єдиному математичному апараті рекурентних V_k –послідовностей. Це дозволило не лише підвищити рівень безпеки системи тестування, але й підвищити швидкість виконання в ній криптографічних перетворень.

Список використаної літератури: ***1.*** *Кабанова Т. А., Новиков В. А. Тестирование в современном образовании. Учебное пособие. – М.: Высшая школа, 2010. – 384 с.* ***2.*** *Винник В. К. Обзор дистанционных электронных платформ обучения // Научный поиск. – № 2.5, 2013. – С. 5–7.* ***3.*** *Алексеев А. Н. Дистанционное обучение инженерным специальностям: Монография. – Суммы: ИТД «Универсальная книга», 2005. – 333 с.* ***4.*** *Агапонов С. В., Джалиашвили З. О., Кречман Д. Л. и др. Средства дистанционного обучения. Методика, технология, инструментарий: ред. З. О. Джалиашвили. – СПб.: БХВ-Петербург, 2003. – 336 с.* ***5.*** *Ибрагимов, И. М. Информационные технологии и средства дистанционного обучения: Учебное пособие для студ. вузов. – М. : Академия, 2005. – 336 с.* ***6.*** *Воронов М. В. Дистанционные образовательные технологии и перераспределение функций профессорско-преподавательского состава // Социология образования. – 2009. – № 5. – С. 40-51.* ***7.*** *Батешов Е. А. Основы технологизации компьютерного тестирования. Учебное пособие. – Астана: ТОО «Полиграф-мир», 2011. – 241 с.* ***8.*** *Калюжный, А. С. Компьютерное тестирование как способ контроля знаний студентов // Высшее образование сегодня. – 2009. – № 7. – С. 67–68.* ***9.*** *Брэдфорд Эд.- Може Лу. Кроссплатформенные приложения для Linux и Windows. – СПб: Питер, 2003. – 672 с.* ***10.*** *Уланов А. В. Многоагентное моделирование механизмов защиты от атак "распределенный отказ в обслуживании": дис. ... канд. техн. наук : 05.13.18, 05.13.19 / Уланов А. В. – Санкт-Петербург, 2007. – 165 с.* ***11.*** *Ястребов И. С. Математические модели и реализация контроля доступа на основе ролей и контекста для распределенной системы управления физическим экспериментом: дис. ... канд. техн. наук : 05.13.18 / Ястребов И. С. – Ульяновск. 2010. – 139 с.* ***12.*** *Нортроп Тони, Макин Дж. Microsoft. Учебный курс 70-642. – М: Русская Редакция (Microsoft Press), 2008. – 570 с.* ***13.*** *Яремчук Ю.Є. Метод шифрування інформації з відкритим ключем на основі рекурентних послідовностей // Інформаційна безпека. – №3, 2013. – С. 123–129.* ***14.*** *Яремчук Ю. Є. Метод відкритого розподілу секретних ключів на основі рекурентних послідовностей // Інформаційна безпека. – №2, 2013. – С. 177–183.* ***15.*** *Яремчук Ю. Є. Метод автентифікації суб'єктів (об'єктів) взаємодії на основі рекурентних послідовностей // Вісник Вінницького політехнічного інституту. – №3, 2013. – С. 99–104.* ***16.*** *Яремчук Ю. Є. Можливість цифрового підписування на основі рекурентних послідовностей // Інформатика та математичні методи в моделюванні. – Том 3, №1, 2013. – С. 13–21.* ***17.*** *Яремчук Ю. Є. Можливість формування та перевірки цифрового підпису на основі рекурентних послідовностей // Вісник Вінницького політехнічного інституту. – №5, 2013. – С. 91–95.*

Сергей Егоров
Национальный авиационный университет

УДК 004.056:004.451.1(045)

ЗАЩИТА И МОНИТОРИНГ ПОРТОВ ВВОДА ВЫВОДА В ОПЕРАЦИОННЫХ СИСТЕМАХ LINUX

Аннотация: **Разработаны рекомендации относительно анализа состояния портов операционной системы Linux и обеспечения их защиты.**
Summary: **Recommendations regarding the analysis of the status of ports of the Linux operating system and ensure their protection.**
Ключевые слова: **Операционные системы, порты ввода-вывода, безопасность, информационная безопасность, Linux, Unix администрирование Linux.**

I Постановка задачи

Службы – это основная составляющая сетевой рабочей среды компьютеров. Если бы не было служб, то в компьютере можно было бы запускать не более одной программы, к которой можно было бы подключаться клиентам по одному за раз. В основе работы служб лежит концепция портов. Порт – это специальная добавка к IP-адресу, которая позволяет определить серверу, как именно работать с клиентской частью. Большинство пользователей очень плохо знакомы с портами, не говоря уже о механизме их функционирования. Это связано с тем, что для работы с сетью необходимо знать только IP-адрес, а номер порта знать не обязательно, потому что этот номер запрограммирован внутри программы.

Однако знание общих правил работы служб и портов ведёт к пониманию основных концепций безопасности. Целью данной статьи является разработка рекомендаций относительно анализа состояния портов операционной системы Linux и обеспечения их защиты. В качестве примера использовалась операционная система Linux Ubuntu 13.10 (64 bit), которая работала как гостевая операционная система под управлением Windows 7 (64 bit). В качестве менеджера виртуальных машин использовался Oracle VM VirtualBox® версии 4.3.2. Между гостевой и основной операционной системой была настроена виртуальная сеть.

II Анализ публикаций и исследований

Проблема анализа состояния портов операционных систем в существующей литературе не раскрыта или раскрыта очень плохо. Например, в [1, 2] анализ состояния портов отсутствует. Анализировать состояние портов должен уметь каждый специалист, который настраивает и сопровождает операционную систему. Ведь без такого анализа будет невозможно корректно настроить обслуживать фаервол, который предназначен для отражения многих типов хакерских атак. Производители соответствующего программного обеспечения (например, [3]) методик анализа состояния портов тоже не приводят.

III Службы

Для того чтобы соединиться с удалённым сервером, нужно знать его IP-адрес и номер порта. Через этот порт и будет установлено соединение. Эти сведения обычно указываются в формате IP-адрес:порт. Каждая серверная программа настроена на использование определённого порта. Клиент, в свою очередь, знает номер этого порта т. к. этот номер в нём запрограммирован. Этот факт и есть та самая причина, по которой пользователь порой даже не подозревает о существовании такой формы жизни как порт. Итак, клиент пытается подключиться к удалённому серверу с использованием порта. Если сервер поддерживает подключение через этот порт, начинается начальная фаза взаимодействия. Устанавливается соединение, после чего сервер может потребовать аутентификацию (без установки соединения аутентификация невозможна). Если аутентификация невозможна, то сервер может разорвать соединение.

Благодаря механизму портов вы можете запускать на одном компьютере несколько серверов (служб). Например, на одном компьютере можно запустить одновременно FTP, telnet, HTTP и т. п.

Упрощённо, процедура установки соединения с использованием протокола TCP выглядит следующим образом.

1. Клиент вступает в контакт с сервером (серверным процессом), посылая запрос на соединение (пакет с установленным битом SYN)
2. Сервер, который находится в очереди ожидания, посылает в ответ клиенту пакет с установленными битами SYN и ACK. Сервер по-прежнему находится в очереди ожидания, т. к. он ожидает ответ от вступившего с ним в контакт клиента и не принимает никаких попыток соединения от других клиентов.
3. Клиент в ответ на пакет, полученный от сервера (серверного процесса), посылает серверу (серверному процессу) пакет с установленным битом ACK. Сервер (серверный процесс) после получения этого пакета покидает очередь ожидания, освобождая её другому процессу для ожидания соединений от других клиентов. С этого момента соединение считается установленным и сервер (серверный процесс) начинает обмен данными с клиентом, и никаким другим.

В файле /etc/services перечислены некоторые общеизвестные службы (Well Known Service, WKS).

Вы можете добавлять в этот файл новые службы и ставить им в соответствие уникальный порт, незанятый другими службами. Однако не следует добавлять в этот файл общеизвестные службы, которые там уже есть. Например, если вы добавите в конец файла services службу www с портом 8080, то такая запись будет проигнорирована. Это связано с тем, что Linux просматривает файл services в направлении сверху вниз, и выбирает первую попавшуюся подходящую запись. А верхней части списка этого файла службе www поставлен в соответствие порт 80. Будет выбрана эта запись. Изменение порта по умолчанию для любой службы WKS крайне не желательно, т.к. это может повлечь за собой неправильную работу клиента.

Удаление WKS из файла /etc/services – тоже плохая идея. Удаление записи WKS, изменение её имени или порта может повлечь за собой неправильную работы соответствующей службы. Во-первых, благодаря этому файлу некоторые утилиты, например netstat, могут преобразовывать номера портов в символьные имена, а во-вторых, некоторые службы могут работать не правильно в случае отсутствия соответствующей записи в файле /etc/services.

В отношении портов следует учитывать одно очень важное обстоятельство: все порты с номерами ниже 1024 рассматриваются операционной системой Linux как привилегированные. Связать имя службы с привилегированным портом может только учётная запись root (UID0). С номерами выше 1024 – кто угодно.

Общеизвестные службы не обязательно могут использовать привилегированные порты. Такие службы используют эти порты по всеобщему соглашению, а не в жестких рамках RFC.

Все порты лежат в диапазоне от 0 до 65535. Это связано с тем, что для хранения номера порта используется переменная размером 16 бит. Выбирать значение порта, которое больше этого диапазона не следует. Дело в том, что для портов существует такая же проблема «зацикливания», как и для UID. Например, если попробовать связать имя службы с портом 65800, то в переменную будет записано 265 (65800 минус 65535). Это привилегированный порт. Для проведения этого трюка необходима модификация исполнительного кода системы. Однако этот старый трюк иногда срабатывает.

Некоторым пользователям может показаться, что диапазон от 0 до 65535 слишком большой. Однако это не соответствует действительности из-за особенностей установления соединения. Рассмотрим ещё раз вкратце установление соединения TCP. Сервер получает от клиента запрос на соединение через порт по умолчанию и отправляет клиенту случайно выбранный порт из диапазона выше 1024, через который и будет передаваться информация клиенту. Т. е. фактически используется два порта. Например, вы имеете дело с 20 одновременными соединениями через порты со службой www, то фактически используется 21 порт: один привилегированный с номером 80 (порт по умолчанию для этой службы) и 20 портов выбранных случайным образом. Именно методом случайного выбора портов и порядковой нумерации пакетов операционная система получает возможность контролировать одновременно все соединения в режиме реального времени. Если вы видите в этом уязвимое место, значит вы начинаете думать как взломщик – именно так, ставя себя на их место, и следует бороться со злоумышленниками.

Идентифицировать IP-адреса и номера портов – довольно простая задача. А вот вставить себя в этот информационный поток не так-то просто. Для большинства систем Linux это чрезвычайно сложная задача. Для этого нужно угадать номер последовательности. Такие атаки носят название man in the middle (человек посередине). Однако некоторые системы используют для генерации начальной последовательности функцию времени. В таком случае задача несколько упрощается. Несмотря на это, для реализации такой атаки, нужна достаточно высокая квалификация.

IV Анализ состояния сетевых подключений

Лучшая утилита для анализа состояния сетевых подключений в узле – это netstat. Если её запустить без параметров, то на терминале будет отображено состояние всех подключений, а номера портов и IP-адреса будут преобразованы в символьные имена (рис. 1). Для этого используются стандартные процедуры разрешения имён и файл /etc/services.

Первая строка вывода утилиты netstat всегда указывает на то, какого рода информация отображается. В данном случае строка Active Internet connections (w/o servers) (активные соединения с Интернет (без серверов)), говорит о том, что программа (сервер) открыла порт и ожидает запроса на соединение. Другими словами, этот сервер стоит в очереди ожидания (wait queue listening). Сервер при этом не ведёт никакого обмена данными, он просто стоит в очереди и ждёт запроса на соединение. Вывод первой строки будет зависеть от передаваемых параметров netstat через командную строку. Таким образом, в данном случае отображается информация только о существующих соединениях.

Вторая строка содержит заголовки для первой части вывода netstat. Proto обозначает протокол (Protocol). В этой колонке, как правило есть только tcp или udp, но согласно файлу /etc/services могут быть и другие, например, raw. Колонки Recv-Q и Send-Q обозначают Receive Queue (очередь на прием) и Send Queue (очередь на передачу) соответственно. Здесь указывается количество принятых байтов из сети, но не скопированных пользовательской программой, и количество пакетов, приём которых не был подтверждён удалённой системой. Это происходит потому, что когда сокет TCP закрывается, то завершающий пакет ACK не может быть подтверждён.

Следующие две колонки – это Local Address (локальный адрес) и Foreign Address (удалённый адрес). Адреса указываются в форме адрес:порт. Содержимое третьей строки в этих колонках говорит о том, что хост stillphelix-Virtu подключен к хосту 192.168.0.1 через службу netbios-ssn (согласно файлу /etc/services произошло подключение к сетевому сервису netbios-ssn на удалённом хосте посредством порта 139) по протоколу TCP (первая колонка). Этот привилегированный порт предназначен для передачи данных сервису. Этот порт задан жёстко и его нельзя изменить. Возврат данных от сервиса на хост stillphelix-Virtu происходит через случайно выбранный порт 48750. Порт возврата данных не задан жестко. Это может быть любой незанятый порт с номером выше 1024.

Во время первого соединения клиент сообщает серверу номер порта и ждёт прихода на этот порт от сервера ответного SYN_ACK. Если сервер занят и не может ответить, то он посылает клиенту в ответ сообщение ICMP Type 3 Code 3 Port Unreachable (порт недоступен). После такого ответа клиент снова случайным образом выбирает порт, посылает его серверу. Если клиент на этот порт получает пакет TCP от

выбранного сервера с установленными битами SYN и ACK, то этот клиент посылает в ответ серверу пакет TCP с установленным битом ACK. После этого соединение считается установленным и начинается передача данных.

Active Internet connections (w/o servers)

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	0	stillphelix-Virtu:48750	192.168.0.1:netbios-ssn	ESTABLISHED
tcp	0	0	stillphelix-Virtu:48749	192.168.0.1:netbios-ssn	ESTABLISHED
tcp	0	0	stillphelix-Virtu:35217	192.168.0.:microsoft-ds	ESTABLISHED
tcp	0	0	stillphelix-Virtu:48751	192.168.0.1:netbios-ssn	ESTABLISHED

Active UNIX domain sockets (w/o servers)

Proto	RefCnt	Flags	Type	State	I-Node	Path
unix	14	[ ]	DGRAM		6968	/dev/log
unix	3	[ ]	STREAM	CONNECTED	11670	
unix	3	[ ]	STREAM	CONNECTED	11669	
unix	3	[ ]	STREAM	CONNECTED	11668	@/tmp/dbus-htSLArbo2L
unix	3	[ ]	STREAM	CONNECTED	11667	
unix	3	[ ]	STREAM	CONNECTED	11666	@/tmp/.ICE-unix/1055
unix	3	[ ]	STREAM	CONNECTED	11665	
unix	3	[ ]	STREAM	CONNECTED	11664	@/tmp/dbus-htSLArbo2L
unix	3	[ ]	STREAM	CONNECTED	11663	
unix	3	[ ]	STREAM	CONNECTED	11662	@/tmp/.X11-unix/X0
unix	3	[ ]	STREAM	CONNECTED	11661	
unix	3	[ ]	STREAM	CONNECTED	11660	@/tmp/dbus-aK7zdcgTeY
unix	3	[ ]	STREAM	CONNECTED	11659	
unix	3	[ ]	STREAM	CONNECTED	11613	@/tmp/dbus-htSLArbo2L
unix	3	[ ]	STREAM	CONNECTED	11612	
unix	3	[ ]	STREAM	CONNECTED	11584	@/dbus-vfs-daemon/socket-mDtcqFsW
unix	3	[ ]	STREAM	CONNECTED	11583	
unix	3	[ ]	STREAM	CONNECTED	11581	@/dbus-vfs-daemon/socket-t13sxC8n
unix	3	[ ]	STREAM	CONNECTED	11580	
unix	3	[ ]	STREAM	CONNECTED	11558	@/tmp/dbus-htSLArbo2L
unix	3	[ ]	STREAM	CONNECTED	11557	
unix	3	[ ]	STREAM	CONNECTED	11539	@/dbus-vfs-daemon/socket-0TBPeSQ0
unix	3	[ ]	STREAM	CONNECTED	11538	
unix	3	[ ]	STREAM	CONNECTED	11536	@/dbus-vfs-daemon/socket-B7IO8Wi1

Рисунок 1 – Сокращённый вывод утилиты netstat без параметров

С хоста stillphelix-Virtu был получен доступ к сетевым ресурсам удалённого хоста с IP 192.168.0.1. В связи с тем, что на удалённом хосте имеется несколько таких ресурсов, система вынуждена открыть несколько соединений. Каждое соединение устанавливается с использованием выше рассмотренных процедур. В результате выполнения этих процедур открывается новый коммуникационный порт. Разрыв соединения происходит путём прохождения похожих процедур. И в последней колонке State, появляется метка CLOSING (соединение закрывается). Однако вместо этого в этой колонке может быть метка LAST_ACK (последнее подтверждение). Когда клиент серверу посылает пакет TCP с установленным битом LAST_ACK, а сервер его получает, то этот сервер срезу же разрывает соединение. Поскольку соединение уже не существует, то сервер на последний пакет ответить не может.

В последней колонке указывается состояние соединения. В ней могут быть такие метки [4].

1. ESTABLISHED – состояние открытого соединения, принимаемые данные представить пользователю. Это нормальное состояние соединения в фазе передачи данных.
2. SYN-SENT – ожидание парного запроса на установление соединения. С нашей стороны запрос уже сделан.
3. SYN_RECV – ожидание подтверждения после того, как запрос соединения уже принят и отправлен.
4. FIN-WAIT-1 – ожидание запроса от чужой программы TCP, или подтверждения ранее отправленного запроса на закрытие соединения.
5. FIN-WAIT-2 – ожидание запроса на закрытие соединения со стороны чужой программы TCP.

6. TIME-WAIT – ожидание, когда истечет достаточное количество времени и можно быть уверенным, что чужая программа TCP получила подтверждение своего запроса на закрытие соединения.
7. CLOSED – состояние полного отсутствия соединения.
8. CLOSE-WAIT – ожидание запроса на закрытие соединения со стороны своего клиента.
9. LAST-ACK – ожидание запроса на закрытие соединения, ранее отправленного чужой программе TCP (запрос включал также подтверждение получения чужого запроса на закрытие соединения).
10. LISTEN – ожидание запроса на соединение со стороны чужих портов и программ TCP.
11. CLOSING – ожидание подтверждения со стороны чужой программы TCP запроса о закрытии соединения.
12. UNKNOWN – не определено (в [4] этот пункт отсутствует).

Следует отметить, что состояние CLOSED является фиктивным, поскольку оно представляет состояние, когда не существует блока TCP, а потому и нет соединения. Некоторые из этих меток могут появляться только на стороне сервера, другие – клиента. Есть и те, которые появляются на стороне сервера и клиента.

Если в командной строке netstat указать ключ -e, то в таблице появится дополнительная колонка User (пользователь). В ней будет указано UID пользователя, от имени которого было открыто соединение.

Следующая часть вывода netstat начинается со строки Active UNIX domain sockets (w/o servers) (Активные доменные сокеты UNIX (без серверов)). Сокеты UNIX бывают двух типов: локальные потрты и сетевые. Локальному сокету присваивается UNIX-адрес и создаётся специальный файл (файл сокета) по заданному пути. Через этот файл могут сообщаться любые локальные процессы путём простого чтения или записи из него. Сокеты представляют собой виртуальный объект, который существует, пока на него ссылается хотя бы один из процессов. При использовании сетевого сокета создается абстрактный объект, который привязан к слушающему порту операционной системы и сетевому интерфейсу, т. е. ему присваивается INET-адрес, который имеет адрес интерфейса и слушающего порта. Рассмотрим более подробно заголовки колонок.

Proto – обозначает протокол. Для сокетов в стиле UNIX единственным допустимым значением здесь может быть только unix.

RefCn – счётчик ссылок. Обозначает количество процессов, подсоединённых к сокету.

Flags – флаги. Обычно это поле остаётся пустым. Однако если соответствующий процесс, который подключен к сокету, ожидает приём данных, то это поле примет значение ACC (то есть SO_ ACCEPTION - принятие), это означает, что сокет готов к приему запроса на соединение. В некоторых ситуациях могут возникать и другие флаги, например N (SO_NOSPACE - нет места) или W (SO_ WAITDATA – ожидание данных).

Type - тип сокета. Как правило, здесь стоит метка STREAM (сокет с созданием соединения), однако допускаются следующие метки: RDM (Reliably Delivered Message – надежно передаваемое сообщение), DGRAM (Datagram – сокет без создания соединения), SEQPACKET (Sequential Packet – последовательно передаваемый пакет), RAW (сокет передачи данных без транспортного протокола), PACKET (пакет простого доступа к интерфейсу) или UNKNOWN (неизвестный тип сокета для будущих усовершенствований).

State – состояние сокета. Здесь могут использоваться следующие метки: FREE (сокет свободен), LISTENING (ожидание поступления запроса), CONNECTING (соединение устанавливается), CONNECTED (соединение установлено), DISCONNECTING (соединение разрывается) или UNKNOWN (неизвестное состояние). Данное поле может быть также вообще пустым. При нормальном функционировании системы сокет не может находиться в состоянии UNKNOWN.

I-Node – не представляет интереса. Здесь отображается номер дескриптора I-Node, соответствующий соединению. Но этот I-Node существует в каталоге /proc только в том случае, если соединение используется, поэтому поиск этого I-Node не приведет к желаемым результатам.

Path - отображает процесс, подключенный к данному сокету.

В листинге 1 не показано одно из наиболее распространённых состояний сокета: LISTENING (слушание). Это означает, что фактически соединения нет, и процесс стоит в очереди и ждёт запроса на соединение. Для вывода на терминал этих процессов в командную строку netstat необходимо добавить ключ -a (рис. 2).

Заголовки двух основных разделов тоже поменялись на Active Internet connections (servers and established) (активные соединения с Интернет (серверы и установленные соединения)) и Active UNIX domain sockets (servers and established) (активные доменные сокеты UNIX (серверы и установленные соединения)). Вывод тоже существенно увеличился из-за того, что в выводе присутствуют сервера, стоящие в очереди ожидания.

Проанализировав данные, полученные с помощью выполнения командной строки netstat -a -e, сразу становиться видно, через какие порты и процессы ведётся прослушивание сети с целью установки соединения. Каждый порт, для которого нет соответствующей записи в файле /etc/services, будет связан с процессом, который ожидает поступления данных через этот порт. Каждый такой процесс будет иметь соответствующую запись в таблице процессов. Связывать службу с необходимым ей портом с помощью

файла /etc/services вовсе не обязательно. Но если вы это сделаете, то сможете избежать борьбы за определённый порт между службами. Ни одна служба не будет использовать порт, который описан в файле /etc/services, кроме той, которая с ним связана.

Active Internet connections (servers and established)

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State	User	Inode
tcp	0	0	localhost:ipp	*:*	LISTEN	root	7075
tcp	0	0	stillphelix-Virtu:35975	192.168.0.:microsoft-ds	ESTABLISHED	stillphelix	15633
tcp	0	0	stillphelix-Virtu:55504	192.168.0.1:netbios-ssn	ESTABLISHED	stillphelix	15495
tcp	0	0	stillphelix-Virtu:55505	192.168.0.1:netbios-ssn	ESTABLISHED	stillphelix15562	
tcp	0	0	stillphelix-Virtu:55506	192.168.0.1:netbios-ssn	ESTABLISHED	stillphelix15578	
udp	0	0	*:33159	*:*		avahi	7030
udp	0	0	*:mdns	*:*		avahi	7028
udp6	0	0	[::]:49883	[::]:*		avahi	7031
udp6	0	0	[::]:mdns	[::]:*		avahi	7029

Active UNIX domain sockets (servers and established)

Proto	RefCnt	Flags	Type	State	I-Node	Path
unix	2	[ACC]	STREAM	LISTENING	15531	@/dbus-vfs-daemon/socket-LUM0ToC0
unix	2	[ACC]	STREAM	LISTENING	14200	@/dbus-vfs-daemon/socket-Hvfm4thv
unix	2	[ACC]	STREAM	LISTENING	6913	/var/run/dbus/system_bus_socket
unix	2	[ACC]	STREAM	LISTENING	15522	@/dbus-vfs-daemon/socket-YhxkHg5O
unix	2	[ACC]	STREAM	LISTENING	15528	@/dbus-vfs-daemon/socket-iv4ORZKs
unix	2	[ACC]	STREAM	LISTENING	13602	/run/user/stillphelix/keyring-yemEYX/gpg
unix	2	[ACC]	STREAM	LISTENING	13604	/run/user/stillphelix/keyring-yemEYX/pkcs11
unix	2	[ACC]	STREAM	LISTENING	11076	@/dbus-vfs-daemon/socket-7xjGwCNo
unix	2	[ACC]	STREAM	LISTENING	15582	@/dbus-vfs-daemon/socket-B7GR0scW
unix	2	[ACC]	STREAM	LISTENING	6569	@/com/ubuntu/upstart
unix	2	[ACC]	STREAM	LISTENING	13428	@/tmp/dbus-cK7dCu3Puy
unix	14	[ ]	DGRAM		6978	/dev/log
unix	2	[ACC]	STREAM	LISTENING	15497	@/dbus-vfs-daemon/socket-XPoQJlrK
unix	2	[ACC]	STREAM	LISTENING	7036	@/org/bluez/audio
unix	2	[ACC]	STREAM	LISTENING	14037	/tmp/pulse-FsFCiI76z4ga/native
unix	2	[ACC]	STREAM	LISTENING	12222	@/tmp/.X11-unix/X0
unix	2	[ACC]	STREAM	LISTENING	7016	/var/run/avahi-daemon/socket
unix	2	[ACC]	STREAM	LISTENING	7024	/var/run/sdp
unix	2	[ACC]	STREAM	LISTENING	7797	/var/run/acpid.socket
unix	2	[ACC]	STREAM	LISTENING	14149	@/dbus-vfs-daemon/socket-tyxTf75s
unix	2	[ACC]	STREAM	LISTENING	15579	@/dbus-vfs-daemon/socket-wykCetpp
unix	2	[ACC]	STREAM	LISTENING	12223	/tmp/.X11-unix/X0

Рисунок 2 – Сокращённый вывод утилиты netstat с параметрами -a и -e

Во время анализа следует обратить внимание на порты, номера которых не преобразованы в символьные имена. В листинге 2 это порты с номерами 33159 и 49883. Поле User показывает, что соединение открыто от имени пользователя avahi. Avahi – инфраструктура для обнаружения сервисов посредством многоадресной рассылки (Multicast DNS Service Discovery). Она позволяет программам предоставлять и обнаруживать сервисы и хосты в локальной сети, не требуя ввода никаких специальных настроек. Например, можно сразу после подключения к сети найти принтеры, файлы и людей для общения. Avahi – открытая и свободная реализация протокола zeroconf, (который в свою очередь разработан Apple) и призван решать следующие проблемы: выбор сетевого адреса для устройства, нахождение компьютеров по имени, обнаружение сервисов, например, принтеров. Порт с номером 35975 был выбран случайным образом во время установления соединения со службой microsoft-ds, а порты с номерами 55504, 55505, 55506 выбирались случайным образом во время установления соединений со службой netbios-ssn.

Если порт закрыт, т. е. ни одна служба не ожидает запроса на соединение через этот порт, то злоумышленник не сможет проникнуть в вашу систему через этот порт. Если вы остановите все службы, то это будет означать то, что злоумышленник не сможет проникнуть в вашу систему, и вынужден будет

оставить вас в покое. Но это означает так же и то, что ни вы, ни кто-либо ещё с вашим хостом не сможет установить соединение, т. к. ни один серверный процесс не будет стоять в очереди ожидания.

V Сетевые атаки и защита

Итак, взлом системы происходит путём проникновения в вашу систему через сетевые службы. Правда тут есть одно исключение: социальная инженерия. В этом случае информация, которая необходима для проникновения в систему, узнается путём налаживания непосредственных контактов с людьми. При этом учитывается психология людей, особенности их поведения. Например, во многих случаях задача, которая состоит в том, чтобы узнать пользовательский пароль для того, чтобы проникнуть в корпоративную сеть решается очень просто: некоторые люди эти пароли записывают на бумажку и затем приклеивают её на монитор. Или если повезёт, то некоторые системные администраторы такие пароли новым зарегистрированным пользователям объявляют громко вслух. Достаточно просто пройтись по офису. Изучение вопросов социальной инженерии выходит за рамки этой статьи.

Многие взломщики, которые орудуют на просторах Интернета – обычные чайники, с достаточно низкой квалификацией и с большими претензиями на то, что они крутые хакеры, они просто хотят попробовать свои силы: получится у них или нет. Но вы должны понимать, что это только часть Интернета. Среди взломщиков есть специалисты с очень высокой квалификацией. Они прилагают максимум усилий ля того, чтобы замести за собой следы: их зачастую конкретно ваша система не интересует, они ставят перед собой более высокие цели. Вашу систему они могут использовать в качестве плацдарма для атаки на интересующую их систему, для того, чтобы сложилось впечатление, что атаку провёл кто угодно, но только не они. Если вы не можете или не хотите вручную отслеживать состояние системы, или с помощью автоматизированных сценариев, которые отсылают необходимую вам информацию на ваш электронный адрес, то остановите все сетевые службы или закройте к ним доступ всем, кроме тех объектов, которым вы доверяете. Но даже в этом случае нет никаких гарантий полной безопасности.

Сеть нормально может функционировать только в том случае, если существует доверие на определенном уровне. Доверие может быть предоставлено пользователю и/или системе. Системе доверяет root. Вы доверяете своим исполняемым файлам. Именно поэтому взломщики пытаются заменить наиболее важные ключевые файлы, которым вы доверяете. Для того, чтобы скрыть абсолютно все следы пребывания в системе необходимы полномочия root. Если взломщику нужно сохранять контроль над системой в течение продолжительного времени, то ему надо отредактировать файлы /etc/passwd и /etc/shadow, а также установить один или несколько исполняемых файлов («комплект root» (root kit)).

Вы должны сделать всё зависящее от вас для защиты системы, затем нужно наладить и тщательно проверить систему аудита. Чем сложнее найти следы пребывания в системе, тем больше шансов на то, что взлом будет обнаружен. Несколько следов лучше, чем один.

Прежде чем будет совершена попытка взлома системы, вы можете получать некоторые симптомы (но это не всегда так). Для начала взломщик должен тщательно изучить вашу систему. Есть много разных способов сделать это, но самый распространённый – это сканирование. При этом хакер получает сведения, наподобие тех, которые даёт исполнение команды netstat -a -e. В большинстве случаев взломщик может получить сведения о версии вашей операционной системы вплоть до нескольких разрядов. А версия программы даёт информацию об уязвимости. Для того чтобы узнать версию службы, достаточно с ней соединиться. Если такую информацию служба не предоставила во время установления соединения, значит нужно попробовать попросить её, чтобы она такую информацию предоставила.

Однако не всегда атаке предшествует сканирование. Если ваша система предоставляет в Интернет общеизвестные сетевые службы, например www, то можете поверить на слово: для этого никакое сканированию не нужно. Об этом будут знать все. В этой ситуации взломщик может попробовать реализовать атаку типа «скоростные условия» (race conditions). И если в системе есть соответствующая не закрытая дыра в безопасности, то атака пройдёт успешно. И даже если такая атака на вас не пройдёт, то этот взломщик, если он не будет вовремя остановлен и наказан провайдером (а это маловероятно), получит доступ к какой-либо системе. Со стороны эта атака будет выглядеть так, как будто она осуществляется, например, с каких-либо двух разных учебных заведений. Хосты, с которых атаковали, скорее всего, окажутся тоже взломанными.

Самый популярный и наиболее часто используемый метод – прослушивание сети. Взломщик получает доступ к системе, а затем просто «прослушивает» весь трафик, проходящий через сетевой интерфейс, и выделяет из него пары «логин - пароль».

Когда взломщик проникает в систему, он может воспользоваться приемами, основанными на переполнении буфера. Находясь в системе, взломщик может попытаться реализовать подобный прием с любыми бинарными файлами, которые выполняются с использованием SUID root. Однако существуют и

другие уязвимые места, например, те, которые связаны со скоростными условиями (race condition). Скоростные условия – это набор обстоятельств, которые пользователь может использовать для того, чтобы непривилегированное обращение к системе было выполнено на уровне привилегий root.

Также злоумышленник может воспользоваться одним из «защищенных» каталогов, таких как /tmp. В этом каталоге процесс, работающий на уровне root, может поместить файл с известным именем. Взломщик, который заранее знает это имя, может заблаговременно разместить там такой файл, и тогда процесс root будет использовать созданный злоумышленником файл вместо своего собственного.

VI Выводы

Отслеживание состояния портов операционной системы позволяет корректно настроить фаервол, а также в некоторых случаях и предотвратить хакерскую атаку или выявить канал утечки информации и принять адекватные меры. Для адекватной настройки и сопровождения операционной системы также следует знать и методы взлома операционных систем. Единственный способ адекватно настроить и сопровождать фаерфол - использовать методику отслеживания состояния портов.

Литература: ***1.*** *Немет, Эви, Снайдер, Гарт, Хейн, Трент, Уэйли, Бэн.Unix и Linux: руководство системного администратора, 4-е изд. : Пер.с англ. — М .: ООО "И Д Вильямс", 2012. — 1312 с.* ***2.*** *Бруй В. В., Карлов С. В. LINUX-сервер: пошаговые инструкции инсталляции и настройки. –М.: СИП РИА, 2003. – 572 с.* ***3.*** *Introduction to Comodo Internet Security [Electronic resource]: © Comodo Group, Inc. – Mode of access: http://help.comodo.com/topic-72-1-451-4685-Introduction-to-Comodo-Internet-Security.html. — Title from the screen.* ***4.*** *RFC 793 [электронный ресурс]. — Режим доступа: http://helpsite.narod.ru/rfc/793/26.htm. – Название с экрана.*