОС (syslog);
- запись в файл в формате Unified 2.

Общий порядок работы в среде Snort SP. При запуске snortsp можно указать сценарий, который будет выполняться в процессе запуска. Для указания файла сценария используется опция –L. Обычно файл сценария называется snort.lua.

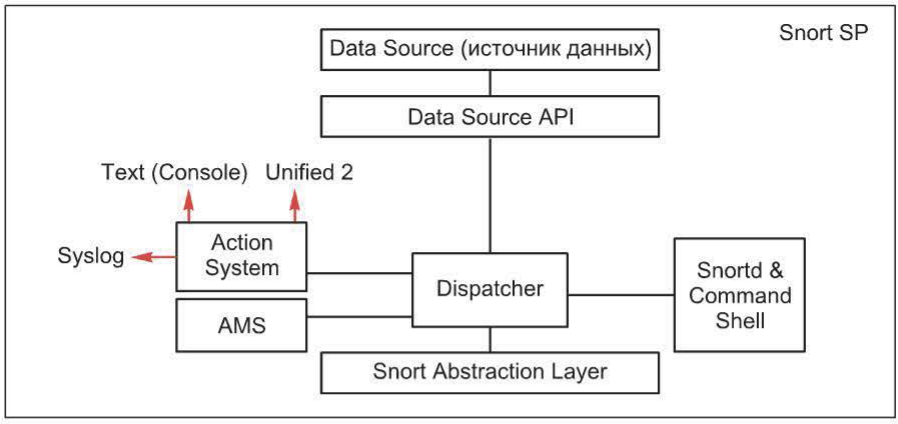


Рис. 22.16. Варианты реагирования

По умолчанию snortsp при запуске ищет файл с таким названием в каталогах /etc, /etc/snort или /usr/local/snortsp/etc (в указанном порядке). Образец файла snort.lua можно найти в каталоге .../snortsp-3.0.0b2/etc

Таким образом, можно либо подготовить файл, откуда snortsp будет брать необходимую информацию при запуске, либо после запуска «построить» нужную конфигурацию вручную.

Общий порядок работы в среде Snort SP следующий:

1) конфигурирование модулей source, engine, analyzer, output. В результате конфигурирования получается набор объектов;

2) «связывание» созданных объектов между собой;

3) запуск модуля engine;

4) после запуска можно управлять модулем engine, пользуясь интерфейсом командной строки.

Контрольные вопросы

1. Опишите использование уязвимостей как признак атаки.
2. В чем заключается признак атаки «отклонение от пороговых значений»? Приведите примеры.
3. Опишите использование известных техник и инструментов для проведения атак.
4. Расскажите о системе обнаружения атак Snort. Какие IDS используются на практике? Каковы их достоинства и недостатки?

Глава 23. МЕТОДЫ ОБНАРУЖЕНИЯ АТАК

Существуют два метода обнаружения атак:

- на основе знания всех возможных атак и их модификаций;
- на основе понимания ожидаемого поведения контролируемого объекта.

23.1. Обнаружение «злоупотреблений»

Первый метод называется обнаружением «злоупотреблений», источниками данных в нем служат журналы, сетевой трафик.

Сигнатура (signature) — совокупность параметров, «отпечаток» (pattern), соответствующий известной атаке.

Обнаружение «злоупотреблений» — процесс сопоставления сигнатур и прошедших предварительную обработку данных (полученных из соответствующих источников) для идентификации возможных инцидентов.

Примеры сигнатур:

- попытка получения по протоколу ftp файла /etc/passwd;
- появление в журнале аудита события с идентификатором 645;
- попытка подключения к закрытому в данный момент TCP-порту.

В качестве простейшего примера можно привести сетевую систему обнаружения атак, занимающуюся синтаксическим анализом отдельных пакетов.

Метод синтаксического анализа применялся в первых сетевых IDS. Позже он был усовершенствован путем добавления новых возможностей (рис. 23.1).

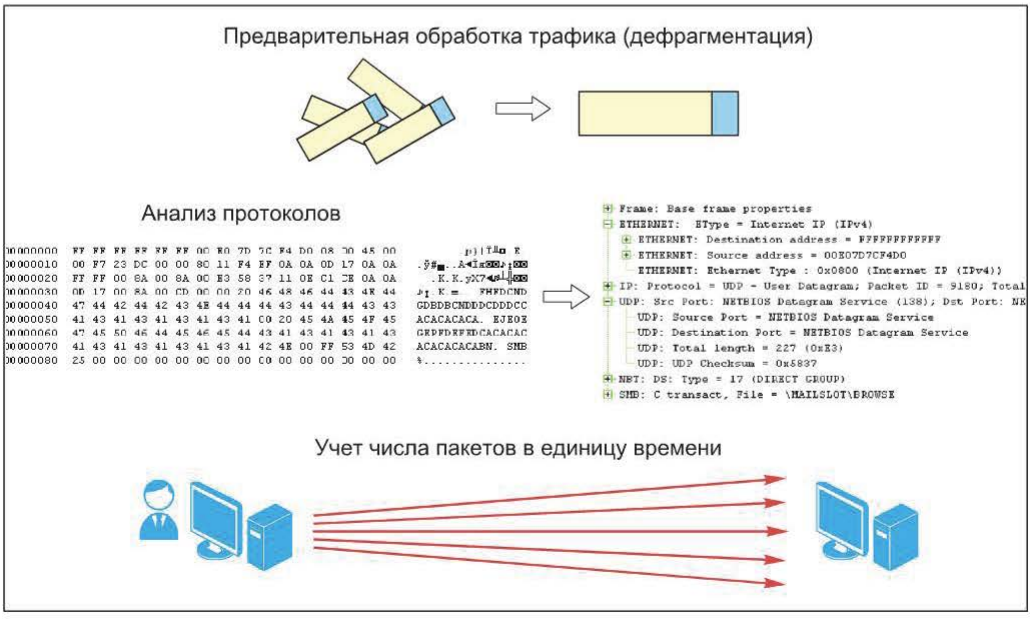


Рис. 23.1. Дополнительные возможности к синтаксическому анализу отдельных пакетов

Дополнительным эффектом такого усовершенствования стала проверка используемых протоколов на соответствие стандарту.

Следующий шаг — анализ протоколов с учетом состояния сетевого взаимодействия.

Анализ протоколов с учетом состояния — это процесс сопоставления данных, полученных системой обнаружения атак и известных моделей поведения протоколов с учетом возможных состояний сетевого взаимодействия. Контроль состояния осуществляется на сетевом, транспортном и прикладном уровнях. Модель поведения определяется:

- вендором;
- стандартами RFC;
- стандартами IETF.

Учет состояния может быть выполнен с помощью различных приемов:

- сопоставления запросов и ответов;
- проверки правильности последовательности команд;
- учета числа однотипных пакетов в единицу времени.

Анализ протоколов с учетом состояния может потребовать сборки потока данных из отдельных TCP-сегментов.

Сопоставление запросов и ответов — один из простейших методов учета состояния, основанный на анализе ответа с учетом перехваченного ранее запроса (рис. 23.2).

Рис. 23.2. Сопоставление запросов и ответов

Этот прием обеспечивает следующие возможности:

- отличать успешные попытки атак от неудачных. Например, при отслеживании попыток доступа к определенному файлу на сервере FTP код ответа сервера, начинающийся с цифры «2», сигнализирует об успешной попытке;
- подсчета однотипных запросов (например, неудачных попыток входа). Это позволяет, например, обнаруживать попытки подбора пароля методом «грубой силы».

Одним из приемов, используемых для учета состояния, является учет разных фаз сессии прикладного уровня. Например, данные по протоколу FTP

не могут быть переданы, если не были согласован режим передачи (активный или пассивный) и передана информация об используемом номере порта.

Добавление возможности подсчета событий в единицу времени позволяет сетевым IDS обнаруживать такие события, как сканирование портов или посылка большого числа запросов на установление соединения (например, SynFlood) (рис. 23.3). Такие сигнатуры чувствительны к некоторому порогу срабатывания.

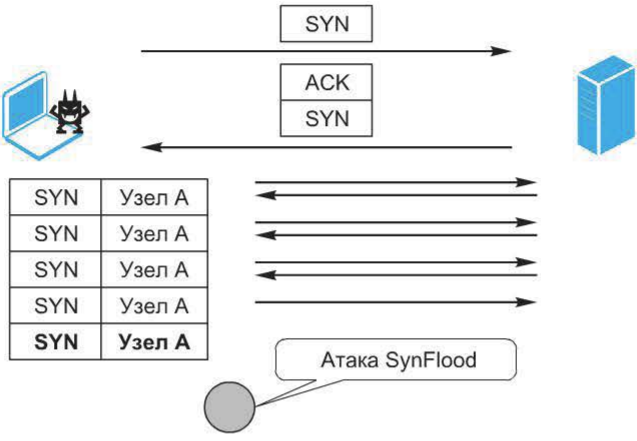


Рис. 23.3. Подсчет событий в единицу времени

Используется так называемый расширенный анализ протоколов, который обеспечивает следующие возможности:

- «нормализацию» трафика;
- учет специфики работы прикладных служб;
- моделирование работы службы;
- детализированной настройки сигнатур.

23.2. Обнаружение аномалий

Рассмотренный выше метод обнаружения «злоупотреблений» — это метод обнаружения атак, имеющий, однако, следующие ограничения:

- обнаружение только известных атак;
- возможные пропуски модификаций атак;
- отсутствие учета специфики сети.

Рассмотрим показатели качества систем обнаружения атак, т. е. модулей слежения (агентов). Очевидно, что к агентам можно применить классический подход, основанный на ошибках первого и второго рода (пропусках и ложных срабатываниях) (рис. 23.4).

Ложными срабатываниями (False Positives) следует признать оповещения о событиях, не происходивших в действительности. Причинами ложных срабатываний, как правило, являются:

По результатам мониторинга	В действительности	
	Факт атаки имел место	Факта атаки не было
Атака обнаружена	True Positive	False Positive
Атака не обнаружена	False Negative	True Negative

Рис. 23.4. Ошибки первого и второго рода

- неудачный выбор признака атаки;
- ошибка реализации сигнатуры.

Ложные срабатывания следует отличать от ложных оповещений (False Alarms). Это, как правило, оповещения о событиях, которые не являются значимыми в данном конкретном случае. Они обусловлены некорректной настройкой системы или особенностями признака атаки. Ложные срабатывания легко отличить от ложных оповещений: ложные оповещения исходят от пользователя системы, а не от разработчика.

Ложные срабатывания и ложные оповещения легко отличить от «правильных» срабатываний (True Positives). Гораздо сложнее выявить пропуски атак (False Negatives). Это можно сделать только в том случае, если точно известно о наличии атаки.

Из приведенных четырех показателей путем несложных вычислений могут быть получены показатели качества модуля слежения. Например, интуитивно понятно, что ложные срабатывания и пропуски должны быть минимизированы.

Первый показатель — точность обнаружения — определяется как отношение «правильно» обнаруженных атак ко всем обнаруженным атакам:

```
Precision= True Positive/(True Positive + False Positive)
```

Второй показатель — чувствительность (Sensitivity) — характеризует процент пропусков:

```
Sensitivity=True Positive/(True Positive + False Negative)
```

Этот показатель зависит прежде всего от полноты базы сигнатур и отчасти от их качества.

Можно рассчитать и итоговый показатель — суммарную точность работы (accuracy):

```
A = (TP+TN)/(TP+TN+FP+FN)
```

Рассмотрим альтернативный методу обнаружения «злоупотреблений» метод обнаружения аномалий (Anomaly-based detection). Это процесс сопоставления прошедших предварительную обработку данных (журналы, сетевой трафик, включая NetFlow, деятельность субъектов системы) и набора

профилей «поведения» (соглашений о том, какая активность считается нормальной) для обнаружения подозрительных ситуаций.

Профиль «поведения» определяет нормальное поведение пользователей, узлов сети, приложений и других субъектов (рис. 23.5). Он создается на основе мониторинга характеристик в течение определенного периода времени.

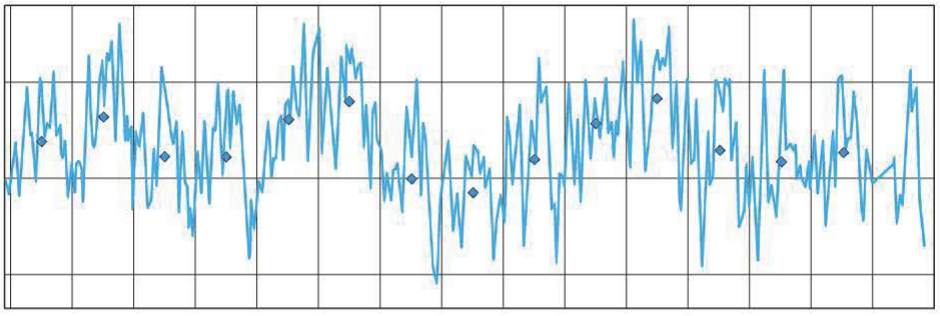


Рис. 23.5. Профиль «поведения»

Характеристики профиля «поведения» могут быть самыми разнообразными (рис. 23.6):

- загрузка отдельного участка сети;
- число писем, отправленных пользователем;
- число неудачных попыток входа в систему.

Данными для построения профиля поведения могут служить:

- объемы трафика;

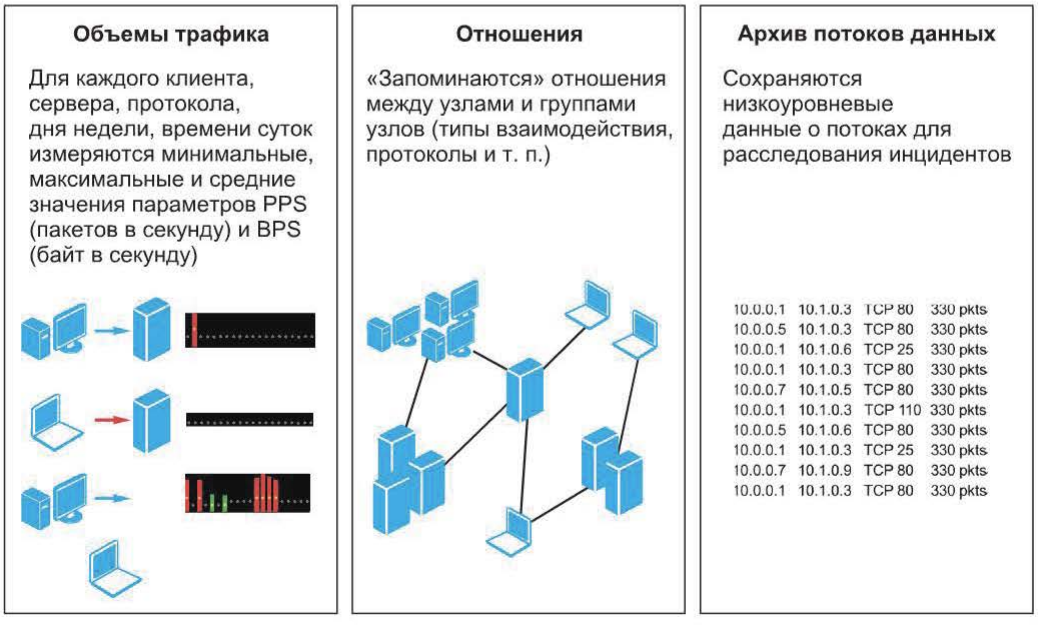


Рис. 23.6. Характеристики профиля «поведения»

• отношения между узлами и группами узлов;
• архив потоков данных.

Пример модели отношений между узлами сети приведен на рис. 23.7.

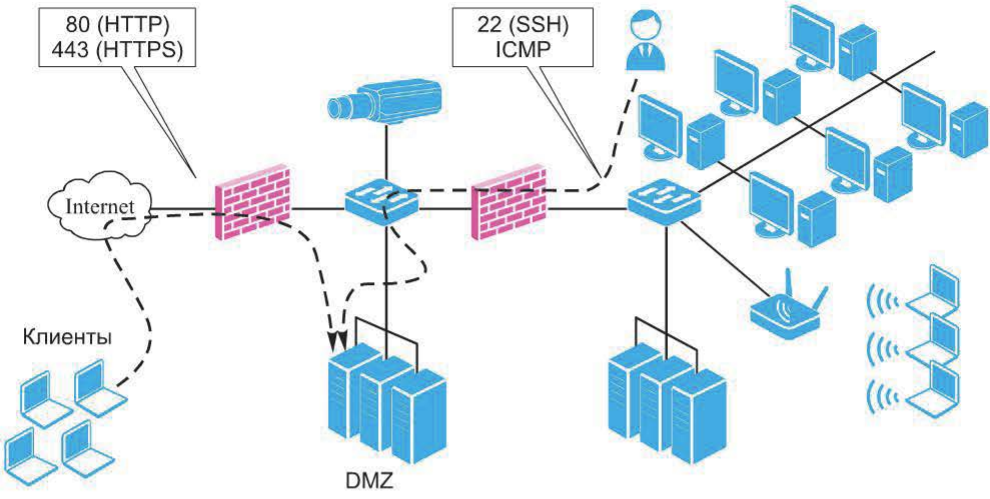


Рис. 23.7. Модель отношений между узлами сети

Метод обнаружения аномалий может быть использован как дополнение к методу обнаружения «злоупотреблений» для выявления:

• отклонений в трафике по времени и объему;
• нетипичных подключений;
• недоступных узлов и сервисов.

Контрольные вопросы

1. Опишите достоинства и недостатки метода обнаружения «злоупотреблений».
2. В чем заключается алгоритм обнаружения атак метода обнаружения аномалий? Назовите его достоинства и недостатки.

Глава 24. МЕХАНИЗМЫ РЕАГИРОВАНИЯ

24.1. Обзор механизмов реагирования

Как отмечалось выше, механизмы реагирования отличаются разнообразием. Тем не менее их можно использовать в качестве критерия для деления систем на два типа: обнаружения атак и противодействия атакам.

На рис. 24.1 представлены различные варианты оповещения.

Варианты регистрации событий представлены на рис. 24.2.

Рис. 24.1. Варианты оповещения

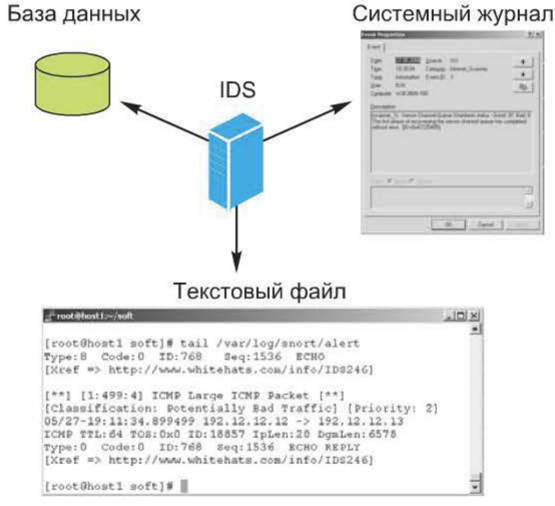


Рис. 24.2. Варианты регистрации событий

24.2. Варианты блокировки

Блокировка предполагает «активное вмешательство» системы обнаружения атак и может быть выполнена следующими способами:

- аварийное завершение TCP-соединения (рис. 24.3);
- посылка ICMP Destination Unreachable для блокировки взаимодействия по протоколу UDP (рис. 24.4);
- блокировка трафика, содержащего признаки атаки (рис. 24.5);
- карантин (рис. 24.6–24.8).

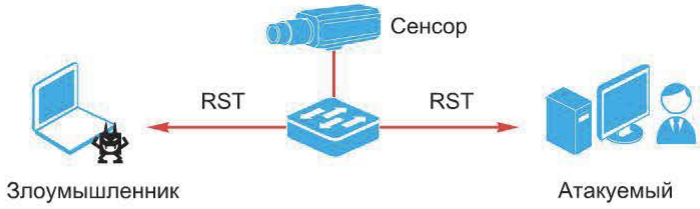


Рис. 24.3. Аварийное завершение TCP-соединения

Рис. 24.4. Посылка ICMP Destination Unreachable

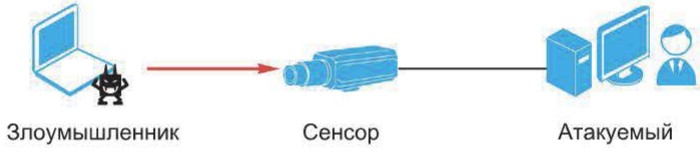


Рис. 24.5. Блокировка трафика, содержащего признаки атаки

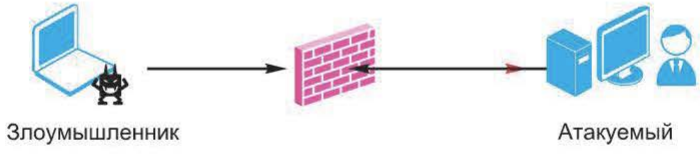


Рис. 24.6. Карантин и изоляция нарушителя

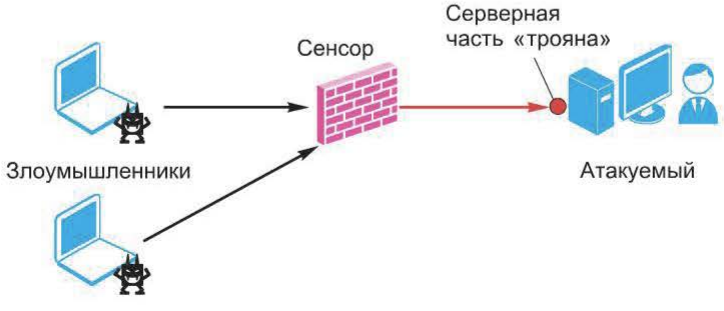


Рис. 24.7. Карантин и изоляция «трояна»

Рис. 24.8. Карантин и изоляция «червя»

Видно, что механизм блокировки «превращает» систему обнаружения атак в систему противодействия атакам. Это накладывает дополнительные требования, в частности:

- наличие сценария действий для Network IPS в случае выхода его из строя;
- наличие «мягкого» режима;
- отсутствие влияния на производительность;
- качество сигнатур, минимизация ложных срабатываний.

Для блокировки нарушителя могут быть применены и стандартные решения по обеспечению отказоустойчивости, с использованием встроенного или внешнего модуля Bypass (рис. 24.9).

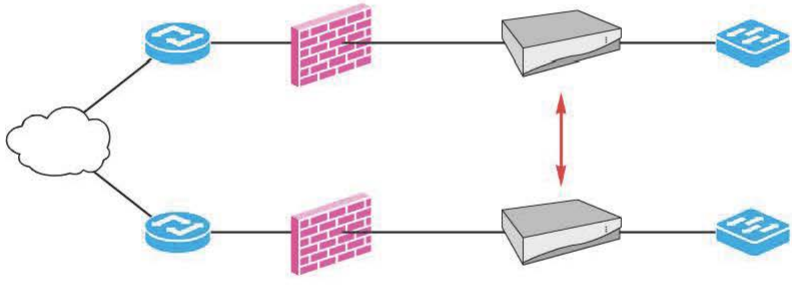


Рис. 24.9. Стандартные решения по обеспечению отказоустойчивости

Наличие «мягкого» режима означает возможность относительно простого и быстрого выключения механизма блокировки и (или) замены его записью в журнал.

Контрольные вопросы

1. Какие механизмы реагирования могут быть реализованы в системах обнаружения атак?
2. Какими способами может быть выполнена блокировка нарушителя?

Глава 25. ОБНАРУЖЕНИЕ АТАК В БЕСПРОВОДНЫХ СЕТЯХ

25.1. Угрозы, связанные с использованием беспроводных сетей

На уровне IP для беспроводных сетей характерны те же проблемы безопасности, что и для обычных сетей. Но на канальном и физическом уровнях особенность беспроводных сетей — передача данных «по воздуху» и отсутствие кабелей для подключения к сети — вносят некоторую специфику. Относительная легкость подключения означает, что доступ к сети может быть получен, например, из соседней комнаты, коридора и, возможно, с улицы.

В связи с этим возникает первая угроза, представляющая собой несанкционированное подключение к ресурсам беспроводной сети и использование ее ресурсов (рис. 25.1).

Вторая угроза — возможность пассивного прослушивания передаваемых в эфире данных (рис. 25.2). Для этого достаточно перевести беспроводной адаптер в неселективный режим (для беспроводных сетей этот режим называют режимом мониторинга), настроить на нужный канал и осуществлять перехват трафика беспроводных сетей, находящихся в радиусе действия антенны.

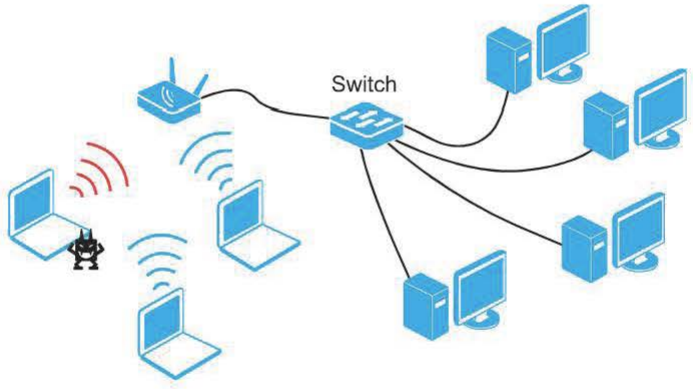


Рис. 25.1. Несанкционированное подключение

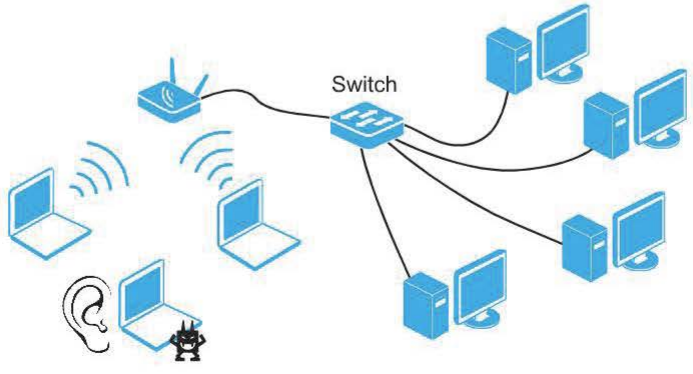


Рис. 25.2. Прослушивание трафика беспроводной сети

Таким образом, в беспроводных сетях модель угроз включает в себя две позиции: несанкционированное подключение и прослушивание. К ним следует добавить также нарушение доступности путем создания ситуации отказа в обслуживании (Denial of Service, DoS). В беспроводных сетях это возможно уже на физическом уровне (например, намеренное создание помех в заданном диапазоне частот).

На качество сигнала могут влиять даже погодные условия, развертывание беспроводной сети соседями по офису и многие другие факторы.

Следует добавить, что приведенные угрозы относятся к корпоративному доступу, для гостевого доступа модель угроз может быть несколько иной.

25.2. IEEE 802.11i — нерешенные проблемы

Построение беспроводной сети, отвечающей требованиям стандарта 802.11i, обеспечивает защиту ее трафика и стойкую аутентификацию, фактически нейтрализуя основные угрозы. Становятся бессмысленными попытки анализа трафика беспроводной сети с целью его расшифрования, затруднена возможность использования ее ресурсов.

Однако возможность прослушивания трафика сети сохраняется, как и возможность влияния на него.

В сети, отвечающей требованиям стандарта 802.11i, остаются нерешенными следующие проблемы:

- несанкционированное использование беспроводных устройств;
- атаки на устройства и сервисы беспроводной сети.

25.3. Несанкционированное использование беспроводных устройств

В ряде случаев появление в сети беспроводного устройства представляет собой угрозу безопасности, поскольку повышает вероятность несанкционированного подключения к обычной (проводной) сети, ведь беспроводная точка доступа может быть соединена с обычной сетью (рис. 25.3).

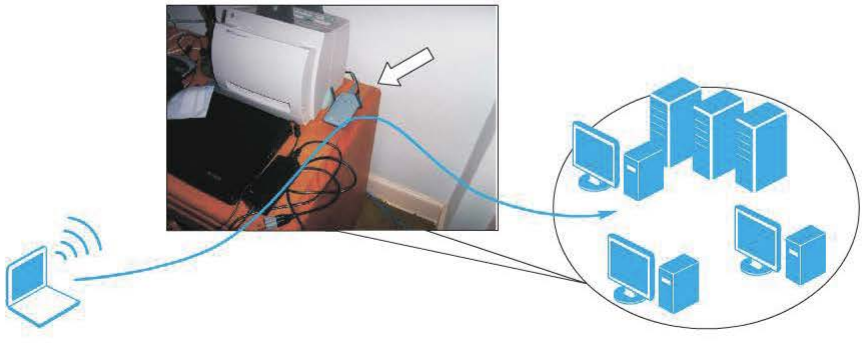


Рис. 25.3. Беспроводная точка доступа

Беспроводное устройство может быть легко замаскировано среди офисной техники. Следует также обратить внимание на то, что эта угроза актуальна и для сетей, где не используются беспроводные технологии. Беспроводные сети могут появиться в любой момент и создать угрозу безопасности корпоративной сети.

25.4. Атаки на устройства и сервисы

Любая, даже самая защищенная беспроводная сеть подвержена атакам, среди них:

- сбор информации о беспроводных сетях;
- атаки на отказ в обслуживании;
- атаки на механизм аутентификации 802.1x;
- атаки на клиентов беспроводных сетей.

Сбор информации о беспроводных сетях. Перед тем как предпринять попытки атак на беспроводную сеть, нарушитель должен собрать информацию о ней. Когда беспроводные технологии только начали широко применяться, возник специальный термин «WarDriving» — перемещение по какой-либо территории для поиска беспроводных точек доступа. Перемещение обычно осуществляется на автомобиле или общественном транспорте, отсюда и название: WarDriving. С технической точки зрения процесс представляет собой перехват трафика беспроводных сетей, например широковещательных фреймов «beacon». Из них может быть получена базовая информация о беспроводной сети: SSID, параметры защиты и т. п. WarDriving не направлен на какую-то конкретную сеть, его цель — построение карт беспроводных сетей с информацией о них. В настоящее время эта тема стала менее актуальной, поскольку беспроводных сетей стало слишком много, к тому же их большая часть защищена от несанкционированных подключений.

В зависимости от используемого ПО сбор информации может быть активным и пассивным. Так, программы NetStumbler, dStumbler и MiniStumbler, как отмечалось выше, передают широковещательные запросы Probe Request и слушают ответы на них. Это увеличивает шансы обнаружения беспроводной сети, т. е. эти программы используют активные методы.

Пассивные методы предполагают только прослушивание трафика (в режиме мониторинга) и анализ радиосигнала с целью обнаружить беспроводные сети.

Сбор информации предполагает только обнаружение беспроводных сетей, но не подключение к ним, и тем более использование их ресурсов. С точки зрения закона это не является нарушением.

Тем не менее для его обнаружения могут быть реализованы следующие меры:

- визуальный контроль (видеонаблюдение) — для обнаружения клиентов, использующих пассивные методы;
- обнаружение фреймов Probe Request и RTS и их анализ (этот метод пересекается с методами обнаружения несанкционированных клиентов, которые более подробно будут рассмотрены далее).

Программы для обнаружения беспроводной сети, подобные Netstumbler (использующие активные методы), не осуществляют перевод адаптера в режим мониторинга, поэтому позволяют себя обнаружить по фреймам Probe Request. Поскольку фреймы Probe Request с некоторой периодичностью рассылает и беспроводной адаптер, задача обнаружения нарушителя в этом случае просто сводится к задаче обнаружения несанкционированных клиентов.

Следует отметить, что задача может быть усложнена, если поставлена цель — идентификация ПО, используемого для обнаружения беспроводных сетей. В литературе представлены методы обнаружения использования программ NetStumbler, dStumbler, MiniStumbler и Wellenreiter.

NetStumbler (а также windows-клиент) помимо фреймов, содержащих SSID «нулевой» длины, отправляет фреймы probe request макимальной длины (32), содержащие неотображаемые символы (рис. 25.4).

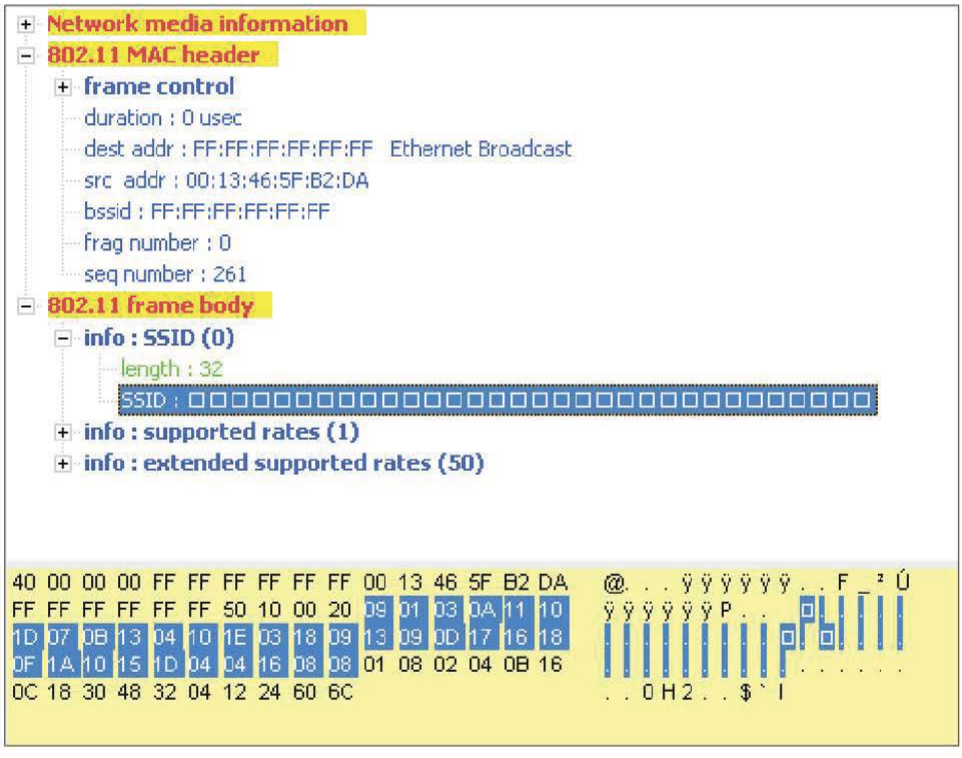


Рис. 25.4. Фреймы probe request

Некоторое время это поведение считалось ошибкой реализации, но, поскольку все осталось без изменений, эта особенность используется системами обнаружения атак в качестве признака атаки.

Технология Honeynet для беспроводной сети. В качестве одной из превентивных мер защиты можно использовать технологию Honeynet для раннего обнаружения несанкционированной активности в беспроводных сетях. Если используются пассивные методы поиска беспроводных сетей, обнару-

жить того, кто это делает, достаточно сложно. Но в этом случае кроме видеонаблюдения может быть предпринята такая мера защиты, как маскировка путем эмуляции нескольких сетей, одна из которых «настоящая». При этом программа для обнаружения беспроводных сетей выведет информацию о нескольких (возможно, многих) сетях, большая часть которых реально не существует. Ложная сеть — приманка — устанавливается в непосредственной близости от реальной сети и использует менее эффективный метод защиты.

Предположим, основная беспроводная сеть использует стандарт WPA, или 802.11i и SSID corporate. Точка доступа для honeypot сконфигурирована для использования SSID guests и использует технологию WEP. Злоумышленник с высокой степенью вероятности выберет для взлома менее защищенную систему. При реализации honeypot можно эмулировать несколько сетевых служб для возможности протоколировать действия взломщика.

Обычно для эмуляции точек доступа используется специальный режим работы беспроводного адаптера Master. В режиме Master беспроводная карта имеет следующие возможности:

- рассылка фреймов Beacon;
- ответы на фреймы Probe Request;
- поддержка времязависимых функций.

Одна из первых программ для генерации фреймов Beacon с разными параметрами называлась FakeAP и до сих пор доступна по адресу http://www.blackalchemy.to/project/fakeap/ Программа работает в среде Linux и осуществляет периодический запуск сетевого интерфейса (с помощью команды iwconfig) в режиме Master с различными параметрами.

Если драйвер уже поддерживает режим Master, генерация ложных сетей может осуществляться с помощью простого сценария:

```
#!/bin/sh
i=1
while [ $i -le 10 ]
do
iwconfig ath0 mode Master
ifconfig ath0 down
iwconfig ath0 essid "Default SSID"
iwconfig ath0 enc off
iwconfig ath0 channel 1
iwconfig ath0 txpower 20
ifconfig ath0 hw ether 00:20:d8:13:1A:0F up
sleep 0.2
ifconfig ath0 down
iwconfig ath0 essid "linksys"
iwconfig ath0 channel 6
iwconfig ath0 key s:"@ABCD"
iwconfig ath0 txpower 100
ifconfig ath0 hw ether 00:04:5A:34:11:AA up
```

```
sleep 0.3
ifconfig ath0 down
iwconfig ath0 enc off
iwconfig ath0 essid "Wireless"
iwconfig ath0 channel 6
iwconfig ath0 txpower 50
ifconfig ath0 hw ether 00:30:AB:29:22:AE up
sleep 0.1
ifconfig ath0 down
iwconfig ath0 enc off
iwconfig ath0 essid "WLAN"
iwconfig ath0 channel 11
iwconfig ath0 txpower 15
ifconfig ath0 hw ether 00:90:d1:93:F1:AA up
sleep 0.2
done
```

В данном примере в бесконечном цикле сетевому интерфейсу ath0 поочередно присваиваются параметры, эмулирующие наличие четырех сетей (Default SSID, linksys, Wireless, WLAN).

Отказ в обслуживании. Существующие технологии защиты беспроводных сетей большей частью сконцентрированы на обеспечении целостности и конфиденциальности информации и не рассматривают аспект доступности беспроводных сетей.

Однако вопрос доступности остается важным аспектом безопасности беспроводных сетей. Кто будет пользоваться неработающей сетью, даже если трафик в ней защищен самыми стойкими криптографическими алгоритмами?

Сети, построенные на основе стандартов IEEE 802.11, содержат ряд уязвимостей, которые могут быть использованы для создания ситуации отказа в обслуживании.

Их можно разделить на следующие группы:

- уязвимости физического уровня;
- уязвимости метода доступа к несущей;
- уязвимости сервисов канального уровня.

Уязвимости физического уровня. В традиционных проводных сетях для осуществления DoS-атаки на физическом уровне злоумышленник должен находиться в непосредственной близости от атакуемого узла или линии связи. В случае беспроводных сетей среда передачи не привязана жестко к какой-либо физической точке, и злоумышленник может осуществлять атаки на физическом уровне, не имея физического доступа к сети, и оставаться при этом незамеченным.

Спецификация физического уровня определяет диапазон частот, которые могут быть использованы устройствами для взаимодействия. Любой злоумышленник имеет возможность создать устройство, которое будет генерировать сигнал достаточной мощности в этом диапазоне. В результате отно-

шение сигнал/шум для точек доступа и беспроводных клиентов в зоне атаки станет неудовлетворительным, и они потеряют возможность подключения к сети.

Создание подобного устройства не представляет особых проблем. Для этой цели может быть использован генератор радиошума в соответствующем диапазоне. В настоящее время существует достаточное число подобных устройств (рис. 25.5).

Рис. 25.5. Устройства генерации радиошума

Для этих целей может быть применен даже простой беспроводной радиотелефон, работающий в диапазоне частот 2,4 ГГц. Помехи могут создавать и другие беспроводные сети (если они используют тот же канал) или персональные устройства, использующие протокол Bluetooth. Поскольку диапазон частот для Bluetooth 2,4 ГГц, возможны помехи в беспроводных сетях.

Уязвимости метода доступа к несущей. Каждый фрейм протокола 802.11 содержит поле Duration, в котором указывается время в миллисекундах, в течение которого клиент может занимать канал. Это значение используется для расчета значения NAV (Network Allocation Vector) на каждом из клиентов. Когда значение NAV достигает нуля, клиент получает возможность передавать в сеть. Такой подход применяется в процессе согласования параметров RTS/CTS.

Во время согласования параметров узел-отправитель посылает небольшой фрейм RST (request to send), в который входит значение поля duration, достаточное для обмена сообщениями RTS/CTS, включая пакет CTS, данные и завершающий пакет подтверждения. Получатель отвечает на пакет RTS пакетом CTS, в котором указывается новое значение поля Duration, учитывающее время, потраченное на пакет RTS. После отправки пакета CTS все станции в данном радиодиапазоне обновляют свое значение NAV и задерживают все пакеты до окончания указанного интервала.

Злоумышленник может использовать этот механизм для оккупации несущей путем отправки пакетов с большим значением поля Duration. Использование RTS дает злоумышленнику наибольшую возможность по занятию несущей. В ходе атаки используется важная процедура метода

CSMA/CA – CCA (Clear Channel Assessment). За ее выполнение отвечает физический уровень, соответствующий спецификации DSSS, поэтому уязвимыми оказываются беспроводные сети, использующие данную спецификацию. Таким образом, сети 802.11a и 802.11g (вариант для скорости 54 М/с) неуязвимы к данной атаке.

Максимальное значение NAV равно 32 767, или примерно 32 мс для сетей 802.11b. Соответственно, злоумышленнику требуется только 30 пакетов в секунду для того, чтобы подавить все взаимодействие на определенном канале.

Описание этой уязвимости можно найти по адресу http://www.auscert.org.au/render.html?it=4091 (бюллетень AA-2004.02).

Уязвимости сервисов канального уровня. На канальном уровне стандарт 802.11 определяет следующие сервисы: Authentication, Association, Deauthentication, Disassociation, Distribution, Integration, Privacy, Reassociation, MSDU delivery.

Уязвимости сервисов канального уровня связаны с тем, что идентификация клиента и точки доступа основана на MAC-адресе, управляющие фреймы не аутентифицируются.

Поскольку использование переносных компьютеров и устройств является нормой в беспроводных сетях, стандарт 802.11 содержит описание механизма энергосбережения. Рассмотрим уязвимости реализации данной функции.

Атаки на механизмы идентификации. Перед началом передачи данных клиент должен пройти аутентификацию на точке доступа. Если в сети включен WPA, то точка доступа в ответ на запрос аутентификации посылает клиенту случайный текст. Клиент зашифровывает его с помощью WPA и отсылает точке доступа. Если для шифрования использовался тот же ключ, что установлен на точке доступа, она подтверждает успешную аутентификацию (рис. 25.6).

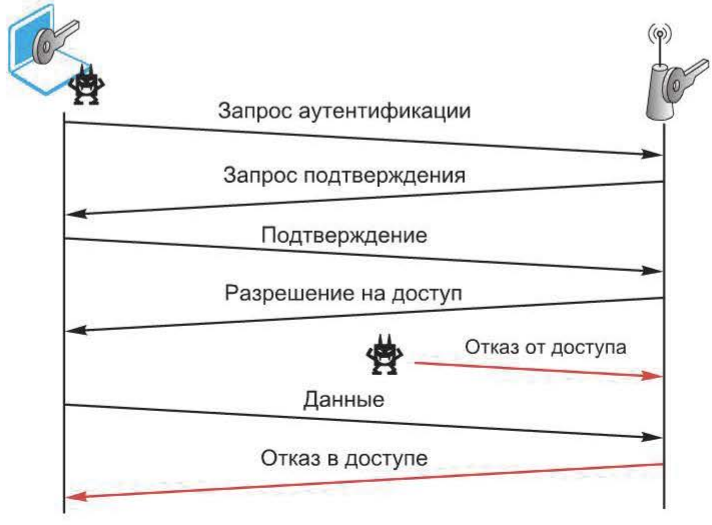


Рис. 25.6. Аутентификация на точке доступа

Злоумышленник, прослушивающий трафик беспроводного сегмента, имеет возможность узнать MAC-адреса точки доступа и ее клиентов. Именно MAC-адреса используются для идентификации управляющих сообщений аутентификации.

Соответственно, злоумышленник имеет возможность сгенерировать и отправить пакет отказа от доступа (Deauthentication) с MAC-адресом клиента в адрес точки доступа, которая примет этот пакет и откажет клиенту в доступе.

Подобной атаке подвержен и механизм установления ассоциации (Association Request — Association Response), когда у злоумышленника появляется возможность послать пакет Disassociation с MAC-адресом клиента, который будет принят и обработан точкой доступа.

Некоторые точки доступа подвержены атаке Authentication Flood, если злоумышленник генерирует большое число запросов на аутентификацию от различных MAC-адресов (рис. 25.7).

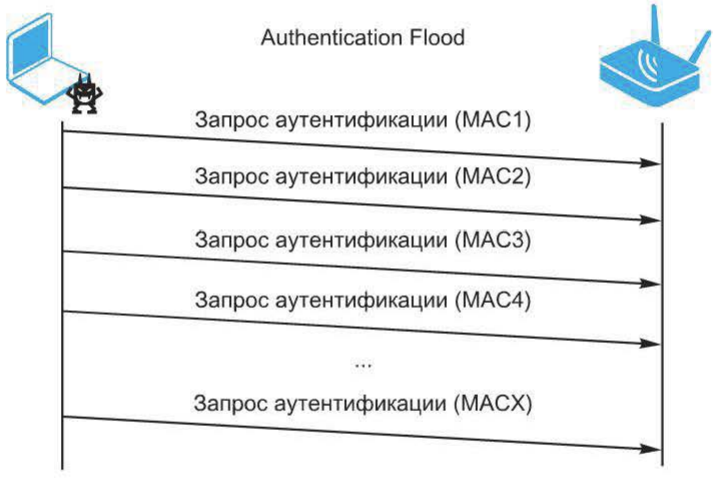


Рис. 25.7. Атака Authentication Flood

Некоторые точки доступа в результате подобной атаки могут полностью потерять возможность обслуживать клиентов в течение некоторого промежутка времени, некоторые не обрабатывают новых клиентов во время атаки.

Атаки на функции энергосбережения. В стандарт 802.11 встроены функции энергосбережения, которые помогают более экономно расходовать электроэнергию беспроводных устройств. В спящем режиме клиент не передает и не получает данные. Перед переходом в спящий режим клиент сообщает об этом точке доступа, после чего она начинает сохранять пакеты, направленные этому клиенту, в буфере.

Периодически клиент переходит в активный режим и передает точке доступа уведомление об этом. Если в буфере есть информация для этого клиента, то точка доступа передает ему данные из буфера.

Злоумышленник может сгенерировать ложный пакет, в котором уведомит точку доступа, что клиент перешел в активный режим, после чего точка доступа передаст данные. После того как реальный клиент перейдет в активный режим, он не получит своих данных (рис. 25.8).

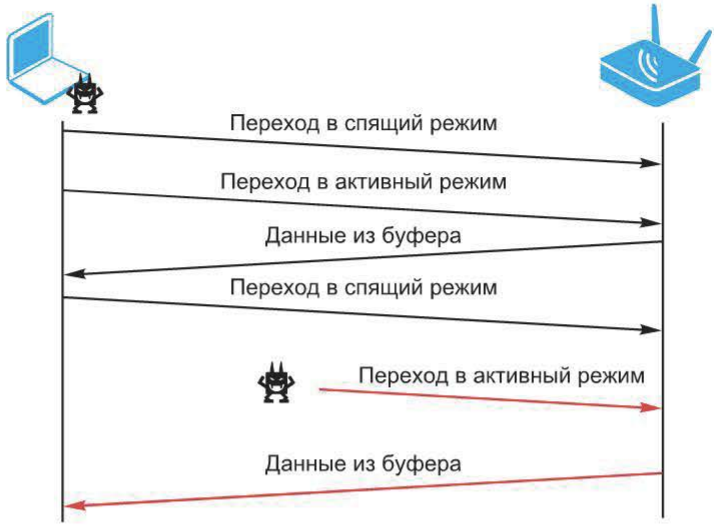


Рис. 25.8. Атаки на функции энергосбережения

Кроме того, злоумышленник может подделывать пакеты TIM (Traffic Indication Map). Эти пакеты периодически передаются точкой доступа, чтобы оповестить клиентов, находящихся в спящем режиме, о наличии данных в буфере. Если клиент получит подделанный пакет, в котором будет указано, что данных для него нет, он опять перейдет в спящий режим.

25.5. Атаки на механизм аутентификации 802.1x

Стандарт 802.11i требует обязательного использования механизма аутентификации 802.1x. Этот механизм, в свою очередь, предполагает использование протокола EAP.

Общие проблемы EAP. Одна из проблем протокола EAP — отсутствие аутентификации и контроля целостности EAP-пакетов. В частности, это означает, что любой неаутентифицированный клиент может начать диалог по протоколу EAP. Рассматриваемые далее DoS-атаки как раз и обусловлены этой особенностью.

Другой вектор угроз направлен на сервер RADIUS. Поскольку неаутентифицированный клиент взаимодействует с сервером RADIUS, нужен тщательный контроль отправляемых клиентом EAP-пакетов.

Самый простой случай — использование аутентификации EAP-MD5. Обмен данными при этом происходит в последовательности, представленной на рис. 25.9.

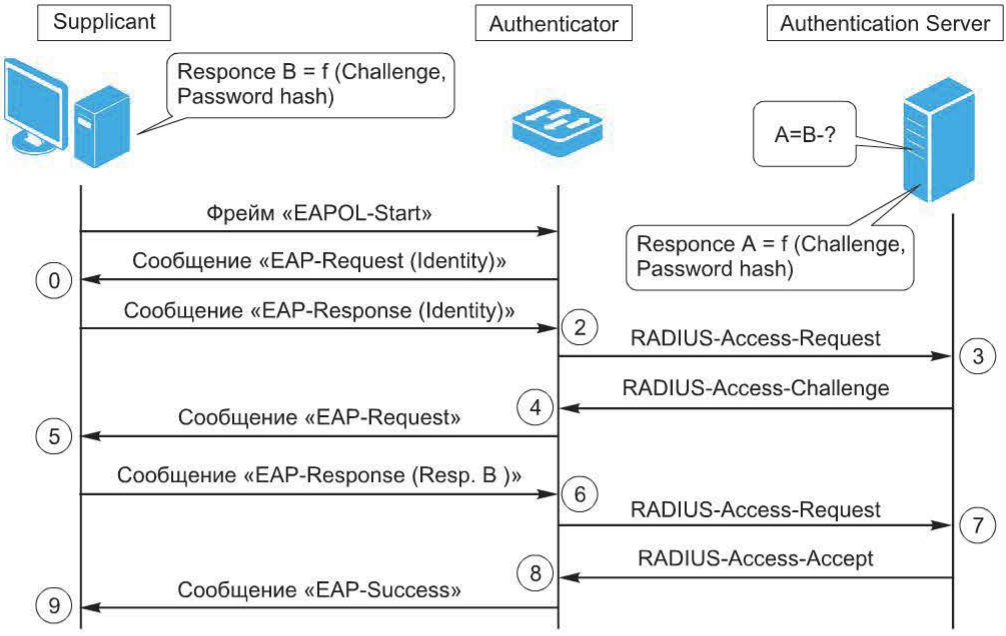


Рис. 25.9. Обмен данными при использовании аутентификации EAP-MD5

После загрузки узел-клиент инициирует процедуру аутентификации посылкой фрейма EAPOL-start. Процедуру аутентификации может инициировать и коммутатор (Authenticator). Это происходит, как только соответствующий порт коммутатора становится активным. В любом случае клиент получает запрос идентификации (сообщение EAP-request/identity). Оно передается как EAP-Packet с кодом 1 (Request). Начинается процедура идентификации клиента.

Клиент отвечает Authenticator фреймом EAP-response/identity, содержащим данные идентификации клиента (обычно это имя пользователя). Authenticator копирует содержимое фрейма EAP-response/identity в атрибут User-Name (для сервера RADIUS) и передает серверу RADIUS сообщение Access-Request.

Сервер RADIUS решает, какой тип аутентификации EAP следует использовать, формирует сообщение Access-Challenge и передает его клиенту. При формировании этого сообщения используются полученные от клиента данные идентификации, на основе которых сервер RADIUS находит в БД пользователей необходимую информацию. Параллельно сервер RADIUS производит вычисления, используя пароль пользователя и переданный ему отклик Access-Challenge. Результат позже будет сравниваться с информацией, полученной от клиента. Фактически начинается процедура аутентификации клиента.

Authenticator направляет сообщение клиенту в пакете EAP-Request.

Клиент, получив сообщение, выполняет аналогичные вычисления, используя свой пароль и полученное сообщение Access-Challenge. Затем результат отправляется Authenticator.

Authenticator перенаправляет результат серверу RADIUS. Если сообщения, полученное от клиента и вычисленное ранее, совпадают, это означает, что данные аутентификации, предоставленные клиентом, корректны. Сервер RADIUS отвечает сообщением об успешности аутентификации, порт коммутатора переходит в состояние authorized. Authenticator передает разрешение на доступ клиенту, и клиент получает доступ к ресурсам сети.

Этот механизм аутентификации уязвим к атакам по словарю. Достаточно перехватить запрос и отклик, и задача сводится к восстановлению пароля по его хэшу. Практическая реализация этой атаки — утилита eapmd5pass. Аналогичная проблема свойственна и протоколу LEAP. Используемый им вариант аутентификации MS-CHAPv1 также уязвим к словарным атакам.

Проблемы остальных вариантов EAP будут приведены далее. Рассмотрим варианты DoS-атак, основанные на том, что приведенные выше сообщения EAP (например, Request, Response, Success, Failure) не аутентифицированы.

Фрейм EAP Logoff. Прежде всего протокол EAP уязвим для DoS-атак. Например, для принудительного отключения клиента нарушитель может отправить от его имени (MAC-адреса) фрейм EAP Logoff (рис. 25.10).

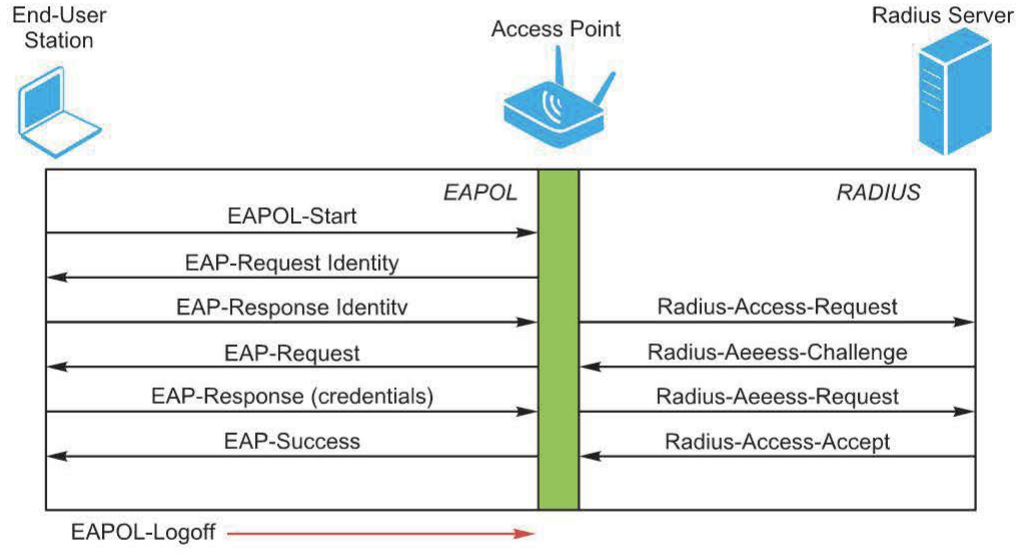


Рис. 25.10. Принудительное отключение клиента

Фрейм EAP Logoff не аутентифицируется, следовательно, для нарушителя достаточно перехватывать трафик беспроводной сети и формировать фреймы.

Фреймы EAP Success и EAP Failure. Если клиент успешно прошел процедуру аутентификации, от точки доступа приходит сообщение EAP Success. При этом стандарт 802.1x требует, чтобы процедура обмена пакета была полностью завершенной. Если нарушитель отправит пакет EAP Success от имени точки доступа раньше времени (до того как полностью завершится процеду-

ра обмена пакетами), клиент не сможет взаимодействовать с сетью, несмотря на то, что было получено сообщение EAP Success (рис. 25.11).

Точно таким же образом может быть отправлен фрейм EAP Failure, приводящий к тому же результату — сбою подключения клиента к точке доступа (рис. 25.12).

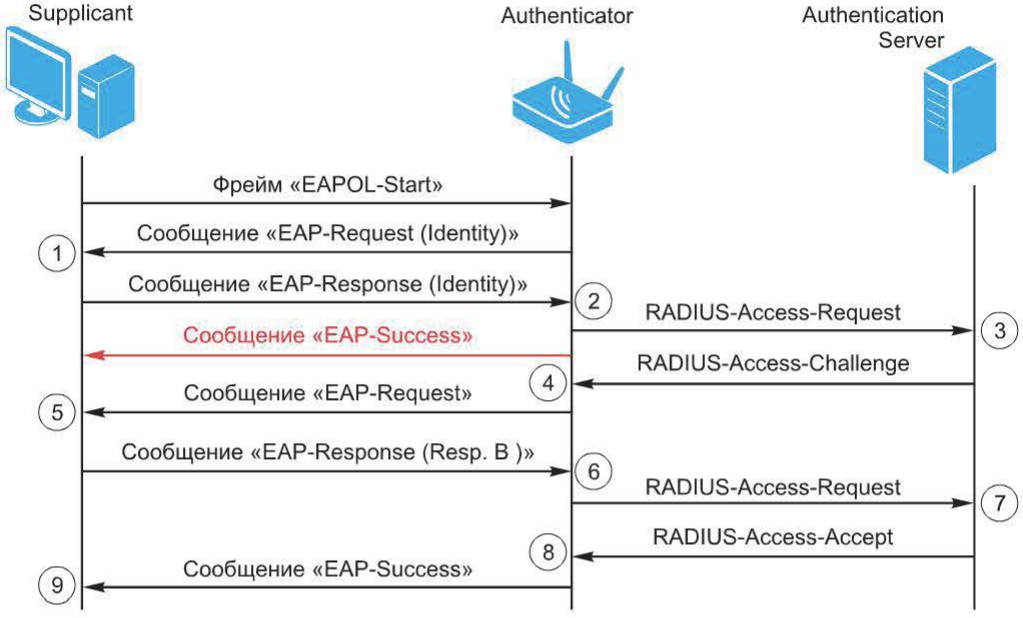


Рис. 25.11. Отправка сообщения EAP Success от имени точки доступа раньше времени

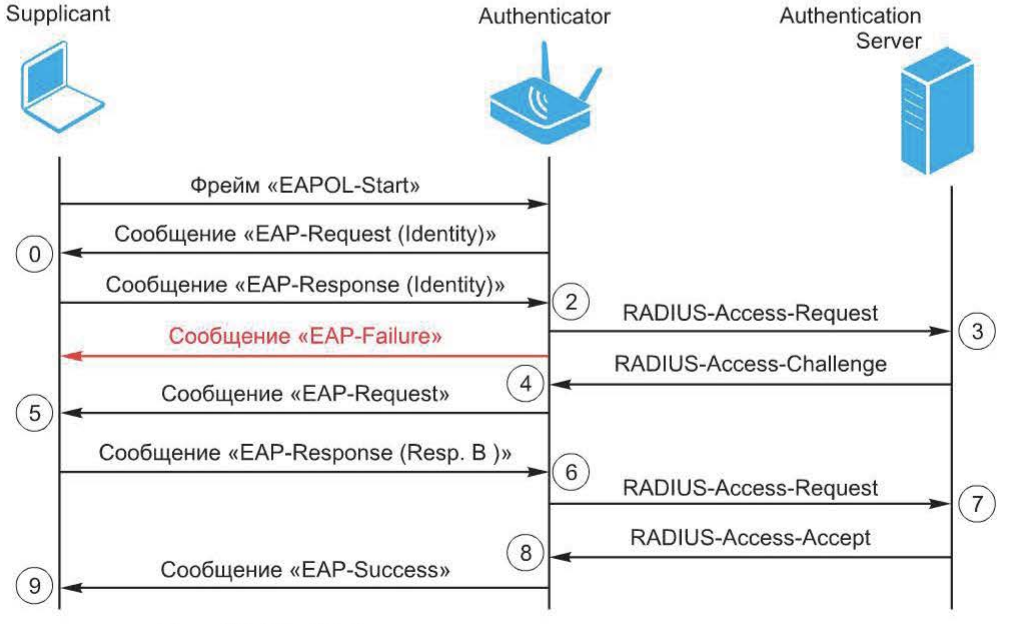


Рис. 25.12. Сбой подключения клиента к точке доступа

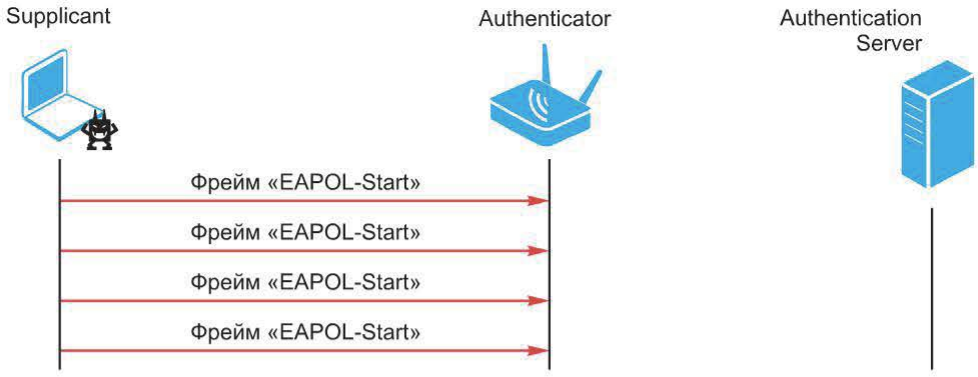


Рис. 25.13. Переполнение точки доступа запросами EAPOL Start

Переполнение запросами EAPOL Start. Как и любой сетевой протокол, EAP уязвим к DoS-атакам, основанным на переполнении точки доступа разного рода запросами. Например, точка доступа может быть переполнена запросами EAPOL Start (рис. 25.13).

25.6. Атаки на клиентов

Векторы атак с точки зрения клиента. Построение беспроводной сети с учетом требований стандарта 802.11i, правильная организация фильтрации трафика позволяют достичь вполне приемлемого уровня защищенности. При этом вектор атак смещается в сторону беспроводных клиентов, поскольку, как и во многих других ситуациях, «слабым звеном» при защите беспроводной сети оказываются пользователи и их рабочие места.

Перечень возможных атак на беспроводных клиентов:

- с использованием уязвимостей ОС и прикладного ПО;
- на защитные механизмы канального уровня (аутентификация, шифрование);
- с использованием уязвимостей драйверов сетевых адаптеров DoS-атаки.

Следует также отметить, что, как и в обычных сетях, атаки на клиентов беспроводных сетей по своей эффективности могут превосходить традиционный взлом защиты точек доступа.

Атаки на ОС и прикладное ПО. Вследствие специфики беспроводного доступа клиент беспроводной сети и потенциальный злоумышленник оказываются по отношению друг к другу в одном сегменте. Аналогично клиентам VPN, для которых характерно подключение к ресурсам корпоративной сети из неизвестного сетевого окружения, для клиентов беспроводных сетей существует угроза атак со стороны нарушителей, находящихся в зоне действия точки доступа (рис. 25.14).

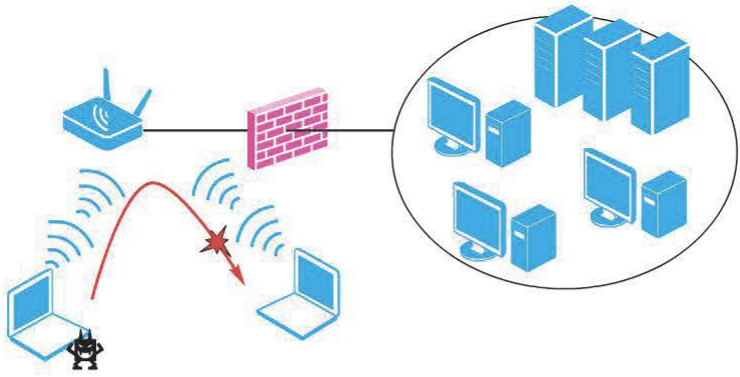


Рис. 25.14. Атака со стороны нарушителей, находящихся в зоне действия точки доступа

Следовательно, в отношении клиента беспроводной сети могут быть выполнены следующие сетевые атаки:

- удаленное изменение таблицы ARP (ARP-Spoofing);
- удаленное изменение таблицы маршрутизации (ICMP Redirect);
- внедрение ложного DHCP сервера;
- подмена DNS-ответов (DNS-spoofing).

В случае успеха приведенных атак могут быть реализованы и атаки типа «человек посередине» на используемые в беспроводной сети криптографические протоколы, например SSH, SSL. Кроме того, используя уязвимости сетевых сервисов, нарушитель может пытаться получить доступ к узлу на уровне ОС или прикладного ПО.

Однако для выполнения таких атак необходимо выполнение одного из следующих условий:

- нарушитель и объект атаки подключены к одной и той же точке доступа;
- объект атаки подключен к точке доступа, контролируемой нарушителем (например, к ложной точке доступа).

Первое условие может быть выполнено в следующих случаях:

- гостевой доступ, при котором аутентификация осуществляется через web-портал;
- корпоративный доступ с использованием технологий VPN. В этом случае обычно на канальном уровне соединение устанавливается без каких-либо препятствий, а все защитные механизмы начинают работать на сетевом уровне и выше.

Таким образом, уязвимыми для таких атак оказываются пользователи «хотспотов» и клиенты беспроводной сети, осуществляющие доступ в корпоративную сеть с использованием VPN-технологий.

В качестве меры защиты от рассмотренных угроз можно использовать набор средств Endpoint Security, в состав которых входят персональный межсетевой экран и система предотвращения атак.

Фактически в этом случае ситуацию следует рассматривать как подключение из «неизвестного» сетевого окружения и принимать соответствующие меры защиты. Кроме того, можно использовать механизм AP Isolation, который препятствует взаимодействию абонентов одной точки доступа между собой. Устройство, подключенное к такой точке доступа, может взаимодействовать только с узлами, находящимися в сети, с которой она соединена (рис. 25.15).

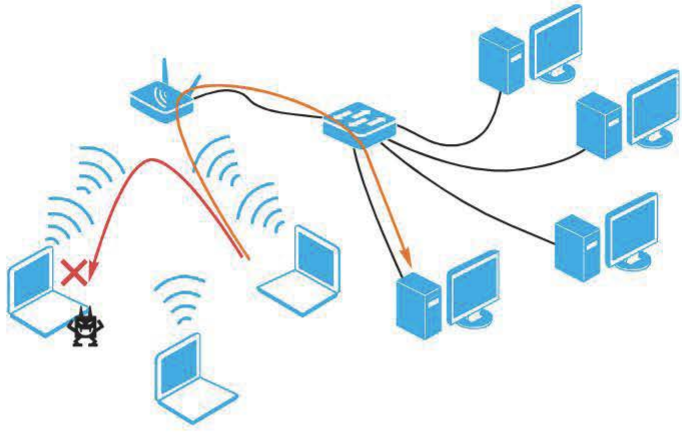


Рис. 25.15. Механизм AP Isolation

Эта функция может иметь различные названия, например:

- Public Secure Packet Forwarding (PSPF) на устройствах Cisco;
- Station Separation на устройствах EnGenius (http://www.engeniustech.com).

Помимо безопасности использование этой функции повышает производительность беспроводной сети, так как точка доступа не проводит ретрансляцию фреймов между ее абонентами.

Ложная точка доступа. Отдельного рассмотрения заслуживает ситуация «поднятия» нарушителем собственной точки доступа с необходимыми параметрами. Это может быть использовано для атак на так называемых «неассоциированных» клиентов, т. е. в том случае, когда клиент находится вне зоны действия «своей» сети, а его ПО рассылает фреймы Probe Request, пытаясь «найти» точку доступа (рис. 25.16).

Разумеется, порядок проведения такой атаки зависит от логики работы встроенного в ОС клиента беспроводной сети. Например, клиент, встроенный в ОС Windows (Wireless Zero Configuration, WZC), работает по сле-

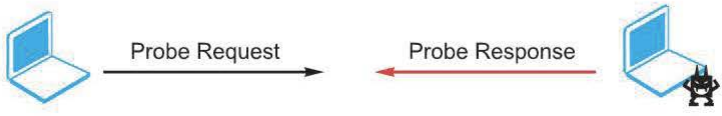


Рис. 25.16. Атака на «неассоциированных» клиентов

дующему алгоритму (http://technet2.microsoft.com/WindowsServer/en/Library/370b019f-711f-4d5a-8b1e-4289db0bcafd1033.mspx).

1. Составляется список доступных сетей путем рассылки запросов Probe Request с пустым значение SSID по всем каналам (клиент WZC «ведет себя» как активный клиент). Результат опроса можно увидеть в списке доступных беспроводных сетей (рис. 25.17).

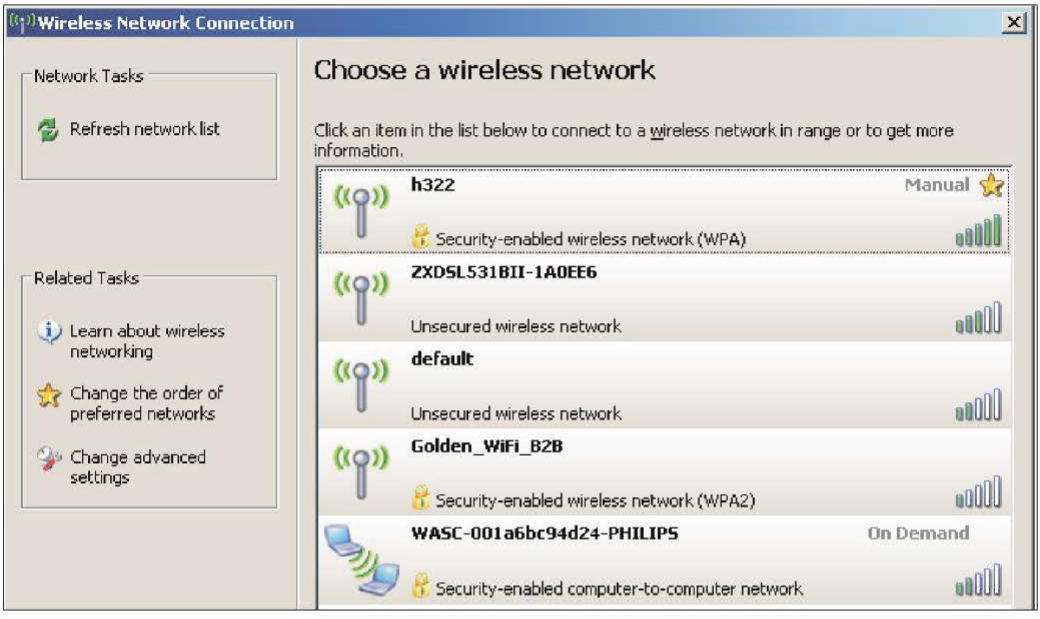


Рис. 25.17. Список доступных беспроводных сетей, полученный в ответ на запрос Probe Request с пустым значение SSID по всем каналам

2. Последовательно рассылаются запросы Probe Request с указанием SSID сетей, сконфигурированных в свойствах клиента (в списке предпочтительных сетей) и присутствующих в списке доступных сетей. Порядок опроса сетей задается их порядком в списке предпочтительных сетей (Preferred networks) (рис. 25.18).

3. Если ни одна из сетей не ответила на запрос или соединение с сетью не было установлено, опрашиваются сети, которые присутствуют в списке предпочтительных сетей, но отсутствуют в списке доступных сетей, что позволяет определить наличие сети, не рассылающей идентификатор во фреймах Beacon и Probe Response.

4. В случае если ассоциация с точками доступа не была установлена и соединение с одноранговыми сетями не запрещено (не включен режим Access Point Networks Only), проводится попытка установить соединение с Ad-Hoc-сетями, которые сконфигурированы в свойствах клиента (все в том же списке предпочтительных сетей).

5. Если одноранговые сети (из списка предпочтительных сетей) не были обнаружены, клиент настраивает беспроводной интерфейс в качестве перво-

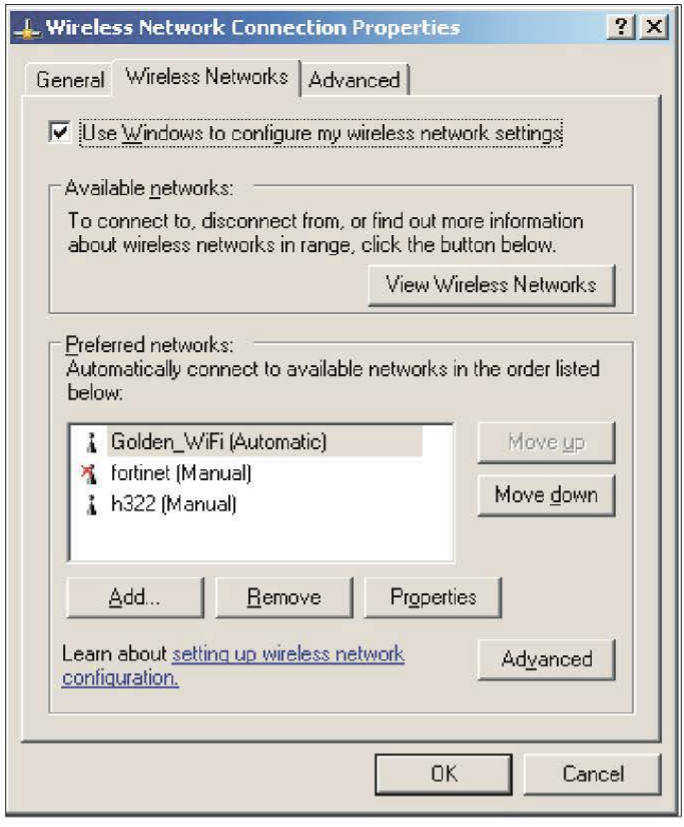


Рис. 25.18. Порядок опроса сетей

го узла такой сети и ожидает подключения других клиентов.

6. Если подключение к сетям Ad-Hoc запрещено, Wireless Auto Configuration проверяет значение параметра Automatically Connect To Non-Preferred Networks (по умолчанию этот режим отключен) (рис. 25.19).

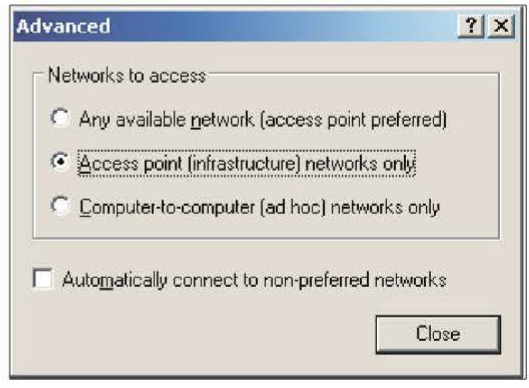


Рис. 25.19. Параметры подключения к беспроводным сетям

Если данная опция отключена, то адаптер переходит в режим инфраструктуры со случайным значением SSID.

7. Если параметр Automatically Connect To Non-Preferred Networks равен единице, клиент беспроводных сетей пытается соединиться с доступными сетями, полученными при опросе радиоэфира (см. п. 1).

Например, такая атака может быть осуществлена вследствие наличия нескольких профилей для подключения к беспроводным сетям (рис. 25.20).

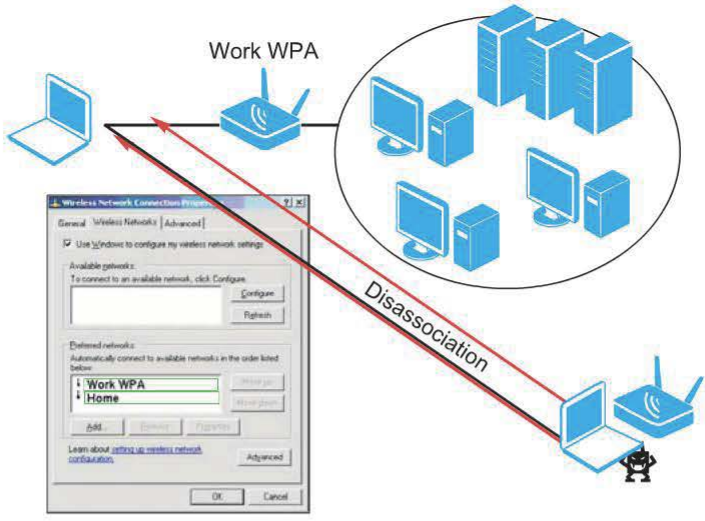


Рис. 25.20. Наличие нескольких профилей для подключения к беспроводным сетям

Довольно типичная ситуация — наличие двух профилей: для подключения к корпоративной сети (где задействованы все необходимые механизмы защиты) и для подключения к «домашней» сети (где возможно отсутствие большей части защитных механизмов).

Нарушитель, зная параметры профиля «домашней» сети (которые он может извлечь из перехваченного фрейма Probe Request), создает ложную точку доступа, имеющую данные параметры, и с помощью приведенных выше методов заставляет клиента ассоциироваться с ложной точкой доступа (рис. 25.21).

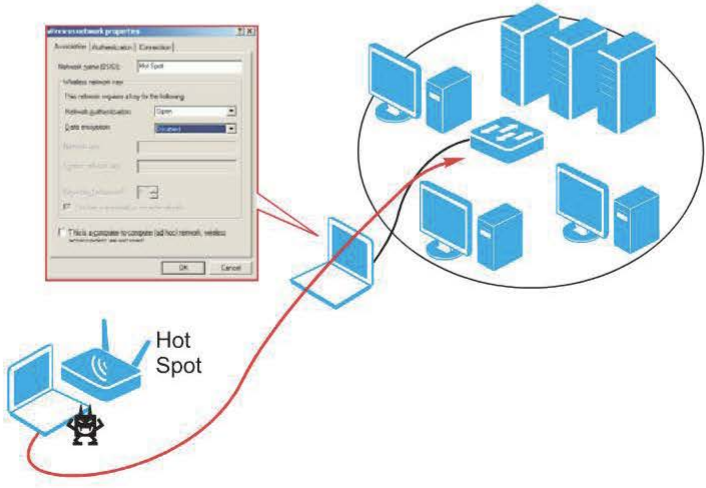


Рис. 25.21. Создание ложной точки доступа (инструменты — Hotspotter, Airsnarf)

Для защиты от подобных атак рекомендуется использовать следующие методы:

• в среде Windows использовать механизм групповых политик для применения параметров сетевого подключения (исключающих возможность использования других профилей подключения к сетям) в момент установления соединения с корпоративной сетью;

• использовать стандартные методы защиты клиентов VPN (фильтрацию трафика и предотвращение атак на уровне узла клиента, функцию «карантина»).

25.7. Мониторинг безопасности

Из приведенных выше примеров атак на беспроводные сети следует, что кроме построения сети по стандарту 802.11i необходимо осуществлять мониторинг трафика с целью обнаружения и предотвращения атак. Причем в отличие от обычной сети в беспроводной сети использование механизма обнаружения атак является необходимостью. В общем виде задача обнаружения атак в беспроводной сети сводится к обнаружению:

• несанкционированных точек доступа;
• несанкционированных беспроводных клиентов;
• атак, характерных для беспроводных сетей.

25.8. Особенности обнаружения атак

Казалось бы, система обнаружения атак сетевого уровня (network-based IDS) в беспроводной сети должна перехватывать сетевые пакеты и искать в них признаки атак, как и IDS для обычной сети. Основной источник данных для Wireless IDS — сетевой трафик (фреймы канального уровня). Но, поскольку в беспроводной сети обычно (как минимум) используется WEP, фреймы канального уровня зашифрованы. В этом случае сетевой сенсор не сможет искать признаки атак в перехваченном трафике, и поэтому атаки сетевого уровня и выше не будут обнаружены.

Сенсор, предназначенный для обнаружения «беспроводных» атак, рационально располагать вблизи точки доступа. Такой сенсор также может быть

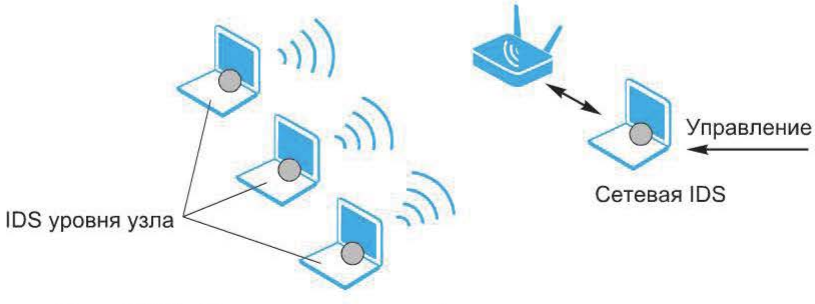


Рис. 25.22. Система противодействия атакам уровня узла

подключен непосредственно к точке доступа и контролировать трафик одного BSS.

Кроме того, на каждом узле-клиенте беспроводной сети рекомендуется поместить систему противодействия атакам уровня узла (рис. 25.22). Интегрированная со стеком TCP/IP защищаемого узла, такая система сможет просматривать трафик уже после его расшифрования и обнаруживать в нем признаки атак.

25.9. Атаки, характерные для беспроводных сетей

Wireless IDS будет обнаруживать следующие *разновидности атак*:

• атаки на физическом уровне;

• DoS-атаки с использованием передатчика, создание помех работе беспроводной сети;

• появление несанкционированного беспроводного устройства.

При этом сенсор руководствуется следующими *событиями физического уровня*:

• появление несанкционированного передатчика (беспроводной точки доступа или клиента, а также любого другого источника сигнала). Следовательно, IDS должна позволять задавать список разрешенных беспроводных устройств и обнаруживать несанкционированные;

• использование канала, отличного от обычного, непредвиденная смена канала;

• перегрузка и перекрытие каналов (overlapping channels);

• слабый сигнал, другие проблемы качества.

***Атаки на сервисы канального уровня*:**

• активный WarDriving;

• DoS-атаки на сервисы канального уровня;

• MAC address spoofing;

• попытка подбора ESSID (в случае выключения его широковещательной рассылки).

***События, связанные с сервисами канального уровня, на основе которых обнаруживаются данные атаки*:**

• фреймы неизвестного типа, нестандартного размера и т. п.;

• большое число фреймов deassociate или deauthenticate;

• фреймы с неизвестным ESSID (появление в сети несанкционированного беспроводного устройства);

• фреймы с ESSID="any";

• нарушение последовательностей фреймов при фрагментации;

• признаки использования инструментов для проведения атак;

• конфликт MAC-адресов;

• частая смена MAC-адреса;

• появление MAC-адреса, отсутствующего в списке разрешенных (если задействован механизм Port Security);

• большое число сообщений probe requests.

***Атаки, характерные для 802.11i*:**

- попытка обхода механизмов защиты 802.11i;
- несанкционированная точка доступа с поддержкой 802.1x;
- DoS-атаки на механизм аутентификации 802.1x.

***События, связанные с использованием 802.1x, на основе которых обнаруживается данный класс атак*:**

- некорректные, поврежденные фреймы 802.1x, а также имеющие нестандартный размер;
- фреймы, содержащие неиспользуемые типы EAP;
- большое число запросов/ответов (EAP authentication Request Response);
- большое число фреймов EAP start и EAP logoff;
- незавершенная процедура аутентификации 802.1x;
- фреймы EAP, связанные с несанкционированной точкой доступа.

25.10. Примеры систем обнаружения атак

В настоящее время имеется три типа решений по мониторингу беспроводной сети.

1. Программно-аппаратные:
   - AirDefense Guard (http://www.airdefense.net/products/airdefense_ids.shtm);
   - Wireless Sentry (http://www.isomair.com/products.html).
2. Программные:
   - WiSentry (http://www.wimetrics.com/products/download_wisentry.php);
   - AirMagnet from Global Secure Systems (http://www.gsec.co.uk/products/wireless_security.htm).
3. Программные свободно-распространяемые решения:
   - WIDZ;
   - Kismet;
   - Snort-Wireless.

Из свободно-распространяемых IDS для беспроводных сетей наибольшими возможностями обладает утилита Kismet.

Для использования системы Snort в беспроводной сети потребуются драйверы AirJack или HostAP. В этой системе предусмотрено специальное правило с ключевым словом wifi:

```
<action> wifi <src mac> -> <dst mac> (<rule options>)
```

Кроме того, предусмотрены два препроцессора: Rogue AP Preprocessor и Anti Stumbler Preprocessor.

25.11. ПО Air Magnet

ПО Air Magnet включает в себя три базовых продукта: Air Magnet Laptop, Air Magnet Surveyor и Air Manget Blue Sweep.

Air Magnet Laptop. Система Air Magnet Laptop представляет собой систему контроля беспроводных сетей. Принцип реализации — пассивный сбор информации с беспроводной сетевой карты в режиме мониторинга. В зависимости от используемой сетевой карты поддерживаются сети на основе 802.11a/b/g.

ПО устанавливается на ОС Microsoft Windows (XP/2003) и может применяться как в качестве отдельного продукта, так и в качестве сенсора распределенной системы контроля беспроводной сети Air Magnet Distributed.

Система поддерживает работу с GPS, что позволяет определять текущее положение рабочего места. Данная функция может использоваться для построения карты беспроводной сети, а также при локализации устройств.

По результатам сбора трафика отображается информация об обнаруженных клиентах и точках доступа, используемых сетевых протоколах и технологиях защиты. Присутствует возможность сохранять и анализировать трафик беспроводной сети, используя встроенный сетевой анализатор или внешние программы.

В поставку программы входят дополнительные утилиты, позволяющие измерять уровень сигнала для выбранной точки доступа или клиента, проводить диагностику качества связи в беспроводной сети и т. д.

Air Magnet Surveyor. Программа Air Magnet Surveyor используется для построения карты беспроводной сети. План помещения в виде графического файла импортируется в программу (поддерживаются различные растровые и векторные форматы), после чего проводится замер уровня сигнала из различных точек помещения. По результатам измерения соотношения сигнал/шум в каждой из точек строится карта распределения зон доступности для различных беспроводных точек доступа.

Поддерживаются активный и пассивный варианты оценки качества сигнала. В пассивном режиме данные собираются в режиме мониторинга на основе трафика в сети. Активный режим предполагает установление ассоциации с точками доступа для проверки качества соединения. При работе в активном режиме существует возможность указывать параметры соединения (ssid, ключи WEP и WPA, имя пользователя и пароль для EAP).

Для определения координат может использоваться система GPS, а также ручное указание расположения датчика на плане здания. Поддерживаются многоэтажные планы.

Air Manget Blue Sweep. Система представляет собой активный сканер устройств Bluetooth, поддерживающих подключения со всех устройств. По результатам сканирования выводится информация об обнаруженных устройствах и поддерживаемых ими сервисах. В настоящее время утилита не поддерживается производителем.

25.12. Обнаружение несанкционированных беспроводных устройств

Обнаружение беспроводных клиентов. В качестве клиента может выступать любое беспроводное устройство. Появление беспроводного устройства в сети необходимо обнаруживать. С точки зрения обнаружения беспроводные устройства можно разделить на две категории: пассивные и активные.

Пассивные: сетевые адаптеры таких устройств находятся в режиме мониторинга, они ничего не передают в эфир. Такие устройства, по сути, клиентами не являются.

Активные: сетевые адаптеры таких устройств осуществляют периодические попытки подключения к сетям, находящимся в зоне охвата антенны. Задача обнаружения пассивных устройств сложна и может быть выполнена только визуально.

Активные адаптеры периодически посылают фреймы Probe Request (рис. 25.23). Именно по ним и обнаруживаются беспроводные клиенты.

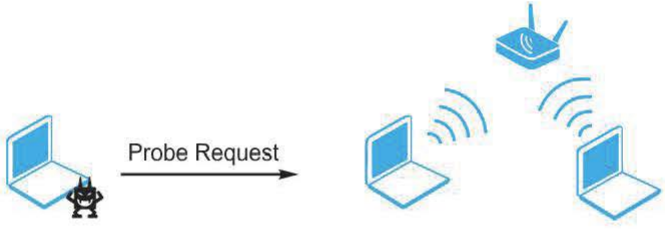


Рис. 25.23. Обнаружение беспроводных клиентов по фреймам Probe Request

Обнаружение точек доступа. Несанкционированные точки доступа (Rogue Access Points) представляют собой угрозу безопасности для всей корпоративной сети, прежде всего они делают уязвимым ее периметр. Пользователь может просто принести с собой точку доступа и установить ее в корпоративной сети, делая возможным подключение к сети, например, за пределами офиса. Кроме того, точка доступа может иметь различный функционал, например, DHCP-сервер, который позволит потенциальному нарушителю сразу получить правильный IP-адрес.

Для обнаружения несанкционированной точки доступа можно использовать следующие методы:

- визуальный контроль;
- обнаружение фреймов Beacon (или других фреймов, сигнализирующих о присутствии точки доступа);
- Nmap TCP Fingerprinting.

Первый метод предполагает осмотр тех мест, где потенциально могут быть установлены точки доступа: столы, подоконники и т. п.

Второй метод, по сути, сводит задачу к рассмотренной ранее задаче обнаружения беспроводных сетей, которая решалась с помощью программы Network Stumbler. Но в данном случае одной только программы Network Stumbler недостаточно. Нарушителем могут быть предприняты меры по снижению вероятности обнаружения, например может быть выключена широковещательная рассылка фреймов Beacon. Необходимо использование программ, переводящих сетевой адаптер в режим мониторинга и обнаруживающих любую активность, связанную с несанкционированно установленной точкой доступа.

Задача обнаружения беспроводных устройств (клиентов или точек доступа) может быть решена, например, с помощью следующих программ:

- Omni Peek;
- Air Magnet;
- Air Defence.

Наконец, еще один метод обнаружения точек доступа — сканирование диапазона адресов с помощью сканера портов. Точка доступа имеет IP-адрес для подключения к ней с целью управления. Обычно это осуществляется через web-интерфейс, следовательно, должен быть открыт 80-й порт. Кроме того, часто возможно управление с помощью протоколов telnet, ssh, snmp. Наиболее эффективным приемом является идентификация ОС по методу TCP Fingerprinting. На рис. 25.24 представлены результаты определения ОС точки доступа с помощью сканера nmap.

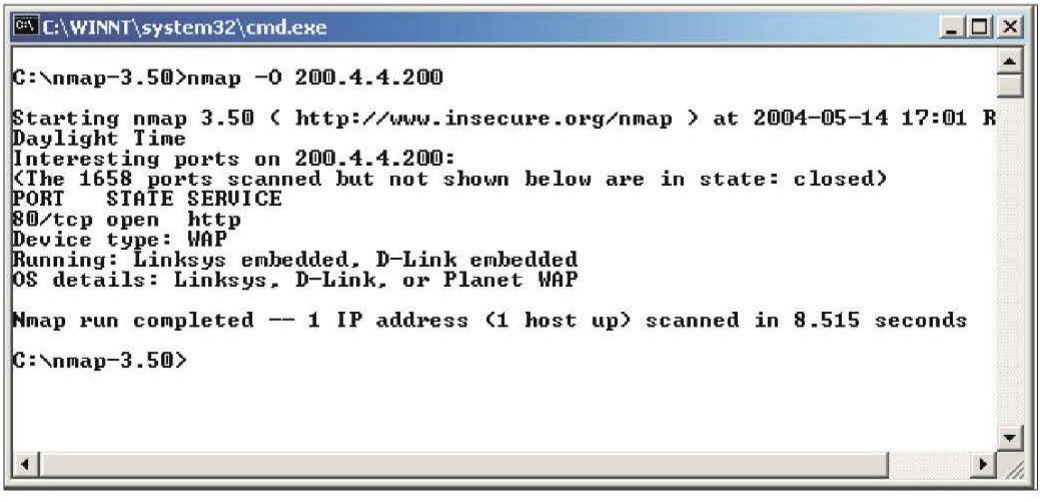


Рис. 25.24. Определение ОС точки доступа с помощью сканера nmap

В целом задачу обнаружения несанкционированных беспроводных устройств можно разделить на следующие этапы:

- сбор информации об используемых беспроводных сетях;
- анализ результатов, выбор разрешенных сетей и клиентов;

• ввод информации о разрешенных сетях и клиентах;
• обнаружение несанкционированных беспроводных устройств.

Эта задача может быть решена с помощью программы Air Magnet.

25.13. Контроль политики безопасности беспроводной сети системой Air Magnet

В системе Air Magnet предусмотрены следующие возможности по контролю принятой в компании политики безопасности беспроводной сети (рис. 25.25), в частности, обнаружение:

• несанкционированных клиентов;
• несанкционированных точек доступа;
• нарушений принятой политики защиты трафика.

Проверки каждой из групп могут применяться либо ко всем обнаруживаемым коммуникациям, либо для определенных групп точек доступа на основе SSID либо списков контроля доступа по MAC-адресам. Это позволяет задавать разные правила для различных сетей, например, контролировать клиентов, точки доступа и применение 802.1x и WPA в основной сети и не обращать внимания на незащищенные взаимодействия в расположенной рядом сети соседней компании. Кроме того, в разных частях корпоративной

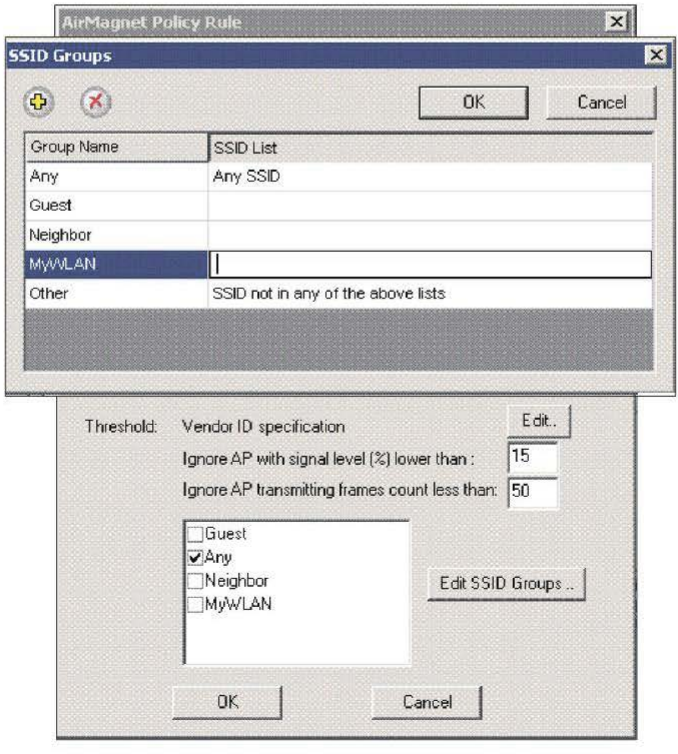


Рис. 25.25. Настройки системы Air Magnet

WLAN могут действовать различные политики безопасности (основная сеть, гостевая сеть).

Система Air Magnet может использовать следующие конфигурируемые критерии для определения принадлежности точки доступа и клиента сети компании:

- используемый канал 802.11b/g/a;
- идентификатор производителя в MAC-адресе (IEEE OUI);
- реализации протокола 802.11 (802.11a, 802.11b, 802.11g или различные сочетания);
- адрес канального уровня MAC-адрес;
- идентификатор сети (SSID).

Для каждой из категории можно задать «белые списки» MAC-адресов, идентификаторов OUI и т. д., обнаружение которых не вызывает срабатывания сигнатуры.

При этом контролируется использование следующих технологий защиты беспроводных сетей:

- шифрование (любое);
- аутентификация Open System/Shared Key;
- виртуальные частные сети на основе L2TP, IPSec, PPTP, SSH;
- технология 802.1x (динамические ключи WEP);
- шифрование TKIP (WPA);
- аутентификация Protected EAP (PEAP);
- аутентификация на общих ключах (WPA-PSK, 802.11i-PSK);
- аутентификация EAP-FAST;
- шифрование AES (802.11i);
- шифрование Fortress;
- шифрование Cranite.

Контрольные вопросы

1. Перечислите угрозы безопасности по отношению к беспроводным сетям.
2. В чем заключается несанкционированное использование беспроводных устройств?
3. Какие задачи решаются в ходе мониторинга безопасности беспроводной сети?
4. Каковы особенности обнаружения атак в беспроводных сетях?
5. Сформулируйте перечень атак, специфичных для беспроводных сетей.
6. Приведите примеры систем обнаружения атак в беспроводных сетях.

Глава 26. ИНТЕГРАЦИЯ СРЕДСТВ ОБНАРУЖЕНИЯ И ПРЕДОТВРАЩЕНИЯ АТАК В ЕДИНУЮ СИСТЕМУ И ВЗАИМОДЕЙСТВИЕ С ДРУГИМИ СРЕДСТВАМИ ЗАЩИТЫ

Во многих случаях решение по обнаружению и предотвращению атак строится с использованием различных (рассмотренных выше) технологий. Наряду с сетевыми системами обнаружения атак могут применяться системы защиты серверов и рабочих станций, а также специализированные системы защиты от атак.

В дополнение к этому одна и та же технология может быть реализована в разных продуктах (например, от разных производителей). Совместное применение таких продуктов может быть обусловлено желанием повысить отказоустойчивость системы в целом или уменьшить число ложных срабатываний.

Наконец, наряду с системами обнаружения и предотвращения атак в корпоративной сети применяются и другие средства защиты. Часто взаимодействие этих средств между собой повышает эффективность защиты в целом.

26.1. Интеграция средств обнаружения и предотвращения атак в единую систему

В целях защиты достаточно часто используются продукты и технологии от одного вендора, например, совместное применение средств обнаружения атак уровня сети (network-based) и уровня узла (host-based). В этом случае их интеграция в единую систему — это централизованное управление средствами обнаружения различных атак с помощью единой консоли (рис. 26.1).

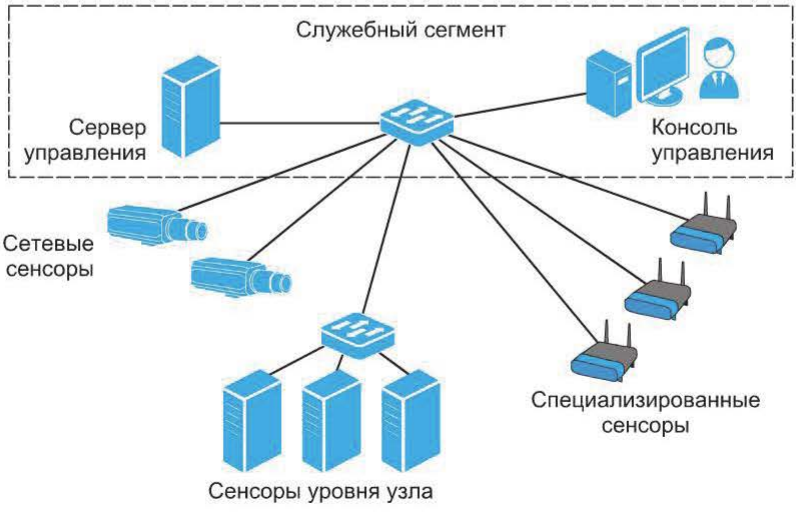


Рис. 26.1. Совместное применение средств обнаружения атак уровня сети и уровня узла

События, обнаруженные сенсорами различного типа, хранятся в единой базе, что упрощает процесс их анализа и сопоставления.

Другой способ интеграции — подключение средств обнаружения и предотвращения атак к системе централизованного мониторинга и управления событиями безопасности SEM (Security Event Management) и SIM (Security Information Management). Обычно в такую систему «стекается» информация из журналов (логов), в которые попадают события безопасности (рис. 26.2).

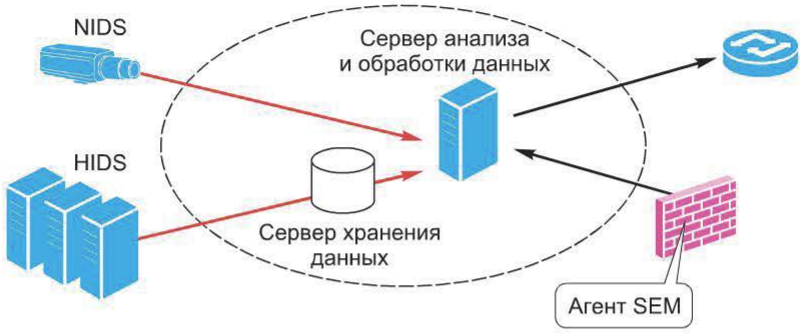


Рис. 26.2. Подключение IDS/IPS к системе централизованного мониторинга и управления событиями безопасности

При этом возможна схема подключения, которая предполагает прямое взаимодействие между серверами управления средствами обнаружения атак и SEM. Чаще всего такое взаимодействие осуществляется путем установки агента SEM на сервер управления средствами обнаружения атак (рис. 26.3).

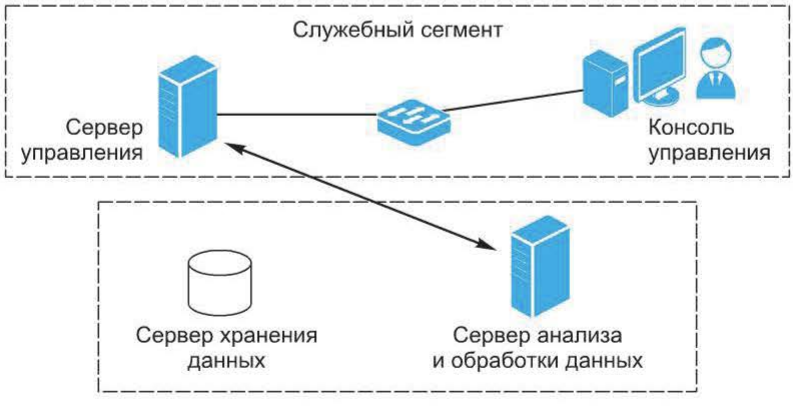


Рис. 26.3. Установки агента SEM на сервер управления средствами обнаружения атак

26.2. Примеры корреляции данных

Системы обнаружения атак, разумеется, не единственные средства защиты, применяемые в корпоративной сети.

Это делает возможной корреляцию данных, полученных из различных источников. Например, понять, была ли успешной атака на web-сервер, можно, проанализировав результаты недавнего сканирования (рис. 26.4).

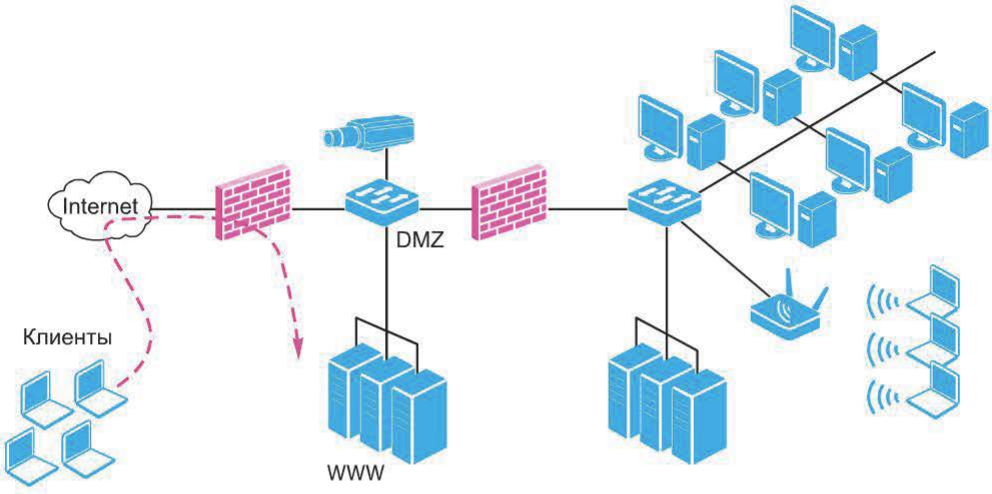


Рис. 26.4. Сканирование web-сервера

Для того чтобы найти источник атаки, иногда требуется проанализировать журналы прокси-сервера (рис. 26.5).

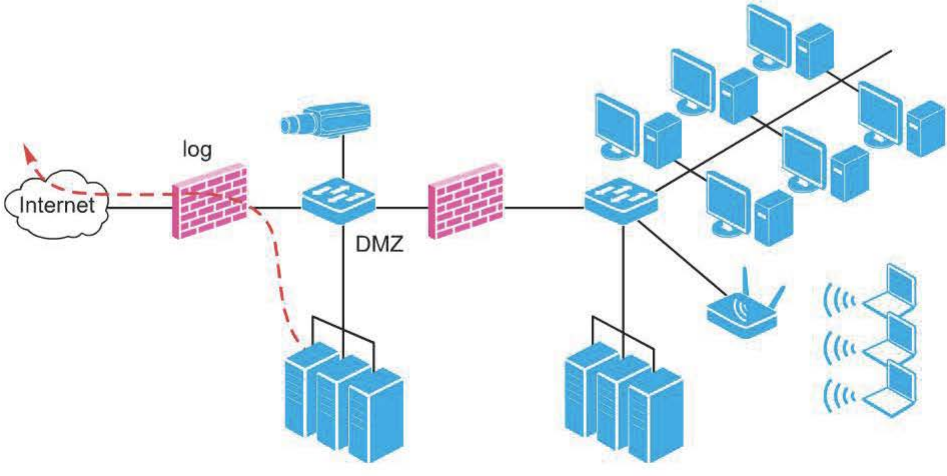


Рис. 26.5. Обнаружение источника атаки с помощью анализа журналов (log) прокси-сервера

В целом требуется сопоставлять следующие данные:

- другие события, обнаруженные тем же сенсором;
- журналы посредников (proxy), межсетевых экранов;
- результаты работы сканеров безопасности;
- журналы антивирусных систем.

Контрольные вопросы

1. Каким способом можно провести интеграцию IDS/IPS в единую систему?
2. Приведите примеры корреляции данных, полученных из различных источников, для обнаружения атак.

Литература

Абрамов Е.С., Сидоров И.Д. Метод обнаружения распределенных информационных воздействий на основе гибридной нейронной сети // Известия ЮФУ. Технические науки. 2009. № 11 (100). С. 154–164.

Аткина В.С. Применение иммунной сети для анализа катастрофоустойчивости информационных систем // Известия ЮФУ. Технические науки. 2011. № 12 (125). С. 203–210.

Баранова Е.К., Бабаш А.В. Информационная безопасность и защита информации: учеб. пособие. 3-е изд. М.: ИЦ РИОР; НИЦ ИНФРА-М, 2016. 322 с.

Будько М.Б., Будько М.Ю. Отслеживание изменений в структуре сети и решение задач повышения безопасности на основе анализа потоков данных // Научно-технический вестник Санкт-Петербургского государственного университета информационных технологий, механики и оптики. 2009. № 59. С. 78–82.

Жук А.П., Жук Е.П., Лепешкин О.М., Тимошкин А.И. Защита информации: учеб. пособие. 2-е изд. М.: ИЦ РИОР; НИЦ ИНФРА-М, 2015. 392 с.

Ищейнов В.Я., Мецатунян М.В. Основные положения информационной безопасности: учеб. пособие. М.: ИД «Форум»; НИЦ ИНФРА-М, 2015. 208 с.

Максимова Е.А., Корнева В.А. Оптимизация технологии безопасного информационного взаимодействия в корпоративных системах // Матер. XII Междунар. науч.-практ. конф. «ИБ-2012». Ч. II. Таганрог: Изд-во ТТИ ЮФУ, 2012. С. 124–129.

Максимова Е.А., Корнева В.А. Формализация действий злоумышленника при прогнозировании вторжений в корпоративную информационную систему // Актуальные вопросы информационной безопасности регионов в условиях глобализации информационного пространства. Матер. II Всеросс. науч.-практ. конф., Волгоград, 26 апреля 2013 г. Волгоград: Изд-во ВолГУ, 2013. С. 71–78.

Масленников Д. Развитие информационных угроз в первом квартале 2013 г. [Электронный ресурс] // Лаборатория Касперского. Аналитика от 15 мая 2013 г. URL: http://www.securelist.com/ru/analysis/208050801/Razvitie_informatsionnykh_ugroz_v_ pervom_kvartale_2013_goda (дата обращения 15.10.2013).

Никишова А.В. Кооперация агентов многоагентной системы обнаружения атак // Актуальные вопросы информационной безопасности регионов в условиях глобализации информационного пространства. Матер. II Всеросс. науч.-практ. конф., Волгоград, 26 апреля 2013 г. Волгоград: Изд-во ВолГУ, 2013. С. 118–120.

Никишова А.В. Архитектура типовой информационной системы для задачи обнаружения атак // Известия ЮФУ. Технические науки. 2011. № 12 (125). С. 104–109.

Платонов В.В. Программно-аппаратные средства защиты информации: учеб. для вузов по напр. подготовки «Информационная безопасность». М.: ИЦ «Академия», 2014. 331 с.

Партыка Т.Л., Попов И.И. Информационная безопасность: учеб. пособие. 5-е изд., перераб. и доп. М.: ИД «Форум»; НИЦ ИНФРА-М, 2016. 432 с.

Шаньгин В.Ф. Информационная безопасность компьютерных систем и сетей: учеб. пособие. М.: ИД «Форум»; НИЦ ИНФРА-М, 2014. 416 с.

McAfee Threats Report: Second Quarter 2013 [Электронный ресурс] // McAfee Labs. Reports.
URL: http://www.mcafee.com/ca/resources/reports/rp-quarterly-threat-q2-2013.pdf

Muller Jorg P. The Design of Intelligent Agents: a Layered Approach / Jorg P. Muller. Berlin; Heidelberg; New York: Springer, 1996. Vol. 1177.

Shoham Y., Leyton-Brown K. Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations. New York: Cambridge University, 2009.

Оглавление

Предисловие ... 3
Обозначения, используемые в книге ... 5
Глава 1. Постановка задачи анализа защищенности компьютерной системы ... 6
1.1. Корпоративная сеть как объект защиты ... 6
1.2. Событие безопасности ... 9
1.3. Понятие уязвимости ... 9
1.4. Классификация уязвимостей ... 11
1.5. Источники информации по уязвимостям ... 13
1.6. Принятые обозначения уязвимостей ... 19
1.7. National Vulnerability Database ... 21
1.8. Уязвимости и безопасность промышленных систем управления ... 26
Контрольные вопросы ... 32
Глава 2. Методы выявления уязвимостей и системы анализа защищенности ... 32
2.1. Основные приемы выявления уязвимостей ... 32
2.2. Выявление «известных» уязвимостей ... 35
2.3. Системы анализа защищенности ... 35
2.4. Примеры средств анализа защищенности ... 38
Контрольные вопросы ... 39
Глава 3. Сетевые сканеры безопасности ... 39
3.1. Размещение сетевых агентов сканирования в сети ... 40
3.2. Сетевые агенты и сбор информации ... 42
Контрольные вопросы ... 42
Глава 4. Способы сбора информации о сети. Предварительное изучение цели ... 43
4.1. Способы сбора информации о сети ... 43
4.2. Предварительное изучение цели ... 44
Контрольные вопросы ... 49
Глава 5. Идентификация сетевых объектов ... 50
5.1. Использование протокола ICMP ... 50
5.2. Идентификация узлов с помощью протокола ARP ... 57
Контрольные вопросы ... 58
Глава 6. Определение топологии сети ... 58
6.1. Отслеживание маршрутов ... 58

6.2. Отслеживание маршрутов и фильтрация 60
6.3. Утилита traceproto 61
Контрольные вопросы 61
Глава 7. Идентификация статуса порта 62
7.1. Сканирование портов 62
7.2. Сканирование портов TCP 63
7.3. Сканирование портов UDP 66
Контрольные вопросы 68
Глава 8. Идентификация сервисов и приложений 68
8.1. Идентификация TCP-служб 68
8.2. Идентификация UDP-служб 71
8.3. Сканирование протоколов 71
Контрольные вопросы 72
Глава 9. Идентификация операционных систем 73
9.1. Простейшие методы определения ОС 73
9.2. Опрос стека TCP/IP 75
9.3. Инструменты 78
9.4. SinFP 78
9.5. Использование протокола ICMP 79
9.6. Retransmission Timeout 81
9.7. Port 0 OS Fingerprinting 82
9.8. Активная идентификация ОС — перспективы 83
Контрольные вопросы 83
Глава 10. Идентификация уязвимостей по косвенным признакам 83
10.1. Методы идентификации уязвимостей по косвенным признакам 83
10.2. Баннерные проверки 84
10.3. Сетевые сервисы как объект сканирования 84
10.4. Локальные проверки 87
10.5. Механизмы взаимодействия с системами Windows 87
Контрольные вопросы 89
Глава 11. Passive Fingerprinting 89
11.1. Анализ сетевого трафика 89
11.2. Анализ запросов от сканируемого узла 93
Контрольные вопросы 94
Глава 12. Выявление уязвимостей с помощью тестов 94
12.1. Эксплойты и их разновидности 95
12.2. Использование техники запуска кода 95
12.3. Простые эксплойты 97
12.4. Удаленный подбор пароля 97
12.5. Оценка стойкости паролей 97
12.6. Тестирование 99
12.7. Анализ результатов 100

12.8. Отказ в обслуживании ... 101
Контрольные вопросы ... 101
Глава 13. Сетевой сканер Nessus ... 101
13.1. Обзор возможностей сканера ... 101
13.2. Архитектура сканера ... 102
13.3. Получение и установка сканера ... 103
13.4. Работа со сканером ... 105
Контрольные вопросы ... 109
Глава 14. Язык описания атак NASL ... 109
14.1. Структура сценария ... 109
14.2. Синтаксис языка и подключаемые библиотеки ... 113
Контрольные вопросы ... 116
Глава 15. Сканеры безопасности компании Positive Technologies ... 116
15.1. Краткая историческая справка ... 116
15.2. Архитектура и основные возможности сканера XSpider ... 117
15.3. Этапы работы сканера XSpider ... 118
15.4. Сбор информации о сети ... 118
15.5. Идентификация уязвимостей ... 121
15.6. Локальные проверки систем Windows ... 123
15.7. Выявление уязвимостей web-приложений ... 124
Контрольные вопросы ... 126
Глава 16. Анализ защищенности на уровне узла ... 126
16.1. Задачи локального сканирования ... 126
16.2. Архитектура ... 127
16.3. Сбор информации и идентификация уязвимостей ... 128
16.4. Сканер Assuria Auditor ... 129
Контрольные вопросы ... 130
Глава 17. Специализированные средства анализа защищенности ... 130
17.1. Классификация сканеров безопасности по назначению ... 130
17.2. Угрозы и уязвимости СУБД ... 132
17.3. Особенности анализа защищенности СУБД ... 133
17.4. Примеры программ-сканеров уязвимостей СУБД ... 136
Контрольные вопросы ... 136
Глава 18. Методология анализа защищенности Ethical Hacking ... 136
18.1. Необходимость методологии анализа защищенности ... 136
18.2. Penetration Testing — общие сведения ... 137
18.3. Разновидности Penetration Testing ... 138
18.4. Структура Penetration Testing ... 139
Контрольные вопросы ... 143
Глава 19. Централизованное управление уязвимостями ... 144
19.1. Необходимость централизованного управления уязвимостями ... 144
19.2. Инвентаризация информационных активов ... 145

19.3. Мониторинг состояния защищенности ... 145
19.4. Устранение уязвимостей и контроль ... 147
Контрольные вопросы ... 147
Глава 20. Контроль защищенности беспроводных сетей ... 148
20.1. Особенности сканирования беспроводных сетей ... 148
20.2. Сканеры для беспроводных сетей ... 148
20.3. Сканирование точки доступа на сетевом уровне ... 150
20.4. Методология аудита ... 152
Контрольные вопросы ... 155
Глава 21. Источники данных для систем обнаружения атак ... 155
21.1. Составляющие технологии обнаружения атак ... 155
21.2. Сетевой трафик как источник данных ... 157
21.3. Обнаружение атак на уровне узла ... 163
21.4. Host IDS — контроль действий субъектов системы ... 165
21.5. Составляющие обнаружения атак уровня узла ... 166
21.6. Анализ данных о потоке ... 167
Контрольные вопросы ... 168
Глава 22. Признаки атак ... 169
22.1. Использование уязвимостей как признак атаки ... 169
22.2. Отклонения от пороговых значений ... 171
22.3. Использование известных техник и инструментов для проведения атак ... 173
22.4. Система обнаружения атак Snort ... 174
Контрольные вопросы ... 179
Глава 23. Методы обнаружения атак ... 180
23.1. Обнаружение «злоупотреблений» ... 180
23.2. Обнаружение аномалий ... 182
Контрольные вопросы ... 185
Глава 24. Механизмы реагирования ... 185
24.1. Обзор механизмов реагирования ... 185
24.2. Варианты блокировки ... 186
Контрольные вопросы ... 188
Глава 25. Обнаружение атак в беспроводных сетях ... 189
25.1. Угрозы, связанные с использованием беспроводных сетей ... 189
25.2. IEEE 802.11i — нерешенные проблемы ... 190
25.3. Несанкционированное использование беспроводных устройств ... 190
25.4. Атаки на устройства и сервисы ... 191
25.5. Атаки на механизм аутентификации 802.1x ... 198
25.6. Атаки на клиентов ... 202
25.7. Мониторинг безопасности ... 208
25.8. Особенности обнаружения атак ... 208
25.9. Атаки, характерные для беспроводных сетей ... 209

25.10. Примеры систем обнаружения атак 210
25.11. ПО Air Magnet 211
25.12. Обнаружение несанкционированных беспроводных устройств 212
25.13. Контроль политики безопасности беспроводной сети системой Air Magnet 214
Контрольные вопросы 215
Глава 26. Интеграция средств обнаружения и предотвращения атак в единую систему и взаимодействие с другими средствами защиты 216
26.1. Интеграция средств обнаружения и предотвращения атак в единую систему 216
26.2. Примеры корреляции данных 218
Контрольные вопросы 219
Литература 220

Учебное издание

Бондарев Валерий Васильевич

Анализ защищенности
и мониторинг компьютерных сетей

Методы и средства

Редактор *Л.Т. Мартыненко*
Художник *Я.М. Асинкритова*
Корректор *Н.В. Савельева*
Компьютерная графика *О.В. Левашовой*
Компьютерная верстка *Т.В. Батраковой*

Оригинал-макет подготовлен
в Издательстве МГТУ им. Н.Э. Баумана.

В оформлении использованы шрифты
Студии Артемия Лебедева.

Подписано в печать 16.10.2017. Формат 70×100/16.
Усл. печ. л. 18,525. Тираж 100 экз. Изд. № 106-2016. Заказ

Издательство МГТУ им. Н.Э. Баумана.
105005, Москва, 2-я Бауманская ул., д. 5, стр. 1.
press@bmstu.ru
www.baumanpress.ru

Отпечатано в типографии МГТУ им. Н.Э. Баумана.
105005, Москва, 2-я Бауманская ул., д. 5, стр. 1.
baumanprint@gmail.com