УДК 004.056

Тестирование на проникновение как анализ защищенности компьютерных систем

А.С. Плешков, Д.Д. Рудер

Алтайский государственный университет (Барнаул, Россия)

Penetration Testing as a Security Analysis of Computer Systems

A.S. Pleshkov, D.D. Ruder

Altai State University (Barnaul, Russia)

Тестирование на проникновение является одним из приоритетных направлений в информационной безопасности. Оно позволяет получить объективную оценку того, насколько легко осуществить несанкционированный доступ к компьютерной системе, а также взглянуть на саму систему с точки зрения злоумышленника, а именно – понять, каким способом можно скомпрометировать выбранную систему и какие вредоносные действия на ней можно совершить. Описаны способы преодоления злоумышленником установленных средств защиты и набор действий, которые он сможет совершить, получив несанкционированный доступ к компьютерной системе. Наиболее популярными действиями, которые осуществляются злоумышленниками на скомпрометированной системе, являются следующие: локальное повышение прав, кража конфиденциальной информации и данных учетных записей пользователей, проведение слежки за скомпрометированным пользователем с использованием установленного на системе-жертве оборудования, к которому относятся микрофон и web-камера. Для защиты пользователей от компрометации их рабочих станций описаны не только актуальные методы проникновения злоумышленников в системы, но и основные правила и рекомендации о том, как можно повысить защищенность своих компьютерных систем и уменьшить вероятность наступления негативных последствия от сетевых атак.

Ключевые слова: информационные технологии, информационная безопасность, защита информации, компьютерная система, несанкционированный доступ, сетевые атаки, вредоносные программы (вычислительная техника).

DOI 10.14258/izvasu(2015)1.1-31

Penetration testing is one of the priority areas in information security. It allows to perform an objective assessment of how easy it is to make unauthorized access to a computer system, as well as take a look at the system from the attacker's point of view, namely, to understand how he can compromise the selected system, and what malicious actions he can perform. The paper describes the ways of overcoming installed security tools by an attacker, and a set of actions that he can do while having unauthorized access to a computer system. The most popular actions that are performed by attackers on a compromised system are: local elevation of privileges, theft of confidential information and data, user accounts, implementation of shadowing over the compromised user using the equipment installed on the victim's system, which includes a microphone and web camera. To protect users from compromising their workstation, the article describes not only the actual methods of penetration by attackers into systems, but also the basic rules and recommendations on how users can increase security of their computer systems and reduce the probability of negative consequences from network attacks.

Key words: information technologies, information security, protection of information, computer system, unauthorized access, network attacks, malicious application (computer engineering).

В настоящее время информационная безопасность является одной из важнейших компонент в любой организации, так как информация, обрабатываемая в их информационных системах, в большей или меньшей степени принадлежит к категориям коммерческой тайны и персональных данных. Разглашение указанной информации может привести не только к материальным потерям, но и к утрате репутации и имиджа компании,

что в конечном итоге в некоторых случаях может привести к ее полному краху. Поэтому для того, чтобы избежать указанных последствий, необходимо проводить анализ защищенности и надежности информационных систем обработки данных. Одним из самых действенных способов такого анализа является использование методов «тестирования на проникновение».

Под термином «тестирование на проникновение» подразумевается имитация действий реального злоумышленника по реализации несанкционированного проникновения в информационную систему [1].

Во время проведения тестирования на проникновение в качестве атакуемой системы была выбрана операционная система Windows 7 с установленным антивирусным программным обеспечением Eset Smart Security 4. Моделирование действия злоумышленника производилось в дистрибутиве Kali Linux с ядром версии 3.12-kali1-686-pae и установленным пакетом Metasploit Framework версии 4.8.2-2014030501.

Проникновение в систему с использованием атаки на стороне клиента

Вначале производится генерация полезной нагрузки – windows/meterpreter/reverse_tcp с помощью специального модуля – Veil Evasion [2], разработанного известным пен-тестером Кристофером Трунсером. Данный модуль является новой разработкой и активно используется специалистами по защите информации для осуществления тестирований на проникновение. Приложение помогает сгенерировать нагрузку таким образом, чтобы большинство или практически все антивирусные программные обеспечения не нашли в ней угрозу безопасности и не смогли предпринять действий по ее нейтрализации. Одним из достоинств модуля является поддержка работы со специализированным программным обеспечением Metasploit Framework. Интерфейс описанного приложения представлен на рисунке 1.

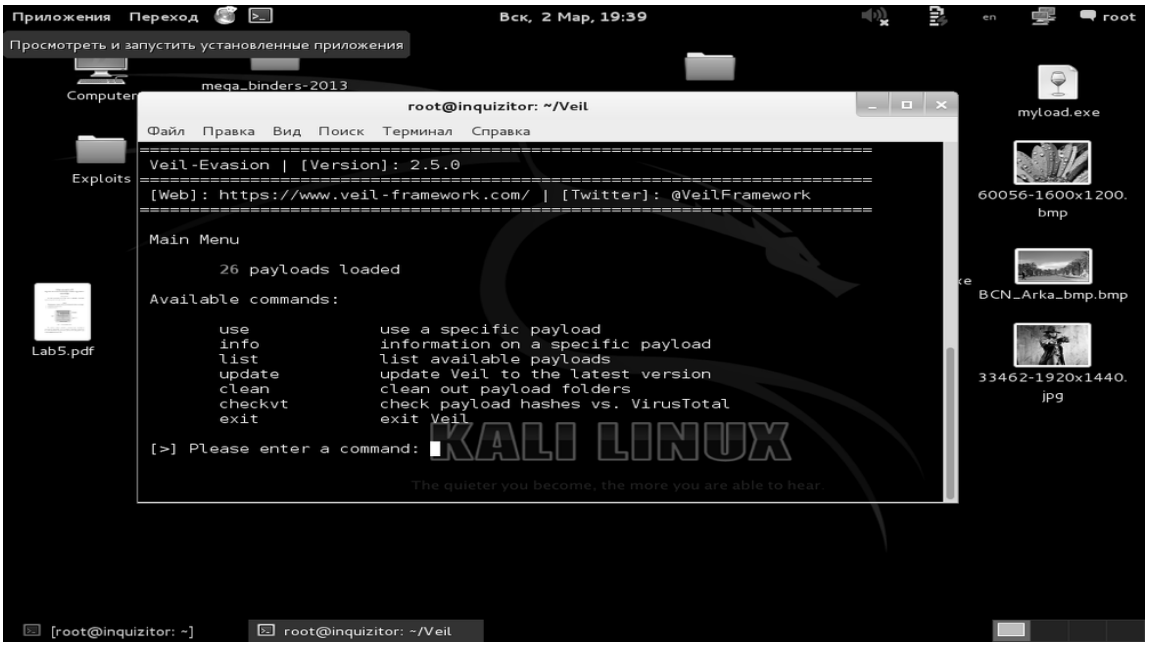

Рис. 1. Интерфейс модуля Veil-Evasion

На рисунке 1 — главное окно модуля Veil, где непосредственно сгенерирована нагрузка. Генерация полезной нагрузки начинается с выбора типа, который она будет иметь. Каждый тип нагрузки отличается от остальных и используется в конкретной, более подходящей ситуации. Для проведения тестирования на проникновение использована полезная нагрузка с типом python/meterpreter/rev_tcp. Особенностями данного типа нагрузки является то, что она предоставляет собой расширяемую многофункциональную командную оболочку, которая может быть динамически расширена во время работы, а инициирование соединения происходит не от рабочей станции злоумышленника, а от рабочей станции жертвы. Для того чтобы просмотреть весь список полезных нагрузок, можно воспользоваться командой list. Далее с помощью команды use необходимо задать выбранный тип нагрузки, после чего приложение попросит ввести основные параметры через команду set: LHOST и LPORT–ip-адрес и порт рабочей станции атакующего, через которые будет устанавливаться соединение. На финальной стадии генерирования нагрузки модуль попросит ввести имя файла. В итоге получается полностью готовая полезная нагрузка, которую можно использовать для атаки на стороне клиента.

После генерации нагрузки полученный файл отправляется жертве, для того чтобы она его запустила. Для этого можно воспользоваться различными социо-инженерными и фишинговыми методиками.

После отправки полезной нагрузки в Metasploit в интерфейсе msfconsole необходимо настроить «слушателя» – эксплойт, прослушивающий указанный порт системы атакующего на предмет входящих соединений от созданного shell-кода. В качестве эксплойта используется multi/handler. Как только жертва запустит данный файл, то в фоновом режиме у нее задействуются два процесса, соответствующие запущенной нагрузке.

В результате запуска приложения на системе жертвы у злоумышленника появляется активная сессия со скомпрометированной рабочей станцией, как показано на рисунке 2.

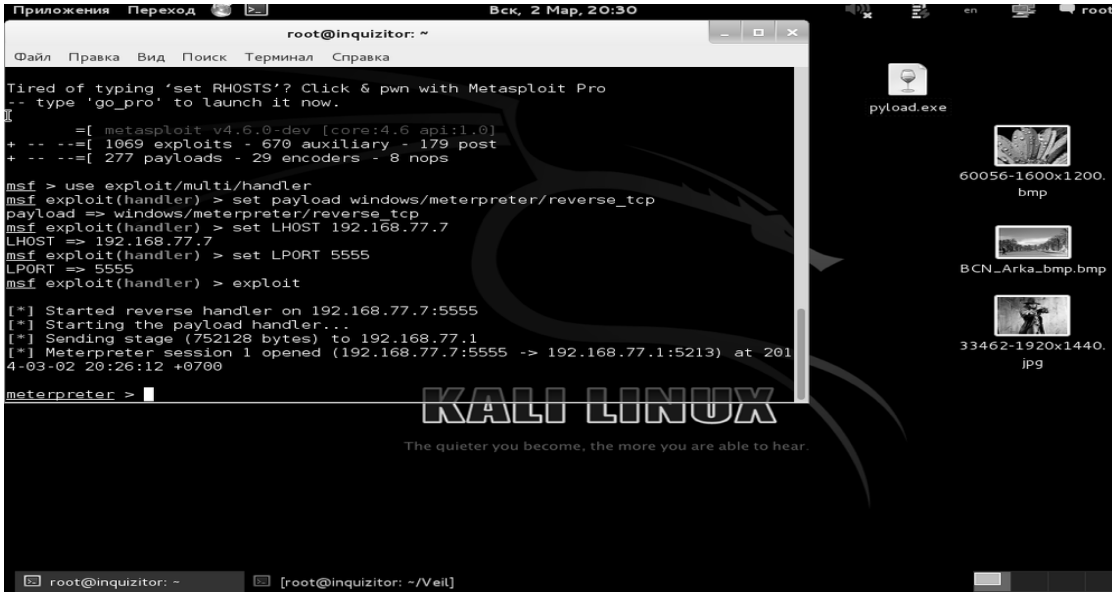

Рис. 2. Получение несанкционированного доступа к системе жертвы

Из рисунка 2 следует, что у злоумышленника появилась активная сессия и управление было передано интерпретатору meterpreter. Далее атакующий может приступить к этапу эксплуатации скомпрометированной системы.

Для того чтобы защитить компьютерные системы от компрометации данным способом, необходимо осуществлять мониторинг и исследование оперативной памяти системы и ее процессов на предмет наличия в ней динамически расширяемой полезной нагрузки meterpreter с помощью утилиты Antimeter2. Данная утилита написана на языке программирования python и имеет ряд опций, которые описаны в прилагаемом к утилите текстовом файле.

После запуска программы с необходимыми настройками начинается процесс сканирования памяти, который приведен на рисунке 3.

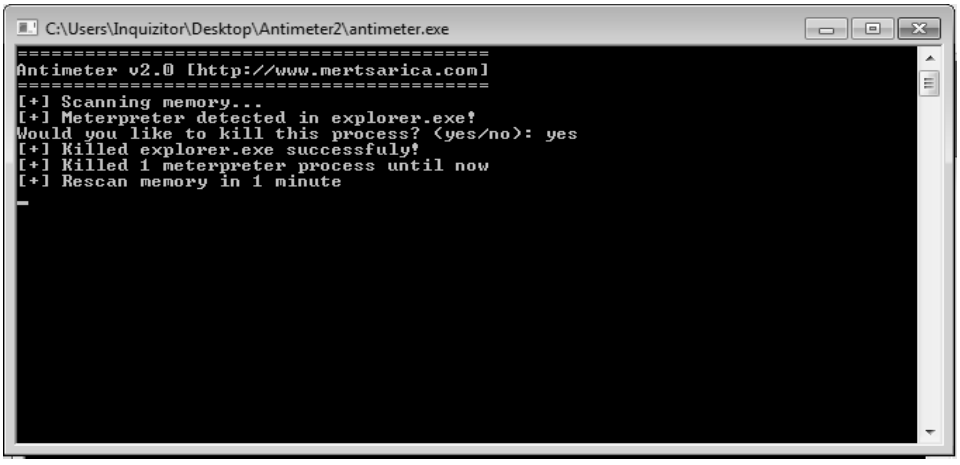

Рис. 3. Сканирование памяти с помощью Antimeter2

Из рисунка 3 следует, что при сканировании был найден процесс, инфицированный полезной нагрузкой meterpreter, после чего пользователю было предложено «убить» его. Также все обнаруженные процессы, инфицированные meterpreter, фиксируются в специальном файле antimeter.txt вместе с пометкой о времени обнаружения для непосредственного анализа и получения статистики инфицирования процессов.

Мониторинг и исследование памяти с помощью утилиты Antimeter2 позволяет не только обнаружить полезную нагрузку meterpreter на скомпрометированной системе, но и предотвратить дальнейшую ее эксплуатацию посредством деактивации инфицированного процесса.

Локальное повышение прав

После проникновения в систему жертвы атакующий должен узнать, какую систему он скомпрометировал и какие права на ней он имеет в данный момент, для этого можно воспользоваться командами sysinfo и getuid. На рисунке 4 представлена информация, которую удалось получить, благодаря применению данных команд.

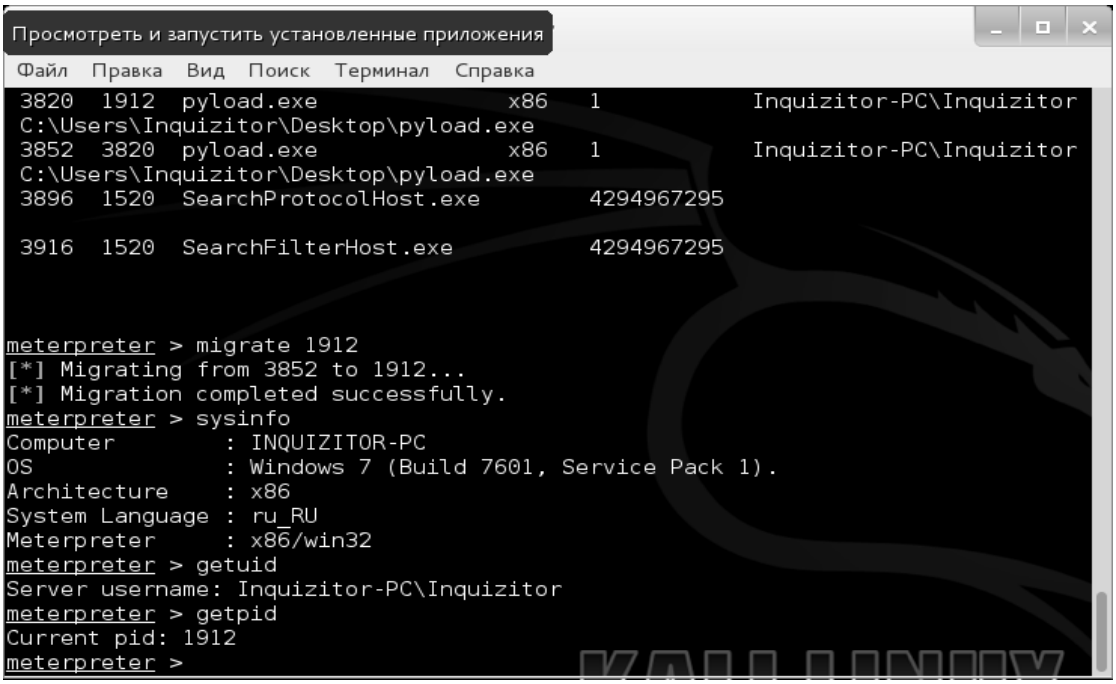

Рис. 4. Получение сведений о системе и проверка прав пользователя

Из рисунка 4 следует, что текущие права не являются правами пользователя «System», поэтому атакующий может попробовать повысить свои привилегии в системе, т.е. совершить локальное повышение прав. Для повышения привилегий используется команда getsystem интерпретатора команд meterpreter. Команда getsystem использует четыре техники (метода) повышения прав: две техники Named Pipe Impersonation (требуются права администратора), снятие копии токена с правами системы (права администратора) и эксплойт KiTrap0D (достаточно прав пользователя). Ниже описаны подробности данных техник локального повышения прав.

Named Pipe — это механизм, который делает возможным межпроцессную связь для приложений. Приложение, создающее «трубу» (pipe), называется pipe-сервер, а приложение, которое подключается к pipe-серверу, известно как pipe-клиент, что позволяет серверному потоку выполнять действия от имени клиента, но в пределах контекста безопасности клиента. Проблема возникает, когда клиент имеет больше прав, чем сервер. Этот сценарий создаст атаку повышения привилегий, которая называется Named Pipe Impersonation [3].

Каждый пользователь в операционной системе обладает уникальным идентификатором (token ID). Этот ID используется для проверки уровней привилегий у пользователей. Суть техники заключается в копировании (дубликате) ID пользователя более высокого уровня пользователем более низшего уровня, т.е. пользователь более низшего уровня наследует права и привилегии более высшего пользователя [3].

Эксплойт KiTrap0D, выпущенный в 2010 г., сделал уязвимой каждую ОС Windows. Предпосылкой уязвимости является то, что когда доступ к 16-битным приложениям поддерживается на 32-битной (x86) платформе, то она ненадлежащим образом проверяет определенные вызовы BIOS. Это позволяет локальному пользователю получить привилегии, обрабатывая VDM_TIB структуру данных в Thread Environment

Block (TEB). Уязвимость существует из-за неправильной обработки исключения в обработчике ловушек #GP (nt!KiTrap0D) [3].

При попытке повышения прав атакующий может увидеть сообщение: «priv_elevate_getsystem: Operation failed: Access is denied». Данное сообщение свидетельствует о том, что уязвимость, которую использует эксплойт KiTrap0D, устранена, а на скомпрометированной системе включена служба UAC (User Account Control), которая следит за изменением прав пользователей в системе и блокирует их несанкционированное изменение. Выход из данной ситуации существует. Для обхода UAC применяется эксплойт bypassuac_injection, требующий наличия активной сессии на скомпрометированной системе. Он позволяет обойти Windows UAC через процессы инъекции, используя сертификат надежного издателя. Эксплойт bypassuac_injection будет порождать другую полезную нагрузку, для которой флаг UAC отключен, применяя технику «рефлексивной dll инъекции» [4], позволяющей использовать только dll двоичной полезной нагрузки вместо трех отдельных файлов. Эксплойт bypassuac_injection дает возможность обойти UAC даже при активно работающем антивирусном программном обеспечении, так как данная методика не использует исполняемых файлов, и при ее применении происходит минимальное воздействие на целевой процесс. После отключения флага UAC атакующий может повысить свои права до пользователя «System» через оставшиеся три техники команды getsystem.

Эксплойт bypassuac_injection требует установки следующих параметров: активная сессия (set session <сессия>), ip-адрес машины атакующего (set lhost <ip-адрес>), тип полезной нагрузки (set payload <тип нагрузки>). Применение эксплойта представлено на рисунке 5.

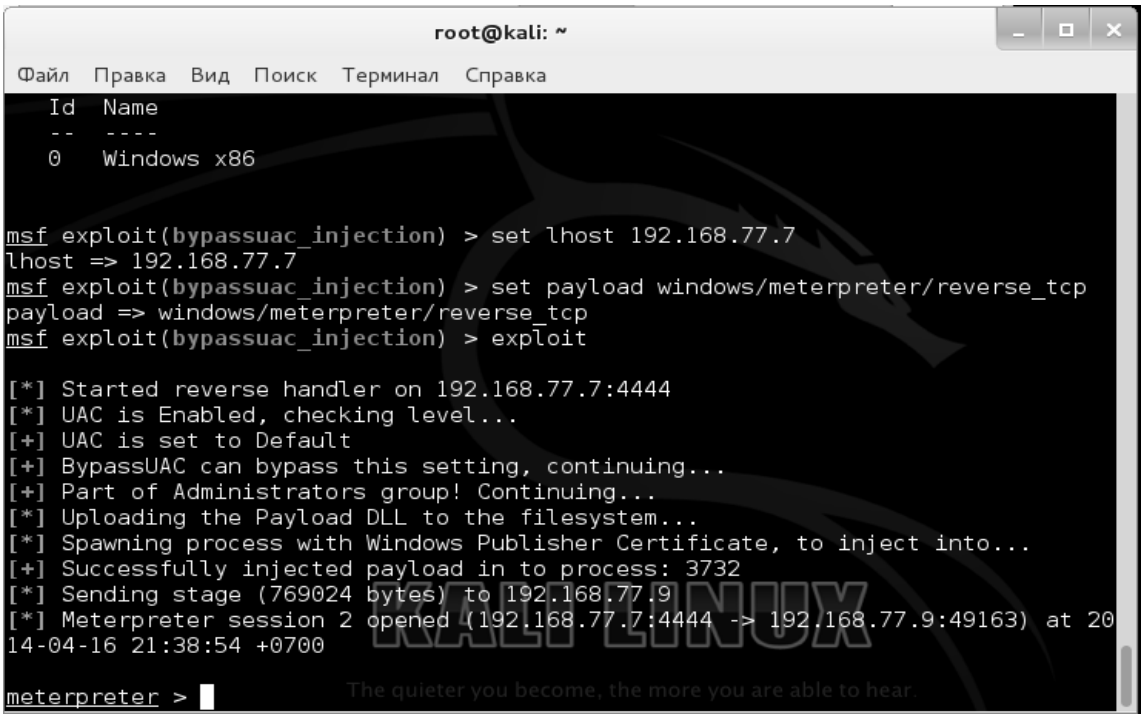

Рис. 5. Обход службы UAC

Из рисунка 5 следует, что эксплойт выполнился удачно на скомпрометированной системе: загрузил dll полезной нагрузки на удаленную систему, породил процесс с сертификатом надежного издателя, осуществил инъекцию полезной нагрузки в процесс с pid 3732 и открыл новую активную сессию, для которой UAC был отключен. После выполнения эксплойта атакующий может локально повысить свои права, используя команду getsystem. Повышение текущих прав до привилегий пользователя «System» представлено на рисунке 6.

Из рисунка 6 следует, что атакующий получил максимальные привилегии на скомпрометированной системе, обойдя службу UAC, об этом свидетельствует надпись «NT AUTHORITY».

Для защиты от данной уязвимости необходимо использовать специализированные программные средства исследования и мониторинга памяти, для того чтобы обнаружить и пресечь выполнение вредоносного кода внутри адресного пространства процессов.

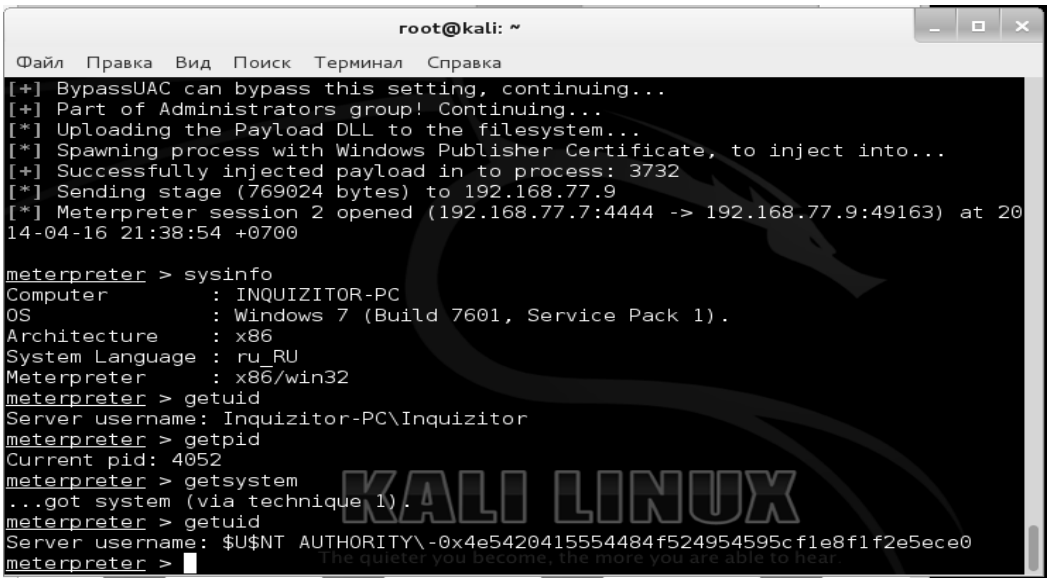

Рис. 6. Локальное повышение прав

Получение подробных сведений о системе и конфиденциальной информации

После получения наивысших привилегий и перехода на системный процесс можно получить много полезной информации о скомпрометированной системе: сведения о реестре, хэши паролей, информацию о сети, список запущенных служб, сведения о системе, сохранить все это на рабочей станции атакующего. Для получения такой информации можно воспользоваться скриптом scraper.rb, который сохраняет ее в директории /root/.msf4/logs/scripts/.

В ходе выполнения тестирования на проникновение была получена следующая информация: HKCR, HKCU, HKLM, HKU – информация о реестре; nethood, network – информация о сети; system, system info, env, group, localgroup, users, service, shares – информация о системе, окружении пользователей, группах, службах, списке пользователей, общих папках; и самое главное, hashes – информация о списке пользователей и хэши их паролей. Все файлы можно открыть, просмотреть и изучить для принятия дальнейших действий. Например, содержимое файла hashes.txt представлено на рисунке 7.

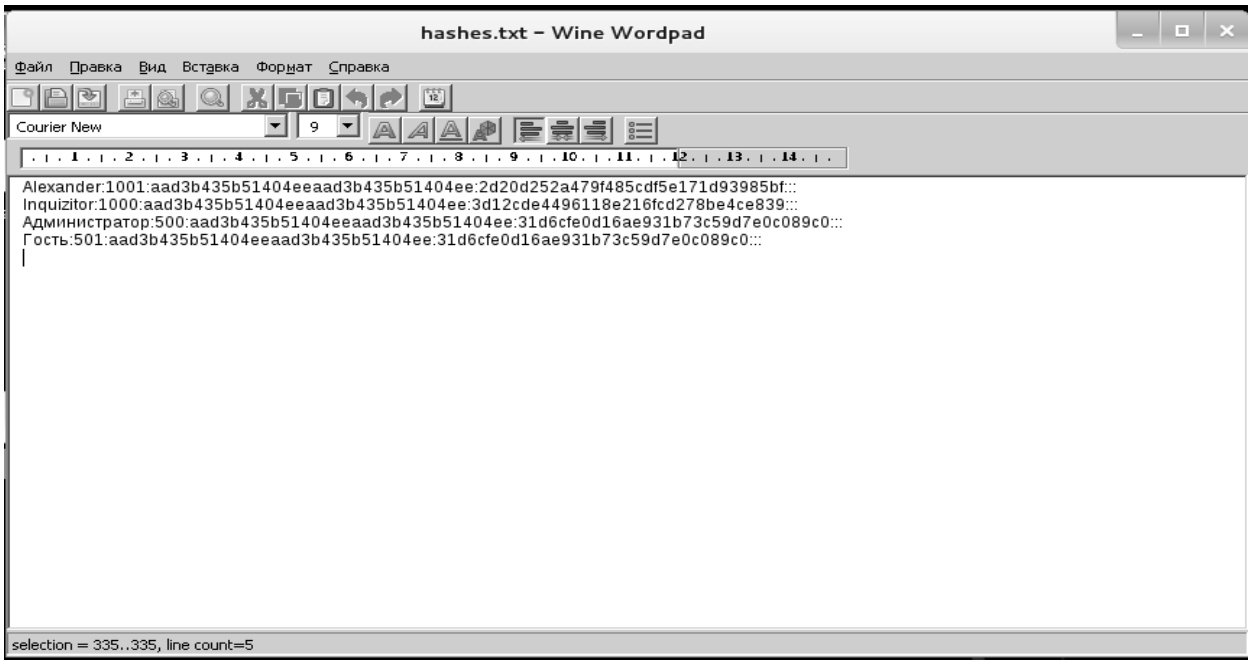

Рис. 7. Содержимое файла hashes.txt

Из рисунка 7 следует, что на скомпрометированной системе присутствуют четыре пользователя, для которых указаны идентификаторы и хэши их паролей. Поскольку скомпрометированной системой является Windows 7, то хеширование паролей производилось по алгоритму NTLM, установленному в системе по умолчанию, т.е. хэши являются NTLM-хэшами.

Получив хэши паролей, можно попытаться произвести их расшифровку, используя специализированные программы (password crackers). Для демонстрации расшифровки хэшей можно воспользоваться программой ophcrack. Для работы этого приложения необходимо скачать и использовать в процессе расшифровки радужные таблицы. Радужные таблицы выбираются в зависимости от того, какой тип имеют хэши. В ходе тестирования в программе ophcrack была использована радужная таблица с названием vista_free размером 482 МБ. На рисунке 8 представлен результат работы программы.

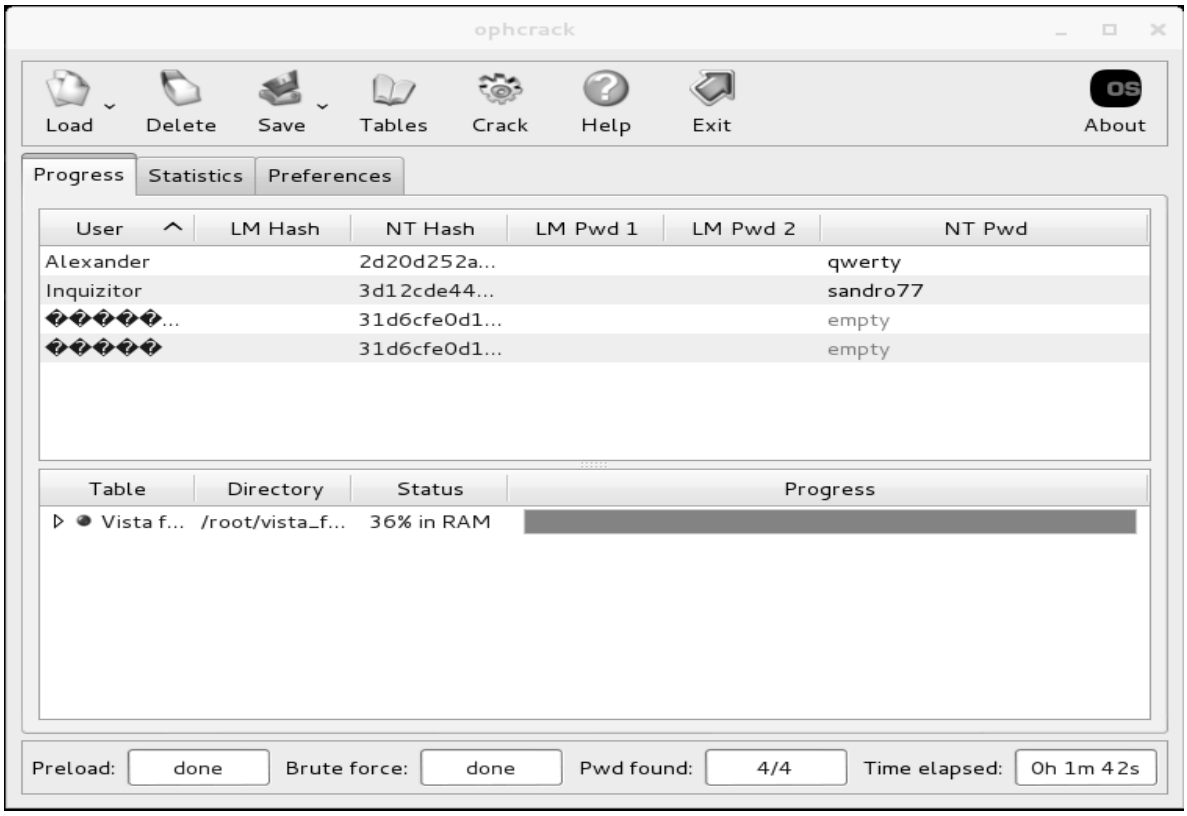

Рис. 8. Расшифровка хэшей

Из рисунка 8 следует, что все хэши были расшифрованы, а в последней колонке отображены пароли каждого пользователя. Для проведения расшифровки хэшей потребовалась 1 минута и 42 секунды.

Для того чтобы избежать или усложнить расшифровку паролей при утечке хэшей, пользователи должны не только руководствоваться общими правилами выбора пароля, но и использовать вместо стандартного алгоритма NTLMv1, осуществляющего хэширование паролей по умолчанию, более защищенный и устойчивый к атакам NTLMv2.

В ходе работы была исследована проблема защищенности компьютерных систем от несанкционированного доступа. Во время проведения исследования была осуществлена атака на стороне клиента с использованием специализированного модуля для генерации shell-кода, совершено локальное повышение прав до привилегий пользователя «System», после чего были получены имена и пароли всех пользователей системы. В итоге после проведения тестирования на проникновение установлено, что развитие информационных технологий привело к тому, что пользователю с каждым днем становится все труднее защищать свою компьютерную систему от различных типов сетевых атак, и в случае, если злоумышленник получил несанкционированный доступ к системе пользователя и расширил свои права на ней, то он сможет делать практически любые действия на этой системе, включая и слежку за скомпрометированным пользователем.

Библиографический список

1. Тестирование на проникновение: инструментальный анализ уязвимостей или имитация действий злоумышленника? [Электронный ресурс]. — URL: http://www.nobunkum.ru/ru/pentest (дата обращения: 12.04.2014).

2. Truncer C. Veil – A Payload Generator to Bypass Antivirus [Electronics resource]. — URL: https://www.christophertruncer.com/veil-a-payload-generator-to-bypass-antivirus/ (дата обращения: 15.04.2014).

3. Левин К.В. Metasploit Penetration Testing Cookbook – часть 5 [Electronics resource]. — URL: http://www.levinkv.ru/bezopasnost/metasploit/metasploit-penetration-testing-cookbook-часть-5.html (дата обращения: 15.04.2014).

4. Fewer S. Reflective DLL Injection v1.0 [Electronics resource]. — URL: http://www.harmonysecurity.com/files/HS-P005_ReflectiveDllInjection.pdf. (дата обращения: 16.04.2014).