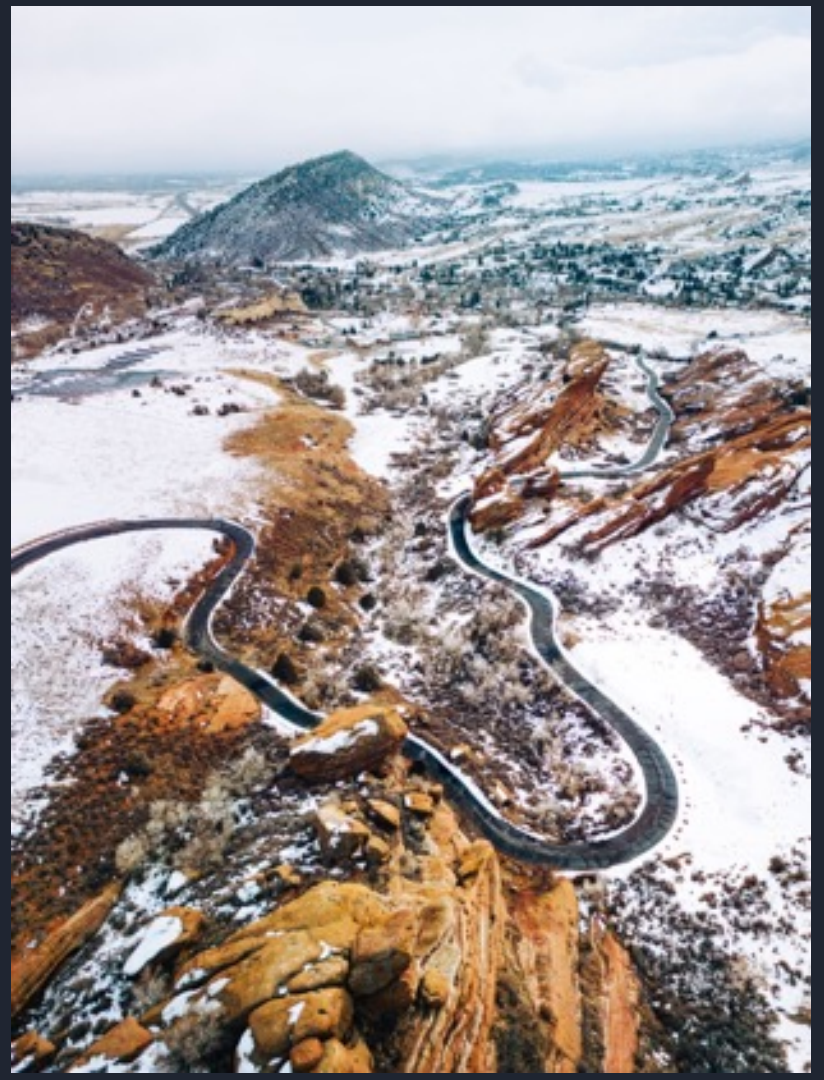
The Truth is Out There

Solving the mysteries of lateral movement paths by feeding logs to the hound

What can you expect today ?

- AI assisted programming, learning a language
- Blue team capability enhancements
- Near-real time path discovery and prediction
- **Tool drop!**



Olaf Hartong

Detection Engineer and Security Researcher

- Purple teaming, Threat hunting
- IR and Compromise assessments

Former documentary photographer
Father of 2 boys
“I like **warm hugs**”

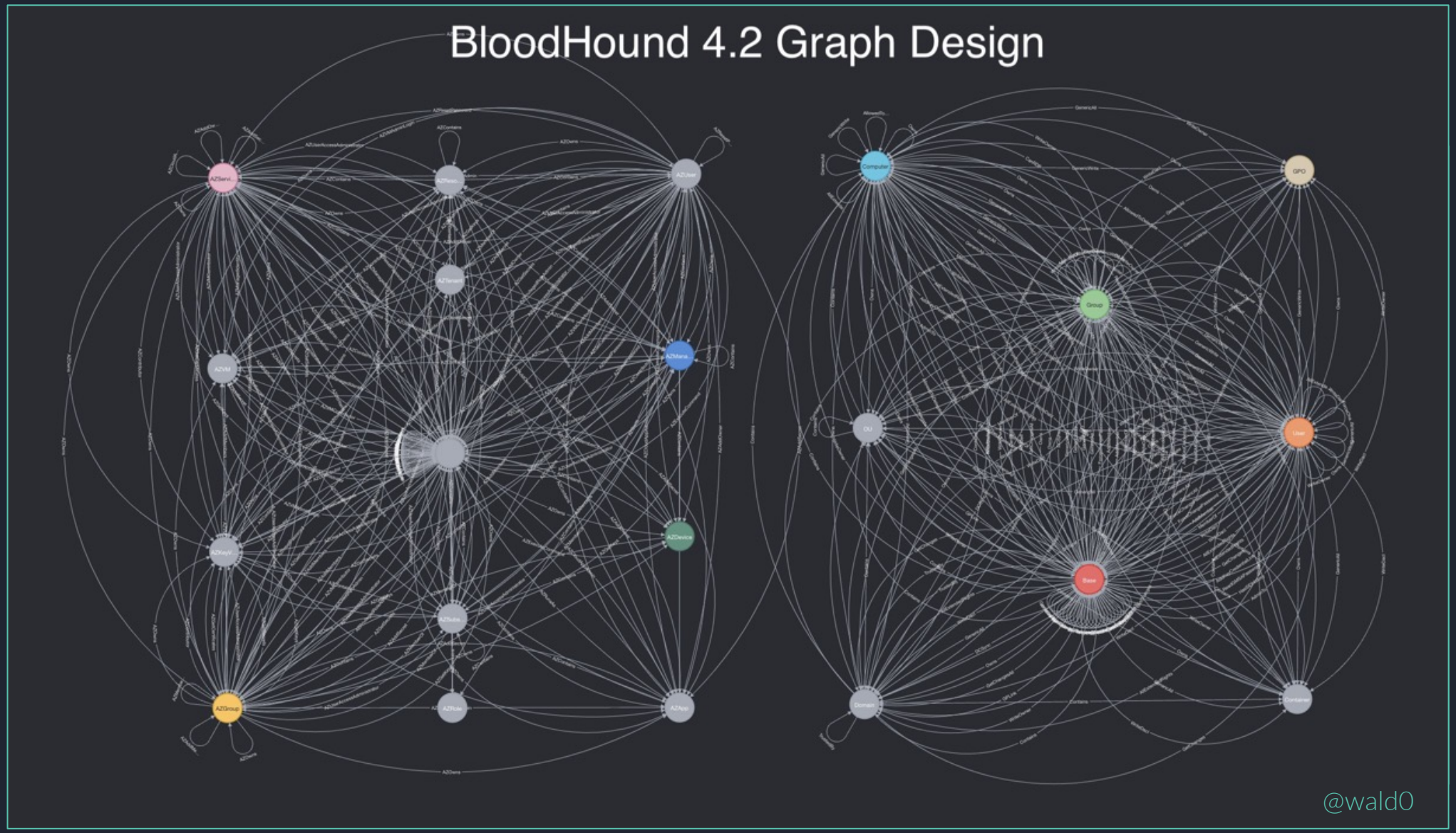
@olafhartong
github.com/olafhartong
olaf@falconforce.nl
olafhartong.nl / falconforce.nl

This has always bugged me

“DEFENDERS THINK IN LISTS.
ATTACKERS THINK IN GRAPHS.
AS LONG AS THIS IS TRUE, ATTACKERS WIN”

John Lambert, 2015

A year later, BloodHound was released



Path based detection and enrichment

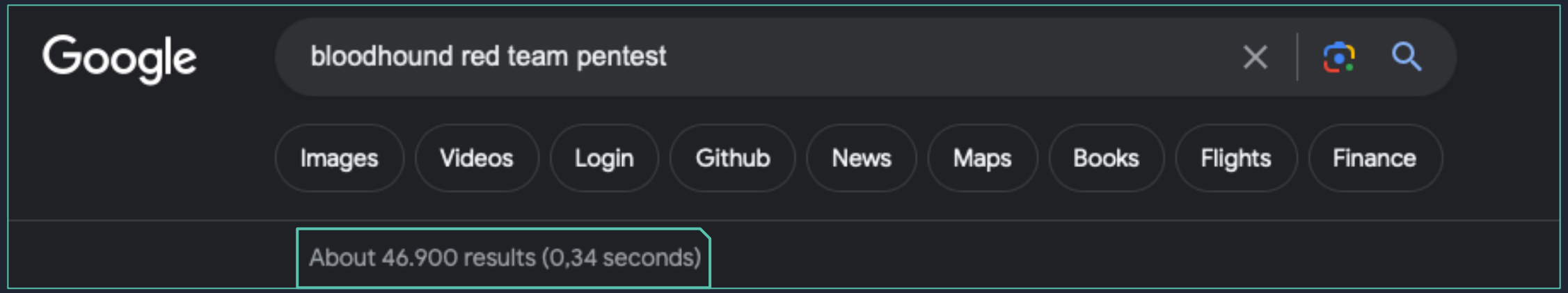
Most red teams rely on BloodHound in their operations for uncovering lateral movement paths.

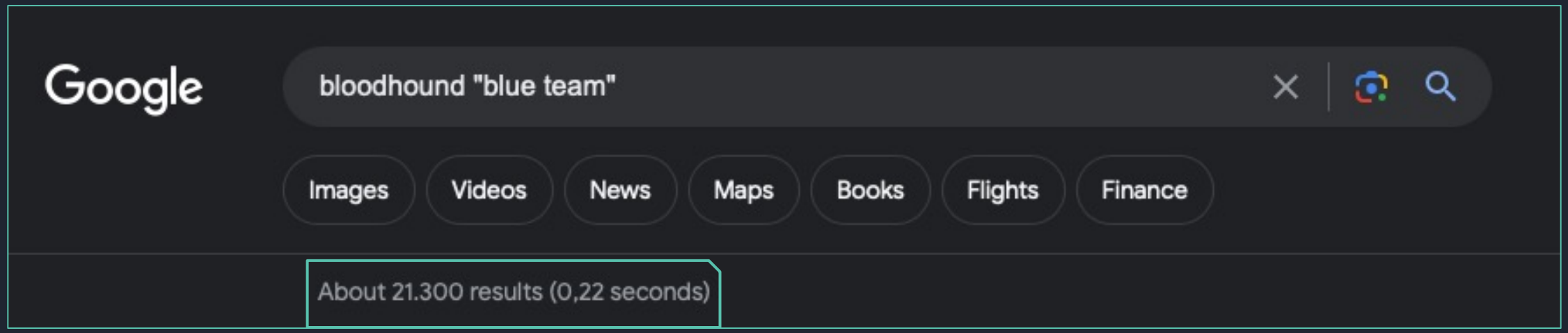
Sadly, we don't see many detection teams leverage it for mitigations and detections, not even talking about enrichment or alert investigations.

In some cases we see it being used, but only a hand-full times a year, in a very manual way.

There must be a better way.

It’s not the lack of documentation

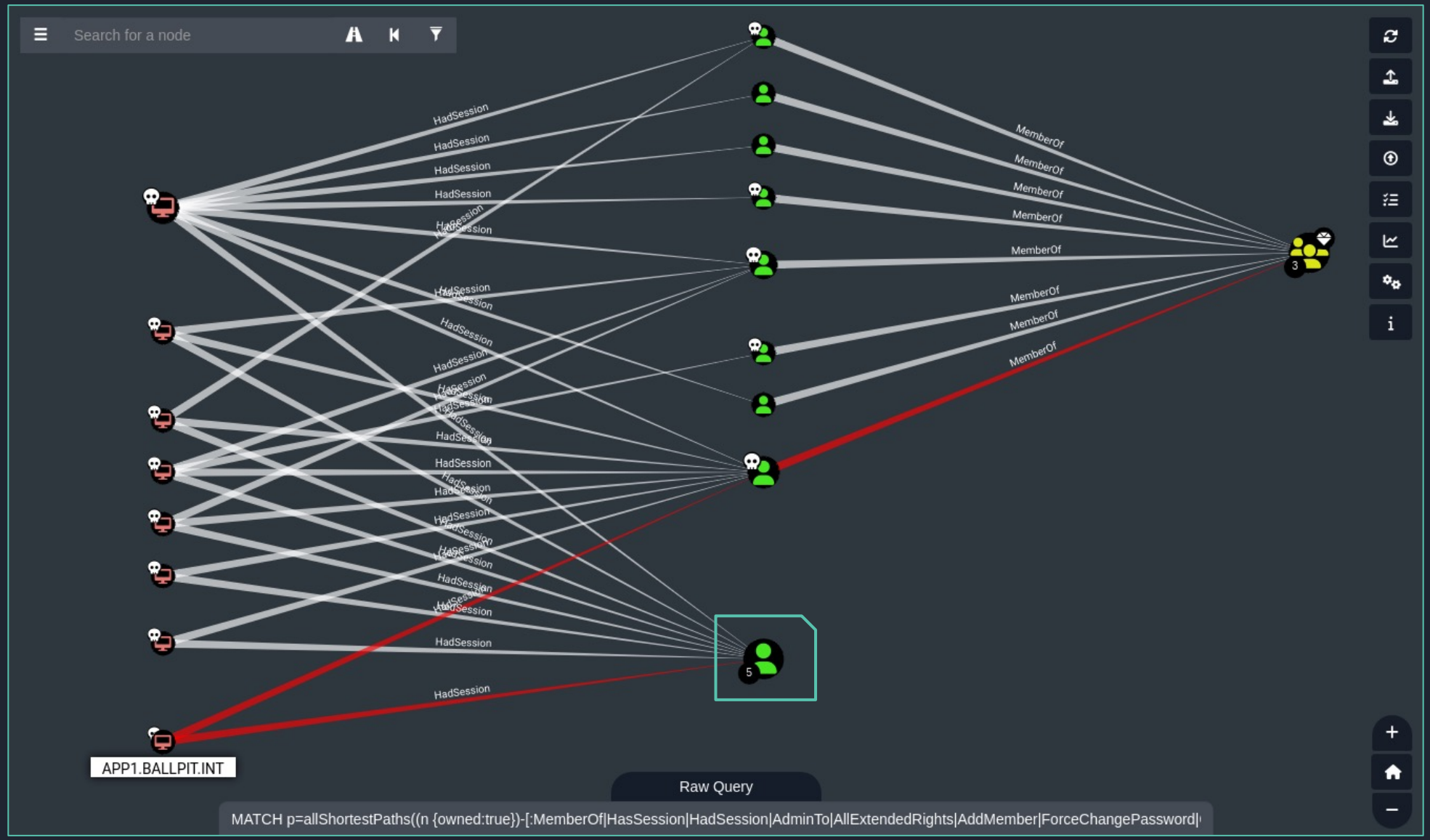




BloodHound challenges for red and blue

- Frequently scanning adds complexity and sometimes friction between teams
- Hard to reach all subnets for session data
- It is always a moment in time snapshot
- Local user and group modifications are mostly invisible
- How do you know a session is still active? Or if there are more?

Why are sessions so important?



The power of ~~*love*~~ logs

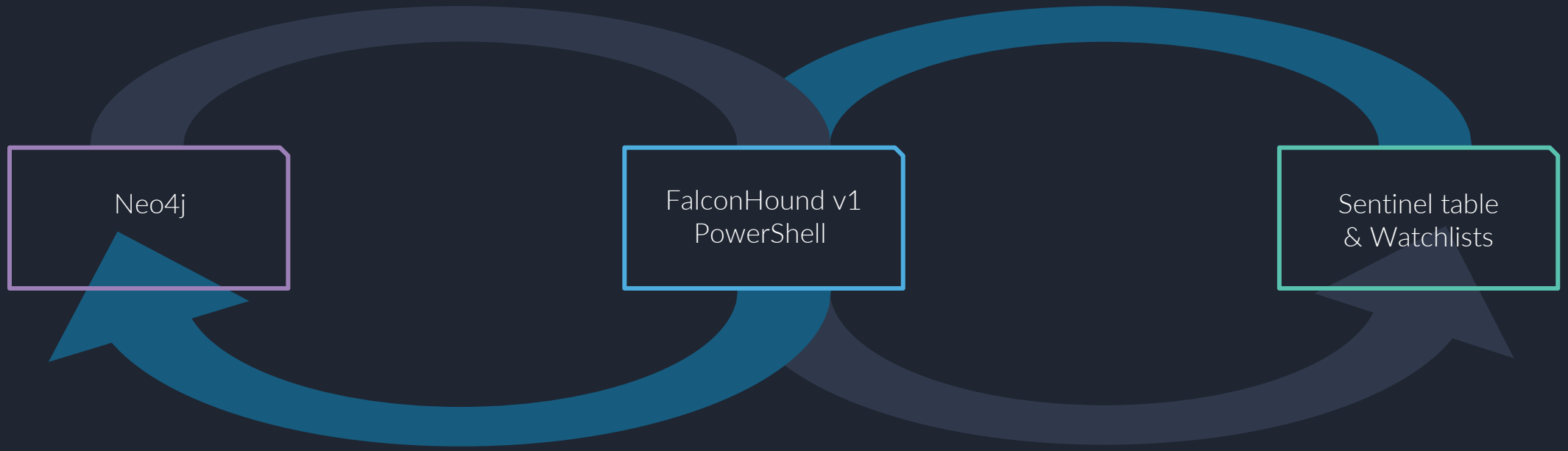
As a blue team, we already have this information!

And with that.... much more..

The power of ~~love~~ *logs*

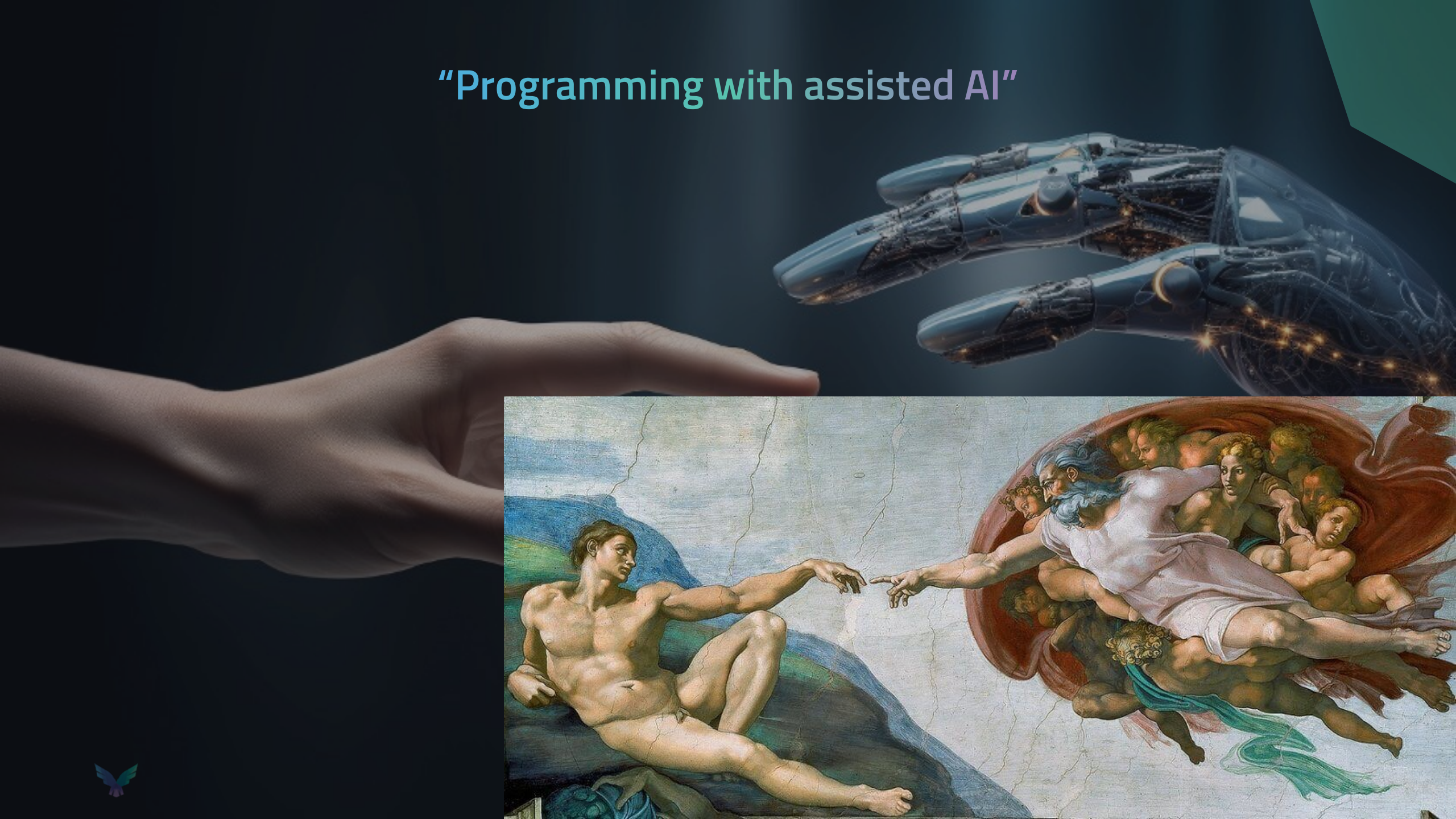
- User logons / unlocks
- User / group / object modifications
- Alerts on entities
- Role assignments
- All these events and much more in near-realtime, *all the time*!

FalconHound POV



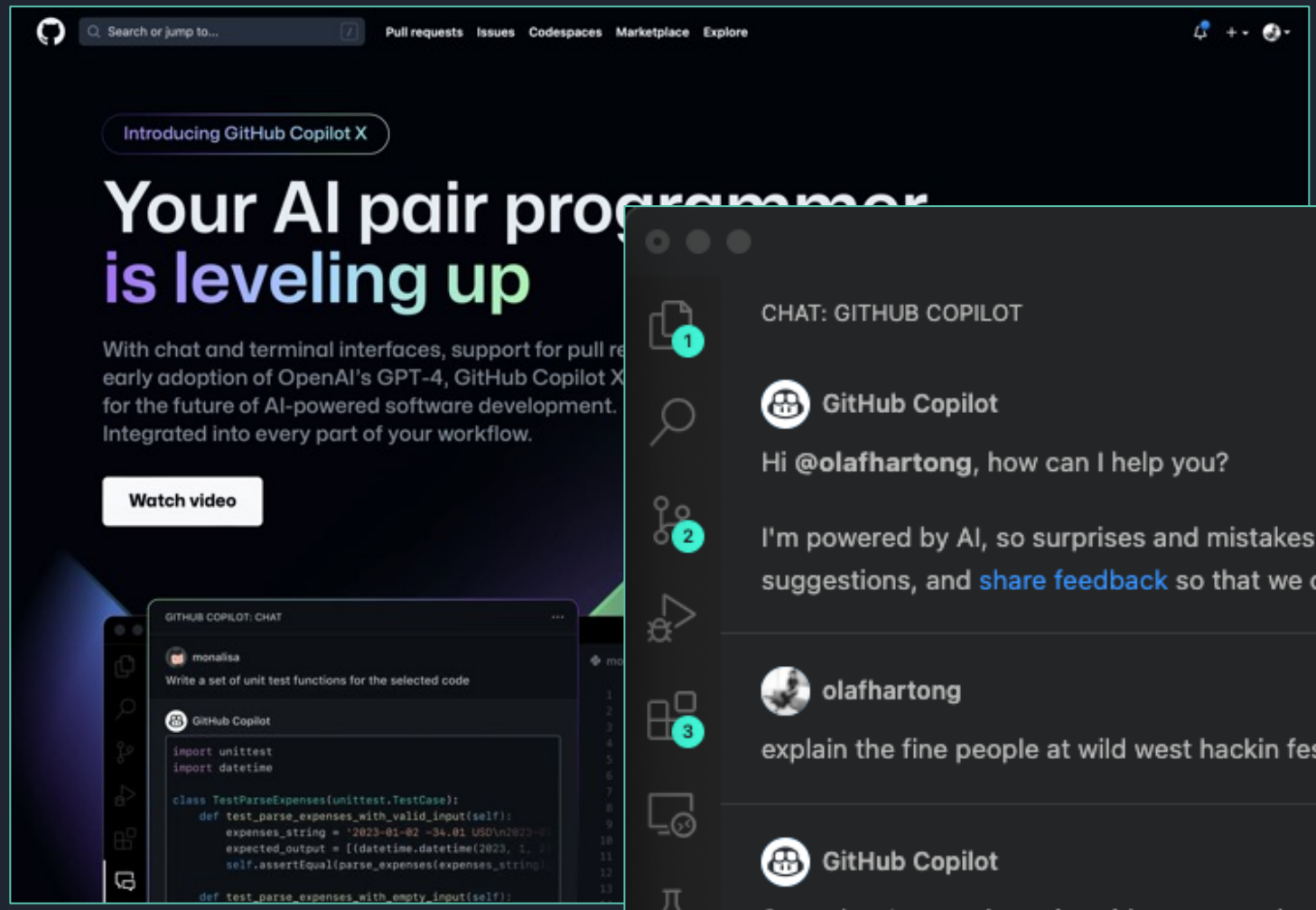
Thanks to @sadprocessor

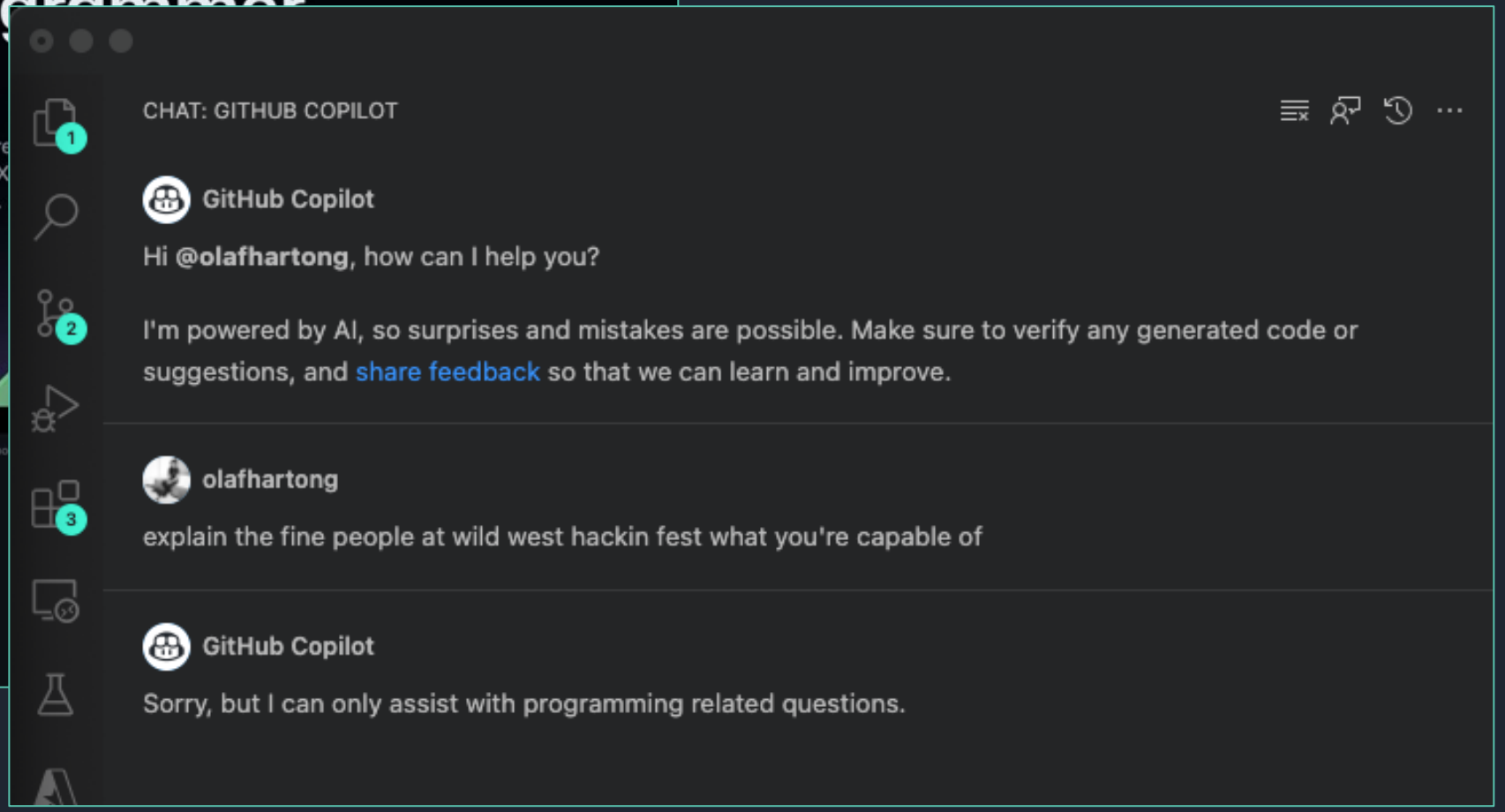
“Programming with assisted AI”



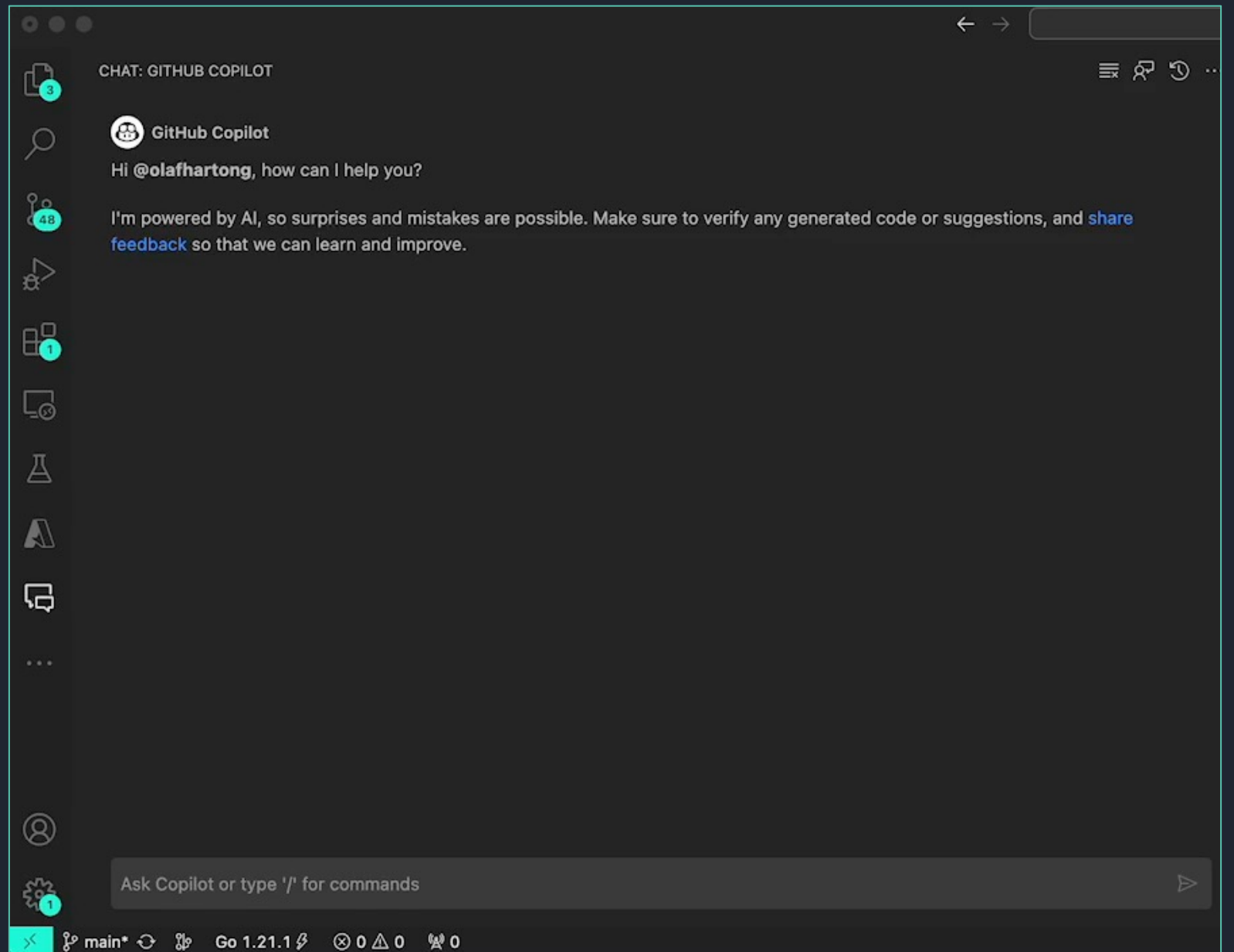
THAT WAS DARK...

Programming with assisted AI

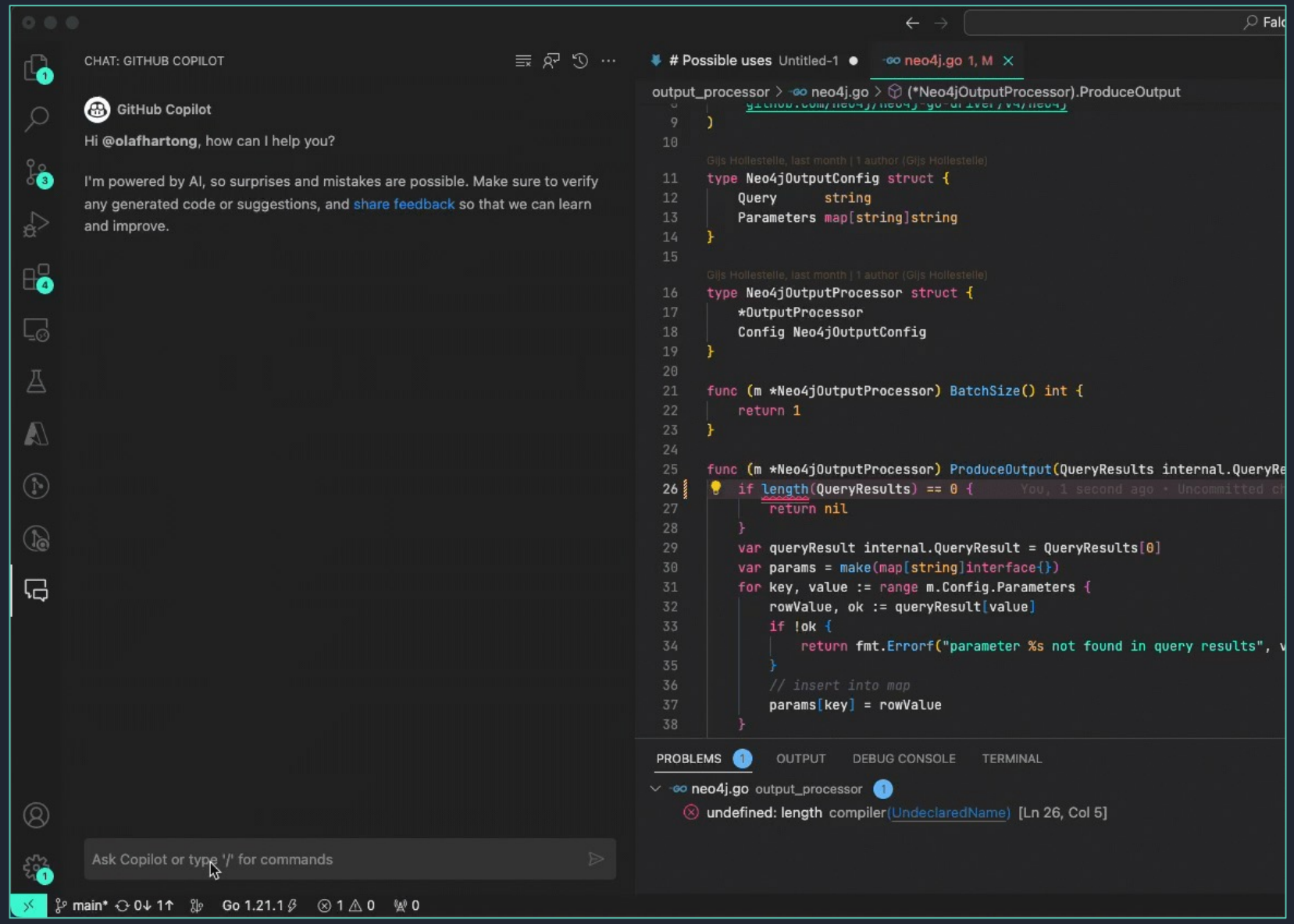




Create building blocks



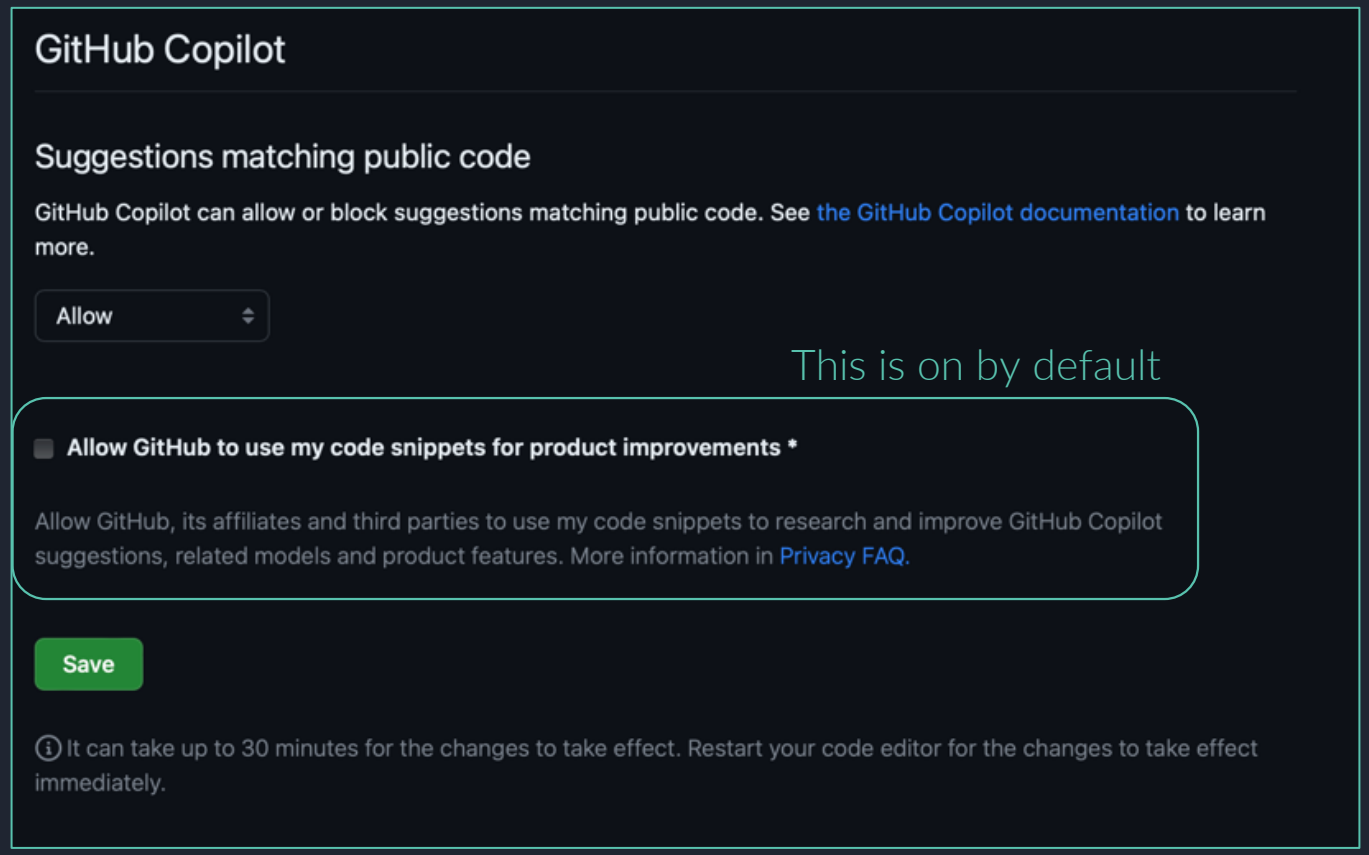
Create building blocks or fix errors



It’s great to learn a language and support you in dev tasks

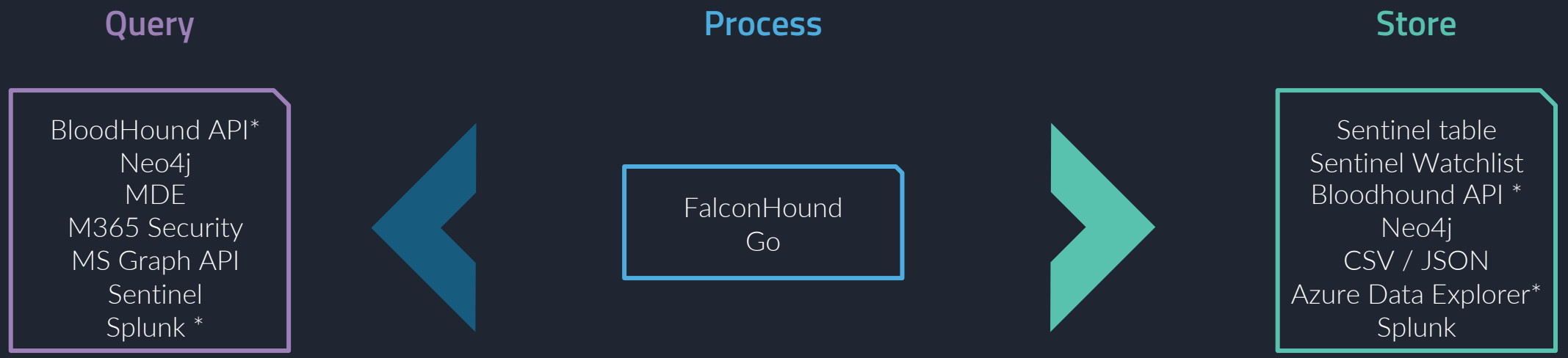
- While this is amazingly useful it will make mistakes… A LOT of them..
- When you really know what you want and prompt it well it provides quite good results.
- Use it only for small blocks, it is aware of your open file(s)
- You still need to architect your project well
- In the Azure case its knowledge is based on libraries that are deprecated

Also be careful with those AI assistants



FalconHound build 2

Redesigned for many-to-many destinations

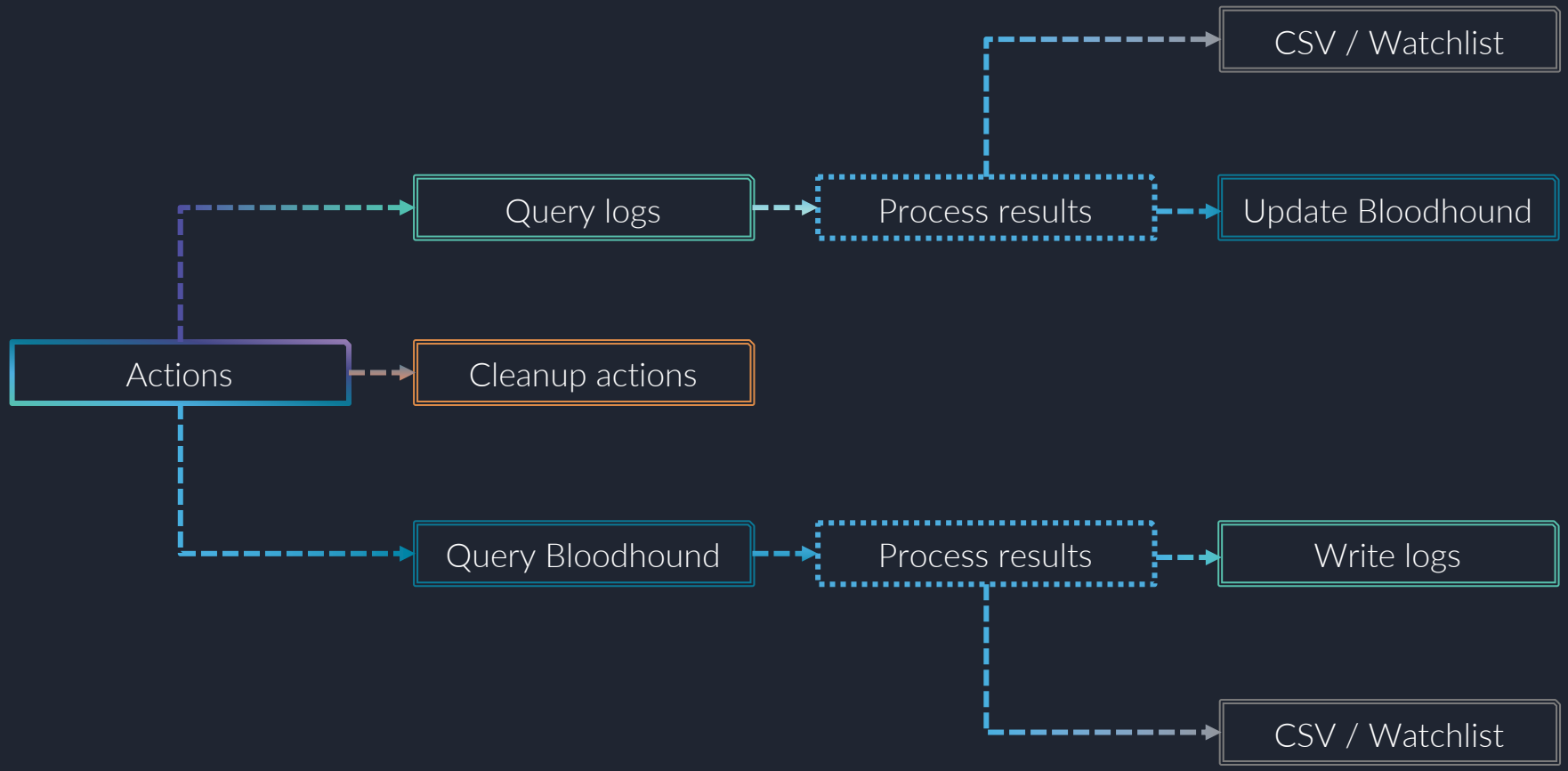


* Under development, more connectors to follow

Built for easy implementation

- Connects to APIs
- Builds for all major operating systems
- Update the config file with your keys
- Define the outputs you want per action
- Schedule the agent to run every 5/10/15 mins, whichever you prefer and your env can take.

Process flow – simplified example



Some example actions

- Keep the session information in BloodHound up to date, without scanning!
- Mark machines/users with alerts as Owned
- Repeatedly query for paths to high value targets, for instance from owned to sensitive resource
- Add new users or group modifications
- Calculate new paths or chokepoints based on this information

YAML based configuration

```yaml
Name:                       # Choose a name that describes the action
ID:                         # Unique ID (short version of the name, no spaces, will end up in the logs)
Description:                # Short description (one-liner)
Author: FalconHound         # Optional: Author of the action
Version: '1.0'              # Optional: Version of the action
Info: |-                    # Optional: Additional information about the action
Active: true                # Enable to run this action
Debug: true                 # Enable to see verbose results on the console
SourcePlatform: MDE         # Supported sources; Sentinel, Neo4j, MDE, Graph
Query: |                    # Query to run against the source platform
  -query here-
Targets:                    # Targets are the platforms that this action will push to (CSV, Neo4j, Sentinel, Wachlist, Splunk)
  - Name: CSV
    Enabled: true
    Path: output/get_sessions_mde.csv
  - Name: Sentinel
    Enabled: true
  - Name: Splunk
    Enabled: true
  - Name: Neo4j
    Enabled: true
    Query: |
      MATCH (x:Computer {name:$device_name}) SET c.exploitable = true, c.exploits = $cve_ids
    Parameters:
      device_name: DeviceName
      cve_ids: CveIds
  - Name: Watchlist
    Enabled: true
    WatchlistName: FH_MDE_Exploitable_Machines
    DisplayName: MDE Exploitable Machines
    SearchKey: DeviceName
    Overwrite: true       # Overwrite the watchlist with the query results, when false it will append the results to the watchlist
```

2 Levels of configuration

APIs in the app config (or keyvault)

Outputs in each action config

```
  with the timestamp of the first logon event.
Active: true                                    # Enable to run this action
Debug: true                                     # Enable to see query results in the console
SourcePlatform: Sentinel                        # Sentinel, Watchlist, Neo4j, CSV, MDE, Graph, Splunk
```

```
Targets:                                        # Targets are the platforms that this action will push to (CSV, Neo4j, Sentinel, Wachlist, Slack, Teams,
  - Name: Neo4j
    Enabled: true
    Query: |
      WITH toUpper($Computer) as Computer, toUpper($TargetUserSid) as TargetUserSid, $Timestamp as Timestamp
      MATCH (x:Computer {name:Computer}) MATCH (y:User {objectid:TargetUserSid}) MERGE (x)-[r:HasSession]->(y) SET r.since=Timestamp SET r.source='falconhound'
    Parameters:
      Computer: Computer
      TargetUserSid: TargetUserSid
      Timestamp: Timestamp
```

Some practical blue uses

- Generate watchlists for users and devices with a known, harder to mitigate path
- Dynamically raise the severity of alerts for these entities
- Generate alerts based on new paths found in Bloodhound
- Tracking uncommon sessions across a path. Lateral Movement indicator
- Uncover risky resources or unintentional paths

Demo

```
                  %????????;;'
                %??++;;;;??????'''+
              ++++%??%%%???;;???''+;
           ???++++;??????%#%?????;;;'';
           +;;+?????????.'#.......?%%??
         +;;;+???????????.??.;%##%%%%%??
      ##??;;;???';;???????%%%#####%%%%#??.
      #?;;;??;;;';;?????+?+?%%... %???????'
     +;;';?'';'''';;;;;;;???????.. +''.????'
    ;##+';;;';';;''''';'''''........   .'++'
     #+;;;';''''''';;;''''........'       +'
     ';'';;';;;;;';;..''....'.''''.      ;
       ;;.''''.'''''..'....''''''..
        . '''..'''''''...'''....'..
          ''''''.''''''..''''......    FalconForce Sentry
           ' '''...'''''.........      FalconHound v0.80
               ;''.'''.........'.      ------------------
                 ''''.......'?         Usage: FalconHound -[options]
                    '''.''''           https://github.com/FalconForceTeam/FalconHound
                      ''
Usage: FalconHound -[options]
Options:
  -actionlist
        Get a list of all enabled actions, use in combination with -go
  -actionsdir string
        Path to the actions directory (default "actions/")
  -config string
        config file name (default "config.yml")
  -go
        Run all actions in the actions directory
  -help
        Print this help message
  -ids string
        comma separated list of action IDs to run
  -keyvault
        Use the keyvault specified in the config for secrets
```

Demo

```
olafhartong ~/0xFF-Code/Sentry/FalconHound main 1.21.1 07:22:39 ./falconhound -actionlist -go
```

Demo

```
olafhartong ~/0xFF-Code/Sentry/FalconHound main 1.21.1 07:39:33 ./falconhound -go -config configs/config-customer.yml -ids MDE_Exploitable_Hosts,N4
_Exploitable_Device_to_HighValue
```

Debug mode

```
Name: Hosts with public available code for CVE
ID: MDE_Exploitable_Hosts
Description: Find Machines in MDE with exploitable software
Author: FalconHound
Version: '0.1'
Info: |-
Active: true                # Enable to run this action
```

```
Name: Hosts with public available code for CVE
ID: MDE_Exploitable_Hosts
Description: Find Machines in MDE with exploitable software
Author: FalconHound
Version: '0.1'
Info: |-
Active: true                # Enable to run this action
Debug: true                 # Enable to see query results in the console      You, 6 seconds ago • Uncommitted changes
SourcePlatform: MDE         # Sentinel, Watchlist, Neo4j, CSV, MDE, Graph, Splunk
Query: |
    let Timestamp = datetime(now)-1d;
    let vulnerablesoftware=DeviceTvmSoftwareVulnerabilities
    | where isempty(CveId)==false;
```

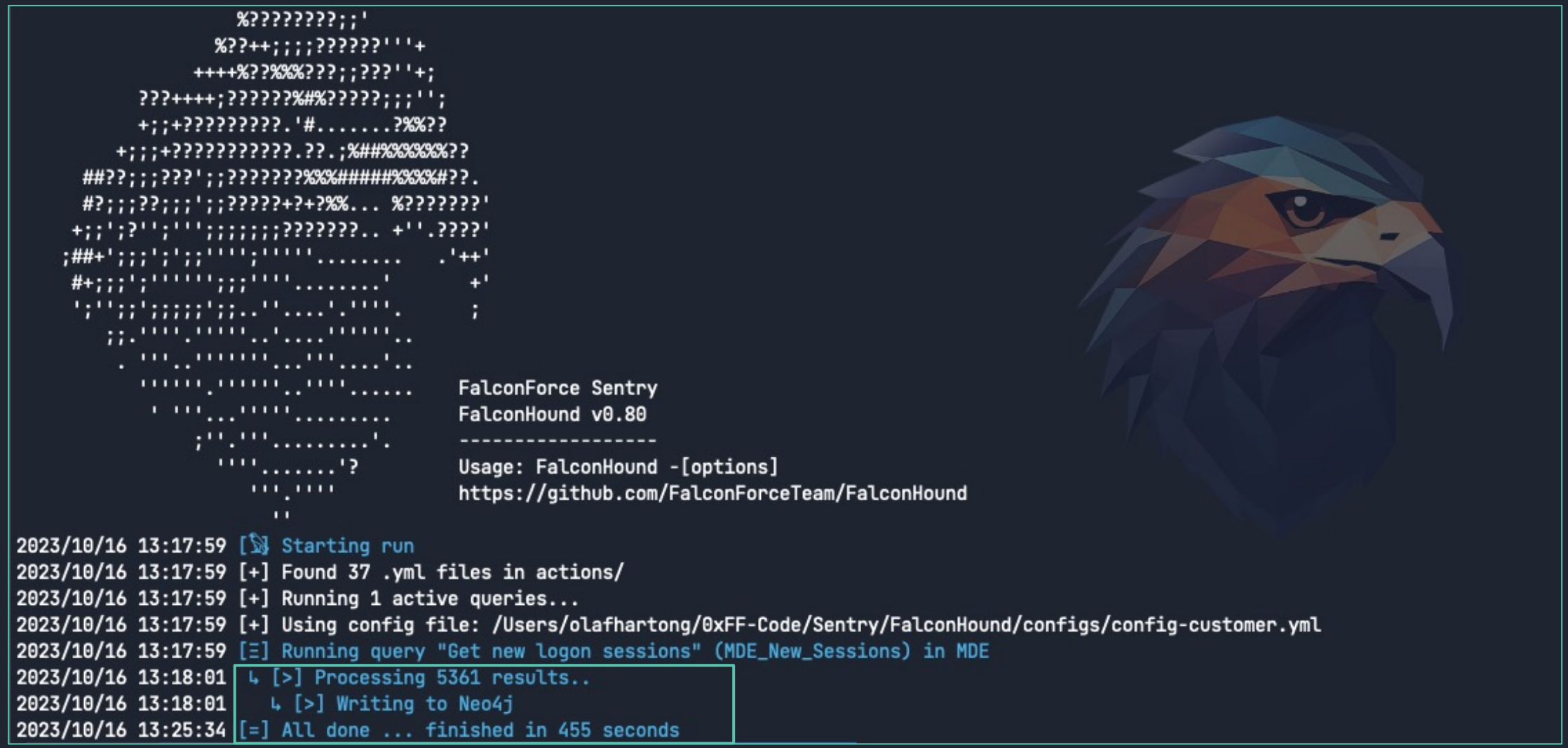
```
targets:
  - Name: Neo4j
    Enabled: true
    Query: |
      MATCH (c:Computer {name:$DeviceName}) SET c.exploitable = true, c.exploits = $CveIds
    Parameters:
      DeviceName: DeviceName
      CveIds: CveIds
```

Debug mode

-zsh

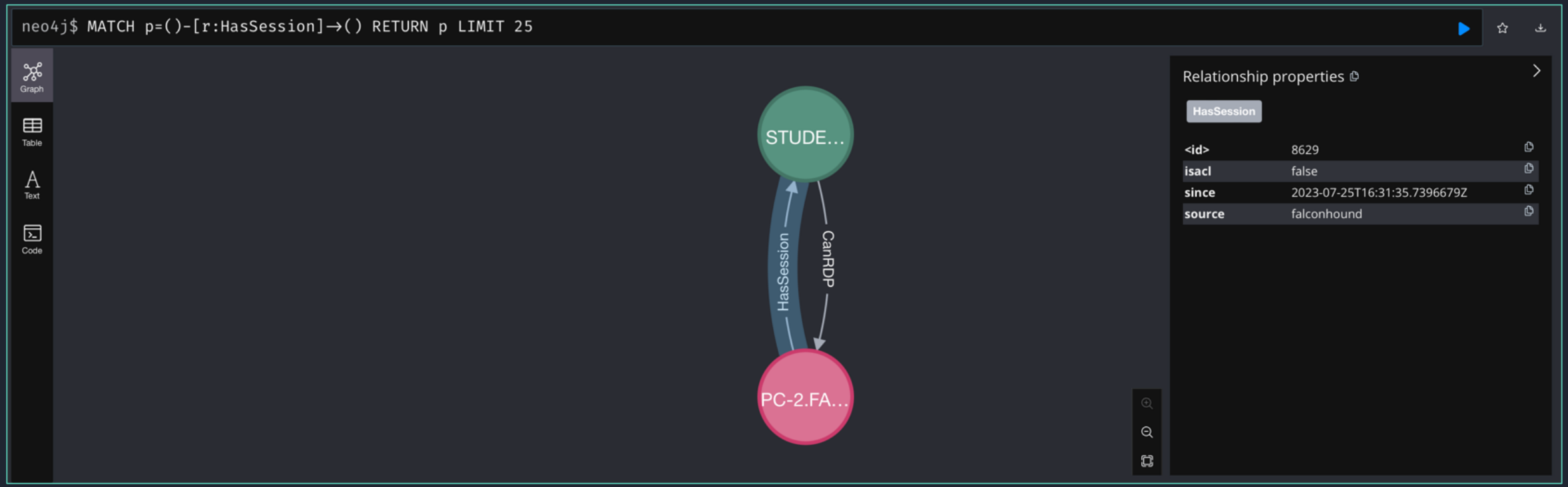
```
olafhartong ~/0xFF-Code/Sentry/FalconHound main 1.21.1 18:45:35 ./falconhound -go -config configs/config.yml -ids MDE_Exploitable_Hosts
```

Running all actions at a big corp



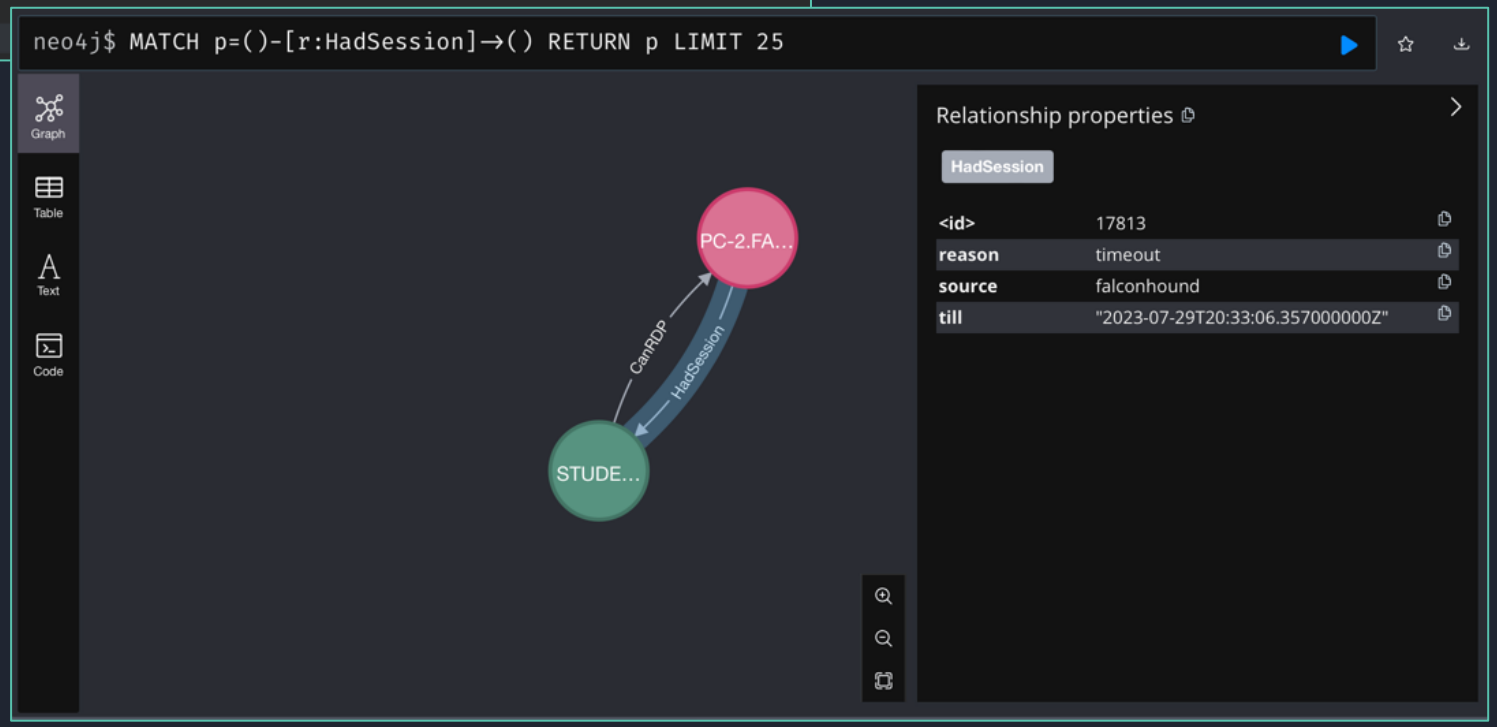
Creating new sessions

```
2023/07/28 22:30:10 [ ] Starting run
2023/07/28 22:30:10 [+] Found 9 files in actions/test/
2023/07/28 22:30:10 [+] Running 3 active queries...
2023/07/28 22:30:10 [<] Running query "Get Sessions from MDE" in MDE
{"AccountDomain":"FALCONFORCE","LogonType":"Interactive","Timestamp":"2023-07-25T16:31:35.7396679Z","min_Timestamp_arg1":0,"DeviceName":"PC-2.FALCONFORCE.LOCAL","AccountSid":"S-1-5-21-2953915480-1169422843-688089779-1103","AccountName":"STUDENT-USER"}
2023/07/28 22:30:10 [>] Writing to Neo4j
MATCH (x:Computer {name:'PC-2.FALCONFORCE.LOCAL'}) MATCH (y:User {objectid:'S-1-5-21-2953915480-1169422843-688089779-1103'}) MERGE (x)-[r:HasSession]->(y) SET r.since='2023-07-25T16:31:35.7396679Z' SET r.source='falconhound'
```



Time out sessions and set new edge

```
Name: N4J_CLN_Remove_Older_Sessions
Synopsis: Removes the HasSession relation and replaces it with HadSession if the session is older than 3 days
ID: N4J_CLN_Remove_Owned
Description: Removes the HasSession relation and replaces it with HadSession if the session is older than 3 days
Author: FalconHound
Version: '0.8'
Info: |
  tbd
Active: true                # Enable to run this action
Debug: false                # Enable to see query results in the console
SourcePlatform: Neo4j
Query: |
    MATCH (c)-[R:HasSession]->(u)
    WHERE duration.between(datetime(R.since), datetime()).days > 3
    MERGE (c)-[r:HadSession]->(u) SET r.till=datetime() SET r.source='falconhound' SET r.reason='timeout' DELETE R
Enhance: |
Targets:
```



And even fancier, via MDE

```yaml
Name: Get Active Sessions from MDE
ID: MDE_Get_Active_Sessions
Description: test
Author: FalconHound
Version: '0.1'
Info: |
  This query will look for all active sessions in MDE and add them to the graph,
  also if there are still sessions in the graph that are older than 2 hours, they will be removed and set to HadSession
Active: false                # Enable to run this action
Debug: false                 # Enable to see query results in the console
SourcePlatform: MDE          # Sentinel, Watchlist, Neo4j, CSV, MDE, Graph, Splunk
Query: |
    DeviceInfo
    | extend LoggedOnUsers=parse_json(LoggedOnUsers)
    | mv-expand LoggedOnUsers
    | extend UserName=toupper(tostring(LoggedOnUsers.UserName)), AccountSid=toupper(tostring(LoggedOnUsers.Sid))
    | summarize Timestamp=arg_min(TimeGenerated,*) by DeviceName, UserName, AccountSid
# Targets are the platforms that this action will push to (CSV, Neo4j, Sentinel, Wachlist, Slack, Teams, Splunk, Markdown
Targets:
  - Name: CSV
    Enabled: false
    Path: output/get_sessions_mde.csv
  - Name: Neo4j
    Enabled: true
    Query: |
      WITH '%v' as DeviceName, '%v' as AccountSid, '%v' as Timestamp
      MATCH (x:Computer {name:DeviceName}) MATCH (y:User {objectid:AccountSid}) MERGE (x)-[r:HasSession]->(y) SET r.since
      WITH DeviceName
      MATCH (c)-[R:HasSession]->(u)
      WHERE c.name=DeviceName
      WHERE duration.between(datetime(R.since), datetime()).hours > 2
      MERGE (c)-[r:HadSession]->(u) SET r.till=datetime() SET r.source='falconhound' SET r.reason='timeout' DELETE R
    ReplaceItems:                                    # Replace items in the query with values from the query res
      DeviceName : DeviceName
      AccountSid: AccountSid
      Timestamp: Timestamp
```

How can we use this?

With this data we can build detections based on for example an increase in direct or nested paths from for example:

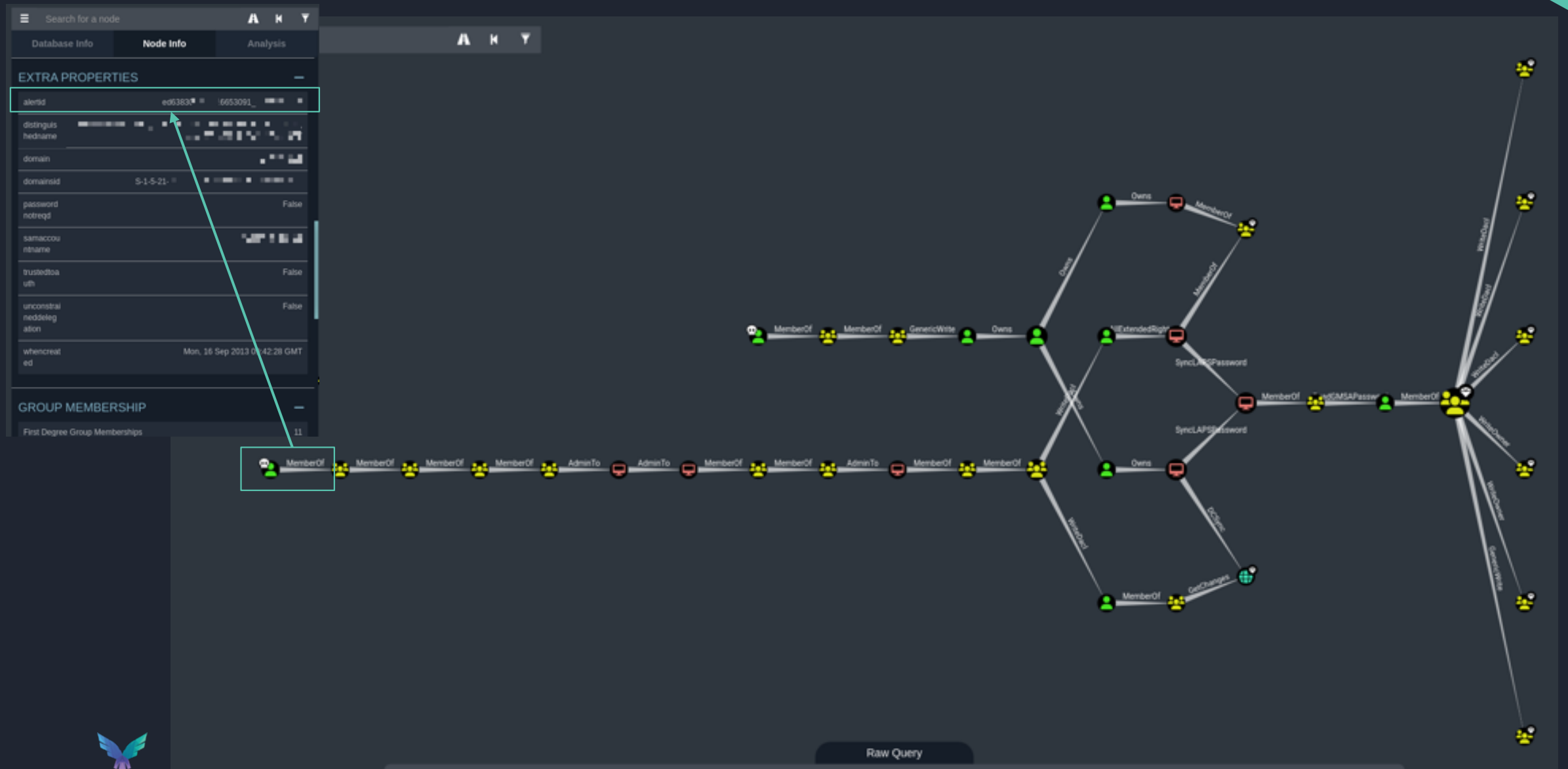
Owned users / computers to high value resources

```
DemoFalconHound_CL
| where EventID_s == "N4J_Owned_User_to_HighValue"
| where EventData_s != "[]"
| extend EventData=parse_json(EventData_s)
| sort by TimeGenerated asc
| extend PrevEventData=prev(EventData)
| extend diff=set_difference(EventData, PrevEventData)
| where diff!="[]"
| mv-expand diff
| extend TargetName=diff.Name,DirectCount=diff.Direct, DirectPath=diff.DirectNames, NestedCount=diff.Nested, NestedPath=diff.NestedNames
| project-away EventData,diff, EventData_s, PrevEventData
```

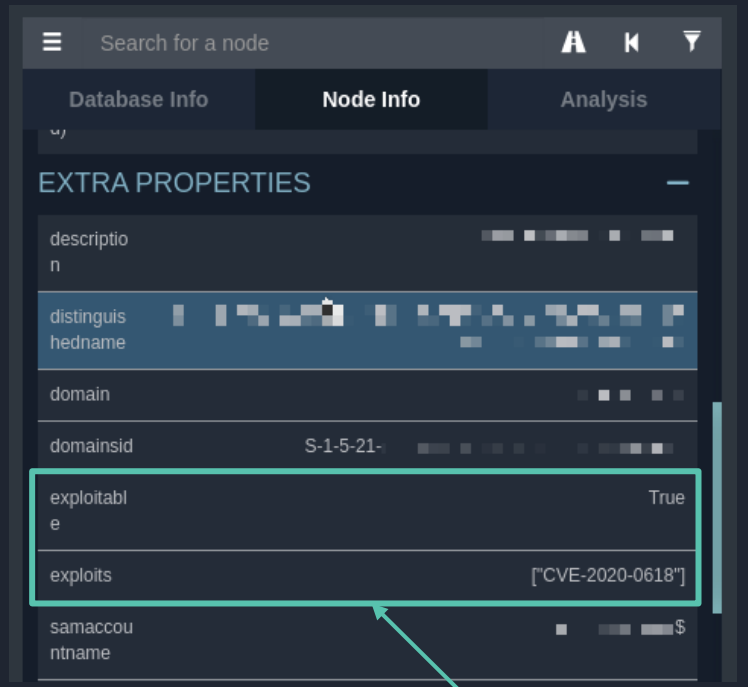
Results | Chart | Add bookmark

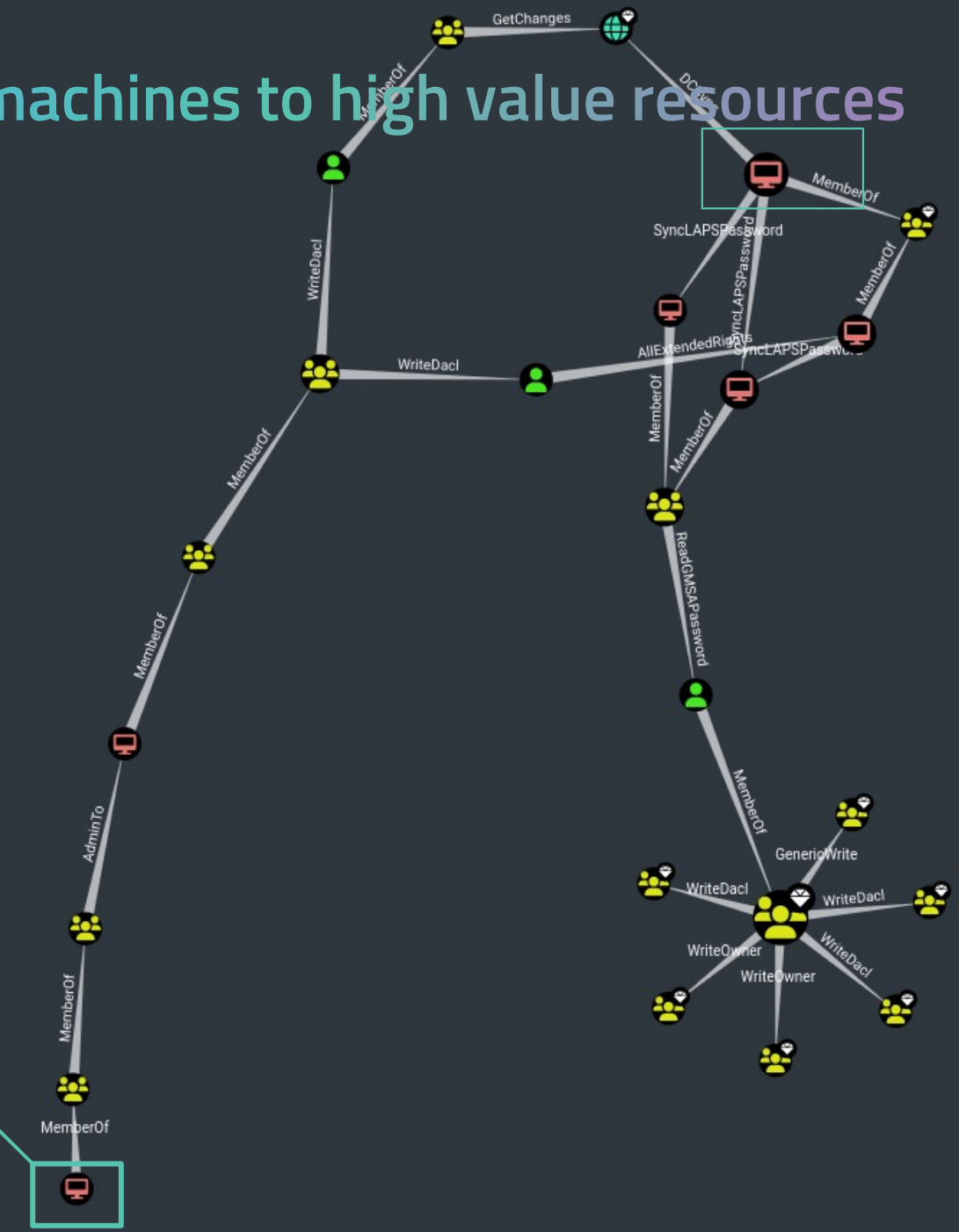
TimeGenerated [UTC]	TargetName	DirectCount	DirectPath	NestedCount	NestedPath	EventID_s	Description_s
01/09/2023, 13:04:18.788	ADMINISTRATORS@BALLPIT...	1	["ADMINISTRATOR@BALLPIT.INT","ADMINISTRA...	1	["ADMINISTRATOR@BALLPIT...	N4J_Owned_User_to_HighValue	Counts all direct and
01/09/2023, 13:04:18.788	DOMAIN ADMINS@BALLPIT.I...	1	["LYNX-ADM@BALLPIT.INT","DOMAIN ADMINS@...	1	["LYNX-ADM@BALLPIT.INT",...	N4J_Owned_User_to_HighValue	Counts all direct and
01/09/2023, 13:04:18.788	DOMAIN ADMINS@BALLPIT.I...	1	["SEAL-ADM@BALLPIT.INT","DOMAIN ADMINS@...	1	["SEAL-ADM@BALLPIT.INT","...	N4J_Owned_User_to_HighValue	Counts all direct and

Owned user to high value targets



Exploitable machines to high value resources





How can we use this?

Name	Alias	Source	Created time	Last updated
MDE Exploitable Machines	FH_MDE_Exploitable_Machines	Local file	25/09/23, 21:14	25/09/23, 21:14
AD Domain Controllers	FH_DomainControllers	Local file	25/09/23, 18:48	25/09/23, 18:48
VMs with Managed Identity	FH_AZ_VM_Managed_Identity	Local file	25/09/23, 18:48	25/09/23, 18:48
Unconstrained Delegation	FH_AD_Unconstrained_delegation	Local file	25/09/23, 18:48	25/09/23, 18:48
Kerberoastable Users	FH_Kerberoastable_Users	Local file	25/09/23, 18:48	25/09/23, 18:48

Edit watchlist items

FH_MDE_Exploitable_Machines | SearchKey field: DeviceName

Refresh | Add new | Save | Delete | Columns

Timestamp	DeviceName	OSPlatform	JoinType	IsInternetFacing	SoftwareNames	IsAzureADJoined	DeviceType	ExposureLevel	CveIds
2023-09-13T11:38:15.6629076Z	GH-DC-HZ.GHLABHZ.INTERNAL	WindowsServer2019	Domain Joined						
2023-09-25T18:34:57.1909488Z	APP3-WIN2K16.BALLPIT.INT	WindowsServer2016	Hybrid Azure AD						
2023-09-22T05:33:06.0849036Z	0XFF-ADFS.HATCHERY.LOCAL	WindowsServer2019	Hybrid Azure AD						
2023-09-25T18:51:34.4969976Z	DESKTOP-607LVB7	Windows10							
2023-09-22T11:28:03.8431747Z	ADM-WP-WIN10.TESTADOR.LO...	Windows10	Domain Joined						
2023-09-21T07:44:23.8176323Z	0XFF-PC-OLAF.HATCHERY.LOCAL	Windows10	Hybrid Azure AD						
2023-09-25T19:03:16.727273Z	0XFF-MAC-OS11	macOS	AAD Registered						
2023-09-13T11:46:24.0560652Z	GH-DET11-HZ.GHLABHZ.INTER...	Windows11	Domain Joined						
2023-09-13T10:33:25.1789422Z	GH-DET-HZ.GHLABHZ.INTERNAL	Windows10	Domain Joined						
2023-09-22T05:01:09.3891084Z	0XFF-DC1.HATCHERY.LOCAL	WindowsServer2019	Hybrid Azure AD						

Edit watchlist items

FH_MDE_Exposed_Machines | SearchKey field: DeviceName

Refresh | Add new | Save | Delete | Columns

PublicIP	LocalIP	LastSeen	PublicPorts	DeviceId
192	107.	2023-08-28T13:20:19.9665756Z	[3389]	4558959c9dff3b
192	198.	2023-08-28T09:47:46.7754769Z	[23]	4558959c9dff3b
95.1	198.	2023-08-28T13:17:43.6229775Z	[3389]	63a2c3426a6e4
61.8	162.	2023-08-28T09:48:01.220882Z	[23]	aa7248f8a88b2
178	192.	2023-08-28T23:53:38.9763465Z	[22]	e5c359777f631a
192	104.	2023-08-28T11:46:24.0805175Z	[21]	4558959c9dff3b
178	192.	2023-08-28T09:48:08.9803218Z	[23]	e5c359777f631a
192	198.	2023-08-28T13:11:49.9038634Z	[445]	4558959c9dff3b
95.1	192.	2023-08-28T11:49:37.6612275Z	[21]	63a2c3426a6e4
95.1	162.	2023-08-28T16:21:44.5711127Z	[389]	63a2c3426a6e4
150	192.	2023-08-28T11:48:10.5636601Z	[21]	b1955df96b780

Follow-up query in the data

Query results can be augmented with BlooHound queries for further analysis

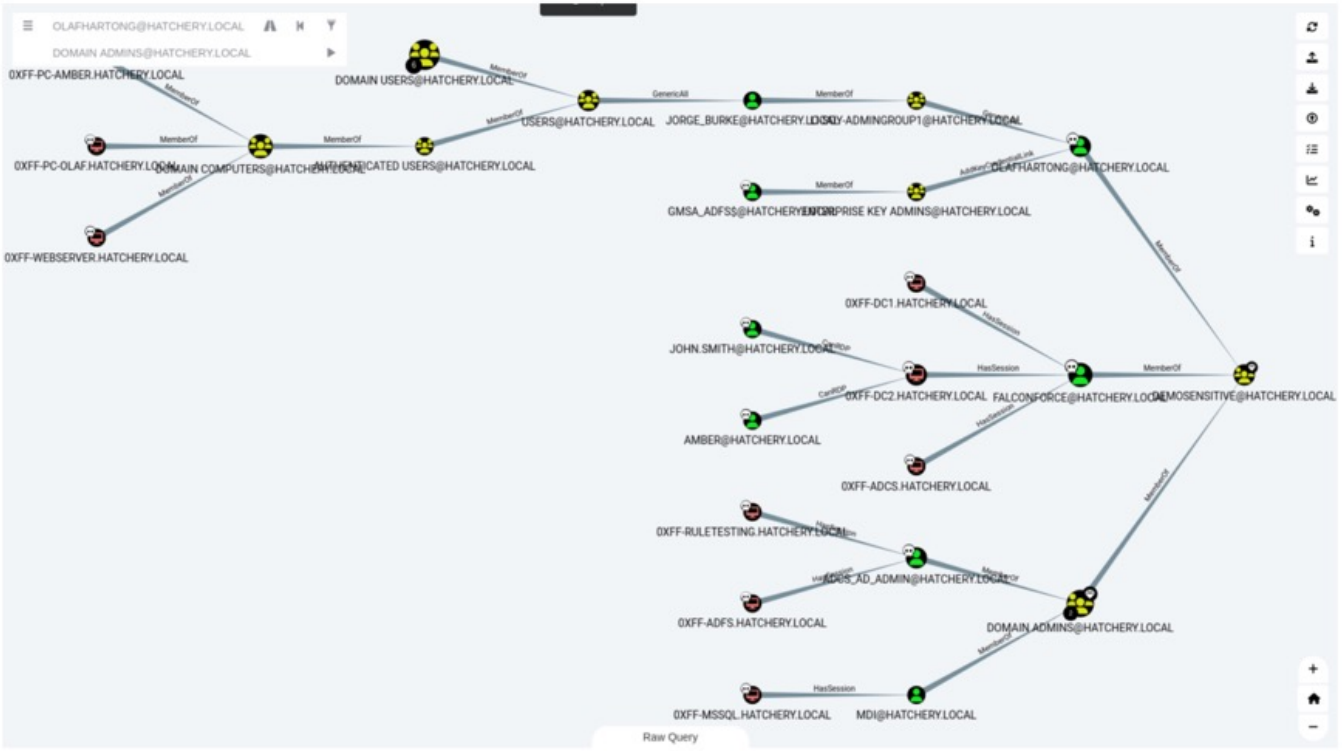
TenantId	fb5d034d-10f1-4497-b0fa-b65412190202
SourceSystem	RestAPI
TimeGenerated [UTC]	2023-10-13T14:55:07.6708278Z
BHQuery_s	MATCH a=shortestPath((u:User {owned:true})-[*..]->({highvalue:TRUE})) RETURN a
EventID_s	N4J_Owned_User_to_HighValue
Description_s	Counts all direct and nested shortest paths to HighValue nodes from owned users
> EventData_s	[{"Direct":1,"DirectNames":[' [redacted] T","Nested":1,"N
Name_s	AD Owned User with a path to high value assets
Type	DemoFalconHound_CL

How can we use this?

BloodHound details

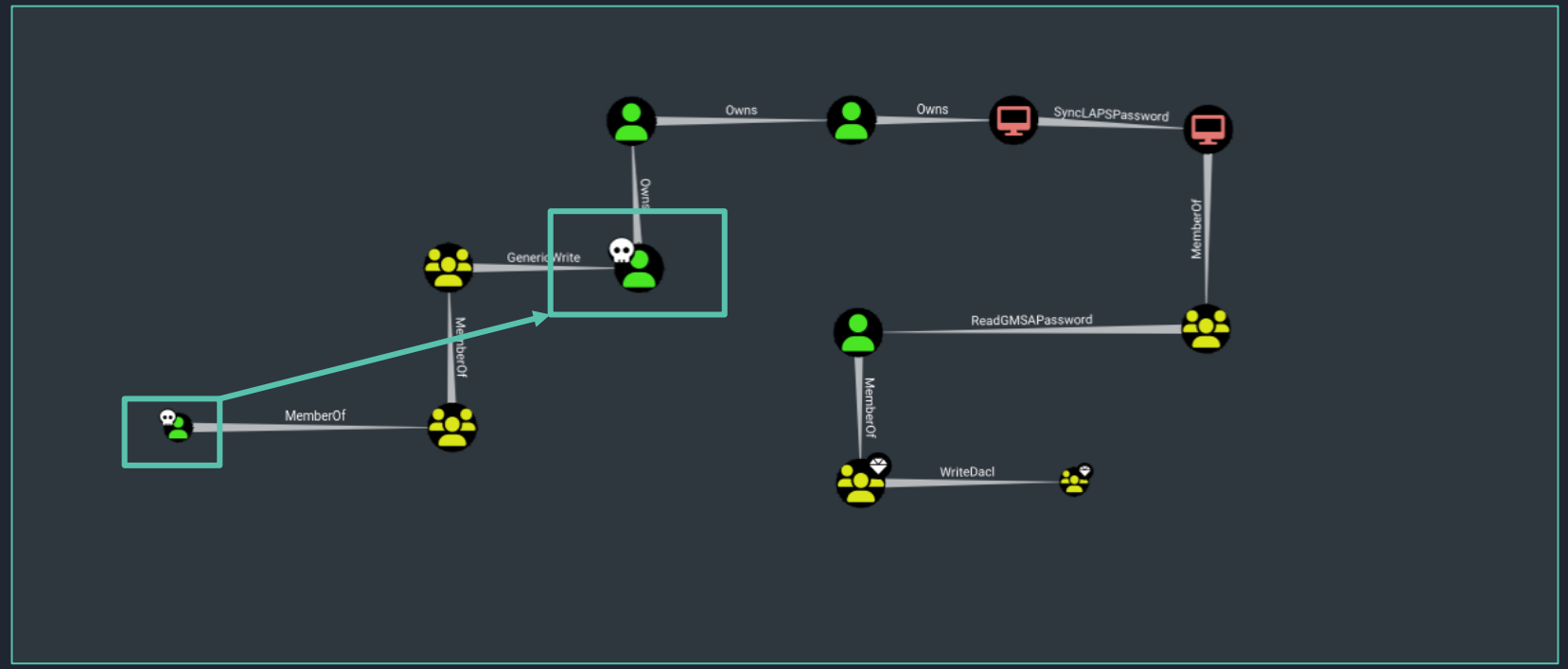
Name	Value
Outbound transitive objects	5024
Inbound transitive objects	6
Total objects in BloodHound	7203
Path to sensitive resources	TRUE

Bloodhound paths to sensitive resources:



How can we use this?

- Alert on a lateral movement path being traversed
- AND *(attempt to)* predict the next steps to investigate



Implementation

Sharphound & Azurehound
Periodically

- Not all details are in the logs
- Incremental collections with LDAP filter
- Sessions not needed
- Run it every week/month

FalconHound
Every 15 minutes

- Set as a scheduled task / cron job
- Collects sessions, new objects
- New / updated path calculations
- Watchlists / Lookuplists for detections

BloodHound Community/Enterprise

- BloodHound CE / Enterprise APIs rely on objectids (SIDs)
- Logs don't contain SIDs
- BloodHound CE is currently supported through Neo4j
- BH API support is under development, coming soon :)

Things I've learnt

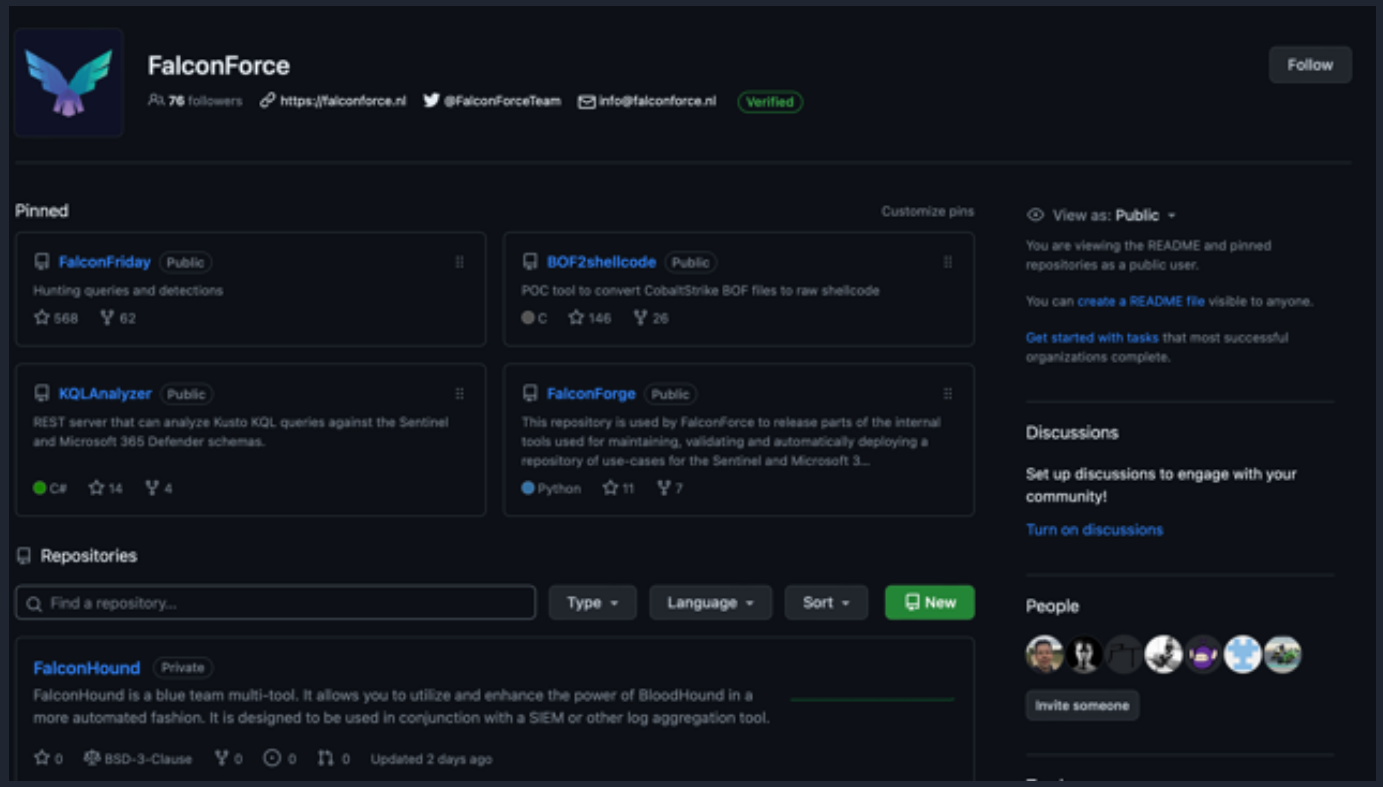
- Juggling with data is both fun and frustrating
- Coding with copilot is pretty comfortable, **IF** prompted well
- Graph databases are an amazing asset for detection engineers
- We see the same as attackers, and way more..
- I'm still not a developer :P

Defenders think in lists,
Attackers think in graphs.

Defenders can leverage attack graphs to make more lists.

As long as this is true, attackers can't win.

FalconHound, available soon!



https://github.com/FalconForceTeam/FalconHound

Contributions welcome!

We see the same, and more ..

Meet us at the FalconForce booth!

olaf@falconforce.nl

https://falconforce.nl

@olafhartong
@falconforceteam

https://linkedin.com/in/olafhartong
https://linkedin.com/company/falconforce