Разработка методов и средств противодействия начальному этапу сетевого вторжения

Е.С. Сагатов[1], С. Майхуб[1], А.М. Сухов[1], М.А. Баймяшкин[1]

[1]Самарский национальный исследовательский университет им. академика С.П. Королева, Московское шоссе 34А, Самара, Россия, 443086

Аннотация. Настоящая статья посвящена разработке методов противодействия начальному этапу сетевой атаки. Анализ исходящего трафика позволил сформулировать квалификационные признаки сканирования TCP и UDP портов. Появление ICMP 3.3 и TCP RST пакетов откликов свидетельствует о начале процесса сканирования. IP адреса пунктов назначения из этих пакетов блокируются на 5 минут. Разработанные программные комплексы в виде SDN модуля и Linux утилиты протестированы и показали высокую эффективность.

1. Введение

Масштабные вмешательства в сетевую инфраструктуру стали нормой в современном мире. Все чаще угрозы о сетевых атаках звучат на межгосударственном уровне. Современные сетевые атаки [1] носят сложный характер, состоят из нескольких этапов и предусматривают различные варианты нанесения ущерба.

Тем не менее, все существующие типы атак начинаются одинаково. На первом этапе производится разведка сетевого подключения информационной системы. Она включает, прежде всего, сканирование портов [2]. При этом ищутся открытые порты, по их номерам и конфигурации можно сделать выводы об используемой операционной системе и типах установленного программного обеспечения. Опираясь на эти данные, злоумышленники составляют модель атаки.

Если системному администратору удастся затруднить этап разведки, помещать сканированию портов и выдавать данные о первоначальной конфигурации портов сервера, то этот сильно осложнит дальнейшие действия злоумышленников [3].

Настоящая статья будет посвящена разработке мер по противодействию сканированию портов. Решать эту проблему можно по-разному, существует целый ряд соответствующих технологий. В современных технологиях часто используют нейронные сети и при их помощи делают ты выводы о правомерности тех или иных сетевых запросов. Тем не менее, все подходы, основанные на обучении с применением нейронных сетей [4], обладают рядом существенных недостатков. Главный недостаток, это то, что при появлении нового метода атаки пройдет несколько часов или даже дней перед тем, как будут найдены квалификационные признаки атаки. Это время уйдет на сбор статистики и обучение. О проблемах наборов данных для обучающих систем говориться в одной из самых цитируемых за последние пару лет статей [5].

Мы же пойдем другим путем. Для получения необходимых сведений злоумышленники должны проводить тестирование всех портов, причем пакетами обоих протоколов: TCP и UDP. При этом происходит множество обращений к закрытым портам. Изучая отклики с закрытых портов, можно сформировать квалификационные признаки начального этапа атаки – сканирования портов.

Поиск подобных квалификационных признаков [6] это одна из целей настоящего исследования. После того, как эти квалификационные признаки будут сформулированы, можно приступить к разработке программного обеспечения, которое будет противодействовать атакам на этапе разведки.

В результате исследований будут найдены квалификационные признаки сетевой атаки. Новым в нашем подходе является то, что анализироваться будет не входящий трафик, а исходящий, то есть пакеты отклики на атакующие запросы. Квалификационные признаки будут получены в ходе статистического анализа исходящего трафика. В их основу будет положена повторяемость пакетов откликов на атакующие запросы. Эти пакеты будут группироваться по типам и протоколам, в зависимости от типов атакующих запросов. В первую очередь, поиск квалификационных признаков будет осуществлен для противодействия начальному этапу атаки, который состоит в сканировании портов атакуемого сервера.

Данное программное обеспечение будет реализовано в двух вариантах. Первый вариант это SDN модуль [7]. Вторая реализация — это утилита для Linux сервера непосредственного действия. Но действующий алгоритм защиты будет единым и основан на найденном квалификационном признаке. Утилита для Linux легко может быть доработана для ОС Android, необходимо только добавить оригинальные механизмы блокирования трафика со выделенных IP адресов.

Статья организованна следующим образом. Во втором разделе будут сформулированы основные задачи защитной инфраструктуры. Третий раздел стати будет посвящён формулировке квалификационного признака для выделения разведывательных запросов. В четвертом разделе описаны разработка и тестовые испытания защитной утилиты для Linux. Раздел 5 статьи посвящён разработке и тестированию SDN модуля для защиты от сканирования. В этом разделе приведены данные об тестировании разработанного программного обеспечения на реальной сети. Раздел 4 описывает утилиту, которая запускается на Linux сервере и противодействует разведке. В этом разделе также обсуждается возможность доработки этой программы для ОС Android и использование этого программного пакета для защиты смартфонов.

2. Основные задачи защитной инфраструктуры

Выявление атакующих IP адресов и блокировка трафик с них – это одна из важнейших задач данного исследования. Выявление можно проводить, анализируя входящие или исходящие пакеты от защищаемого ресурса. Наше исследование предполагает анализ начального этапа любой сетевой атаки (этап разведки). На этом этапе злоумышленник пытается понять какие порты открыты и по этим портам определить типы сервисов и программного обеспечения. Наша задача состоит в том, чтобы затруднить злоумышленнику сбор данных на первом этапе вторжения. Для этого будут найдены особенности сканирования портов и сформулированы соответствующие квалификационные признаки. Новизна нашего подхода будет заключаться в том, что мы планируем анализировать пакеты-отклики от защищаемого сервера.

Вторая задача в рамках общей проблемы сетевой безопасности состоит в создании защитной инфраструктуры. Это необходимо для того, чтобы обнаруживать сетевые вторжения. Эта инфраструктура должна измерять некоторые сетевые параметры и на основе их значений делать выводы об аномальной сетевой активности, которая исходит от подозрительных IP-адресов. Эти выводы делаются на основе квалификационных признаков. Формулировка ключевых признаков рассматривается в разделе 3 данной статьи.

Третья задача состоит в создании специализированного программного обеспечения для блокировки атакующего трафика. Данное ПО будет использовать различные защитные механизмы на основе свободно распространяемого программного обеспечения. В качестве базы

выбрана технология iptables межсетевого экрана Linux. Но особое внимание будет уделено технологиям программно-конфигурируемых сетей (SDN) так, как только эти технологии позволяют организовать взаимодействие между провайдерами разного уровня. Как правило, защитные механизмы требуется размещать на два уровня вверх от защищаемого ресурса.

3. Квалификационные признаки начального этапа сетевой атаки

При подготовке сетевой атаки злоумышленник, как правило, начинает вторжение с поиска открытых портов, а также изучения откликов на различные типы ICMP запросов. Как правило порты привязаны к интернет сервисам стандартным образом, получение списка открытых портов позволяет предположить, какое программное обеспечение установлено на данном ресурсе. Благодаря этому злоумышленник может составить список возможных уязвимостей для приложений, установленных на атакуемом сервере. В результате добывается информация, которую можно использовать для последующего взлома.

Существует множество программ, которые используются злоумышленниками для проведения сетевых атак. Отметим наиболее популярные из них: Nmap, Hping3 и LOIC.

Nmap является стандартной утилитой для сканирования портов. Она может быть использована для проверки безопасности, или же просто для определения сервисов запущенных на узле, для идентификации ОС и приложений, определения типа фаервола используемого на сканируемом узле.

Hping3 – генератор пакетов и анализатор для TCP/IP протокола. Он поддерживает протоколы TCP, UDP, ICMP, имеет режим traceroute, возможность отправки файлов между закрытым каналом и многими другими функциями.

LOIC - утилита, предназначенная для осуществления DDoS-атак, написанная на языке программирования C#. Осуществляет атаки по протоколам TCP, UDP или HTTP.

При сканировании портов утилитами атакуемый сервер даёт отклики. Эти отклики были записаны нами во время тестовой атаки и проанализированы. Как результат этого анализа, были сформулированы квалификационные признаки сканирования портов.

В ходе поисков мы применим принципиально новый поход. Квалификационные признаки можно искать как для атакующих пакетов, так и для пакетов-откликов на атакующие запросы. Анализ пакетов откликов проще, так как он содержит меньше типов пакетов. То есть в качестве квалификационных признаков будем рассматривать пакеты отклики определенных типов. В заголовке подобных пакетов содержится информация от источниках атаки.

Для портов UDP таким квалификационным признаком является отклик, содержащий пакет ICMP типа 3.3 (порт недостижим). При повторном появлении на защитном устройстве такого пакета с одного и того же внешнего IP адреса, получение пакетов с этого IP адреса должно быть блокировано.

При сканировании портов пакетами TCP SYN, TCP NULL, TCP FIN и TCP XMAS отправляется пакет TCP с флагами ACK и RST, последний сбрасывает соединение. При сканировании пакетами TCP ACK и TCP сканировании Меймона в ответ посылается TCP пакет с флагами RST. Фактически при любом сканировании пакетами TCP квалификационным признаком сканирования является исходящий от сервера пакет с флагом RST. При повторном появлении подобного пакета трафик с адреса получателя должен быть заблокирован.

Для ICMP пакетов мы не стали искать квалификационные признаки, так как их достаточно трудно сформулировать, а полный запрет на передачу пакетов этого типа может привести к блокированию проверок по работоспособности сети.

4. Защитная утилита для Linux

На основе квалификационных признаков разработаны алгоритмы противодействия сетевых вторжениям разных типов. Эти алгоритмы положены в основу разработанного программного обеспечения. Для различных категорий пользователей требуется программное обеспечение разной производительности. Наивысшая производительность достигается при использовании SDN технологий. Поэтому первая реализация нашего защитного механизма будет доступна в виде SDN модуля, но об этом говориться в следующем разделе этой статьи.

На другом полюсе потребителей находятся сетевые пользователи со смартфонами, ноутбуками и устройствами интернет вещей, для них не нужны защитные механизмы большой производительности. Основное требование к персональному защитному программному обеспечению – это низкая требовательность к вычислительным ресурсам, потребляемым мощности процессора и оперативной памяти. Поэтому вторая реализация будет выполнена в виде простой утилиты, которая запускается в операционной систем Linux. В ходе выполнения проекта будет произведено первичное тестирование разработанного программного обеспечения на созданном экспериментальном полигоне.

Эта утилита написана на языке C, она анализирует заголовки исходящих пакетов. Если соответствующие биты показывают, что исходящий пакет является пакетом TCP с флагом RST или пакетом ICMP версии 3.3, то из этого пакета извлекается адрес назначения, а специальный счетчик увеличивает значение на единицу. После того, как значение счетчика достигнет трех утилита *iptables* прекращает прием любых входящих пакетов с этого адреса на 5 минут (300 секунд).

Тестирование проводилось на специальном сервере под управлением ОС Linux, на котором была инсталлирована разработанная утилита. Сканирующая утилита *nmap* запускалась с IP адреса из другого сегмента локальной сети.

На тестируемом сервере открыто по 100 портов TCP и UDP, которые наиболее часто используются приложениями. Частота использования портов была определена с помощью данных с серверов ловушек [спросить Сагатова].

В ходе эксперимента осуществляется сканирование портов и происходит поиск открытых и закрытых портов TCP и UDP. Причем сканирование осуществляется 2 раза, в ходе первого сканирования защитные механизмы отключены. Второе сканирование происходит при активированных защитных механизмах. Сведения о портах, полученные в ходе обоих экспериментов, сравниваются между собой, а также данными о портах, полученными непосредственно с сервера при помощи команды netstat (спросить Сагатова).

Результаты измерений сведены в Таблицу 1.

Таблица 1. Результаты тестирования утилиты.

Режим тестирования	UDP			TCP		
	Открыто (open)	Закрыто (closed)	Ответ не получен (open|filtered)	Открыто (open)	Закрыто (closed)	Ответ не получен (filtered)
Без защитной утилиты	0	65 435	101	99	65 436	1
С включённым защитным механизмом	0	1333	64203	14	1262	64260

Сканирование UDP одушевляется путем отправления пустого заголовка UDP (UDP-пакет без нагрузки) на каждый порт сканируемой машины. На основании ответа, или его отсутствия, порт присваивается одному из четырех состояний. Получение любого UDP отклика от сканируемого порта означает, что порт открыт. Получение пакета ICMP типа 3.3 означает, что UDP порт закрыт. Сканирование ограничивается этими двумя случаями, если защитная утилита отключена. Включение защитной утилиты ограничивает поступление любых откликов от сканируемого сервера. Если ответ не поступил, то *nmap* относит такое состояние к неопределенным (filtered). При включенной защите абсолютное большинство портов (65165 из 65536) распознается как неопределенные.

Сканирование TCP осуществляется путем отправления пакет TCP с флагом SYN. Этот метод сканирования TCP-портов часто называют "полуоткрытым" сканированием, поскольку полное TCP-соединение с портом не устанавливается. На основании ответного пакета или его

отсутствия делается заключение о статусе порта. Он может находиться в одном из трёх состояний (open-closed-filtered). Получение отклика TCP с флагами SYN/ACK от сканируемого порта означает, что порт открыт. Получение отклика TCP с флагами RST от сканируемого порта означает, что порт закрыт. Отсутствие ответа или получение пакета ICMP любого типа означает, что TCP порт находиться в неопределенном состоянии (filtered).

Данные Таблицы 1 свидетельствуют о том, что разработанная нами утилита кардинально искажает данные о состоянии TCP и UDP портов. В случае UDP портов определить, какие порты открыты, практически невозможно. Точно можно утверждать, что 2% портов закрыты. Разработанная утилита дает возможность правильно определять статус открытого порта только для 14% TCP портов. В то же время статус закрытого порта подтверждается только для 1.9% сканируемых TCP портов.

5. SDN модуль для противодействия сканированию портов

Утилитой, которая блокирует сканирование портов, наши разработки не ограничились. Наиболее перспективной технологией для защиты от сетевых атак является, на наш взгляд, технология программно-конфигурационных сетей. Она не имеет аналогов по производительности, быстродействию, универсальности и спектру распознаваемых угроз.

В этом разделе мы представляем специализированный модуль для SDN контроллера F*loodlight*, разработанный нами. Этот модуль предназначен для блокирования трафика с IP адресов, с которых ведется сканирование портов. Код модуля написан на языке *java*. Он обрабатывает весь исходящий трафик с защищаемого сервера. При попадании на коммутатор пакета нового типа, он переправляет его на контроллер для создания соответствующего правила. Правило определяет порядок действий с пакетом.

Правила основываются на квалификационных признаках, сформулированных в разделе 3. Их программная реализация содержится в специализированном модуле. Созданные правила загружаются на коммутатор сообщениями FlowMod.

Весь входящий трафик обрабатывается. Первоначальные правила предписывают пересылку всех пакетов ICMP и пакетов TCP с флагом RST с коммутатора на контроллер. Для этого выделен специально зарезервированный порт (CONTROLLER). Пакеты ICMP и TCP RST перенаправляются на контроллер в формате сообщений типа Packet-in.

Модуль содержит счетчики и таймеры. Счетчик считает количество пришедших с IP адреса пакетов ICMP и TCP RST. Каждые 150 секунд он обнулит значение счетчика, если оно меньше 3. Если этот счетчик достигает значения 3 для фиксированного IP адреса, то контролер создает новое правило с более высоким приоритетом и пересылает его на коммутатор. Это правило предписывает блокировать все входящие и исходящие пакеты на этот фиксированный IP адрес. Эта блокировка по умолчанию длится 5 минут.

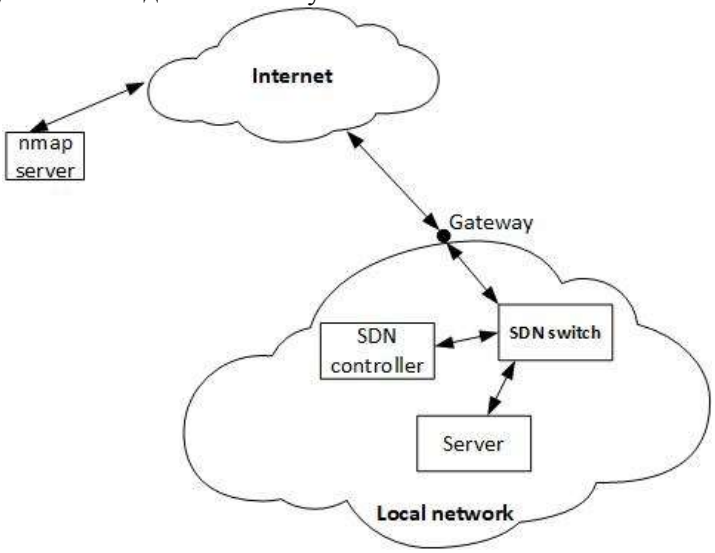

Рисунок 1. Схема полигона с защитной SDN инфраструктурой.

Для проведения тестовых испытаний был построен специализированный полигон для анализа сетевых атак и тестирования разработанных защитных механизмов. Этот полигон будет представлять сегмент локальной сети, подключенный к интернет, в котором будет инсталлировано несколько серверов на публичных IP адресах (см. рис. 1а). Один из серверов сможет предоставлять виртуальные машины, в том числе и все заданного сегмента. Сегмент будет управляться SDN коммутатором, а на серверах будет инсталлирован контроллер. Вне сегмента на одном из серверов будет собран комплект программ, применяемых хакерами для нападения.

В ходе тестирования *nmap* сервер, расположенный все локальной сети, сканировал TCP и UDP порты защищаемого ресурса. Сканирования также производилось дважды. Во время первого тестирования SDN модуль был отключен, второе тестирование проводилось при включенном SDN модуле. Результаты тестирования приведены в Таблице 2.

Таблица 2. Результаты тестирования SDN модуля.

Режим тестирования	UDP			TCP		
	Открыто (open)	Закрыто (closed)	Ответ не получен (open|filtered)	Открыто (open)	Закрыто (closed)	Ответ не получен (filtered)
Без защитного модуля	0	65 435	101	99	65 436	1
С включённым защитным механизмом	0	357	65179	15	135	65386

Сканирование при отключенном модуле позволило определить, открыт или закрыт тот или иной TCP или UDP порт с высокой точностью. В процессе тестирования была обнаружена только одна ошибка для портов обоих типов. Активация защитного SDN модуля позволила снизить точность сканирования во много раз. Так, точность определения открытых TCP портов составила 15%. Для закрытых UDP портов правильно определен статус лишь в 0.5% случаев. Для закрытых TCP портов эта точность еще меньше и составляет 0.2%. Следует отметить, что эффективность SDN модуля выше в три четыре раза, чем для утилиты. То есть, он позволяет точно установить статус для числа портов, которое в три-четыре раза меньше, чем при защите с помощью утилиты.

6. Выводы

В настоящей статье был представлен метод противодействия начальному этапу любой сетевой атаки. На этом этапе производится сканирование TCP и UDP портов и определяется, какие из них открыты. Эта информация позволяет собрать общие сведения об атакуемом сервере, тип операционной системы, установленное программное обеспечение, особенности удаленного управления и т.д.

Для того, чтобы разработать алгоритмы, которые определяли IP адреса, с которых ведется сканирование, были сформулированы квалификационные признаки. Для того, чтобы их сформулировать изучался исходящий трафик от сканируемого сервера. При проведении сканирования UDP портов, закрытые порты будут отправлять пакеты ICMP типа 3.3. Отличительной особенностью сканирования TCP портов являются отклики в виде пакетов TCP RST. Число подобных пакетов откликов считается отдельно для каждого IP адреса. Если число пакетов любого из этих типов достигнет трех, то трафик с данного IP адреса блокируется на 5 минут.

Данные квалификационные признаки были положены в основу алгоритмов для специализированных программных комплексов. На первом этапе были разработаны два

различных программных продукта. Один э этих продуктов реализован в виде Linux утилиты, другой представляет собой SDN модуль. При этом Linux утилита не требовательна к вычислительным ресурсам, но вполне справляется с защитой отдельно информационного сервера или пользовательской машины.

Для того, чтобы понять эффективность защиты с помощью разработанных нами программных комплексов была проведена серия тестов. Для проведения этих тестов был создан специальный полигон, на котором защищаемый сервер сканировался при помощи утилиты nmap. Созданное программное обеспечение пыталось исказить результаты атаки. В целом искажение сведений о состоянии портов было достаточно сильным. Точность определения открытых портов TCP не превысила 15%, а для остальных портов (закрытых TCP, UDP любого типа) не достигла 2%.

Однако, мы полагаем, что спектр действия наших программных комплексов не ограничен только сканированием портов. Эти комплексы могут быть применены против различных типов атак, например, DDoS атак по типу TCP и UDP флуда, ибо квалификационные признаки этих атак совпадают со сканированием портов. Этому будут посвящены дальнейшие исследования и эксперименты.

Еще одной областью применения наших усилий станет разработка отдельного программного комплекса для смартфонов, прежде всего под управлением ОС Android. Исходные коды данной утилиты во многом совпадут с кодами уже разработанной утилиты под Linux. Трудность состоит в том, что необходимо найти простую и эффективную замену межсетевому экрану iptables для блокировки трафика с фиксированных IP адресов.

7. Благодарности

Работа выполнена при поддержке гранта РФФИ № 16-07-00218А.

8.Литература

[1] Марков, А.С. Систематика уязвимостей и дефектов безопасности программных ресурсов / А.С. Марков, А.А. Фадин // Защита информации. Инсайд. – 2013. – Т. 3. – С. 56-61.

[2] Sridharan, A. Connectionless port scan detection on the backbone / A. Sridharan, T. Ye, S. Bhattacharyya // IEEE International Performance Computing and Communications Conference. – 2006. – Vol. 10. – P. 576.

[3] Gadge, J. Port scan detection / J. Gadge, A.A. Patil // 16th IEEE International Conference on Networks, 2008. – P. 1-6.

[4] Javaid, A. A deep learning approach for network intrusion detection system // Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS) –Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, 2016. – P. 21-22.

[5] Sharafaldin, I. Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization / I. Sharafaldin, A.H. Lashkari, A.A. Ghorbani // ICISSP, 2018. – P. 108-116.

[6] Morin, B. Correlation of intrusion symptoms: an application of chronicles / B. Morin, H. Debar // International Workshop on Recent Advances in Intrusion Detection – Springer, Berlin, Heidelberg, 2003. – P. 94-112.

[7] Yan, Q. Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges // IEEE Communications Surveys & Tutorials. – 2015. – Vol. 18(1). – P. 602-622.

Development of methods and means of counteracting the initial stage of network intrusion

E.S. Sagatov[1], S. Mayhoub[1], A.M. Sukhov[1], M.A. Baymyashkin[1]

[1]Samara National Research University, Moskovskoe Shosse 34A, Samara, Russia, 443086

Abstract. This article is devoted to the development of methods to counteract the initial stage of a network attack. Analysis of outgoing traffic allowed us to formulate the qualification features of scanning TCP and UDP ports. The appearance of ICMP 3.3 and TCP RST response packets indicates the beginning of the scanning process. Destination IP addresses from these packets are blocked for 5 minutes. The developed software systems in the form of an SDN module and Linux utilities are tested and have shown high efficiency.