

BloodHound Community Edition (BHCE) Guide (RU)

 [blog.taipanbyte.ru/2024/BloodHound-Community-Edition-\(BHCE\)-Guide-\(RU\)](http://blog.taipanbyte.ru/2024/BloodHound-Community-Edition-(BHCE)-Guide-(RU))



Эта статья является русскоязычной адаптацией оригинальной статьи, с которой вы можете ознакомиться перейдя по [ссылке](#).



Я сталкивался со многими хакерами, которые хотели бы научиться использовать BloodHound, но по каким либо причинам не могут начать. В результате они теряются и разочаровываются.

В этой статье я расскажу о настройке, сборе данных, их анализе и предоставлении связей на основе этих данных. В качестве образца данных для сбора и анализа будет использован [лес GOADv2](#). Я уже собрал несколько образцов данных и [поместил их в репозиторий](#) для анализа и практической работы.

Огромная благодарность [SpecterOps](#) за создание BloodHound и предоставление его сообществу!

Зачем это все §

Прежде чем мы начнем, нам нужно понять, зачем это нужно. Active Directory печально известна своей сложностью, а в любой сложной системе есть много возможностей для ошибок. AD интересна тем, что в большинстве сред не существует традиционных эксплойтов на основе CVE - вместо этого большинство эскалационных атак основывается на неправильной конфигурации. Наша задача - перечислить как можно больше таких сценариев эскалации, определить их риск и представить рекомендации по снижению риска до минимально возможного уровня.

Если вы хотите, чтобы вам **платили** за эту работу, вы должны уметь эффективно представлять эти риски в письменном виде!

Background §

BloodHound был разработан компанией SpecterOps как способ визуализации отношений между объектами в AD. Из-за масштаба и сложности большинства сетей AD аудит этих связей вручную - сущий кошмар. Вместо этого в оригинальной BloodHound для визуализации этой информации использовалась [теория графов Neo4j](#), позволяющая передавать информацию между объектами.

В настоящее время существует [три версии BloodHound](#), о которых вам необходимо знать:

[BloodHound Legacy](#): Оригинальная версия BloodHound, которая больше не поддерживается. Построена на базе приложения Electron, несколько сложна в настройке.

[BloodHound Community Edition](#): Выпущена в августе 2023 года, активно поддерживается. Использует docker compose для управления набором контейнеров, исключительно прост в развертывании. Плавный интерфейс веб-приложения.

[BloodHound Enterprise](#): Платная версия BloodHound для управления путями атак. Основное отличие заключается в том, что [эта версия используется для управления рисками](#) и проверки.

Основные компоненты §

Есть несколько различных компонентов, о которых мы должны знать. Во-первых, само приложение BloodHound - это не более чем фронт-энд для визуализации, представления и анализа данных. Нам нужно собрать данные об окружающей среде с помощью коллектора, чтобы они попали в приложение для анализа.

Коллектор §

Нам нужно собрать данные из среды AD, чтобы передать их в BloodHound для анализа. Существует два основных коллектора, о которых вам нужно знать:

[SharpHound](#): Это официально поддерживаемый инструмент коллектора для BloodHound, написанный на C#. Для сбора информации его необходимо запускать с компьютера под управлением Windows, подключенного к домену.

[BloodHound.py](#): Сценарий на языке python, разработанный сообществом и используемый для сбора данных AD. Может быть запущен с машины на базе Linux, например Raspberry Pi.

Важно понимать, что на момент написания этой статьи `bloodhound.py` не поддерживает BloodHound-CE. Вы должны использовать [ветку](#) `bloodhound-ce` коллектора `bloodhound.python`, если решите использовать его. Не стоит смешивать устаревшие коллекторы с коллекторами Community Edition - это приведет к сбою при вводе (а это очень неприятно!).

Фронтенд [§](#)

BloodHound - это веб-приложение, используемое для интерпретации данных, полученных от коллектора. Это приложение с графическим интерфейсом, с которым мы взаимодействуем, чтобы интерпретировать данные для выявления рисков и путей эскалации. Фронтенд хорош лишь настолько, насколько хороши данные, полученные от коллектора.

Ввод данных [§](#)

Внутри графического интерфейса фронтенда находится File Ingest. С его помощью данные, полученные от коллектора, помещаются в базу данных Neo4j. После разбора эти данные будут доступны GUI-приложению для анализа.

API [§](#)

Одним из самых интересных моментов в BloodHound-CE является доступный HTTP API для запроса данных. Это поможет нам автоматизировать и быстро извлекать данные, чтобы доказать их ценность в ходе пентестинга.

Устаревшая версия BloodHound [§](#)

Мы уже рассказывали об оригинальной версии BloodHound и о том, почему она важна, но это очень важно понимать. Старые коллекторы НЕ РАБОТАЮТ с Community Edition BloodHound. Любой желающий может использовать устаревшую версию, и она по-прежнему будет отлично работать, однако она не будет соответствовать последним угрозам. Поскольку мы доказываем ценность игры для клиентов, нам нужно использовать предметы, которые могут оценить наиболее применимые для них риски.

Начало работы [§](#)

Отлично, теперь, когда мы знаем все о каждой части головоломки, мы можем приступить к установке BloodHound-CE и сбору данных для нашего анализа.

Запуск контейнеров §

Community Edition [использует docker compose через набор контейнеров](#). Это значительно упрощает запуск и управление инфраструктурой для BloodHound, поскольку все контейнеры находятся в сети docker.

Нам нужно лишь загрузить файл `docker-compose.yml` и дать команду docker на сборку контейнеров.

```
wget https://raw.githubusercontent.com/SpecterOps/bloodhound/main/examples/docker-compose/docker-compose.yml -O docker-compose.yml
```

```
docker-compose up
```

```
(root㉿kali)-[~/home/kali/bloodhound]
└─# wget https://raw.githubusercontent.com/SpecterOps/bloodhound/main/examples/docker-compose/docker-compose.yml -O docker-compose.yml
--2024-04-16 05:21:52-- https://raw.githubusercontent.com/SpecterOps/bloodhound/main/examples/docker-compose/docker-compose.yml
Resolving raw.githubusercontent.com (raw.githubusercontent.com) ... 185.199.111.133, 185.199.110.133, 185.199.109.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.111.133|:443... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 3618 (3.5K) [text/plain]
Saving to: 'docker-compose.yml'

docker-compose.yml                                100%[=====] 3618/3618

2024-04-16 05:21:53 (11.0 MB/s) - 'docker-compose.yml' saved [3618/3618]

└─# ls
docker-compose.yml

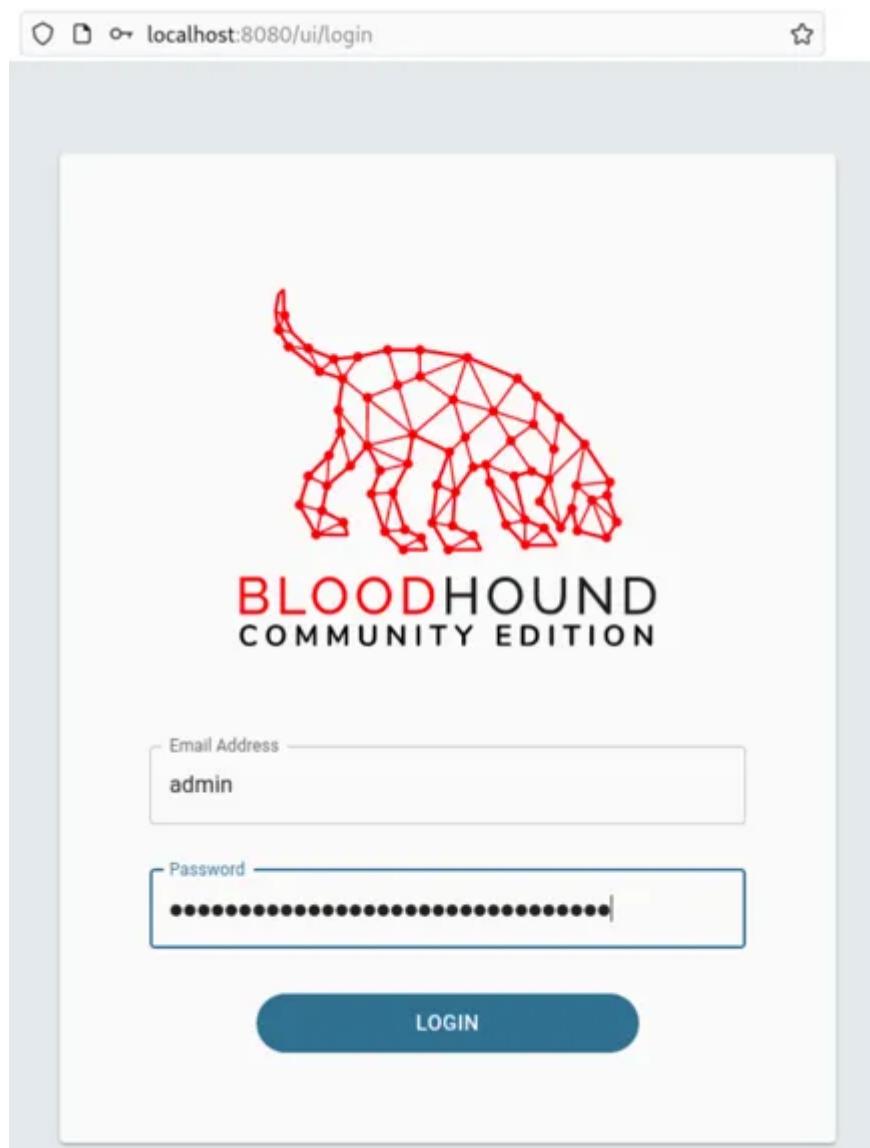
└─# docker-compose up
Creating network "bloodhound_default" with the default driver
Creating volume "bloodhound_neo4j-data" with default driver
Creating volume "bloodhound_postgres-data" with default driver
Pulling app-db (docker.io/library/postgres:13.2)...
13.2: Pulling from library/postgres
69692152171a: Pull complete
a31b993d5cc6: Pull complete
f65921886500: Pull complete
b9c1a94e4ca8: Pull complete
435dd99ceb68: Pull complete
d3ee8e88c67c: Pull complete
84b08674f942: Pull complete
7d358e850d3e: Pull complete
```

```
4f4fb700ef54: Pull complete
87f37be7a7b9: Pull complete
46a62ee4571b: Pull complete
3455a632654e: Pull complete
4b7a7ddb117c: Pull complete
Digest: sha256:9f40fa306b389f7d5d64ba2be651fbb9859990cbd6a2502d0d7b5e454c3e8155
Status: Downloaded newer image for specterops/bloodhound:latest
Creating bloodhound_app-db_1 ... done
Creating bloodhound_graph-db_1 ... done
Creating bloodhound_bloodhound_1 ... done
Attaching to bloodhound_app-db_1, bloodhound_graph-db_1, bloodhound_bloodhound_1
bloodhound_1  | {"level":"info","time":"2024-04-16T09:26:55.272171177Z","message":"
bloodhound_1  | {"level":"info","time":"2024-04-16T09:26:55.272468567Z","message":"
app-db_1      | The files belonging to this database system will be owned by user
app-db_1      | This user must also own the server process.
app-db_1      |
app-db_1      | The database cluster will be initialized with locale "en_US.utf8".
app-db_1      | The default database encoding has accordingly been set to "UTF8".
app-db_1      | The default text search configuration will be set to "english".
```

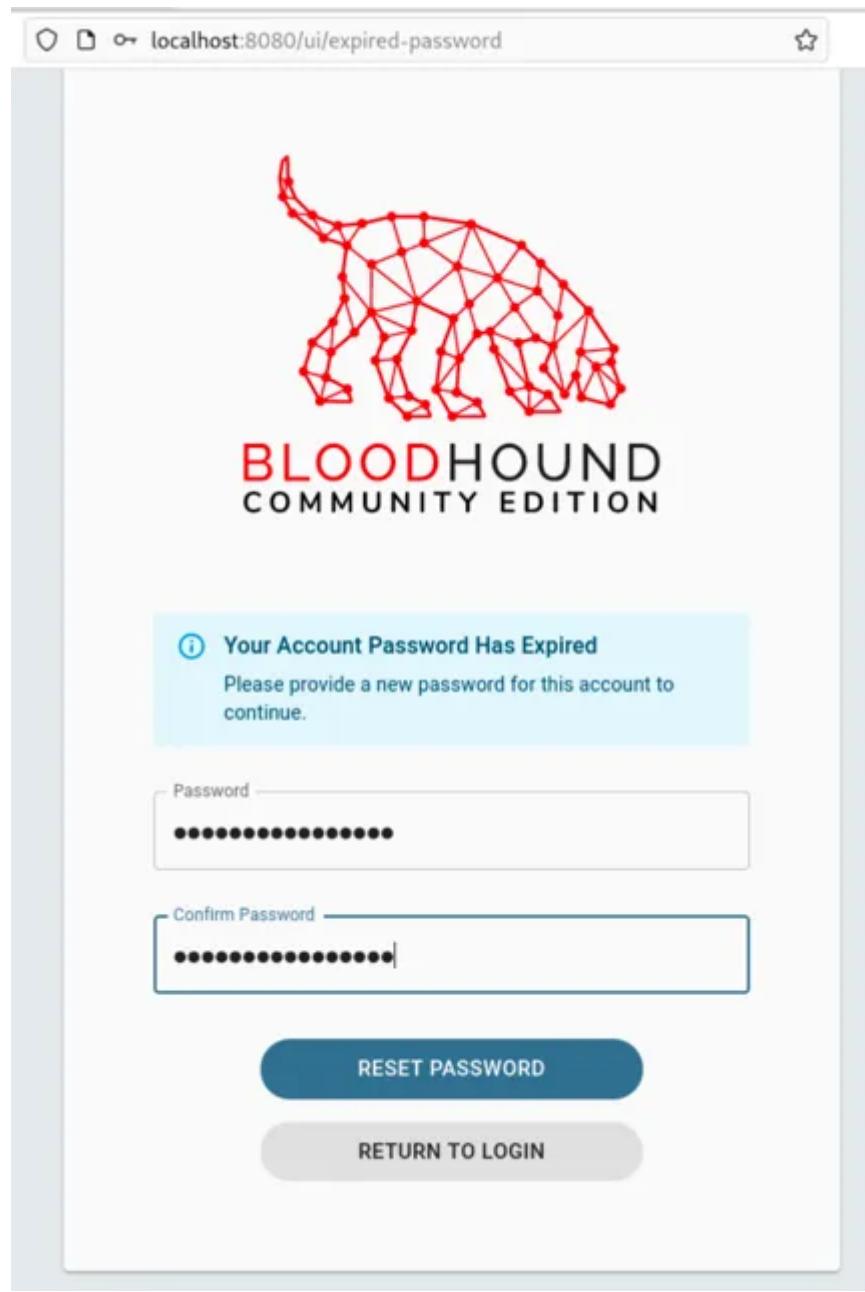
Обратите внимание, что в журналах будет указан начальный пароль, который нам нужно использовать для входа в систему.

```
, "message": "Feature flag risk_exposure_new_calculation created"
,"message": "Feature flag butterfly_analysis created"
,"message": "Feature flag azure_support created"
,"message": "Feature flag clear_graph_data created"
,"message": "#####
,"message": "#"
,"message": "# Initial Password Set To:      Udl2_L7wMVjdMPTaEYdITS9WQIqy4J7B      #"
,"message": "#"
,"message": "#####
,"message": "Adding index azresourcegroup_user_tags_index to labels AZResourceGroup on pr"
,"message": "Adding index azapp_user_tags_index to labels AZApp on properties user_tags us"
,"message": "Adding index azbase_user_tags_index to labels AZBase on properties user_tags"
,"message": "Adding index issuancepolicy_system_tags_index to labels IssuancePolicy on pr
```

Скопируйте этот пароль и перейдите по адресу <http://localhost:8080>, чтобы войти в графический интерфейс. Пользователем будет **admin**.



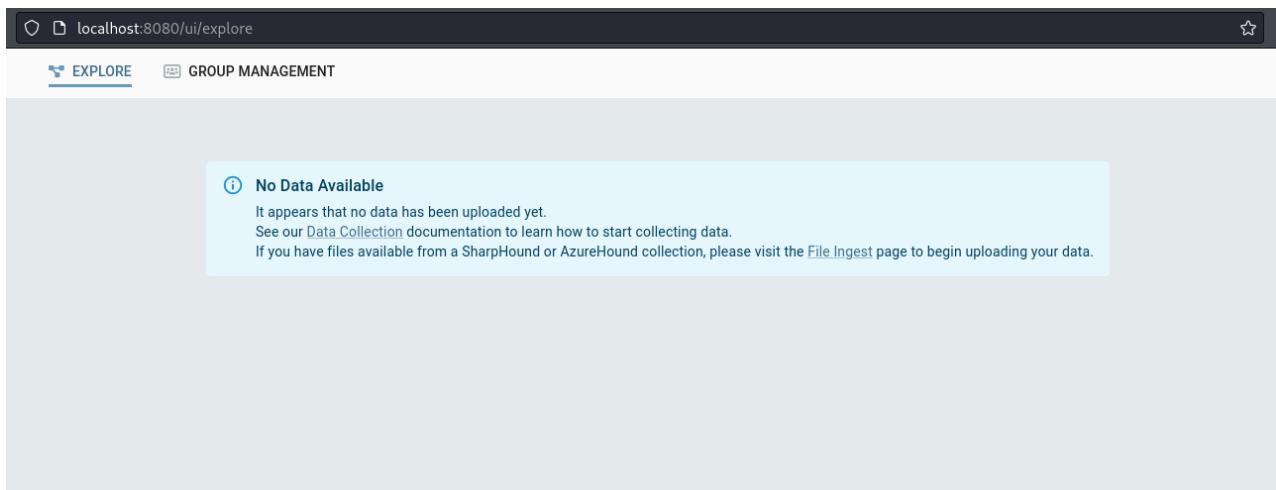
Когда вы войдете в систему, вам будет предложено изменить пароль.



Причем легкий пароль вам поставить не позволяют.

A screenshot of a web page titled "Password Requirements" with an exclamation mark icon. It lists five requirements: "must have at least 12 characters", "must have at least one lowercase", "must have at least one uppercase", "must have at least one number", and "must have at least one of (!@#\$%^&*)". Below this is a form field for "Password" containing three dots, with a red border indicating it does not meet the requirements. The background has a light pink gradient.

Отлично, мы вошли. Но подождите, здесь нет данных! Как же нам приступить к анализу? Сначала нам нужно загрузить коллекторы.

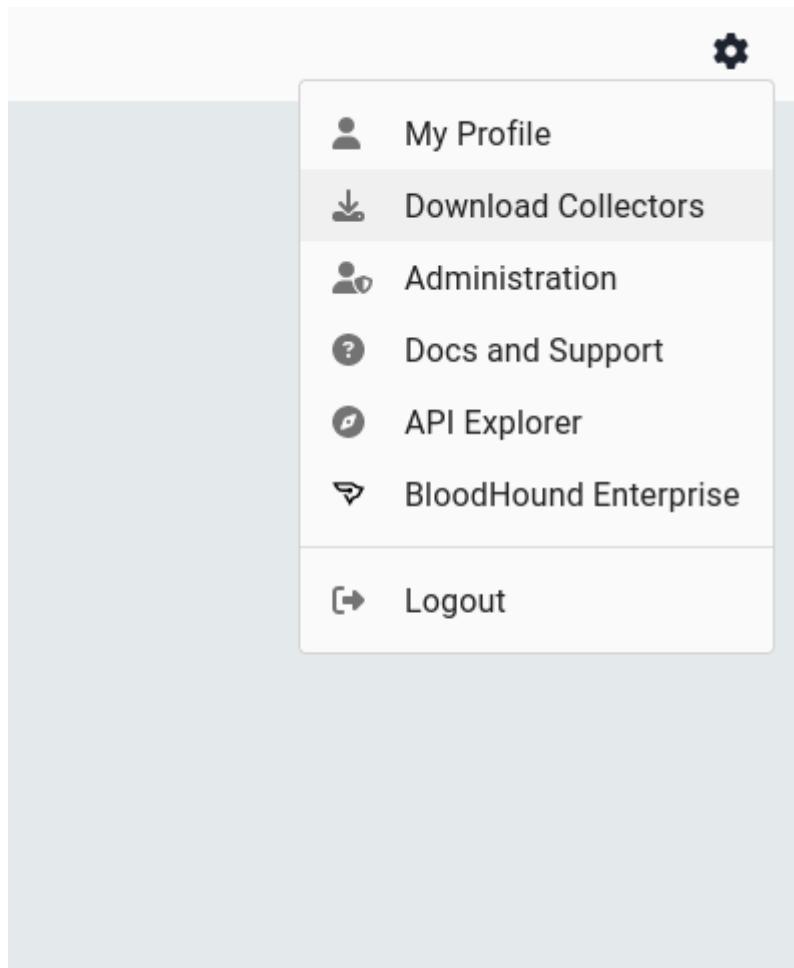


The screenshot shows a web browser window with the URL 'localhost:8080/ui/explore'. At the top, there are tabs for 'EXPLORE' (which is underlined in blue) and 'GROUP MANAGEMENT'. Below the tabs, a message box displays the text: 'No Data Available. It appears that no data has been uploaded yet. See our Data Collection documentation to learn how to start collecting data. If you have files available from a SharpHound or AzureHound collection, please visit the File Ingest page to begin uploading your data.' The rest of the page is mostly blank.

Сбор данных

У нас есть несколько способов сделать это. Мы можем использовать коллекторы SharpHound.exe C#, PowerShell или Python для сбора этой информации.

Чтобы получить копию поддерживаемых коллекторов, мы можем загрузить их прямо из графического интерфейса BHCE. Щелкните на шестеренке, а затем на “Загрузить коллекторы”. Откроется страница, на которой мы можем загрузить коллектор.





Download Collectors

SharpHound

SharpHound v2.3.3 (Latest)

SHA-256: [78b0faf9c2d4afca5873ccc2f04bf9dbffdf76cf1b854f954d20a7335782ec95](#)

AzureHound

AzureHound v2.1.8 (Latest)

SHA-256: [5c1e3fe624225de409ade1b9406238f4fc49022497452821e45ae1a44131611f](#)

После распаковки мы можем запустить этот инструмент сбора на удаленном хосте. Выберите удобный для вас способ выполнения этой операции, будь то маяк для последовательного выполнения или интерактивная RDP-сессия.

Коллектор на C#_S

Использование SharpHound.exe очень простое - мы можем просто запустить его без каких-либо дополнительных флагов, и он с радостью соберет стандартную информацию о текущем домене с текущим пользователем.

.\SharpHound.exe

Чтобы собрать всю доступную информацию, мы можем указать флаг **-c All**. Это позволит получить такие сведения, как информация ADCS, RDP и DCOM. Однако в больших средах сбор всей информации может привести к перегрузке машины - имейте это в виду!

Лично мне нравится собирать все, а затем шифровать ZIP паролем и присваивать файлам префикс. Хотя эти данные доступны всем пользователям домена, в чужих руках они могут оказаться конфиденциальной информацией.

.\SharpHound.exe -c All --zippassword 'p@ssw0rd' --outputprefix 'NORTH'

```
PS C:\Users\hodor\Downloads> .\SharpHound.exe -c All --zippassword 'p@ssw0rd' --outputprefix 'NORTH'
2024-03-30T12:49:19.1160004-07:00|INFORMATION|This version of SharpHound is compatible with the 5.0.0 Release of BloodHound
2024-03-30T12:49:19.3661323-07:00|INFORMATION|Resolved Collection Methods: Group, LocalAdmin, GPOLocalGroup, Session, LoggedOn, Trusts, ACL, Container, RDP, ObjectProps, DCOM, SPNTTargets, PSRemote, UserRights, CARegistry, DCRegistry, CertServices
2024-03-30T12:49:19.3971043-07:00|INFORMATION|Initializing SharpHound at 12:49 PM on 3/30/2024
2024-03-30T12:49:19.5857918-07:00|INFORMATION|[CommonLib LDAPUtils]Found usable Domain Controller for north.sevenkingdoms.local : winterfell.north.sevenkingdoms.local
```

После завершения мы увидим, что zip-файл был создан и с нашим предполагаемым префиксом. Обратите внимание на количество объектов, это может быть полезным числом в отчете для клиента.

```
2024-03-30T12:50:04.9190383-07:00|INFORMATION|Consumers finished, closing output channel
2024-03-30T12:50:04.9504125-07:00|INFORMATION|Output channel closed, waiting for output task to complete
Closing writers
2024-03-30T12:50:05.2465136-07:00|INFORMATION|Status: 347 objects finished (+347 7.711111)/s -- Using 39 MB RAM
2024-03-30T12:50:05.2465136-07:00|INFORMATION|Enumeration finished in 00:00:45.3863150
2024-03-30T12:50:05.4495239-07:00|INFORMATION|Saving cache with stats: 298 ID to type mappings.
 303 name to SID mappings.
 3 machine sid mappings.
 5 sid to domain mappings.
 0 global catalog mappings.
2024-03-30T12:50:05.5273038-07:00|INFORMATION|SharpHound Enumeration Completed at 12:50 PM on 3/30/2024! Happy Graphing!
PS C:\Users\hodor\Downloads> ls

Directory: C:\Users\hodor\Downloads

Mode                LastWriteTime         Length Name
----                -----        ---- 
d-----        3/12/2024    3:59 PM          SharpHound-v2.3.2
-a----        3/30/2024   12:50 PM       32173 NORTH_20240330125004_BloodHound.zip
-a----        3/30/2024   12:50 PM      57526 OTNKNWRmOGItYjJhYi00MTNmLWIyYTUtMzEwNWVkJWUSNWUw.bin
-a----        2/28/2024   6:20 PM       186604 SharpHound-v2.3.2.zip
-a----        3/30/2024   12:46 PM      1343488 SharpHound.exe
```

Междоменный сбор данных. §

Чтобы собрать данные из других доменов в том же лесу, нам нужно добавить несколько дополнительных флагов. Например, с помощью флага `--domain` нам нужно будет направиться в нужный домен. Далее мы переключимся на домен `sevenkingdoms.local`, работая от имени пользователя `hodor@north.sevenkingdoms.local`. Для работы эта машина должна быть способна разрешить домен в DNS.

```
.\SharpHound.exe -c All --domain sevenkingdoms.local --zippassword 'p@ssw0rd' --
outputprefix 'SEVENKINGDOMS'
```

Сбор от имени другого пользователя. §

Если мы находимся в лесу, но у нас нет доступа к учетной записи, доверенной в отдельном домене, мы всегда можем запустить `SharpHound.exe` с помощью команды `runas.exe command`.

```
runas /netonly /user:khal.drogo@essos.local cmd
.\SharpHound.exe -c All --domain essos.local --zippassword 'p@ssw0rd' --
outputprefix 'ESSOS'
```

В качестве альтернативы можно указать флаги `--ldapusername` и `--ldappassword` для подключения к другому домену. Это не требует работы команды `runas.exe`.

```
.\SharpHound.exe -c All --domain essos.local --ldapusername khal.drogo --
ldappassword horse --zippassword 'p@ssw0rd' --outputprefix 'ESSOS'
```

```
PS C:\Users\hodor\Downloads> .\SharpHound.exe -c All --domain essos.local --ldapusername khai.drogo --ldappassword horse --zippassword 'p@ssw0rd' --outputprefix 'ESSOS'
2024-03-30T13:15:52.2874281-07:00|INFORMATION|This version of SharpHound is compatible with the 5.0.0 Release of BloodHound
2024-03-30T13:15:52.5550766-07:00|INFORMATION|Resolved Collection Methods: Group, LocalAdmin, GPOLocalGroup, Session, LoggedOn, Trusts, ACL, Container, RDP, ObjectProps, DCOM, SPNTargets, PSRemote, UserRights, CARegistry, DCRegistry, CertServices
2024-03-30T13:15:52.6002884-07:00|INFORMATION|Initializing SharpHound at 1:15 PM on 3/30/2024
2024-03-30T13:15:52.7874693-07:00|INFORMATION|[CommonLib LDAPUtils]Found usable Domain Controller for essos.local : meereen.essos.local
2024-03-30T13:15:53.1936146-07:00|INFORMATION|Loaded cache with stats: 365 ID to type mappings.
```

Reflectively Loading SharpHound §

Сталкивались ли вы с ситуацией, когда файл SharpHound.exe отмечается средствами обнаружения как вредоносный? В некоторых сценариях мы можем обойти этот контроль, загрузив исполняемый файл C# в память, а затем выполнив точку входа ([ATT&CK ID T1620](#)).

```
$sh = [System.Reflection.Assembly]::Load([byte[]])
([IO.File]::ReadAllBytes("C:\Temp\SharpHound.exe"));
$cmd = "-c All --zippassword 'p@ssw0rd' --outputprefix REFLECTED"
[Sharphound.Program]::Main($cmd.Split())
```

```
PS C:\Users\hodor\Downloads> $sh = [System.Reflection.Assembly]::Load([byte[]][IO.File]::ReadAllBytes("C:\Temp\SharpHound.exe"));
PS C:\Users\hodor\Downloads> $cmd = "-c All --zippassword 'p@ssw0rd' --outputprefix 'REFLECTED'"
PS C:\Users\hodor\Downloads> $sh.EntryPoint
```

Name	:	<Main>
DeclaringType	:	Sharphound.Program
ReflectedType	:	Sharphound.Program

Обратите внимание, что вам может понадобиться добавить ключ `--outputdirectory`, чтобы обеспечить сохранение в нужном вам месте.

Python Collector §

Далее мы можем использовать инструмент `bloodhound.py` для сбора и этой информации. Как отмечалось ранее, текущий пакет `bloodhound.py` в репозиториях Kali предназначен только для Legacy BloodHound. Вам нужно загрузить [ветку](#) `bloodhound-ce` с [их GitHub](#).

```
sudo apt install bloodhound.py
```

```
bloodhound-python -d north.sevenkingdoms.local -u hodor -p hodor -c All -op default_kali_bloodhoundpy --zip -ns 192.168.56.10
```

Поскольку нас интересуют данные для поддержки BHCE, давайте сосредоточимся на установке этой ветки и ее использовании. В частности, мы можем клонировать эту ветку прямо с GitHub.

```
git clone -b bloodhound-ce https://github.com/dirkjanm/BloodHound.py.git
```

Если мы не хотим устанавливать зависимости, мы всегда можем создать контейнер для запуска. В репозитории уже есть Dockerfile, который мы можем использовать для сборки.

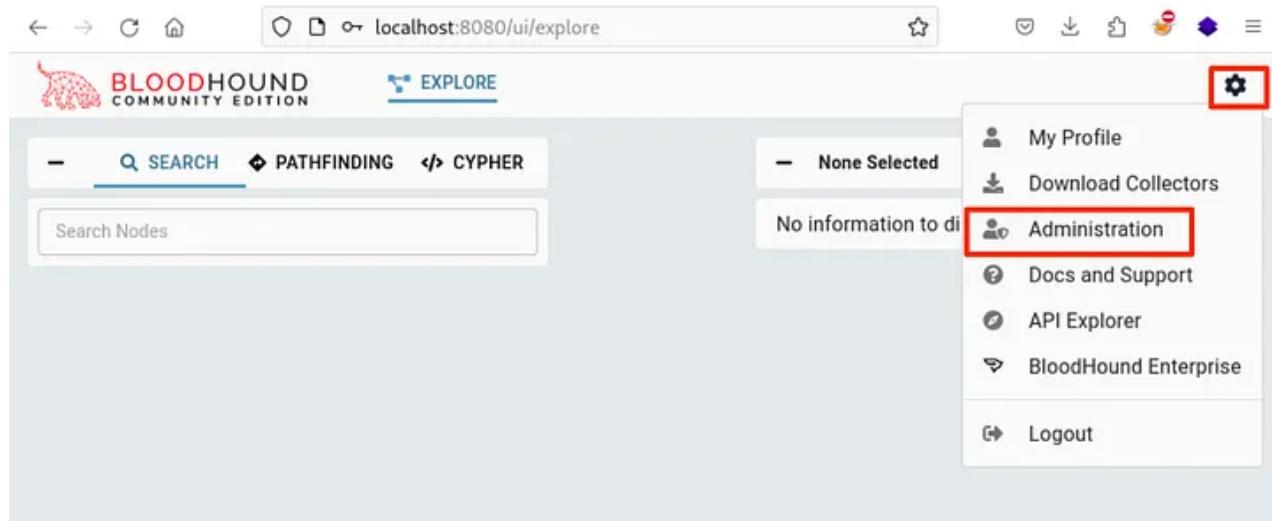
```
cd BloodHound.py
docker build -t bloodhound .
```

```
docker run -v ${PWD}:/bloodhound-data -it bloodhound
a0140a0d356a:/bloodhound-data# bloodhound-python -d north.sevenkingdoms.local -u
hodor -p hodor -c All -op ce_branch_bloodhoundpy --zip -ns 192.168.56.10
```

```
(kali㉿GOAD-Kali)-[~]-[SCI-GOAD-RANGE]
$ bloodhound-python -d north.sevenkingdoms.local -u hodor -p hodor -c All -op default_kali_bloodhoundpy --zip -ns 192.168.56.10
INFO: Found AD domain: north.sevenkingdoms.local
WARNING: Could not find a global catalog server, assuming the primary DC has this role
If this gives errors, either specify a hostname with -gc or disable gc resolution with --disable-autogc
INFO: Getting TGT for user
INFO: Connecting to LDAP server: winterfell.north.sevenkingdoms.local
INFO: Found 1 domains
INFO: Found 2 domains in the forest
INFO: Found 2 computers
INFO: Connecting to GC LDAP server: winterfell.north.sevenkingdoms.local
INFO: Connecting to LDAP server: winterfell.north.sevenkingdoms.local
INFO: Found 17 users
INFO: Found 51 groups
INFO: Found 3 gpos
INFO: Found 1 ous
INFO: Found 19 containers
INFO: Found 1 trusts
INFO: Starting computer enumeration with 10 workers
INFO: Querying computer: castelblack.north.sevenkingdoms.local
INFO: Querying computer: winterfell.north.sevenkingdoms.local
INFO: Done in 00M 02S
INFO: Compressing output into 20240331205852_bloodhound.zip
```

Сбор данных §

Теперь, когда мы смогли собрать данные, нам нужно уметь их использовать. Для этого нужно загрузить их в графический интерфейс, где они будут занесены в базу данных Neo4j. Для этого нужно нажать на кнопку **sog**, а затем на кнопку **Administration**.



На этом этапе мы можем нажать кнопку **UPLOAD FILE(S)**, чтобы загрузить наши данные. Обратите внимание, что мы не можем загрузить zip-файл, но мы можем выбрать несколько JSON-файлов одновременно.

The screenshot shows the BloodHound Community Edition interface at localhost:8080/ui/administration/file-ingest. The left sidebar has sections for Data Collection (File Ingest, Data Quality), Users (Manage Users), Authentication (SAML Configuration), and Configuration (Early Access Features). The main area is titled "Manual File Ingest" and contains a large red arrow pointing to a blue "UPLOAD FILE(S)" button. Below it is a section titled "Finished Ingest Log" with a table header: User, Start Time, End Time, Duration, Status, Status Message. A message at the bottom says "Rows per page: 10" and "0-0 of 0".

В появившемся всплывающем окне мы можем перетаскивать файлы, которые будут приняты. Помните, что мы не можем загрузить zip-архив, но мы можем загрузить все JSON-файлы, извлеченные из zip-архива.

The screenshot shows a modal dialog box for file upload. It has a central area with a folder icon and the text "Click here or drag and drop to upload files". Below this is a table titled "Files" listing several JSON files: NORTH_20240402181513_aiacas.json, NORTH_20240402181513_certtemplates.json, NORTH_20240402181513_computers.json, NORTH_20240402181513_containers.json, NORTH_20240402181513_domains.json, NORTH_20240402181513_enterprisecas.json, NORTH_20240402181513_gpos.json, and NORTH_20240402181513_groups.json. Each file has a green checkmark and the status "Ready". At the bottom are "CANCEL" and "UPLOAD" buttons.

После двойного нажатия на кнопку загрузки мы вернемся на страницу загрузки. Мы видим, что в статусе указано, что загрузка завершена! Мы можем продолжить загрузку дополнительных данных для других доменов в лесу.



EXPLORE



Data Collection

File Ingest

Data Quality

Users

Manage Users

Authentication

SAML Configuration

Configuration

Early Access Features

Manual File Ingest

UPLOAD FILE(S)

Finished Ingest Log

User	Start Time	End Time	Duration	Status	Status Message
spam@example.com	2024-04-02 22:40 EDT (GMT-0400)	2024-04-02 22:41 EDT (GMT-0400)	0 minutes	Complete	Complete

Rows per page: 10 ▾ 1-1 of 1 < >

Ошибки Ingest §

Один из самых неприятных моментов в работе BHCE - отсутствие обратной связи при загрузке устаревшей информации. Именно из-за этого сценария, как мне кажется, многие хакеры, которые учатся использовать этот инструмент, разочаровываются и бросают работу.

В этом случае, если мы соберем информацию с коллекционера для Legacy Bloodhound и импортируем ее в BHCE, есть вероятность, что файлы пройдут первоначальную проверку на загрузку, будут отмечены как завершенные, но на самом деле не будут приняты.

Finished Ingest Log

User	Start Time	End Time	Duration	Status	Status Message
spam@example.com	2024-04-11 08:12 EDT (GMT-0400)	2024-04-11 08:13 EDT (GMT-0400)	0 minutes	Complete	Complete

Вернувшись на страницу Explore, мы видим, что данные все еще не получены. Здесь легко расстроиться - в конце концов, мы же загрузили данные, верно?



No Data Available

It appears that no data has been uploaded yet.

See our [Data Collection](#) documentation to learn how to start collecting data.

If you have files available from a SharpHound or AzureHound collection, please visit the [File Ingest](#) page to begin uploading your data.

Чтобы обнаружить это, нам пришлось заглянуть в журналы с помощью `docker compose logs`, чтобы увидеть ошибку. Здесь показано, что произошла ошибка размаршалинга (процесса преобразования сериализованных данных, часто в определенном формате, например JSON, XML или бинарном, обратно в исходный объект или загруженные данные).

Такое случилось с данными, собранными с помощью `bloodhound-python` из репозитория Kali.

```
bloodhound-1 | {"level": "info", "user_id": "752989a6-846d-444f-9075-bbb620351824", "remote_addr": "192.168.65.1:4371", "proto": "HTTP/1.1", "referer": "http://192.168.40.180:8080/ui/administration/file-ingest", "user_agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36 Edg/123.0.0.0", "request_id": "51731b8c-d40f-4172-a47b-ad1d662bbe4c", "request_bytes": 0, "response_bytes": 330, "status": 200, "elapsed": 4.768083, "time": "2024-04-11T12:13:04.805180636Z", "message": "GET /api/v2/file-upload?skip=0&limit=10&sort_by=-id"}  
bloodhound-1 | {"level": "error", "time": "2024-04-11T12:13:07.84171197Z", "message": "Error decoding ein.Domain object: json: cannot unmarshal number into Go struct field Trust.Trusts.TrustDirection of type string"}  
bloodhound-1 | {"level": "info", "time": "2024-04-11T12:13:08.438585012Z", "message": "Expanding all AD group and local group memberships"}  
bloodhound-1 | {"level": "info", "time": "2024-04-11T12:13:08.439762804Z", "message": "Collected 113 groups to resolve"}  
bloodhound-1 | {"level": "info", "time": "2024-04-11T12:13:08.508270137Z", "message": "Finished post-processing 3 active directory computers"}  
bloodhound-1 | {"level": "info", "time": "2024-04-11T12:13:08.513807346Z", "message": "Finished building adcs cache"}  
bloodhound-1 | {"level": "info", "time": "2024-04-11T12:13:08.530453012Z", "message": "Started Data Quality Stats Collection"}  
bloodhound-1 | {"level": "info", "time": "2024-04-11T12:13:08.533645554Z", "message": "Cache successfully reset by datapipe daemon"}
```

Чтобы исправить это, я бы рекомендовал использовать последние версии коллекторов для BHCE. Я также хотел бы, чтобы графический интерфейс BHCE был обновлен, чтобы отразить эти ошибки размаршалинга или, по крайней мере, предоставить индикацию того, что Ingest не был успешно выполнен.

Ввод данных через API [§](#)

Чтобы получить данные через API, мы можем прочитать документацию. Лично я этого не делаю, поскольку загружаю данные через GUI, однако такая возможность поддерживается.

Просмотр данных [§](#)

Раз уж вы зашли так далеко, мы можем приступить к изучению собранных данных! Это позволит нам найти и понять взаимосвязи между объектами в лесу и то, как их можно использовать. Для начала мы можем использовать встроенные запросы для

изучения данных. Для этого нужно нажать на кнопку CYPHER, затем на значок папки, чтобы открыть запросы.

The screenshot shows the BloodHound Community Edition interface. At the top, there are tabs for EXPLORE and GROUP MANAGEMENT. Below them, there are buttons for SEARCH, PATHFINDING, and CYPHER. A red box labeled '1' highlights the CYPHER button. To its right, another red box labeled '2' highlights a query window containing the following Cypher code:

```
2 MATCH p=(n:Group)<--[:MemberOf*1..]-(m)
2 WHERE n.objectid ENDS WITH
"~512"
3 RETURN p
```

Below the query window are buttons for SAVE QUERY, HELP, and SEARCH. On the left, under Pre-built Searches, the ACTIVE DIRECTORY tab is selected, showing various search options. One option, "All Domain Admins", is highlighted with a red box labeled '3'. To the right, a graph visualization shows three nodes: EDDARD.STARK@NORTH.SEVENKINGDOMS.LOCAL (green), DOMAIN ADMINS@NORTH.SEVENKINGDOMS.LOCAL (yellow), and ADMINISTRATOR@NORTH.SEVENKINGDOMS.LOCAL (green). Arrows labeled "MemberOf" connect EDDARD.STARK to DOMAIN ADMINS and ADMINISTRATOR.

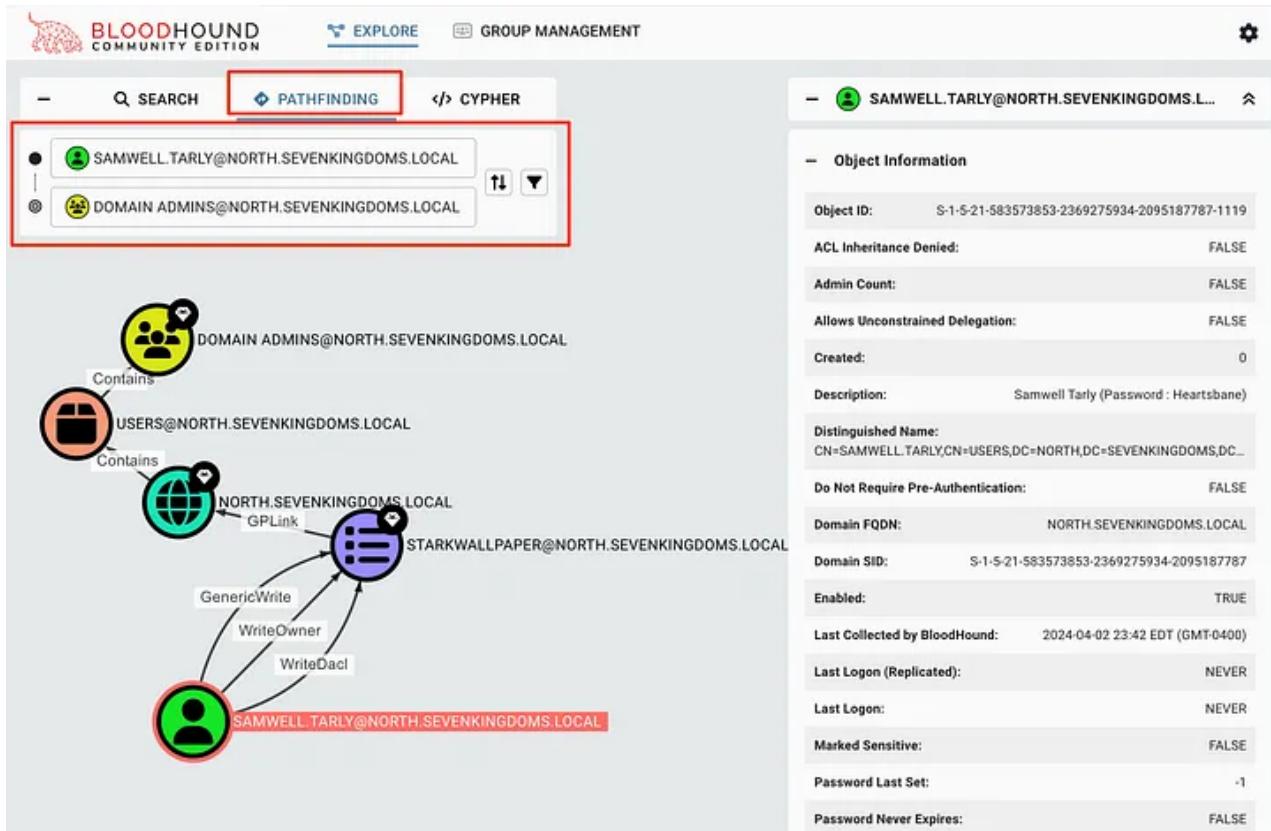
Щелкнув на одном из поисковых запросов, например “Все администраторы домена”, вы добавите запрос Cypher в строку поиска и поищите совпадения в базе данных. Результаты теперь будут на экране! Чтобы получить подробную информацию об объекте, мы можем щелкнуть на нем, чтобы просмотреть его свойства. Это поможет найти дополнительную информацию об учетной записи или домене.

The diagram illustrates the membership chain for the user EDDARD.STARK@NORTH.SEVENKINGDOMS.LOCAL. It shows three objects arranged vertically, each with a green circular icon and a black silhouette icon. The top object is labeled EDDARD.STARK@NORTH.SEVENKINGDOMS.LOCAL. An arrow labeled "MemberOf" points down to the middle object, which is labeled DOMAIN ADMINS@NORTH.SEVENKINGDOMS.LOCAL. Another arrow labeled "MemberOf" points down to the bottom object, which is labeled ADMINISTRATOR@NORTH.SEVENKINGDOMS.LOCAL.

Object Information	
Tier Zero:	TRUE
Object ID:	S-1-5-21-583573853-2369275934-2095187787-1111
ACL Inheritance Denied:	TRUE
Admin Count:	TRUE
Allows Unconstrained Delegation:	FALSE
Created:	2024-02-23 08:27 EST (GMT-0500)
Description:	Eddard Stark
Distinguished Name:	CN=EDDARD.STARK,CN=USERS,DC=NORTH,DC=SEVE...
Do Not Require Pre-Authentication:	FALSE
Domain FQDN:	NORTH.SEVENKINGDOMS.LOCAL
Domain SID:	S-1-5-21-583573853-2369275934-2095187787
Enabled:	TRUE
Last Collected by BloodHound:	2024-04-02 23:42 EDT (GMT-0400)
Last Logon (Replicated):	2024-03-24 19:42 EDT (GMT-0400)
Last Logon:	2024-04-02 22:32 EDT (GMT-0400)

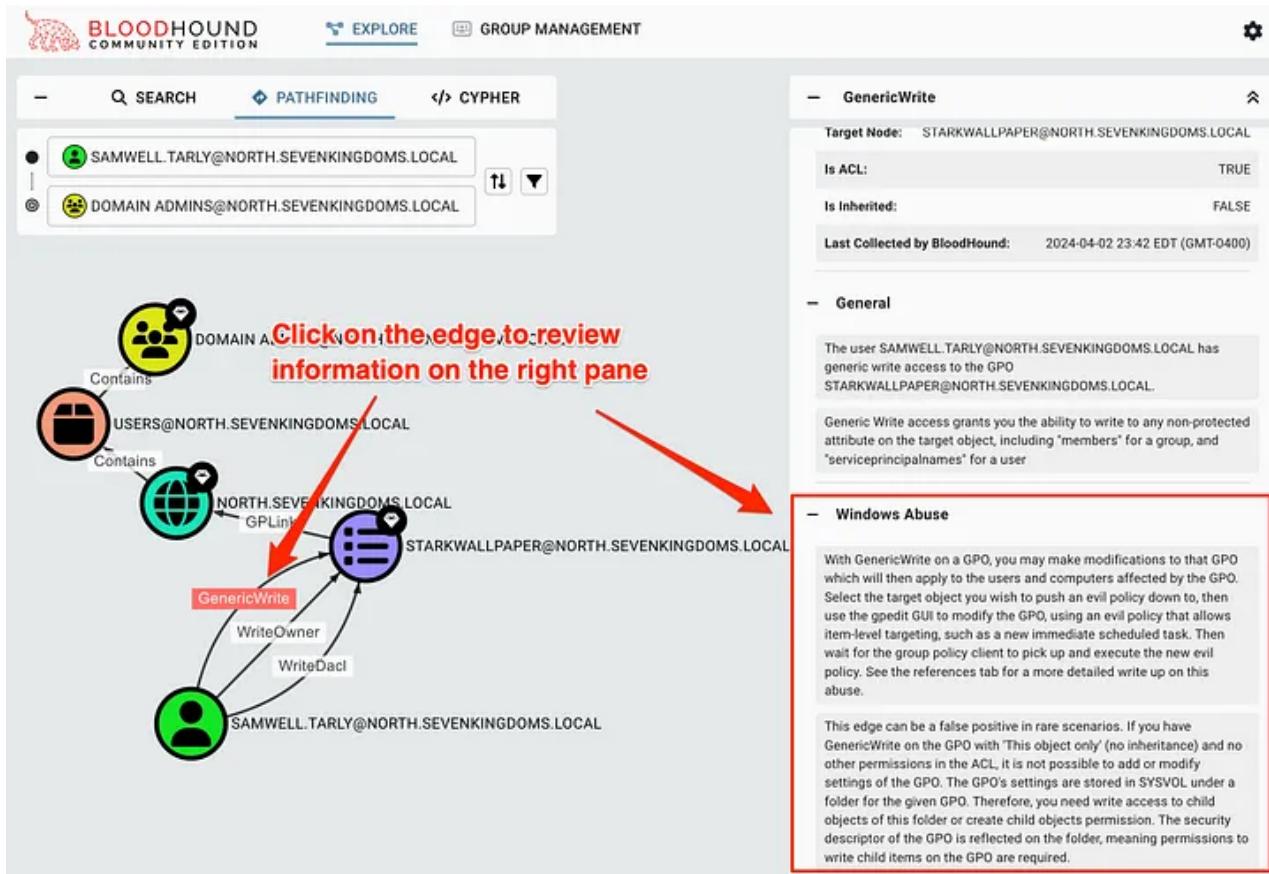
Поиск векторов атаки §

Чтобы найти конкретные пути от одного объекта к другому, мы можем использовать кнопку PATHFINDING. В данном случае мы можем запросить, как пользователь [samwell.tarly](#) может получить доступ к [администраторам домена](#), чтобы определить, как этот пользователь может использовать этот путь.



Edges §

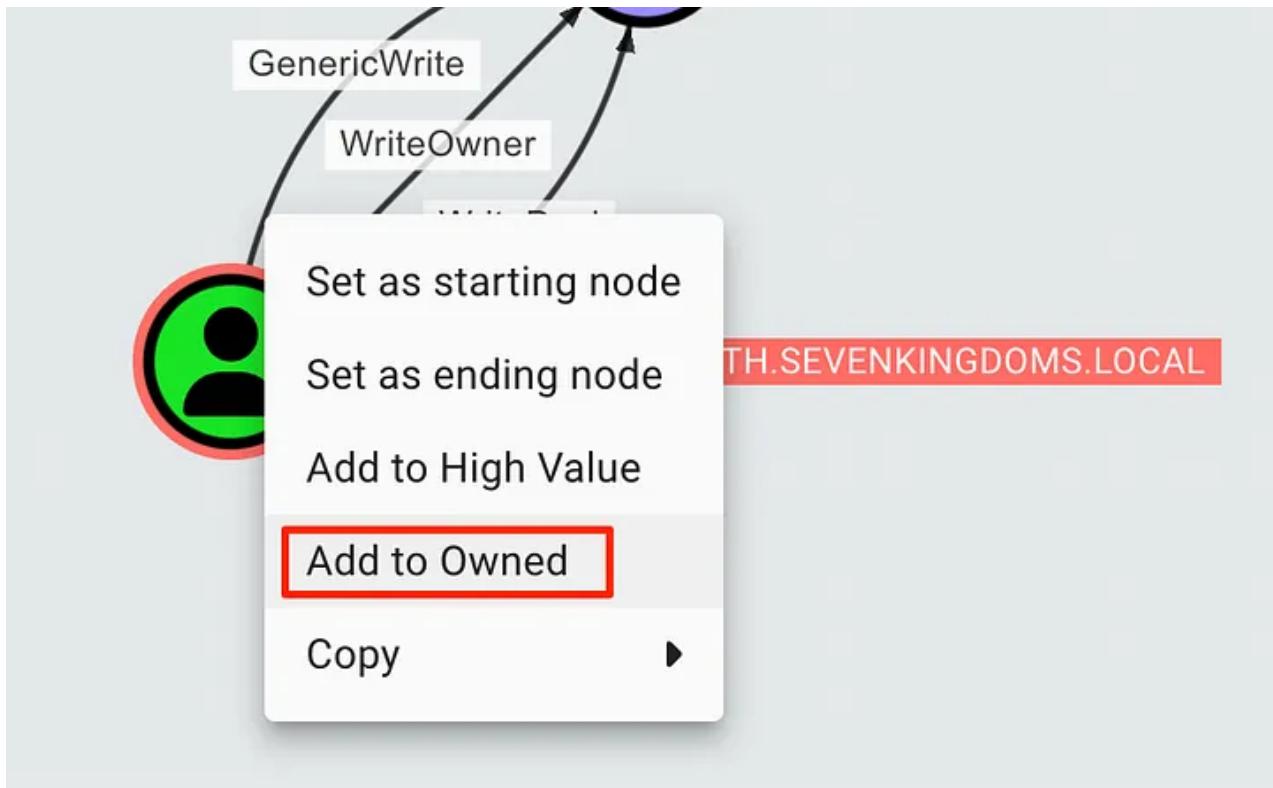
В этом сценарии мы видим, что пользователь `samwell.tarly` имеет разрешения `GenericWrite`, `WriteOwner` и `WriteDacl` на GPO `STARKWALLPAPER`. Если мы не знаем, как это может быть использовано, мы можем щелкнуть на самом краю, чтобы в правой панели открылись свойства. Здесь содержится информация об эдже, в том числе о том, как злоупотреблять этим разрешением на машинах под управлением Windows или Linux.



Насколько это круто? Здесь рассказывается, что и как мы можем использовать для доказательства воздействия. Эти свойства также включают в себя несколько отличных ссылок, которые всегда стоит прочитать, чтобы узнать больше о конкретном сценарии злоупотребления.

Пометка объектов “Owned” §

Чтобы пометить объект как Owned, мы можем щелкнуть на нем правой кнопкой мыши в графическом интерфейсе. При этом на объекте появится значок черепа, что позволит нам выполнять дополнительные запросы на основе принадлежащих объектов. Это поможет нам отслеживать, как мы можем маневрировать в среде по мере получения доступа.



Чтобы просмотреть объекты, помеченные как Owned, мы можем нажать на кнопку “GROUP MANAGEMENT” в верхней части страницы. Вы заметите, что сначала страница будет пустой, и нам придется нажать несколько кнопок, чтобы получить нужную информацию.

Name	Custom Member
No members in selected Asset Group	

Начните с нажатия второй кнопки, чтобы выбрать сущность для поиска. В большинстве случаев достаточно выбрать “Все домены Active Directory”.

BLOODHOUND
COMMUNITY EDITION

EXPLORE GROUP MANAGEMENT

HIGH VALUE

UNKNOWN DOMAIN OR TENANT

Name

Custom Member

No members in selected Asset Group

Name	Custom Member
ESSOS.LOCAL	
NORTH.SEVENKINGDOMS.LOCAL	
SEVENKINGDOMS.LOCAL	
All Active Directory Domains	
All Azure Tenants	

После этого мы можем нажать на верхнее выпадающее меню и выбрать “OWNED”.

BLOODHOUND
COMMUNITY EDITION

EXPLORE GROUP MANAGEMENT

HIGH VALUE

OWNED

HIGH VALUE

FILTERS

Total Count 52

Name

TOR@ESSOS.LOCAL

На этом этапе мы сможем увидеть учетную запись, которую мы отметили как захваченную. Это может быть полезно для отслеживания объектов по мере получения доступа, а также для отчетности перед клиентом.

The screenshot shows the BloodHound Community Edition interface. At the top, there are navigation links: 'EXPLORE' and 'GROUP MANAGEMENT'. The 'GROUP MANAGEMENT' link is underlined, indicating it is the active page. On the left side, there are several buttons and sections: 'OWNED' (highlighted in blue), 'ALL ACTIVE DIRECTORY DOMAI...', 'FILTERS', 'Total Count 1', and 'Add or Remove Members'. Below these are two buttons: 'User 1' and 'Computer 0'. The main content area displays a table with one row. The table has columns for 'Name' and 'Custom Member'. The row contains the name 'SAMWELL.TARLY@NORTH.SEVENKINGDOMS.LOCAL' and a green checkmark icon in the 'Custom Member' column. At the bottom of the table, there are pagination controls: 'Rows per page: 25', '1-1 of 1', and navigation arrows.

Запросы Cypher §

Хотя готовые поисковые запросы помогут быстро найти интересные объекты, нередки случаи, когда нам нужно найти что-то, не охваченное этими запросами. Чтобы решить эту проблему, мы можем создавать собственные запросы, отправляемые в базу данных Neo4j. Так мы можем убедиться, что поиск работает именно так, как мы задумали, и впоследствии сохранить его в пользовательском поиске.

Лично я считаю, что в готовых запросах BHCE не хватает некоторых критически важных операций поиска, которые помогают сопоставить принадлежащие объекты с цennыми целями. В BHCE и Legacy это решается по-другому, и для этого требуются специальные запросы в Cypher. Ниже приведен способ получения списка всех принадлежащих объектов в BHCE:

```
MATCH (n) WHERE "owned" in n.system_tags RETURN n
```

Его можно поместить в панель запросов Cypher, нажать кнопку поиска и увидеть все захваченные объекты. Обратите внимание, что на изображении ниже объекты User и Computer захвачены, но не имеют пути друг к другу из этого поиска.



-

SEARCH

PATHFINDING

</> CYPHER



```
$ MATCH (n) WHERE "owned" in  
n.system_tags RETURN n
```



SAMWELL.TARLY@NORTH.SEVENKINGDOMS.LOCAL



CASTELBLACK.NORTH.SEVENKINGDOMS.LOCAL

Вот несколько полезных запросов, которые я использовал в прошлом и которые помогли мне найти неправильную конфигурацию:

Найти все неограниченные делегирования от не-DC

```
MATCH (c1:Computer)-[:MemberOf*1..]->(g:Group) WHERE g.objectid ENDS WITH '-516'  
WITH COLLECT(c1.name) AS domainControllers MATCH (c2  
{unconstraineddelegation:true}) WHERE NOT c2.name IN domainControllers RETURN c2
```

Найдите пользователей, в описании которых есть слово “pass”.

```
MATCH p = (d:Domain)-[r:Contains*1..]->(u:User) WHERE u.description =~ '(?  
i).*pass.*' RETURN p
```

Список всех захваченных объектов

```
MATCH (n) WHERE "owned" in n.system_tags RETURN n
```

Найдите все пути от владения к объектам уровня 0

```
MATCH p = allShortestPaths((o)-[*1..]->(h)) WHERE 'owned' in o.system_tags AND  
'admin_tier_0' in h.system_tags RETURN p
```

Если вы хотите узнать больше, в документации BloodHound есть несколько фантастических ресурсов о том, как создавать и понимать глубокие запросы Cypher:
[Поиск с помощью Cypher - BloodHound \(bloodhoundenterprise.io\)](http://bloodhoundenterprise.io)

Пользовательские запросы §

Когда мы хотим сохранить результаты поиска для дальнейшего использования, мы можем сохранить его с помощью кнопки “Save Query”. Это приведет к появлению категории “Пользовательские поиски” в той же иконке папки для последующего использования.

The screenshot shows the BloodHound Community Edition interface. At the top, there's a logo for "BLOODHOUND COMMUNITY EDITION" and navigation links for "EXPLORE" and "GROUP MA". Below that, a search bar has "SEARCH" selected. The main area contains a Cypher query:

```
$ MATCH p = (d:Domain)-[r:Contains*1..]->(u:User) WHERE u.description =~ '(?i).*pass.*' RETURN p
```

Below the query are three buttons: "SAVE QUERY" (highlighted with a red box), "HELP", and "SEARCH".

Under "Pre-built Searches", there are tabs for "ACTIVE DIRECTORY", "AZURE", and "CUSTOM SEARCHES" (also highlighted with a red box). Under "User Saved Searches", there's a section for "Find users with password in description" with a trash can icon.

Однако на данный момент не существует способа загрузки пользовательских запросов с диска, как в BloodHound Legacy. Об этом несколько раз писали в BloodHound Slack, предлагая использовать API вместо этого.

Тем не менее, мы можем воспользоваться документацией по API, чтобы попытаться понять, что нам нужно сделать. Полезно отметить, что есть несколько различных вещей, которые мы можем использовать для чтения и размещения новых пользовательских запросов в системе.

The screenshot shows the Bloodhound Community Edition interface. At the top, there are links for 'EXPLORE' and 'GROUP MANAGEMENT'. Below that, a 'Cypher' section is visible. Under 'Cypher', there are two API endpoints listed: 'GET /api/v2/saved-queries' (Get all saved queries for the current user) and 'POST /api/v2/saved-queries' (Create a User saved query). The 'POST' endpoint is highlighted with a red box. The 'Create a new saved query' form is displayed below it, with fields for 'Name' and 'Description' (both currently empty), and a 'Request body' field containing the JSON string '"string"'. A 'Cancel' button is also present.

Веб-консоль Neo4j

Если нам понадобится прямой доступ к веб-консоли Neo4j, мы можем зайти в нее, перейдя по адресу <http://localhost:7474>. Для большинства сценариев в этом нет необходимости, если только нам не нужен прямой доступ к базе данных.

Это веб-интерфейс для базы данных Neo4j, который позволит нам выполнять необработанные запросы к шифру и просматривать данные. Зайдем, используя дефолтные учетные данные `neo4j:bloodhoundcommunityedition`

The screenshot shows the Neo4j Browser interface at the URL `localhost:7474/browser/`. The main area displays the message `neo4j$`. Below this, a blue bar states: 'To help make Neo4j Browser better we collect information on product usage. Review your [settings](#) at any time.' The main content area shows a terminal-like interface with the command `$:server connect`. To the left is a sidebar with icons for database, star, and play. The main content area has two columns: 'Connected to Neo4j' (with the message 'Nice to meet you.') and 'You are connected as user neo4j to neo4j://localhost:7687'. It also notes that 'Connection credentials are stored in your web browser.'

Мы можем поместить необработанные запросы в подсказку и увидеть результаты, а также все свойства для каждого возвращенного объекта.

The screenshot shows the Neo4j web interface. In the top bar, there is a command-line input field with the text: "neo4j\$ MATCH (n) WHERE "owned" in n.system_tags RETURN n". To the right of the input field are several icons: a blue play button, a star, a plus sign, and a downward arrow. On the left side, there is a vertical sidebar with four items: "Graph" (selected), "Table", "Text", and "Code". The main area displays a circular node representation for a user account. The center node is orange and labeled "SAMWE...". It has three gray segments radiating from it, each containing a small icon: a lock, an eye, and a network. To the right of the node is a table titled "Node properties". The table has two tabs at the top: "Base" (selected) and "User". The "User" tab is highlighted with a purple background. The table lists the following properties:

Property	Value
objectid	S-1-5-21-583573853-2369275934-2095187787-1119
passwordnotreqd	false
pwdlastset	1708723691.0
pwdneverexpires	true
samaccountname	samwell.tarly
sensitive	false
serviceprincipalname	
sidhistory	
system_tags	owned
trustedtoauth	false
unconstraineddelegation	false
whencreated	1708723691.0

Это поможет нам создавать и отлаживать пользовательские запросы для BHCE. Опять же, доступ к веб-консоли Neo4j, как правило, не нужен в большинстве сценариев, но его полезно иметь в качестве запасного варианта на случай необходимости.

Очистка данных §

В большинстве консалтинговых сред нам необходимо очистить данные BloodHound, чтобы сохранить разделение данных между клиентами. В последнем выпуске BloodHound 5.8.0 это можно сделать в графическом интерфейсе.

The screenshot shows the BloodHound Community Edition web interface. At the top, there is a logo for "BLOODHOUND COMMUNITY EDITION" and navigation links for "EXPLORE" and "GROUP MANAGEMENT". On the left side, there is a sidebar with several options: "Data Collection", "File Ingest", "Data Quality", "Database Management" (which is highlighted with a red border), "Users", "Manage Users", "Authentication", "SAML Configuration", "Configuration", and "Early Access Features". The main content area is titled "Clear BloodHound Data" and contains the following text: "Manage your BloodHound data. Select from the options below which data should be deleted." Below this is a warning message: "Caution: This change is irreversible and will delete data from your environment." followed by a list of checkboxes for data deletion options. The first option, "Collected graph data (all nodes and edges)", is checked. The other options are: "Custom High Value selectors", "All asset group selectors", "File ingest log history", and "Data quality history". At the bottom of the page is a blue "PROCEED" button.

До версии 5.8.0 нам нужно было удалить том, используемый BloodHound, чтобы очистить его. Я оставил здесь эти инструкции на случай, если кому-то еще понадобится знать, как это сделать, это также описано в [выпуске #107 для BHCE](#).

Для начала давайте перечислим тома, используемые докером:

```
docker volume ls
```

В этих данных мы видим два тома, связанных с BloodHound.

```
(root㉿kali)-[~/home/kali]
# docker volume ls
DRIVER      VOLUME NAME
local      0bec636fcceb115a2e865b24249113e03407855379b56af30a7377
local      04d0f228a228e5bedad6d63de2f5a88d795698fc5c087a1237de
local      5a2cf0e1edc20804937b59d407db0fc94f2db0d7f045f4844cba1
local      5a71769fd758207aaafb180b85959e11a73ed3e96096c36a2e3ad
local      6de83069d9867a2ab02739fd70d38f149059be478e18a5a36176d
local      20bb3c70240646090ecb885d63fe81e4ed9898f9e40dfe7424029
local      34a397b40a3b970736190cf7b417f7e9aae66378a90723d2d64eb
local      37b87e534ad39598fab68b6669ad6655a02880b7cb941bf797843
local      57a1bf41fb1c2ee6ce9c1752542386d85689ea6d6e62615cf0923
local      67a4cf63f498255eaddee2712597e4a079a33ac2bdd51e3f76e295
local      72ec8c33d35e6e9c8f291103ed9a716756c10396de5c97b4b659d
local      91c52ff0a8618d597973e7162f87fa73d7ac39e335b0f91fe1fdc
local      315c5bb9289d1405a42bff84ead724f5d15d82827d2b15d8a34da
local      348f1651ad97e50a26570770546ad15dea673ea04779f4a121f4e
local      486ec9e8c70e6dfae2e00c3af9c295b9b65bf6b04b6dd46251c47
local      582d1b5a6218ac47a315e924f05addebb4df1f9d8c7a65eb29c09
local      885f621d69afdbd61f0c10eb511d58b9c1587f1ecda555e91b36b
local      1168a0be5bd55c4c6013a03a83aa898d65b25e813f9d15190bb64
local      05071f02181eae76df10912c6d5cea169f9da6e0da43a86005375
local      84835b24e2bc48192b2411b25799c7f02244d2803eef582c5a7ad
local      b3bfcc14ef15f9545a76dad16113edc3b6aa026fffc39189c3c53
local      be295837f4a65272f2306b441efffc6f6e4ed0a06d9f8931e86330
local      bloodhound_neo4j-data
local      bloodhound_postgres-data
local      c21d6819ded3e50c2238c265b58ef367a48b2d28a6f7eac71f766
local      c63cbe60e6939a7a371bfabea7b85267bda51942b9b1d4d90597d
local      ccecc70bb5f0bd4acacc7fc5613ccda7846cc47bcbdf0f3cc91d
local      cd20c104ddae180fac7968eee004afdb3533b14f008ecb968086a
```

Определение тома Neo4j, используемого BloodHound

Чтобы удалить эти данные и получить свежий экземпляр данных BloodHound, нам нужно удалить этот том. Обратите внимание, что том `bloodhound_postgres-data` используется для входа в графический интерфейс и веб-приложение. Обычно его не нужно удалять, если только вы не хотите все сбросить.

Давайте удалим этот том, чтобы сбросить данные. Мы используем приведенную ниже команду, чтобы попросить docker удалить этот том.

Чтобы упростить задачу, мы можем выполнить все это одной командой.

```
docker volume rm $(docker volume ls -q | grep neo4j-data)
```

Удалить все §

Если вы хотите удалить все и получить свежую копию всех контейнеров, томов и конфигураций, следуйте этим инструкциям. Предполагается, что файл `docker-compose.yml` находится в рабочей директории.

Сначала нам нужно снести контейнеры и удалить тома с помощью следующей команды:

```
docker-compose down -v
```

После этого мы можем извлечь свежую копию контейнеров с помощью команды `pull`.

```
docker-compose pull
```

Остановка контейнеров §

Остановить контейнеры очень просто: мы можем просто попросить `docker` остановить их.

```
docker compose stop
```

```
~/dev/bloodhound 1m 22s
[› docker compose stop
[+] Stopping 3/3
✓ Container bloodhound-bloodhound-1     Stopped
✓ Container bloodhound-graph-db-1        Stopped
✓ Container bloodhound-app-db-1          Stopped
```

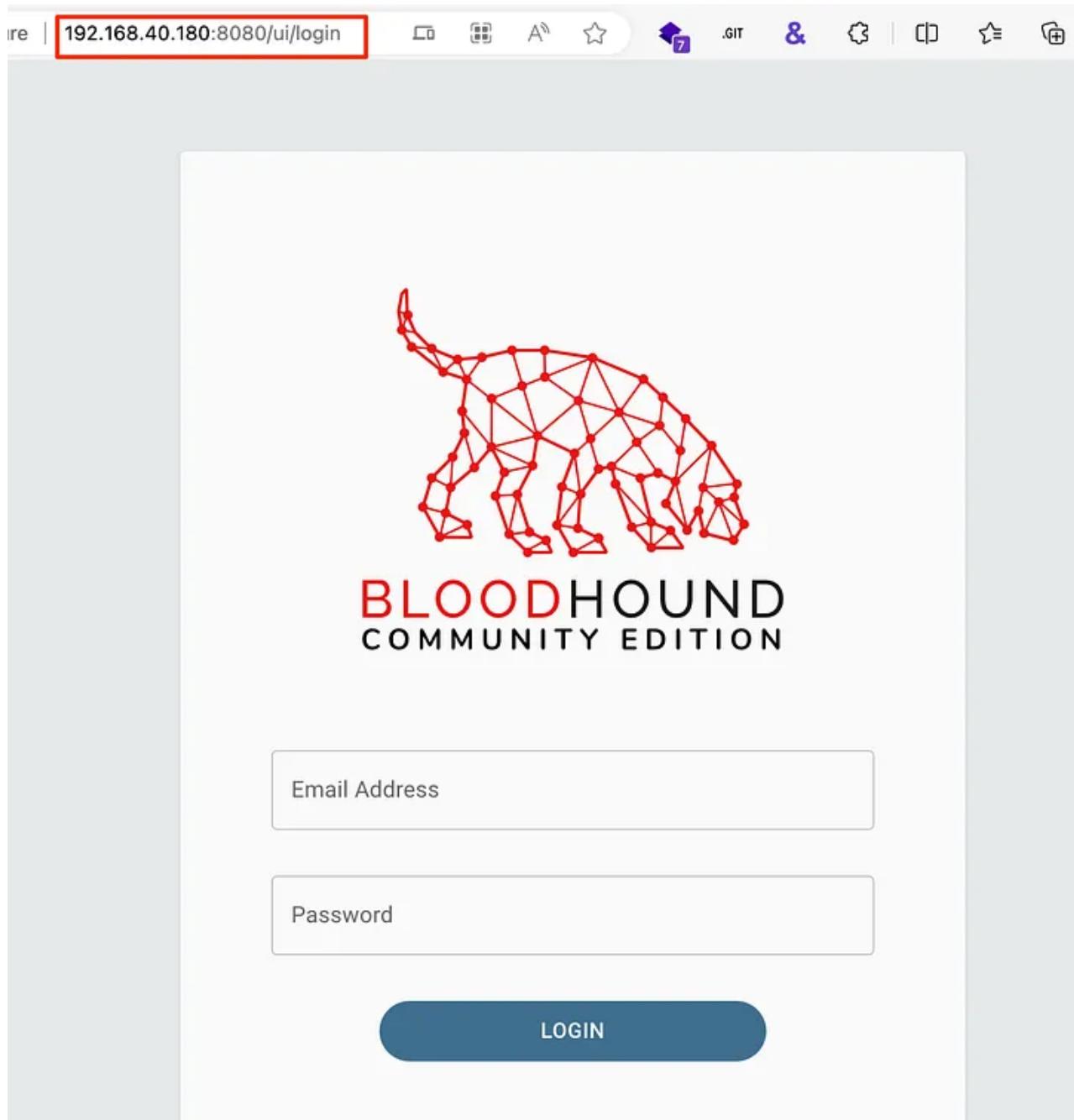
Предоставление другим пользователям §

Во многих средах мы захотим поделиться нашим экземпляром ВНСЕ с другими, чтобы все они могли подключиться к одному экземпляру. По умолчанию ВНСЕ открыт только для `localhost`. Чтобы добиться этого, нам нужно изменить файл `docker-compose.yml`, чтобы открыть его для других.

Изменив приведенную ниже строку в службе `bloodhound`, мы можем указать Docker Compose открывать графический интерфейс для интерфейсов, отличных от `localhost`. Это очень удобно, если мы планируем использовать один сервер для одновременного использования многими пентестерами.

```
bloodhound:
  image: docker.io/specterops/bloodhound:${BLOODHOUND_TAG:-latest}
  environment:
    - bhe_disable_cypher_qc=${bhe_disable_cypher_qc:-false}
    - bhe_database_connection=user=${POSTGRES_USER:-bloodhound} password=${POSTGRES_PASSWORD:-bloodhoundcommunityedit}
    - bhe_neo4j_connection=neo4j://${NEO4J_USER:-neo4j}:${NEO4J_SECRET:-bloodhoundcommunityedition}@graph-db:7687/
    ### Add additional environment variables you wish to use here.
    ### For common configuration options that you might want to use environment variables for, see `*.env.example`
    ### example: bhe_database_connection=${bhe_database_connection}
    ### The left side is the environment variable you're setting for bloodhound, the variable on the right in `${}`'
    ### is the variable available outside of Docker
  ports:
    ### Default to localhost to prevent accidental publishing of the service to your outer networks
    ### These can be modified by your .env file or by setting the environment variables in your Docker host OS
    - ${BLOODHOUND_HOST:-127.0.0.1}:${BLOODHOUND_PORT:-8080}:8080
    ### Uncomment to use your own bloodhound.config.json to configure the application
  # volumes:
  #   - ./bloodhound.config.json:/bloodhound.config.json:ro
  depends_on:
    app-db:
      condition: service_healthy
    graph-db:
      condition: service_healthy
```

Если мы хотим открыть его для всех интерфейсов, измените параметр **BLOODHOUND_HOST** на **0.0.0.0**. Обратите внимание, что если у вас белый IP, ваш сервер будет доступен в Интернете! Обычно лучше привязать его только к VPN-интерфейсу, например, к интерфейсу WireGuard, чтобы ограничить доступ.



Ускорение процесса §

Если вы знаете меня, то знаете, что я люблю ускорять процессы и делать вещи более качественными. Так как же использовать это, чтобы быстрее справляться с задачами?

AD-Miner §

Инструментарий AD-Miner использует данные в Neo4j для поиска множества известных рисков и путей эскалации, а затем представляет эти результаты в HTML-файле с общей оценкой.

Мы можем запустить его, сначала установив инструмент [AD-Miner](#) с помощью pipx, а затем предоставив информацию для подключения к базе данных Neo4j - эти учетные данные хранятся в файле `docker-compose.yml`. По умолчанию это

`neo4j:bloodhoundcommunityedition`. Обратите внимание, что это пароль к базе данных Neo4j, а не к графическому интерфейсу BHCE!

Установка с помощью pipx

```
pipx install 'git+https://github.com/Mazars-Tech/AD_Miner.git'
```

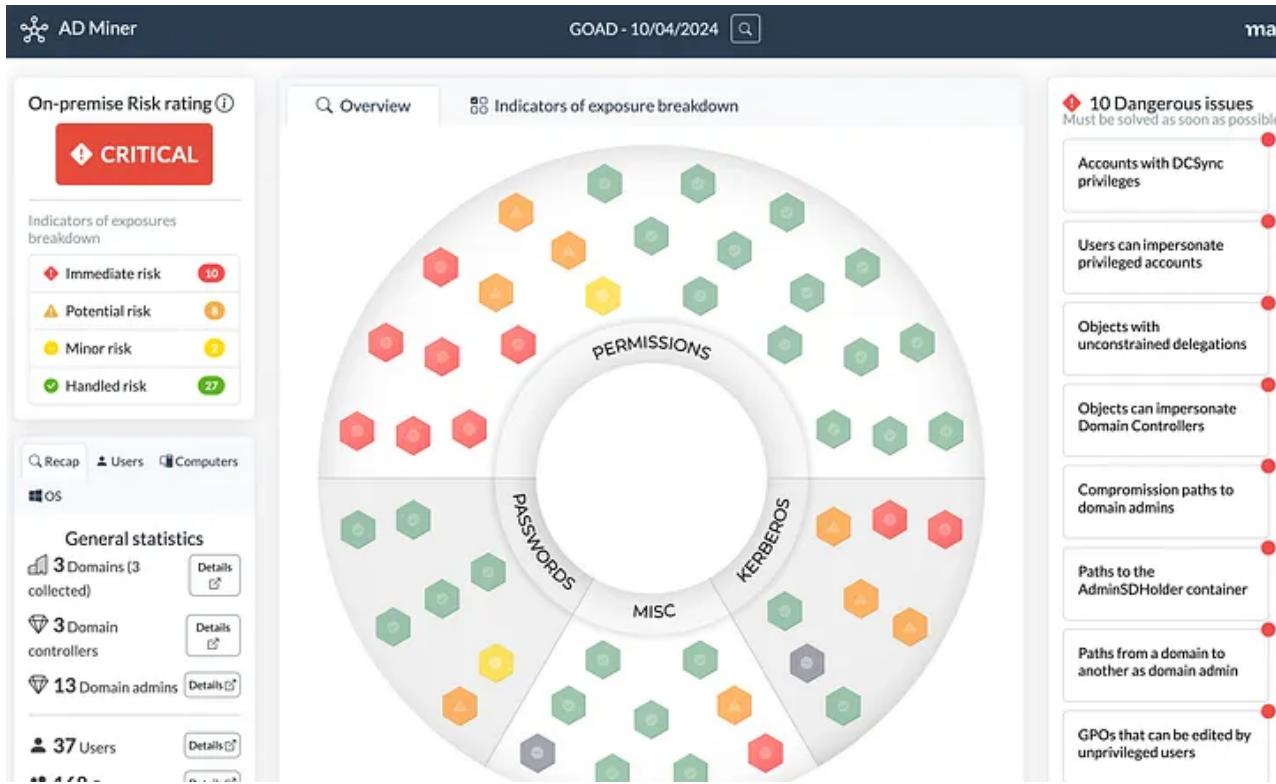
Создание отчета AD-Miner с использованием учетных данных BHCE по умолчанию

```
AD-miner -u neo4j -p bloodhoundcommunityedition -cf GOAD
```

```
~/dev/bloodhound
> AD-miner -u neo4j -p bloodhoundcommunityedition -cf GOAD
[+]MigrationData : 1 | Computer : 5 | ADLocalGroup : 1 | Group : 169 | User : 49
[1/145] [+]Requesting : Checking if Graph Data Science neo4j plugin is installed
[+]GDS plugin not installed.
[+]Not using exploitability for paths computation.
[-]Done in 0.04 s - 1 objects
[2/145] [+]Requesting : Delete orphan objects that have no labels
[-]Done in 0.03 s - 0 objects
[3/145] [+]Requesting : Clean AD Miner custom attributes
[-]Done in 0.04 s - 0 objects
[4/145] [+]Requesting : Delete objects for which SID could not resolved
[-]Done in 0.03 s - 0 objects
[5/145] [+]Requesting : Set domain names to upper case when not the case
```

Это может занять много времени в зависимости от размера среды - в некоторых доменах мне приходилось ждать более 2 часов! После завершения процесса файлы будут находиться в папке `render_GOAD`, используя метку, указанную в ключе `-cf`. Мы можем найти этот HTML-файл в созданной папке.

```
~/dev/bloodhound/render_GOAD
> ll
total 24
drwxr-xr-x 10 chris staff 320B Apr 11 06:42 .
drwxr-xr-x 19 chris staff 608B Apr 11 06:41 ..
drwxr-xr-x 13 chris staff 416B Apr 11 06:15 assets
drwxr-xr-x 14 chris staff 448B Apr 11 06:15 css
drwxr-xr-x 2 chris staff 64B Apr 11 06:41 csv
-rw-r--r-- 1 chris staff 5.2K Apr 11 06:42 data_GOAD_20240410.json
drwxr-xr-x 173 chris staff 5.4K Apr 11 06:42 html
drwxr-xr-x 55 chris staff 1.7K Apr 11 06:15 icons
-rw-r--r-- 1 chris staff 59B Apr 11 06:41 index.html
drwxr-xr-x 18 chris staff 576B Apr 11 06:41 js
```



Как это здорово! Теперь у нас есть интерактивная приборная панель, которая позволяет обратить наше внимание на самые важные ошибки в конфигурации.

AD-Miner - это фантастический инструмент для дополнительного анализа и обогащения данных BloodHound - обязательно ознакомьтесь с ним!

References §

Tags:

[#ad](#) [#bloodhound](#)