# КАК РАБОТАЕТ СКАНЕР БЕЗОПАСНОСТИ?

## Введение

*В последнее время увеличилось число публикаций (в основном, зарубежных), посвященных такому новому направлению в области защиты информации, как адаптивная безопасность сети. Это направление состоит из двух основных технологий - анализ защищенности (security assessment) и обнаружение атак (intrusion detection). Именно первой технологии и посвящена данная статья.*

Сеть состоит из каналов связи, узлов, серверов, рабочих станций, прикладного и системного программного обеспечения, баз данных и т.д. Все эти компоненты нуждаются в оценке эффективности их защиты. Средства анализа защищенности исследуют сеть и ищут "слабые" места в ней, анализируют полученные результаты и на их основе создают различного рода отчеты. В некоторых системах вместо "ручного" вмешательства со стороны администратора найденная уязвимость будет устраняться автоматически (например, в системе System Scanner). Перечислим некоторые из проблем, идентифицируемых системами анализа защищенности:

- "люки" в программах (back door) и программы типа "троянский конь";
- слабые пароли;
- восприимчивость к проникновению из незащищенных систем;
- неправильная настройка межсетевых экранов, Web-серверов и баз данных;
- и т.д.

Технология анализа защищенности является действенным методом реализации политики сетевой безопасности прежде, чем осуществится попытка ее нарушения снаружи или изнутри организации.

Очень часто пишут об уникальных возможностях систем анализа защищенности (сканерах). Они предназначены для обнаружения только известных уязвимостей, описание которых есть у них в базе данных. В этом они подобны антивирусным системам, которым для эффективной работы необходимо постоянно обновлять базу данных сигнатур.

Функционировать такие средства могут на сетевом уровне (network-based), уровне операционной системы (host-based) и уровне приложения (application-based). Наибольшее распространение получили средства анализа защищенности сетевых сервисов и протоколов. Связано это, в первую очередь, с универсальностью используемых протоколов. Изученность и повсеместное использование таких протоколов, как IP, TCP, HTTP, FTP, SMTP и т.п. позволяют с высокой степенью эффективности проверять защищенность информационной системы, работающей в данном сетевом окружении. Вторыми по распространенности являются средства анализа защищенности операционных систем (ОС). Связано это также с универсальностью и распространенностью некоторых операционных систем (например, UNIX и Windows NT). Однако из-за того, что каждый производитель вносит в операционную систему свои изменения (ярким примером является множество разновидностей ОС UNIX), средства анализа защищенности ОС анализируют в первую очередь параметры, характерные для всего семейства одной ОС. И лишь для некоторых систем анализируются специфичные для нее параметры. Средств анализа защищенности приложений на сегодняшний день не так много, как этого хотелось бы. Такие средства пока существуют только для широко распространенных прикладных систем, типа Web-броузеры (Netscape Navigator, Microsoft Internet Explorer), СУБД (Microsoft SQL Server, Sybase Adaptive Server) и т.п.

Помимо обнаружения уязвимостей, при помощи средств анализа защищенности можно быстро определить все узлы корпоративной сети, доступные в момент проведения тестирования, выявить все используемые в ней сервисы и протоколы, их настройки и возможности для несанкционированного воздействия (как изнутри корпоративной сети, так и снаружи). Также эти средства вырабатывают рекомендации и пошаговые меры, позволяющие устранить выявленные недостатки.

Поскольку наибольшее распространение получили средства, функционирующие на уровне сети (системы SATAN, Internet Scanner, CyberCop Scanner, NetSonar и т.д.), то основное внимание будет уделено именно им.

**Механизмы работы**

Существует два основных механизма, при помощи которых сканер проверяет наличие уязвимости - сканирование (scan) и зондирование (probe).

***Сканирование*** - механизм пассивного анализа, с помощью которого сканер пытается определить наличие уязвимости без фактического подтверждения ее наличия - по косвенным признакам. Этот метод является наиболее быстрым и простым для реализации. В терминах компании ISS данный метод получил название "логический вывод" (inference). Согласно компании Cisco этот процесс идентифицирует открытые порты, найденные на каждом сетевом устройстве, и собирает связанные с портами заголовки (banner), найденные при сканировании каждого порта. Каждый полученный заголовок сравнивается с таблицей правил определения сетевых устройств, операционных систем и потенциальных уязвимостей. На основе проведенного сравнения делается вывод о наличии или отсутствии уязвимости.

***Зондирование*** - механизм активного анализа, который позволяет убедиться, присутствует или нет на анализируемом узле уязвимость. Зондирование выполняется путем имитации атаки, использующей проверяемую уязвимость. Этот метод более медленный, чем "сканирование", но почти всегда гораздо более точный. В терминах компании ISS данный метод получил название "подтверждение" (verification). Согласно компании Cisco этот процесс использует информацию, полученную в процессе сканирования ("логического вывода"), для детального анализа каждого сетевого устройства. Этот процесс также использует известные методы реализации атак для того, чтобы полностью подтвердить предполагаемые уязвимости и обнаружить другие уязвимости, которые не могут быть обнаружены пассивными методами, например подверженность атакам типа "отказ в обслуживании" ("denial of service").

На практике указанные механизмы реализуются следующими несколькими методами.

**"Проверка заголовков" (banner check)**

Указанный механизм представляет собой ряд проверок типа "сканирование" и позволяет делать вывод об уязвимости, опираясь на информацию в заголовке ответа на запрос сканера. Типичный пример такой проверки - анализ заголовков программы Sendmail или FTP-сервера, позволяющий узнать их версию и на основе этой информации сделать вывод о наличии в них уязвимости.

Наиболее быстрый и простой для реализации метод проверки присутствия на сканируемом узле уязвимости. Однако за этой простотой скрывается немало проблем.

Эффективность проверок заголовков достаточно эфемерна. И вот почему. Во-первых, вы можете изменить текст заголовка, предусмотрительно удалив из него номер версии или иную информацию, на основании которой сканер строит свои заключения. И хотя такие случаи исключительно редки, пренебрегать ими не стоит. Особенно в том случае, если у вас работают специалисты в области безопасности, понимающие всю опасность заголовков "по умолчанию". Во-вторых, зачастую, версия, указываемая в заголовке ответа на запрос, не всегда говорит об уязвимости программного обеспечения. Особенно это касается программного обеспечения, распространяемого вместе с исходными текстами (например, в рамках проекта GNU). Вы можете самостоятельно устранить уязвимость путем модификации исходного текста, при этом забыв изменить номер версии в заголовке. И в-третьих, устранение уязвимости в одной версии еще не означает, что в следующих версиях эта уязвимость отсутствует.

Процесс, описанный выше, является первым и очень важным шагом при сканировании сети. Он не приводит к нарушению функционирования сервисов или узлов сети. Однако не стоит забывать, что администратор может изменить текст заголовков, возвращаемых на внешние запросы.

**"Активные зондирующие проверки" (active probing check)**

Также относятся к механизму "сканирования". Однако они основаны не на проверках версий программного обеспечения в заголовках, а на сравнении "цифрового слепка" (fingerprint) фрагмента программного обеспечения со слепком известной уязвимости. Аналогичным образом поступают антивирусные системы, сравнивая фрагменты сканируемого программного обеспечения с сигнатурами вирусов, хранящимися в специализированной базе данных. Разновидностью этого метода являются проверки контрольных сумм или даты сканируемого программного обеспечения, которые реализуются в сканерах, работающих на уровне операционной системы.

Специализированная база данных (в терминах компании Cisco - база данных по сетевой безопасности) содержит информацию об уязвимостях и способах их использовании (атаках). Эти данные дополняются сведениями о мерах их устранения, позволяющих снизить риск безопасности в случае их обнаружения. Зачастую эта база данных используется и системой анализа защищенности и системой обнаружения атак. По крайней мере, так поступают компании Cisco и ISS.

Этот метод также достаточно быстр, но реализуется труднее, чем "проверка заголовков".

**"Имитация атак" (exploit check)**

Перевода термина "exploit" в российских публикациях нигде не встречалось, и эквивалента в русском языке также не было найдено. Поэтому воспользуемся переводом "имитация атак". Данные проверки относятся к механизму "зондирования" и основаны на эксплуатации различных дефектов в программном обеспечении.

Некоторые уязвимости не обнаруживают себя, пока вы не "подтолкнете" их. Для этого против подозрительного сервиса или узла запускаются реальные атаки. Проверки заголовков осуществляют первичный осмотр сети, а метод "exploit check", отвергая информацию в заголовках, позволяет имитировать реальные атаки, тем самым с большей эффективностью (но меньшей скоростью) обнаруживая уязвимости на сканируемых узлах. Имитация атак является более надежным способом анализа защищенности, чем проверки заголовков, и обычно более надежны, чем активные зондирующие проверки.

Однако существуют случаи, когда имитация атак не всегда может быть реализована. Такие случаи можно разделить на две категории: ситуации, в которых тест приводит к "отказу в обслуживании" анализируемого узла или сети, и ситуации, при которых уязвимость в принципе не годна для реализации атаки на сеть.

Как известно, многие проблемы защиты не могут быть выявлены без блокирования или нарушения функционирования сервиса или компьютера в процессе сканирования. В некоторых случаях нежелательно использовать имитацию атак (например, для анализа защищенности важных серверов), т.к. это может привести к большим затратам (материальным и временным) на восстановление работоспособности выведенных из строя элементов корпоративной сети. В этих случаях желательно применить другие проверки, например, активное зондирование или, в крайнем случае, проверки заголовков.

Однако, есть некоторые уязвимости (например, проверка подверженности атакам типа "Packet Storm"), которые просто не могут быть протестированы без возможного выведения из строя сервиса или компьютера. В этом случае разработчики поступают следующим образом, - по

умолчанию такие проверки выключены, и пользователь может сам включить их по желанию. Таким образом, например, реализованы системы CyberCop Scanner и Internet Scanner. В последней системе такого рода проверки выделены в отдельную категорию "Denial of service" ("Отказ в обслуживании"). При включении любой из проверок этой группы система Internet Scanner выдает сообщение **"WARNING: These checks may crash or reboot scanned hosts"**.

**Этапы сканирования**

Практически любой сканер проводит анализ защищенности в несколько этапов:

1. ***Сбор информации о сети.*** На данном этапе идентифицируются все активные устройства в сети и определяются запущенные на них сервисы и демоны. В случае использования систем анализа защищенности на уровне операционной системы данный этап пропускается, поскольку на каждом анализируемом узле установлены соответствующие агенты системного сканера.
2. ***Обнаружение потенциальных уязвимостей.*** Сканер использует описанную выше базу данных для сравнения собранных данных с известными уязвимостями при помощи проверки заголовков или активных зондирующих проверок. В некоторых системах все уязвимости ранжируются по степени риска. Например, в системе NetSonar уязвимости делятся на два класса: сетевые и локальные уязвимости. Сетевые уязвимости (например, воздействующие на маршрутизаторы) считаются более серьезными по сравнению с уязвимостями, характерными только для рабочих станций. Аналогичным образом "поступает" и Internet Scanner. Все уязвимости в нем делятся на три степени риска: высокая (High), средняя (Medium) и низкая (Low).
3. ***Подтверждение выбранных уязвимостей.*** Сканер использует специальные методы и моделирует (имитирует) определенные атаки для подтверждения факта наличия уязвимостей на выбранных узлах сети.
4. ***Генерация отчетов.*** На основе собранной информации система анализа защищенности создает отчеты, описывающие обнаруженные уязвимости. В некоторых системах (например, Internet Scanner и NetSonar) отчеты создаются для различных категорий пользователей, начиная от администраторов сети и заканчивая руководством компании. Если первых в первую очередь интересуют технические детали, то для руководства компании необходимо представить красиво оформленные с применением графиков и диаграмм отчеты с минимумом подробностей. Немаловажным аспектом является наличие рекомендаций по устранению обнаруженных проблем. И здесь по праву лидером является система Internet Scanner, которая для каждой уязвимости содержит пошаговые инструкции для устранения уязвимостей, специфичные для каждой операционной системы. Во многих случаях отчеты также содержат ссылки на FTP- или Web-сервера, содержащие patch'и и hotfix'ы, устраняющие обнаруженные уязвимости.
5. ***Автоматическое устранение уязвимостей.*** Этот этап очень редко реализуется в сетевых сканерах, но широко применяется в системных сканерах (например, System Scanner). При этом данная возможность может реализовываться по-разному. Например, в System Scanner создается специальный сценарий (fix script), который администратор может запустить для устранения уязвимости. Одновременно с созданием этого сценария, создается и второй сценарий, отменяющий произведенные изменения. Это необходимо в том случае, если после устранения проблемы, нормальное функционирование узла было нарушено. В других системах возможности "отката" не существует.

В любом случае у администратора, осуществляющего поиск уязвимостей, есть несколько вариантов использования системы анализа защищенности:

- Запуск сканирования только с проверками на потенциальные уязвимости (этапы 1,2 и 4). Это дает предварительное ознакомление с системами в сети. Этот метод является гораздо менее разрушительным по сравнению с другими и также является самым быстрым.
- Запуск сканирования с проверками на потенциальные и подтвержденные уязвимости. Этот метод может вызвать нарушение работы узлов сети во время реализации проверок типа "exploit check".

- Запуск сканирования с вашими пользовательскими правилами для нахождения конкретной проблемы.
- Все из вышеназванного.

**Особенности применения**

Если сканер не находит уязвимостей на тестируемом узле, то это еще не значит, что их нет. И зависит это не только от самого сканера, но и от его окружения. Например, если при тестировании сервиса Telnet или FTP на удаленной машине сканер сообщил, что уязвимостей не обнаружено - это может значить не только, что уязвимостей нет, а еще и то, что на сканируемом компьютере установлен, например, TCP Wrapper. Существует еще ряд причин, например, при попытке получения доступа к компьютеру через межсетевой экран эти попытки блокируются соответствующими фильтрами у провайдера и т.д. Для ОС Windows NT характерен другой случай. Сканер пытается дистанционно проанализировать системный реестр (registry). Однако в случае запрета на анализируемом узле удаленного доступа к реестру, сканер никаких уязвимостей не обнаружит. Существуют и более сложные случаи. И вообще различные реализации одного итого же сервиса по-разному реагируют на системы анализа защищенности. Очень часто на практике можно увидеть, что сканер показывает уязвимости, которых на анализируемом узле нет. Это относится к сетевым сканерам, которые проводят дистанционный анализ узлов сети. И удаленно определить, существует ли в действительности уязвимость или нет, практически невозможно. В этом случае можно порекомендовать использовать систему анализа защищенности на уровне операционной системы, агенты которой устанавливаются на каждый контролируемый узел и проводят все проверки локально.

Для решения этой проблемы некоторые компании-производители пошли по пути предоставления своим пользователям нескольких систем анализа защищенности, работающих на всех указанных выше уровнях, - сетевом, системном и уровне приложений. Совокупность этих систем позволяет с высокой степенью эффективности обнаружить практически все известные уязвимости. Например, компания Internet Security Systems предлагает семейство SAFEsuite, состоящее из четырех сканеров: Internet Scanner, System Scanner, Security Manager и Database Scanner. В настоящий момент это единственная компания, которая предлагает системы анализа защищенности, функционирующие на всех трех уровнях информационной инфраструктуры. Другие компании предлагают или два (Axent) или, как правило, один (Network Associates, NetSonar и др.) сканер.

Компания Cisco, предлагающая только систему анализа защищенности на уровне сети пошла другим путем для устранения проблемы ложного срабатывания. Она делит все уязвимости на два класса:

- *Потенциальные* - вытекающие из проверок заголовков и т.н. активных "подталкиваний" (nudge) анализируемого сервиса или узла. Потенциальная уязвимость возможно существует в системе, но активные зондирующие проверки не подтверждают этого.
- *Подтвержденные* - выявленные и существующие на анализируемом хосте.

Проверки на потенциальную уязвимость проводятся через коллекцию заголовков и использование "несильных подталкиваний". "Подталкивание" используется для сервисов, не возвращающих заголовки, но реагирующих на простые команды, например, посылка команды HEAD для получения версии HTTP-сервера. Как только эта информация получена, система NetSonar использует специальный механизм (rules engine), который реализует ряд правил, определяющих, существует ли потенциальная уязвимость.

Таким образом, администратор знает, какие из обнаруженных уязвимостей действительно присутствуют в системе, а какие требуют подтверждения.

Однако в данном случае остаются уязвимости, с трудом обнаруживаемые или совсем не обнаруживаемые через сеть. Например, проверка "слабости" паролей, используемых

пользователями и другими учетными записями. В случае использования сетевого сканера вам потребуется затратить очень много времени на удаленную проверку каждой учетной записи. В то же время, аналогичная проверка, осуществляемая на локальном узле, проводится на несколько порядков быстрее. Другим примером может служить проверка файловой системы сканируемого узла. Во многих случаях ее нельзя осуществить дистанционно.

Достоинства сканирования на уровне ОС кроются в прямом доступе к низкоуровневым возможностям ОС хоста, конкретным сервисам и деталям конфигурации. Тогда как сканер сетевого уровня имитирует ситуацию, которую мог бы иметь внешний злоумышленник, сканер системного уровня может рассматривать систему со стороны пользователя, уже имеющего доступ к анализируемой системе и имеющего в ней учетную запись. Это является наиболее важным отличием, поскольку сетевой сканер по определению не может предоставить эффективного анализа возможных рисков деятельности пользователя.

Многие сканеры используют более чем один метод проверки одной и той же уязвимости или класса уязвимостей. Однако в случае большого числа проверок использование нескольких методов поиска одной уязвимости привносит свои проблемы. Связано это со скоростью проведения сканирования.

Не все проверки, разработанные в лабораторных условиях, функционируют так, как должны. Даже, несмотря на то, что эти проверки тестируются, прежде чем будут внесены в окончательную версию сканера. На это могут влиять некоторые факторы:

- Особенности конфигурации пользовательской системы.
- Способ, которым был скомпилирован анализируемый демон или сервис.
- Ошибки удаленной системы.
- И т.д.

В таких случаях автоматическая проверка может пропустить уязвимость, которая легко обнаруживается вручную и которая может быть широко распространена во многих системах. Проверка заголовка в совокупности с активным зондированием в таком случае может помочь определить подозрительную ситуацию, сервис или узел. И хотя уязвимость не обнаружена, еще не значит, что ее не существует. Необходимо другими методами, в т.ч. и неавтоматизированными, исследовать каждый подозрительный случай.

**Разница в реализации**

Системы различных производителей могут использовать различные методы поиска одной и той же уязвимости, что может привести к ее нахождению в случае использования одного средства и ненахождения - в случае другого. Кроме того, если в созданном отчете не сказано о той или иной уязвимости, то иногда стоит обратиться к журналам регистрации (log) системы анализа защищенности. В некоторых случаях, когда сканер не может со 100%-ой уверенностью определить наличие уязвимости, он не записывает эту информацию в отчет, однако сохраняет ее в логах. Например, анализ и разбор поля sysDescr в журнале регистрации системы Internet Scanner существенно помогает во многих спорных случаях. Существуют различия и между тем, как влияет одна и та же проверка на различные версии сервисов в различных операционных системах.

**Перспективы развития**

С 1992 года, когда появился первый сканер SATAN, существенно изменились требования к такого рода средствам. Сейчас уже недостаточно, чтобы система анализа защищенности

обладала только обширной базой уязвимостей. Поэтому производители стали расширять функциональность своих продуктов за счет добавления следующих возможностей.

**Единый формат базы уязвимостей**

В целях унификации и возможной интеграции систем анализа защищенности в настоящий момент ведутся работы по созданию единого для всех сканеров формата базы уязвимостей. И хотя работа эта только началась и ей далеко до своего завершения, первые шаги уже сделаны. Например, лаборатория COAST в университете Purdue разработала проект такой базы данных. Одна из проблем, с которой пришлось столкнуться исследователям, - это описание уязвимостей и их проверок (атак).

**Языки описания уязвимостей и проверок**

Попытки добавить механизмы описания уязвимостей и проверок в системы анализа защищенности велись давно. Они предпринимались практически всеми компаниями-разработчиками. Первая такая попытка была предпринята Витсом Венема и Деном Фармером - разработчиками системы SATAN. Описание новых уязвимостей, точнее их проверок, осуществлялось при помощи языка Perl. Это достаточно нетривиальная задача требовала обширных знаний как языка Perl, так и архитектуры стека протоколов TCP/IP и сканируемой операционной системы. По этому же пути (использование Perl) пошли разработчики системы WebTrends Security Analyzer. В приложении 1 приведен пример проверки, позволяющей определить тип операционной системы сканируемого узла. Язык Perl, наряду с языком C, используется и в системе Internet Scanner. Причем помимо возможностей, встроенных в саму систему Internet Scanner, компания ISS поставляет отдельную систему описания атак APX (Advanced Packets eXchange).

Другим языком, используемым при описании осуществляемых проверок, стал Tcl. Модификации этого языка используются в системах APX (бесплатное приложение к системе Internet Scanner), Security Manager и CyberCop Scanner. Компания Network Associates последовала примеру компании ISS и выделила механизм описания уязвимостей в отдельную систему CyberCop CASL (Custom Audit Scripting Language). Также как и APX, система CyberCop CASL может функционировать под управлением ОС Windows NT и Unix (Linux для CASL и Solaris для APX).

В системах APX и CASL описываются параметры сетевых пакетов, при помощи которых моделируются различные атаки. К таким параметрам можно отнести флаги в заголовке IP-пакета, номера портов в заголовке TCP-пакета, поля данных в пакетах различных протоколов и т.д. В качестве примера (Приложение 2) можно привести проверку возможности осуществления подмены пакетов (Spoofing).

Однако наиболее удобным с точки зрения конечного пользователя (не программиста) является язык VDL (Vulnerability Descriptive Language) и VEL (Vulnerability Exploit Language), разработанный компанией Cisco. Проверки, описываемые этими языками, основаны на простых логических утверждениях, и пользователь может добавлять правила, если он видит, что они необходимы. Примером такого правила может быть:

*# Секция описания сервисов: На анализируемом узле найден netstat port 15 using protocol tcp => Service:Info-Status:netstat*

Данная проверка описывает правило, которое определяет наличие сервиса netstat на 15-ом TCP-порту анализируемого узла. Более сложное следующее правило определяет наличие запущенного приложения SuperApp устаревшей версии по заголовку, возвращаемому на запрос, обращенный к портам 1234 или 1235.

*# Пользовательская проверка: Приложение SuperApp 1.0 запущено на сканируемом хосте. (scanfor "SuperApp 1.0" on port 1234) || (scanfor "SuperApp 1.0 Ready" on port 1235) => VULp:Old-Software:Super-App-Ancient:10003*

Данная потенциальная уязвимость (VULp) относится к типу "устаревшее (потенциально уязвимое) программное обеспечение" (Old-Software) и носит название Supper-App-Ancient, задаваемое пользователем. Число 10003 определяет уникальный номер записи в базе данных уязвимостей системы NetSonar (NSDB).

Механизм описания своих проверок и уязвимостей является очень полезной возможностью для администраторов, отслеживающих уязвимости, описанные в Bugtraq и иных списках рассылки.Эта возможность позволяет быстро записать новое правило и использовать его в своей сети. Однако можно заметить, что язык, используемый в системе NetSonar и описывающий эти правила, достаточно элементарен и может помочь только в самых простых случаях. В сложных ситуациях, когда проверку нельзя записать одним правилом, необходимо использовать более сложные сценарии, которые достигаются применением языков Perl, Tcl и C.

Необходимо заметить, что хотя данная возможность и является полезной, ее эффективность достаточно эфемерна. В своей практической деятельности мне не приходилось встречаться с организациями, которые могли бы себе позволить содержать целый штат или одного сотрудника, занимающихся исследованиями в области новых проверок и уязвимостей (я не беру в расчет силовые ведомства и иные организации, работающие в области защиты информации). Как правило, человек, отвечающий за обеспечение безопасности, не обладает глубокими познаниями в программировании. Кроме того, помимо анализа защищенности на нем "висит" еще много других задач (контроль пользователей, установка прав доступа и т.д.), и он просто не имеет времени для такой творческой работы, как описание новых проверок.

**Заключение**

Использование таких средств полезно, но не является решением всех вопросов. Основное назначение сканеров автоматизировать работу, помогая проверить множество узлов, а также подсказать меры по устранению уязвимостей.

При написании данной статьи мною были использованы материалы компании Internet Security Systems, Inc., Cisco Systems и Network Associates.