Cracking the Identity Perimeter

Towards a Unified Access Control Model

Jared Atkinson and Will Schroeder

SpecterOps

Jared Atkinson
@jaredcatkinson

Chief Technology Officer at SpecterOps where he oversees research and development, product discovery, and training.

Former U.S. Air Force officer contributing to the establishment of the first Hunt Operations Team and the establishment of Air Force Cyber Protection Teams.

Blogs about Detection Engineering and Purple Team theory. Wrote many defensively focused PowerShell tools in a past life.

SPECTEROPS
SO-CON
2025

Will Schroeder
@harmj0y

Member of the R&D team at **SpecterOps** where he helps research and develop new offensive techniques and capabilities.

Spoken at several industry conferences including **Black Hat**, **DEF CON**, **Troopers**, and more on topics spanning AV-evasion, Active Directory, red team tradecraft, BloodHound, (offensive) PowerShell, and more.

Cofounder of numerous open-source projects including: **BloodHound**, **Empire**, **Veil-Framework**, **GhostPack**, **Nemesis**, and more.

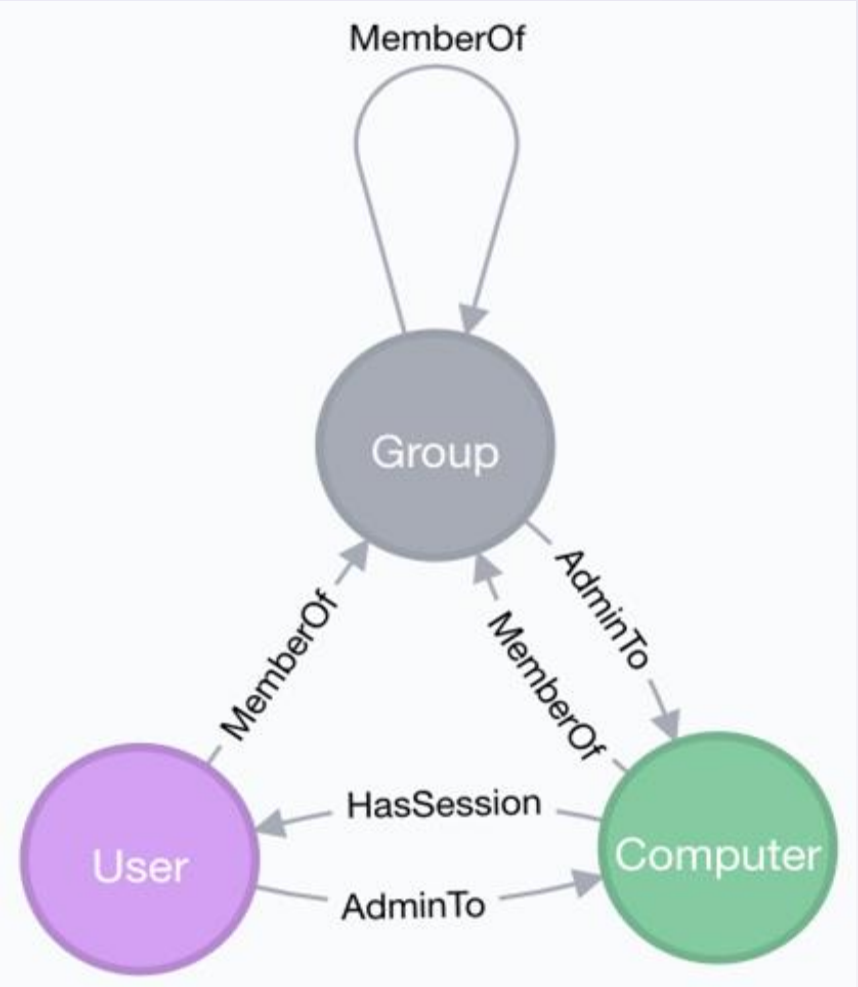
SPECTEROPS
SO-CON
2025

Setting the Stage

The Evolution of the (BloodHound) Attack Graph

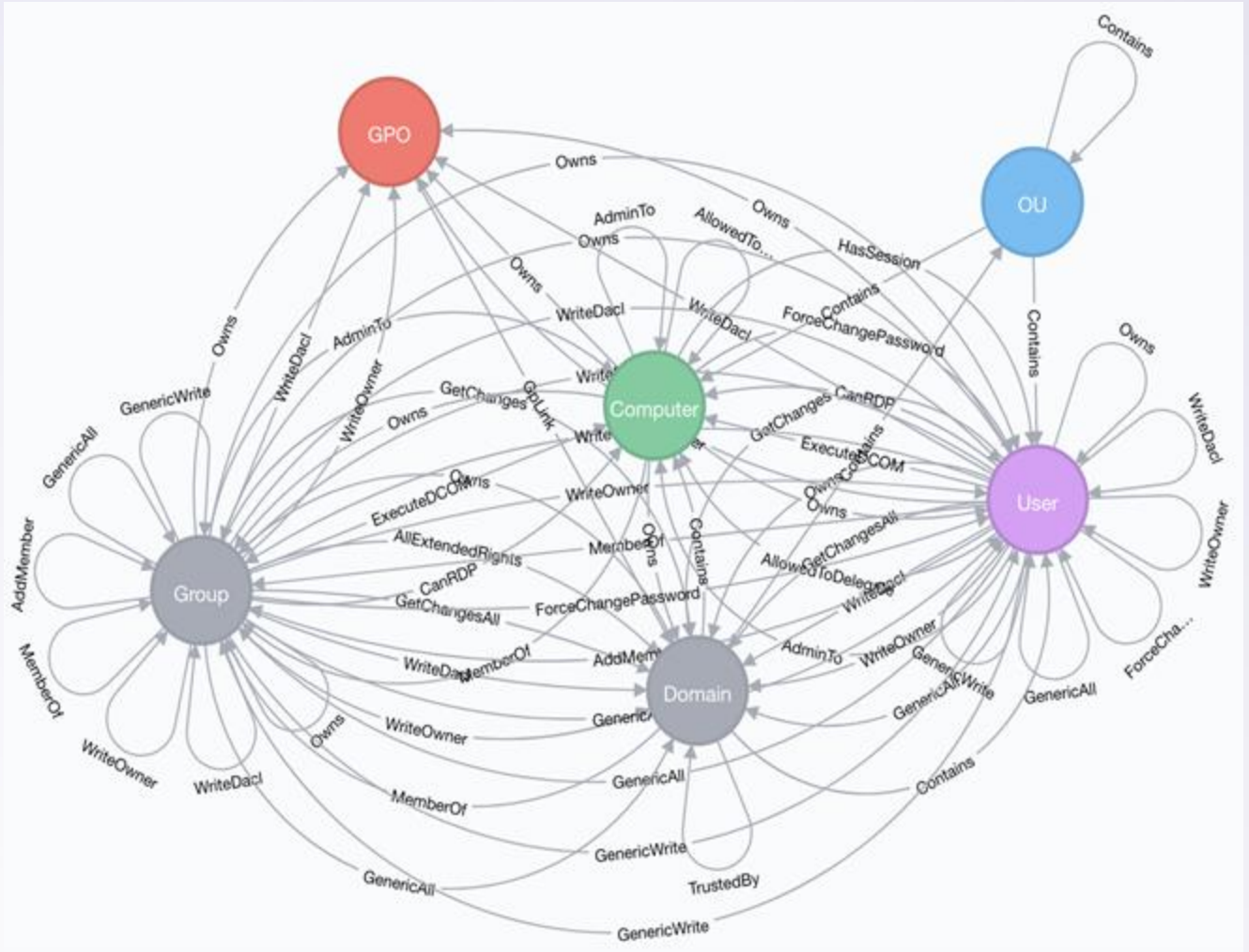
Evolution of the BloodHound Graph

v1.0



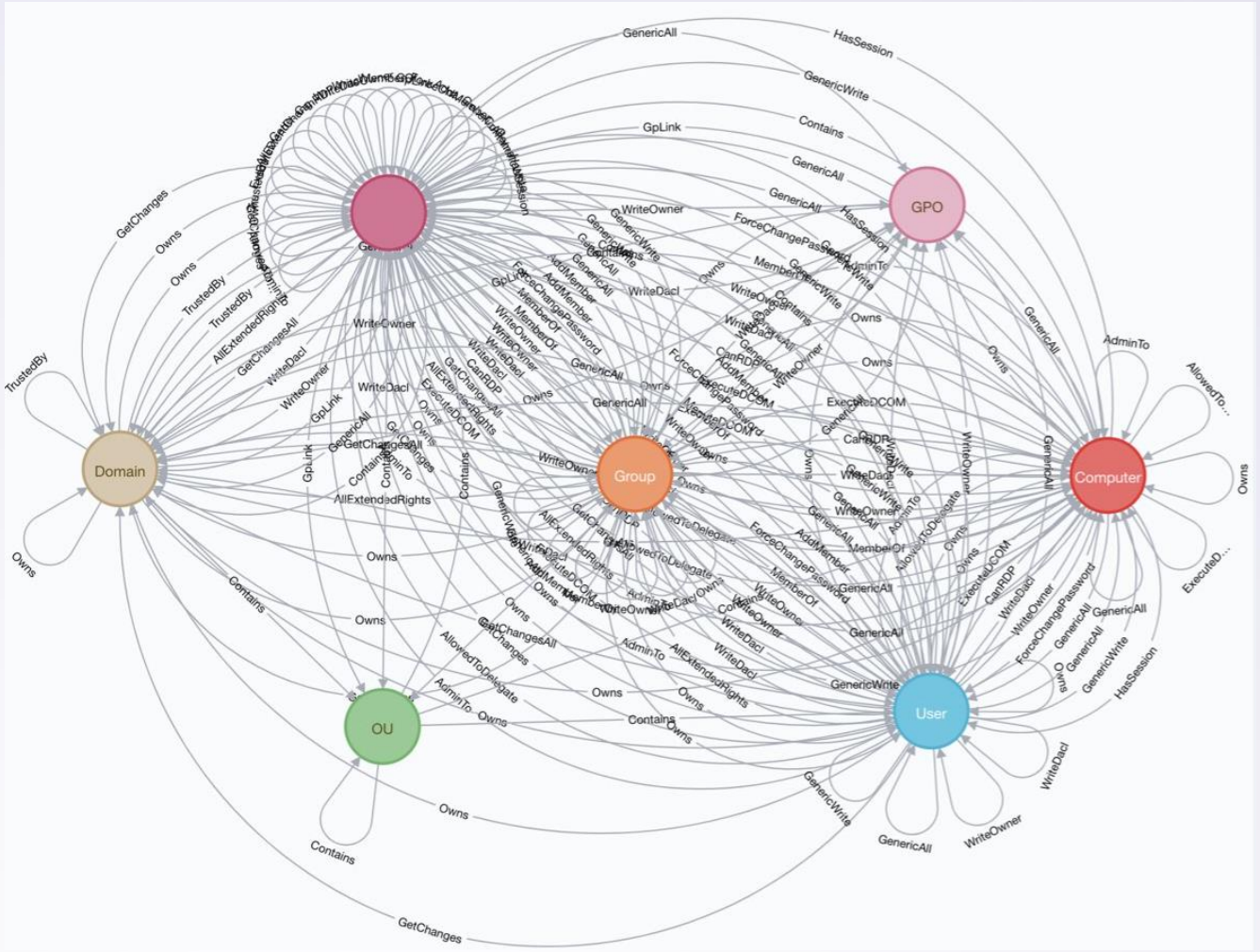
Evolution of the BloodHound Graph

v2.0
ACLs



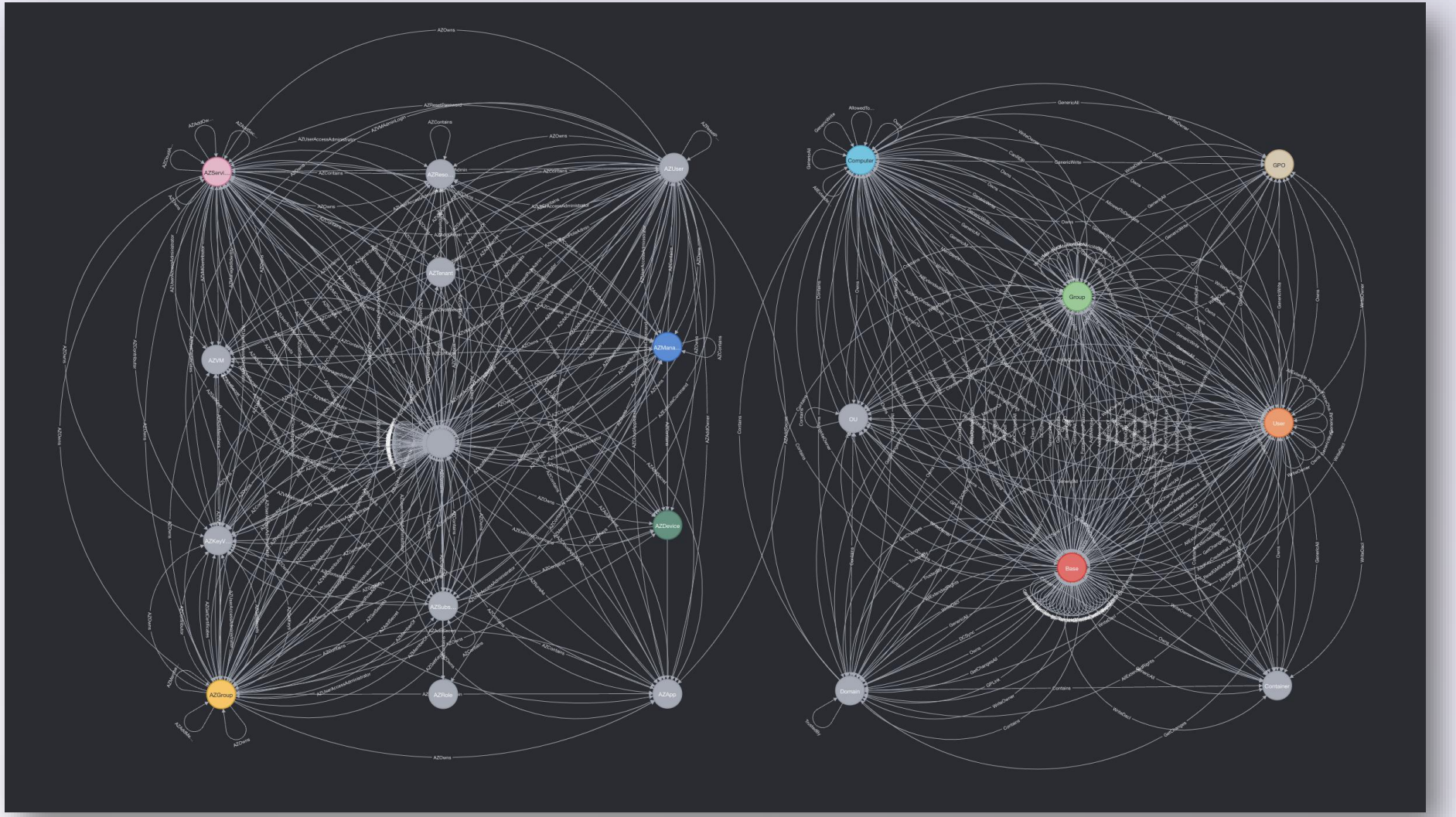
Evolution of the BloodHound Graph

v3.0
GPO Update



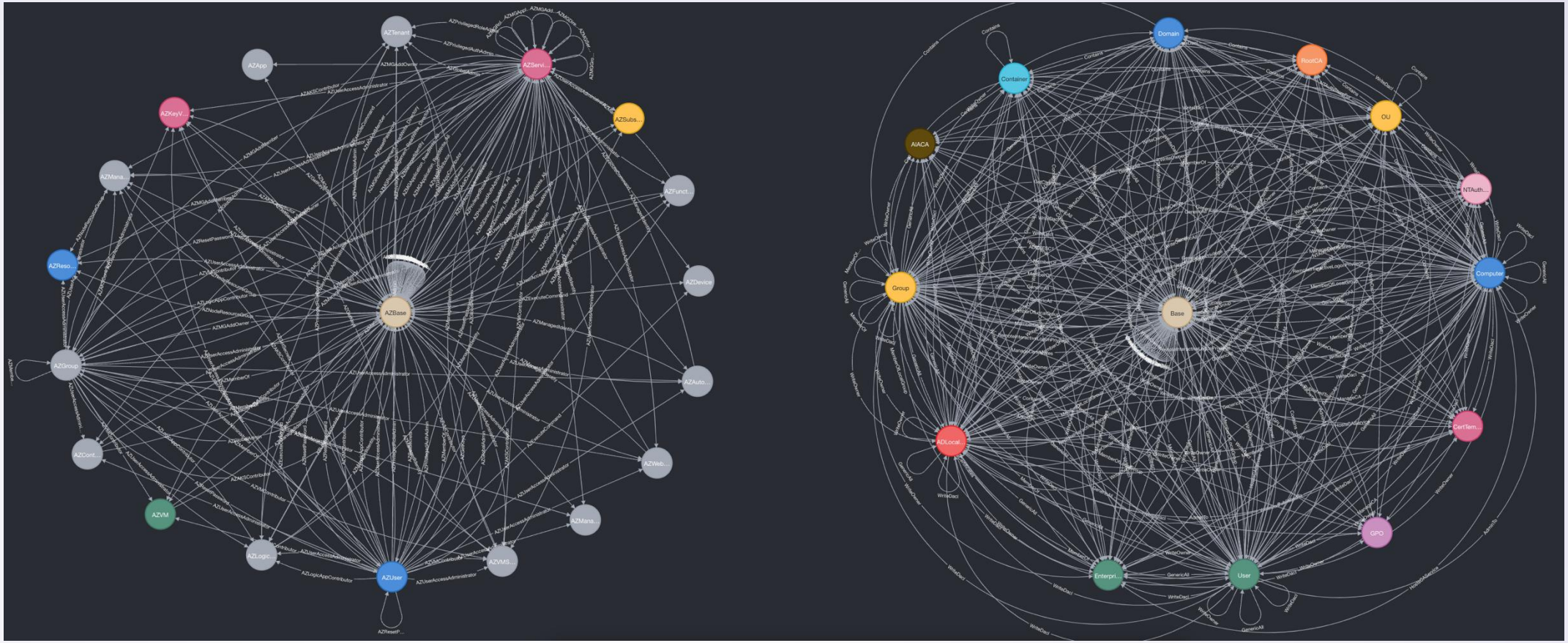
Evolution of the BloodHound Graph

v4.2
Azure+



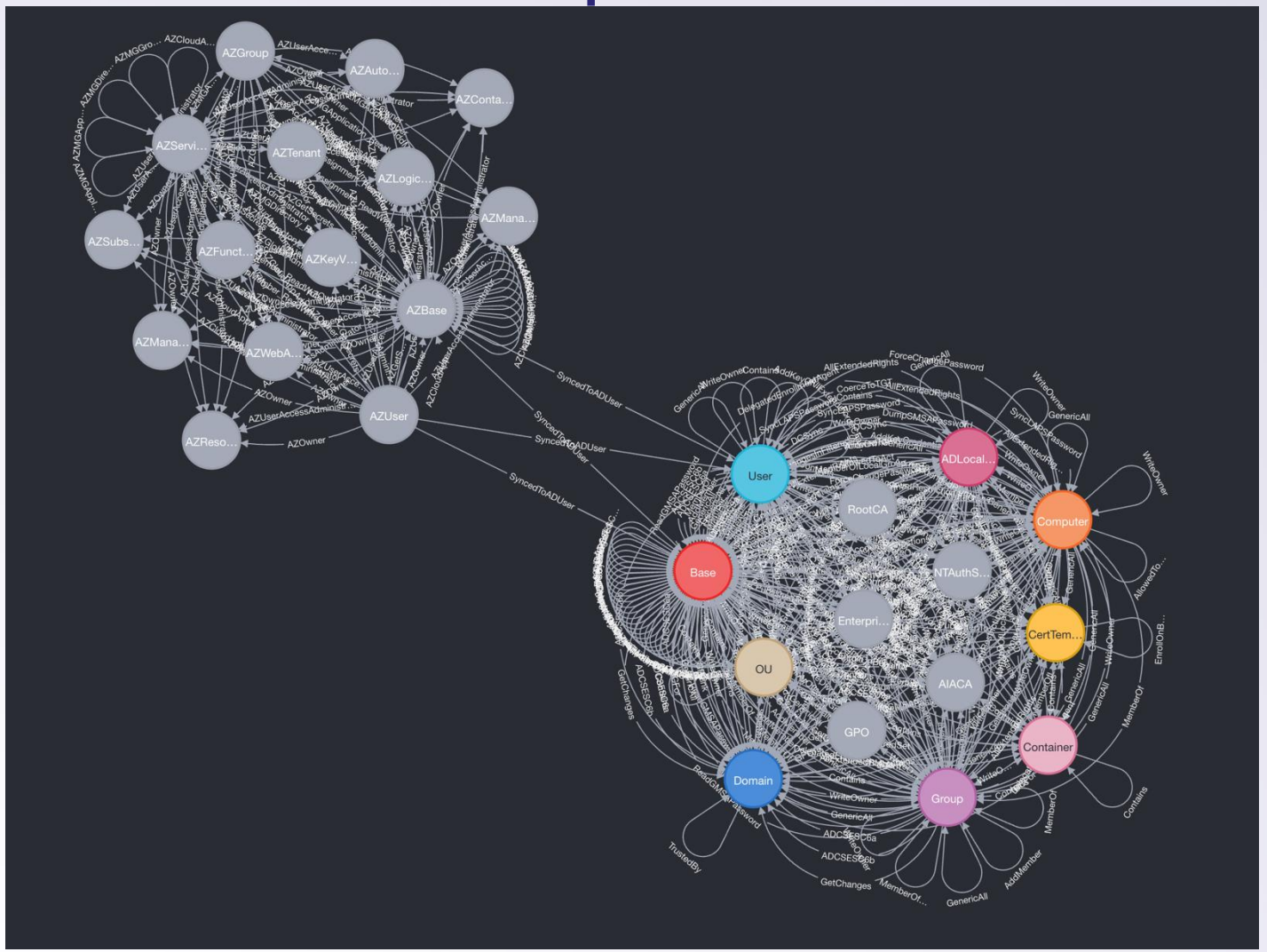
Evolution of the BloodHound Graph

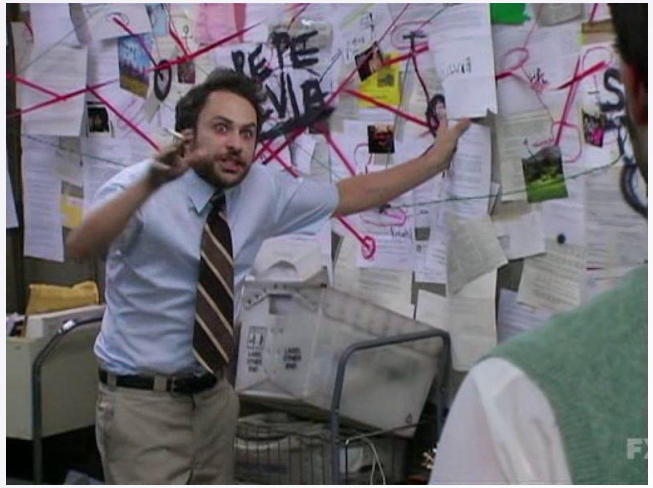
v5.4 – AD CS



Evolution of the BloodHound Graph

v7.1 – Hybrid + NTLM

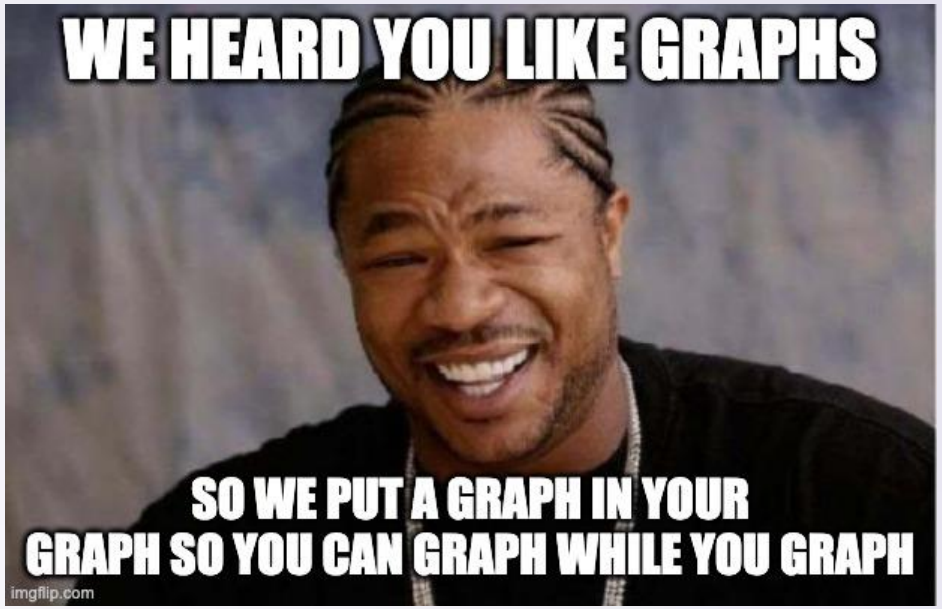




The BloodHound Graph

Current State

- We continue to add additional edges as new attack paths are discovered
  - A March 2025 example: NTLM!
  - After that? There are lots of “local graphs”
    - Kubernetes, GitHub, Snowflake, AWS…
- As the graph grows, a natural question arises:
  - *Is BloodHound a graph-of-graphs (metagraph)?*



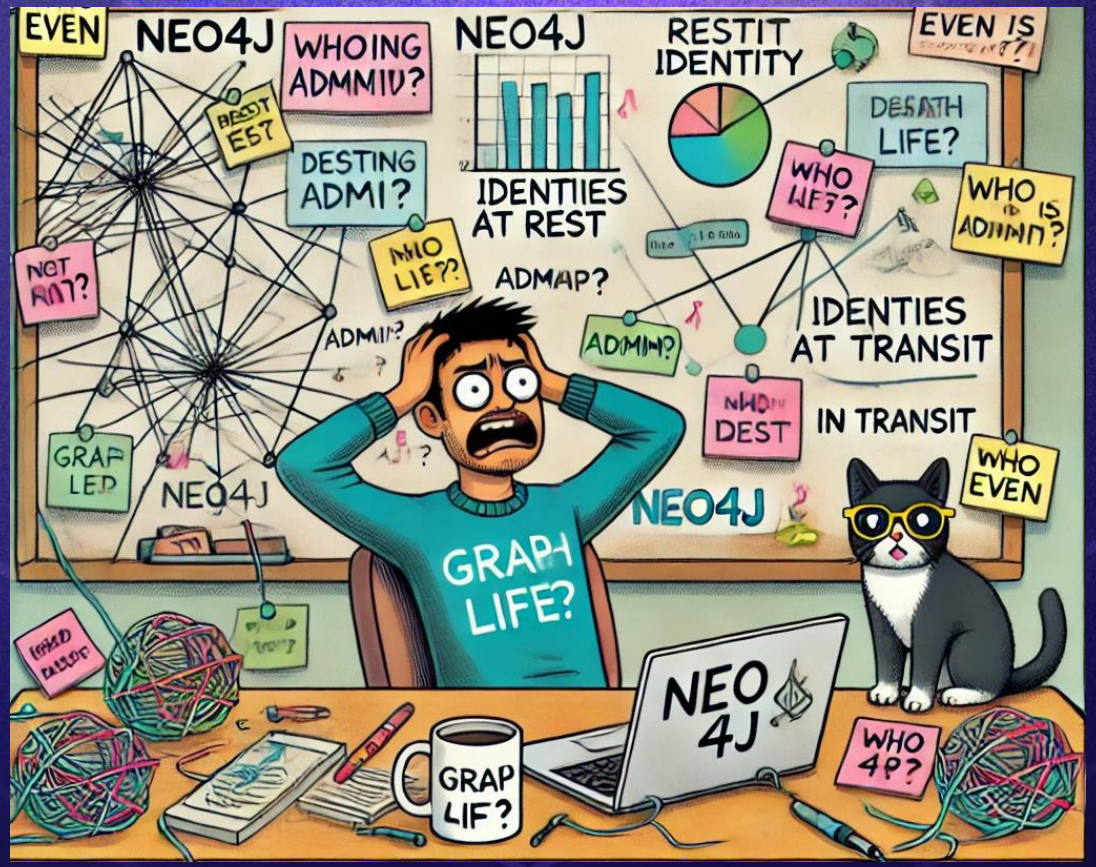
The BloodHound Graph

A Graph of Graphs?

- Formally, a graph is a mathematical structure consisting of a set of *nodes* (or vertices) and connected by *edges*

- A "graph of graphs" refers to a model where the nodes represent graphs, and the edges represent relationships *between* those graphs.
  - We define these as ***local graphs*** vs the ***global graph***
  - While there can be node/edge overlap between subgraphs, each subgraph can be considered “self-contained”

Common (Graph-centric) Issues



How do we define "Admin"?

Harder than you might think…

- ***Local*** Administrator on a system?
- ***Domain*** Administrator for a domain or ***Enterprise*** Administrator for a forest?
- ***Global*** Administrator for an Entra ID tenant?
- ***Database*** Administrator for Snowflake database?
- ***Organization*** Administrator in GitHub?

How do we define "Admin"?

Harder than you might think…

- ***What about…***
- Users one (or more) hops away on an attack graph? (“derivative local admin” anyone?)
- Users with custom roles and very specific (and possibly privileged) actions?
- ***“Regular” domain users logged into privileged cloud SaaS application from their desktop?*** 🤔


IDENTITY IS
THE NEW PERIMETER
INFOSEC
ASSUME
BREACH

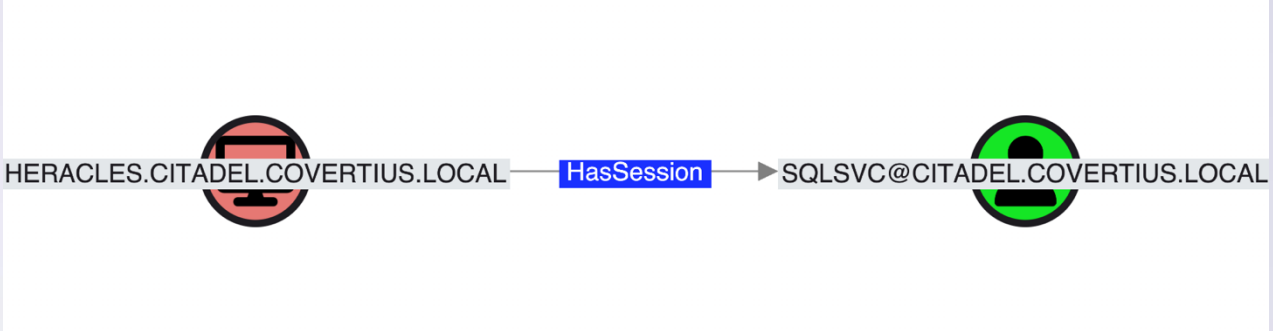
Identity States

Data Protection

- Data Protection – the practices and principles aimed at safeguarding sensitive information from unauthorized access…
  - Encryption is a primary measure used to achieve data protection
- Our approach to applying encryption depends on the state of the data
  - Data At-Rest – data stored on a device or database
  - Data In-Transit – data moving from one system to another

Identity States

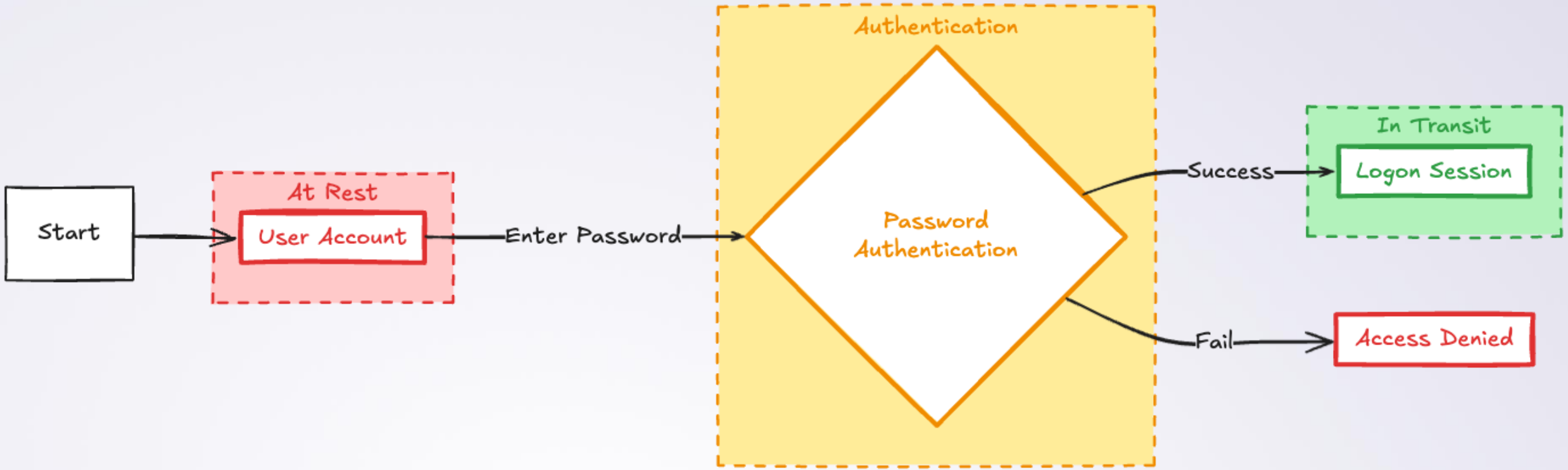
Applying the Analogy to Identity



- Identity At-Rest (Accounts)
  - An account that is registered with the system or an identity provider
  - Attacks: Password Dumping or Brute Forcing
  - BloodHound: User, Computer, AZUser, AZDevice nodes
- Identity In-Transit (Sessions)
  - The session that is the result of successful authentication of an at rest identity
  - Attacks: Token Impersonation or Cookie Theft (Session Hijacking)
  - BloodHound: HasSession edge
- ***There are different approaches in protecting identities at rest vs in transit!***

Authentication vs. Authorization

Putting an At Rest Identity In Transit



Identity States

User Hunting

- Identities exist to be put "in-transit" via authentication
  - With few exceptions, user accounts are created to be used
  - If an identity is valuable, it will likely be in the in-transit state SOMEWHERE in the on-prem environment
  - If an attacker has an on-prem foothold, they can pivot to user hunting
- Attackers simply need to determine where in the on prem environment that in transit identity is (user hunting)

Your Identities are at their most vulnerable

in your on-prem environment!

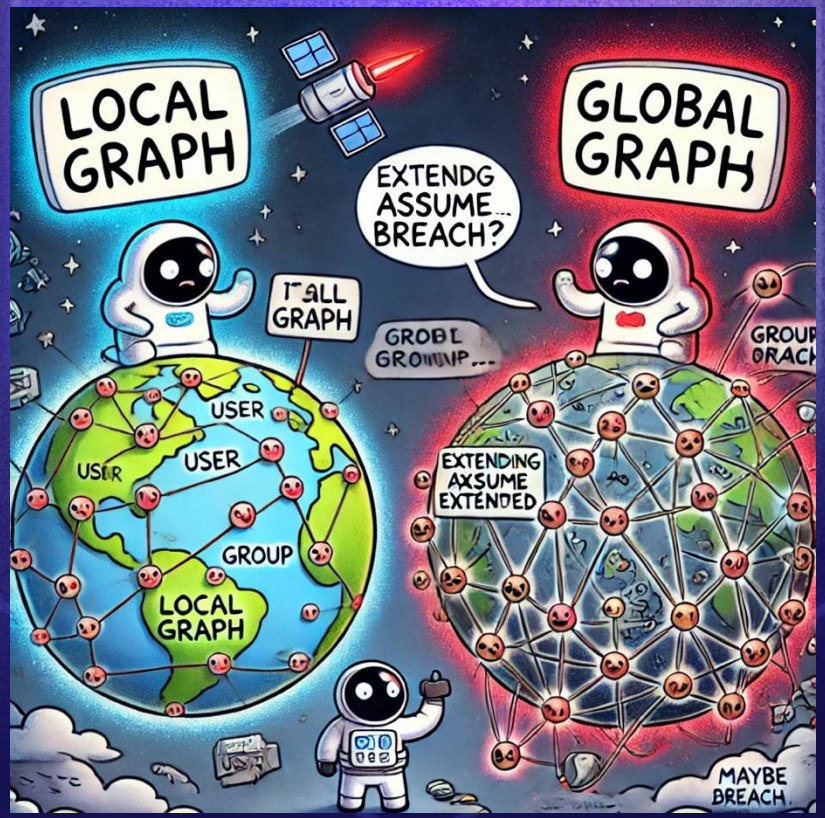
https://blog.harmj0y.net/penetesting/i-hunt-sysadmins/

Identity States

We already know this!

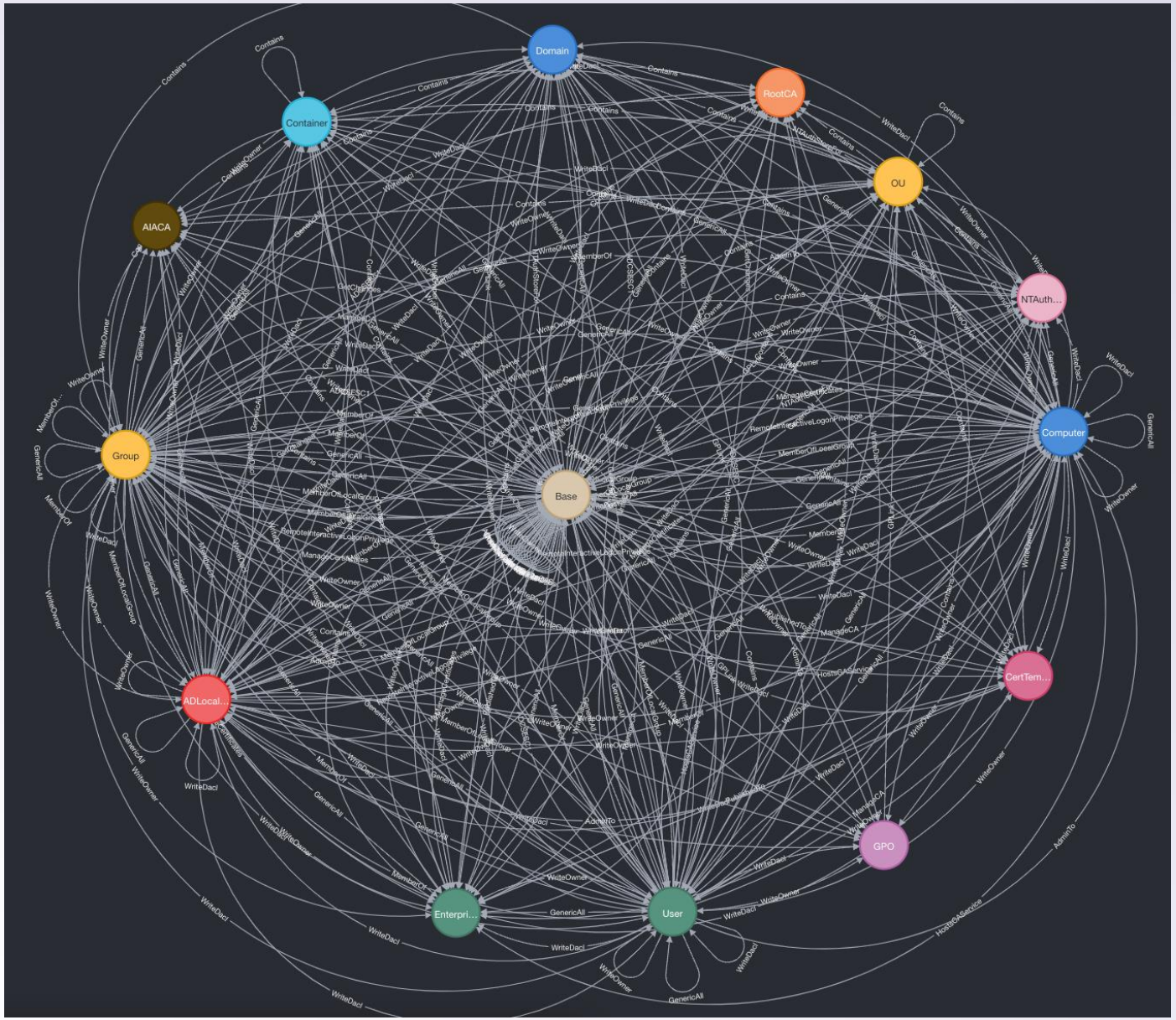
- Every Red Teamer has leveraged on-prem access to compromise a cloud app
  - Identity Providers
    - Andy Robbins and Cody Thomas – Synced User Attack Path Analysis with BloodHound
    - Adam Chester – Identity Providers for Red Teamers
  - Third Party Applications
    - Matt Merrill and Zach Stein – “From AD to SaaS” – Tuesday 2:00pm (Track 1)
  - Code Repositories and CI/CD Pipelines
  - Communication
    - Dan Mayer – SlackPirate Set Sails Again!
  - Documentation
  - Security Tooling

Local vs Global Graphs



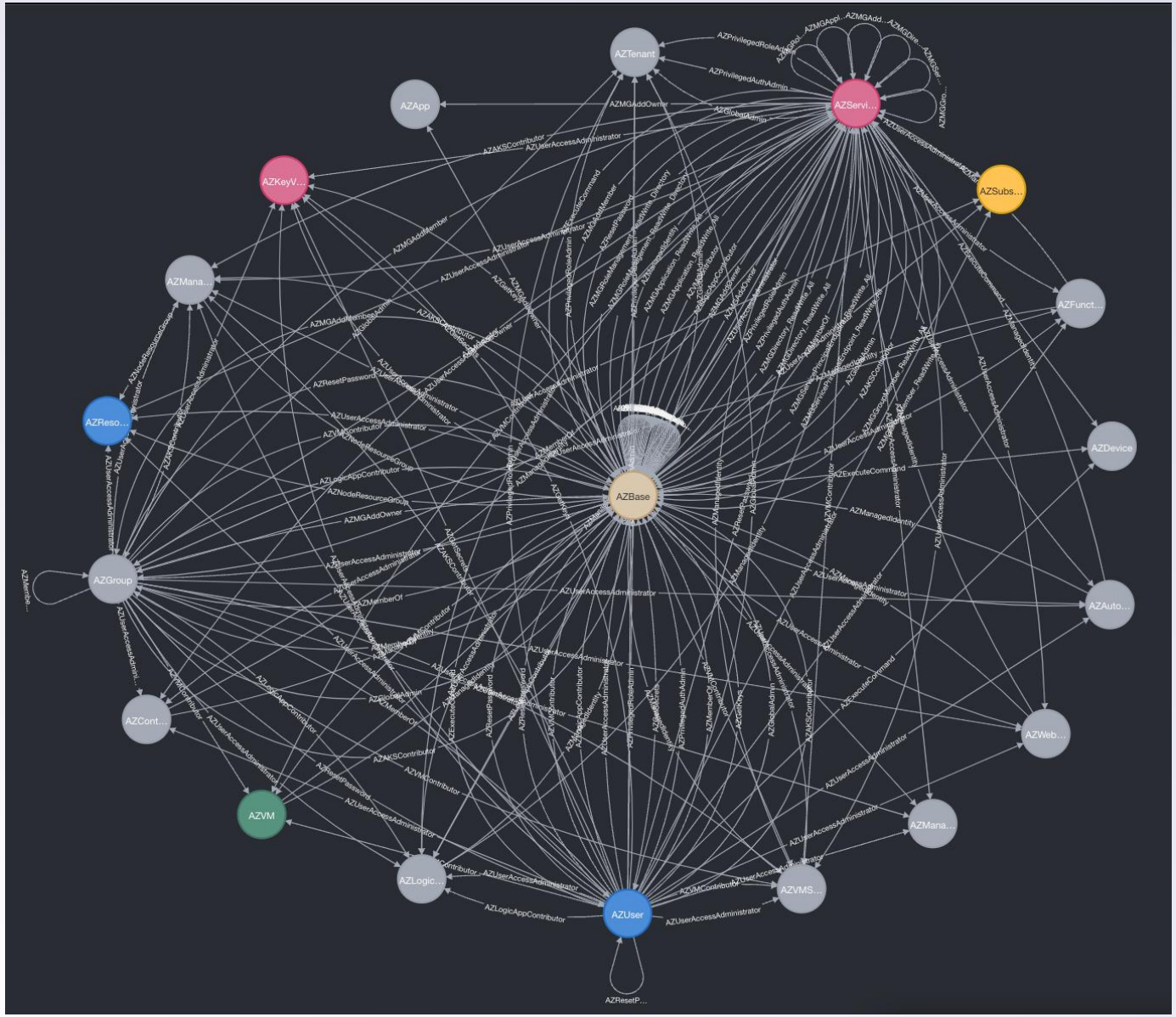
Local Graph

Active Directory



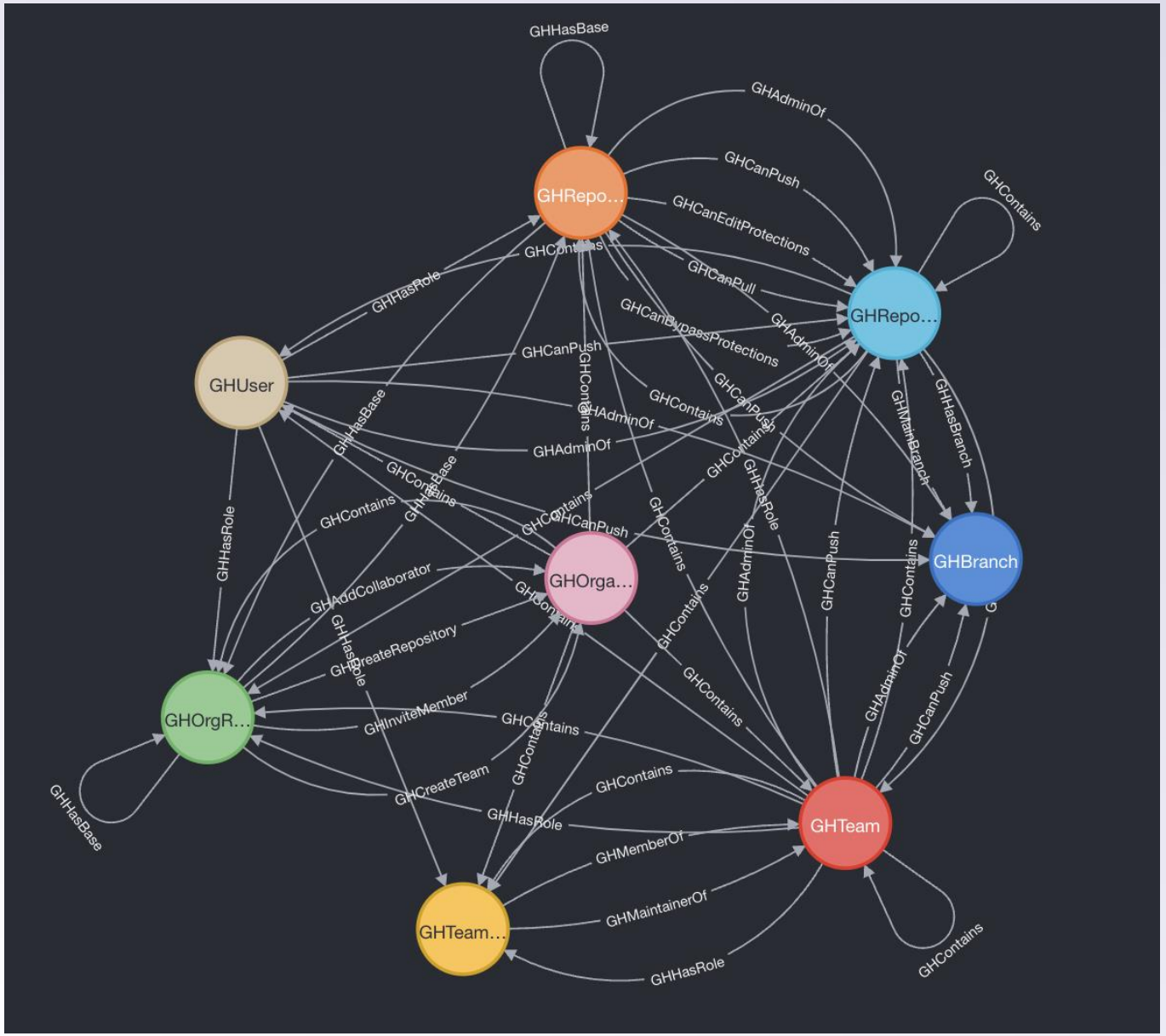
Local Graph

~~Azure~~ Entra ID



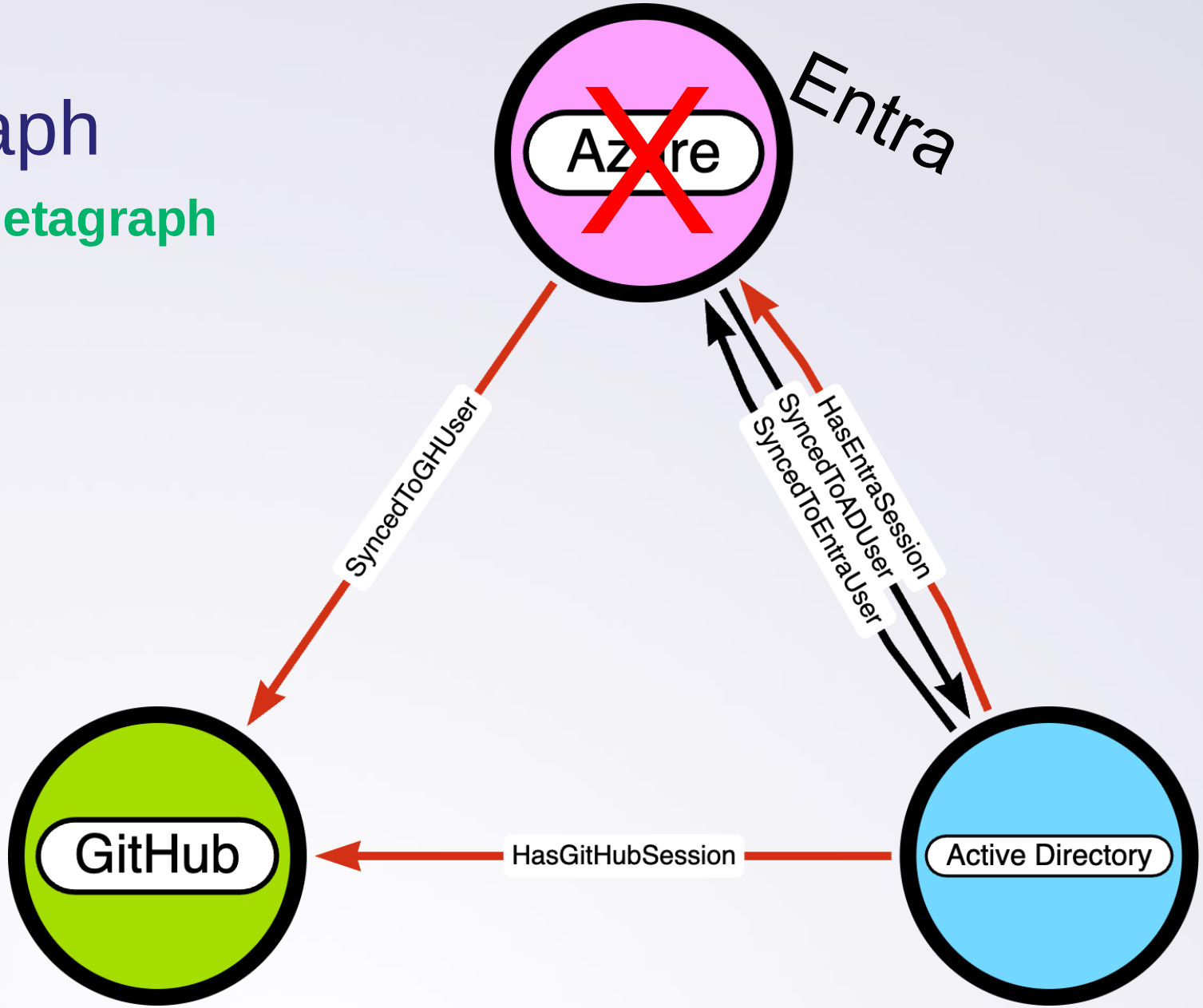
Local Graph

GitHub



Global Graph

Building our Metagraph



Assume Breach ++

Beyond the Perimeter

- **Scenario:**
  - Attackers will almost always achieve initial access to your AD/on-prem environment
  - (Local) Privilege Escalation is (nearly always) a given
    - Local Relay scenarios, AD CS abuses, NTLM relay scenarios, etc.
  - 90+% of AD environments we've seen have 100% exposure of Tier 0
- **Conclusion:**
  - If we accept assume breach, *we must also accept attackers can potentially access any computer and identity in the on-prem environment!*

Assume Breach ++

When the local graph of a SaaS systems is connected to the global graph containing Active Directory, the risk exposure for the SaaS' local graph increases significantly!

We've been here before 🙃

Assume Breach ++

Just logging into a GitHub accounts from your “regular” on-prem Active Directory computer is enough to connect the local GitHub graph to the global graph!

Case Study

Snowflake

Snowflake Case Study

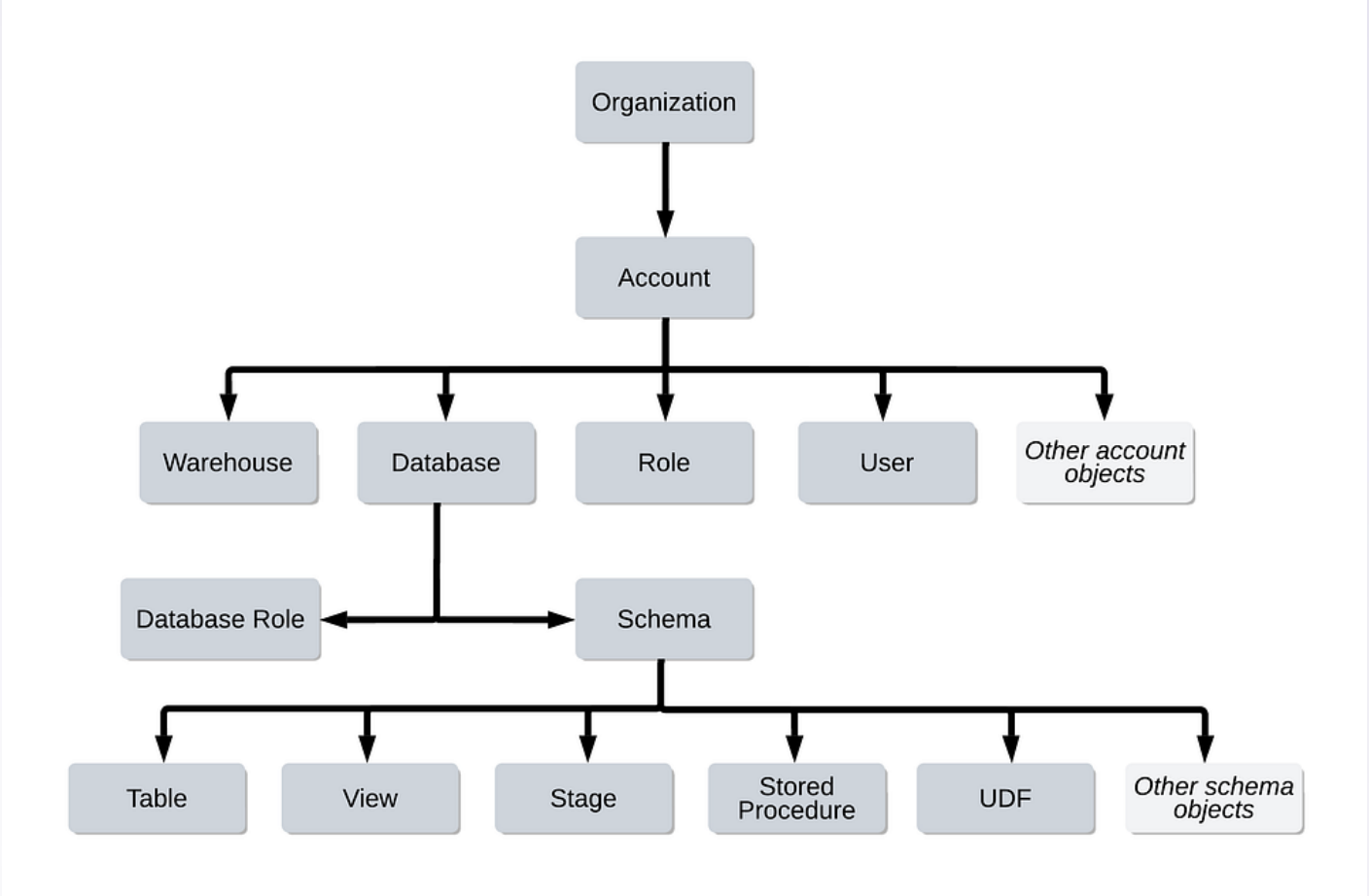
How it happened?

- In April 2024 Mandiant received TI containing data that originated from a Snowflake instance.
- In May 2024, Mandiant identified several additional victims of a larger campaign.
- According to Mandiant, a substantial subset of these victims' Snowflake identities were compromised via a contractor's laptop that was also used for personal activities.

https://cloud.google.com/blog/topics/threat-intelligence/unc5537-snowflake-data-theft-extortion

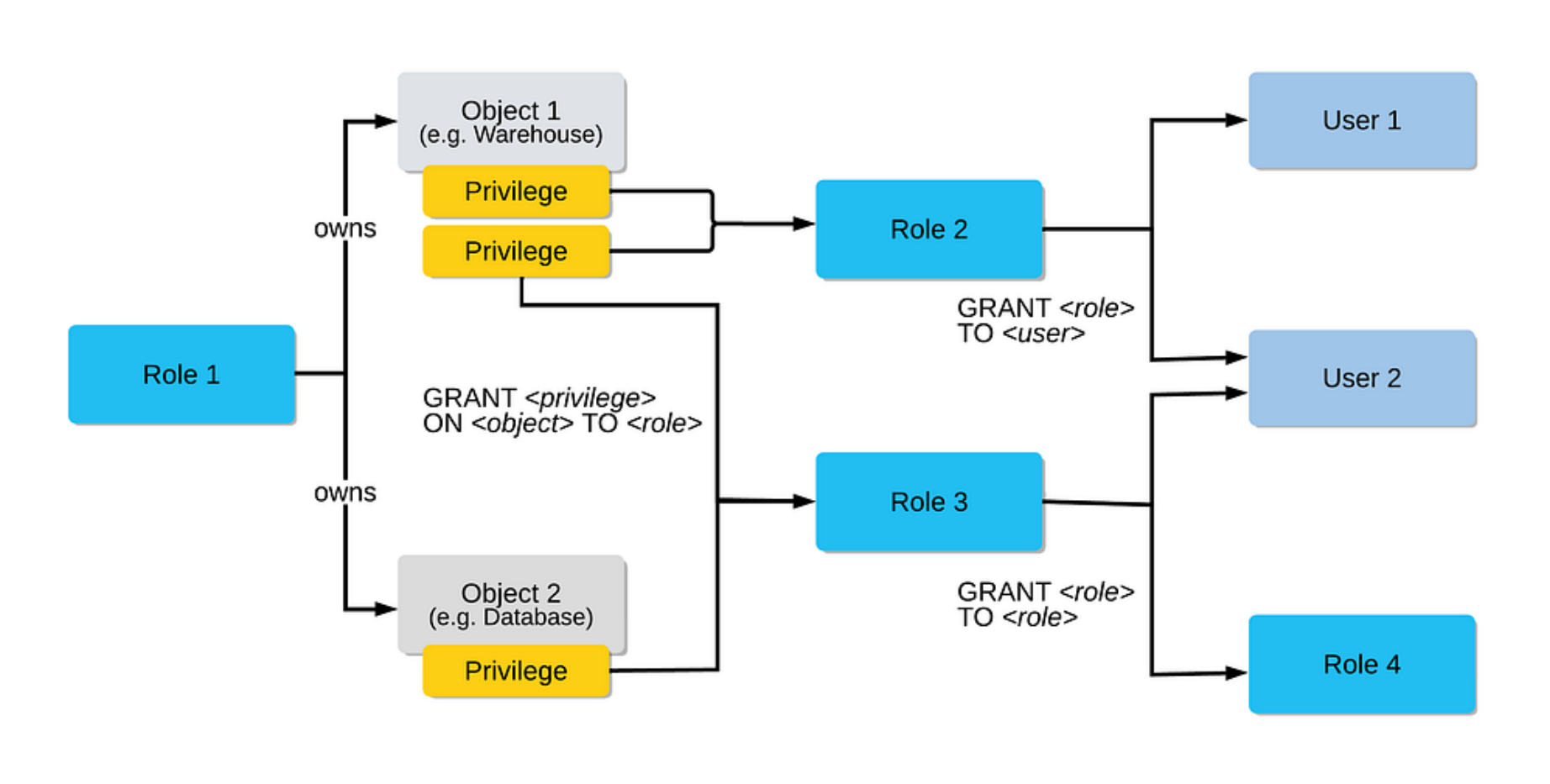
Snowflake Case Study

Object Hierarchy



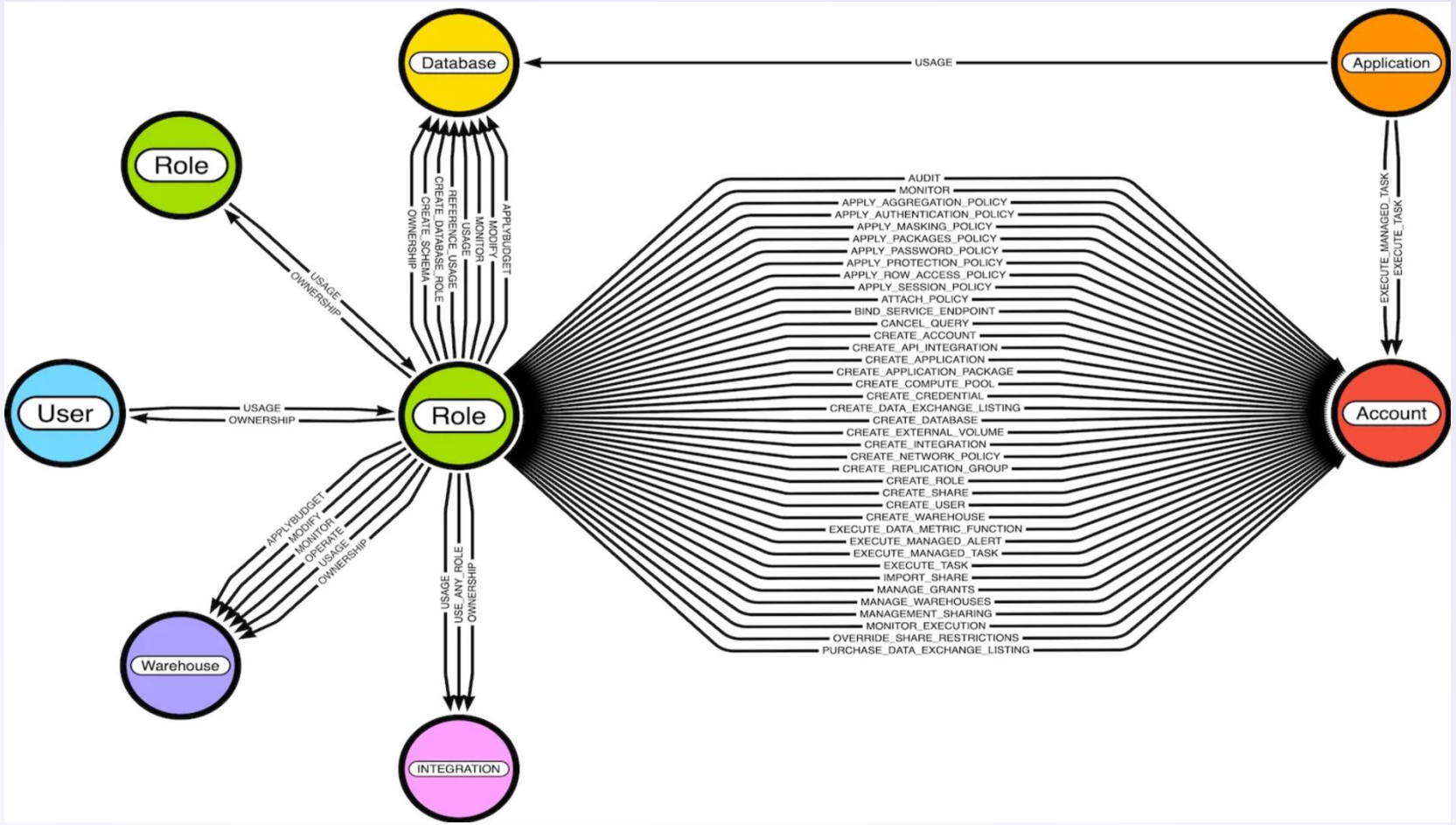
Snowflake Case Study

Access Control Model



Snowflake Case Study

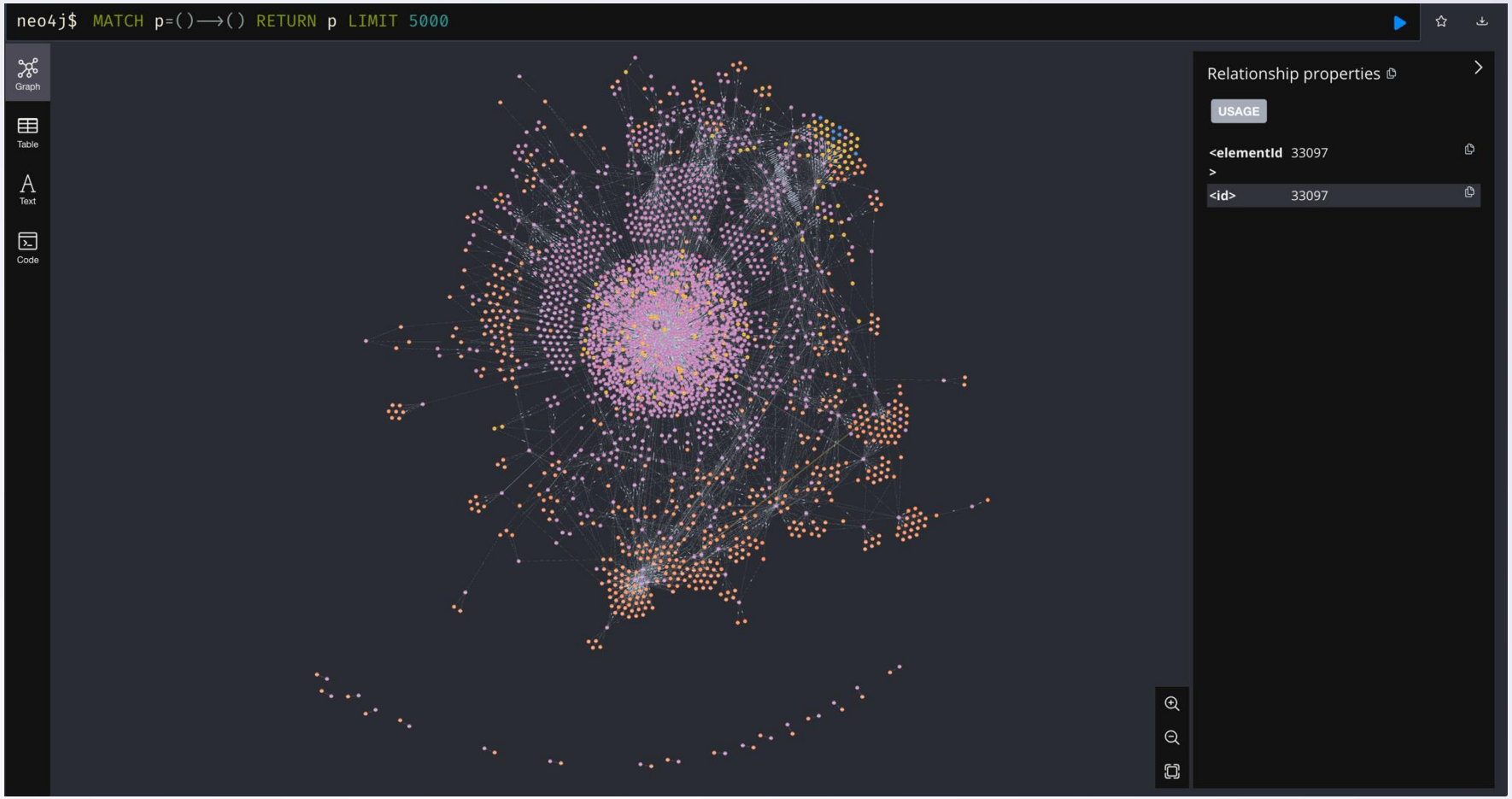
BloodHound Model (Down to Database Level)



https://posts.specterops.io/mapping-snowflakes-access-landscape-3bf232251945

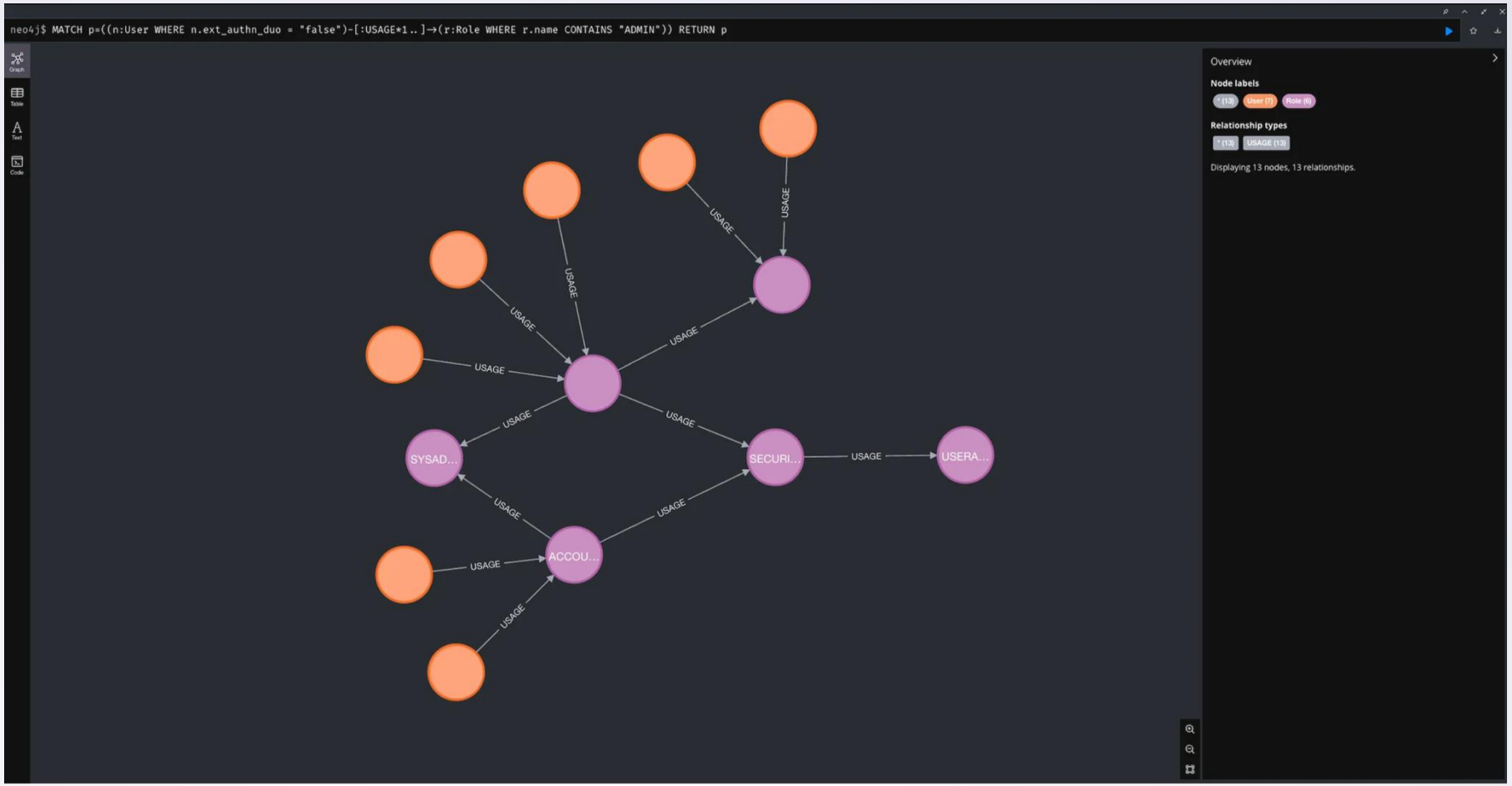
Snowflake Case Study – In Action

The interconnected local graph



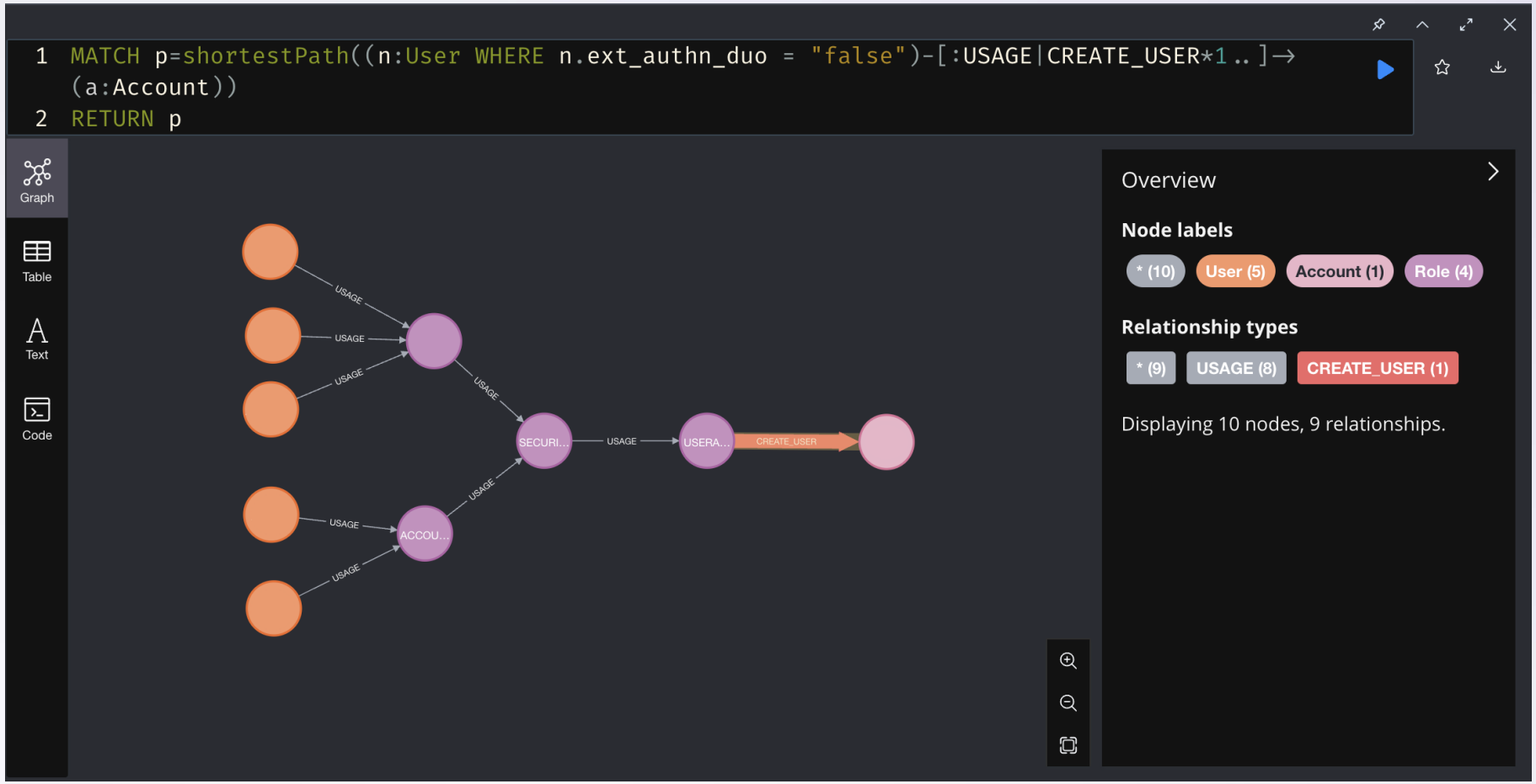
Snowflake Case Study – In Action

Privileged Accounts without MFA



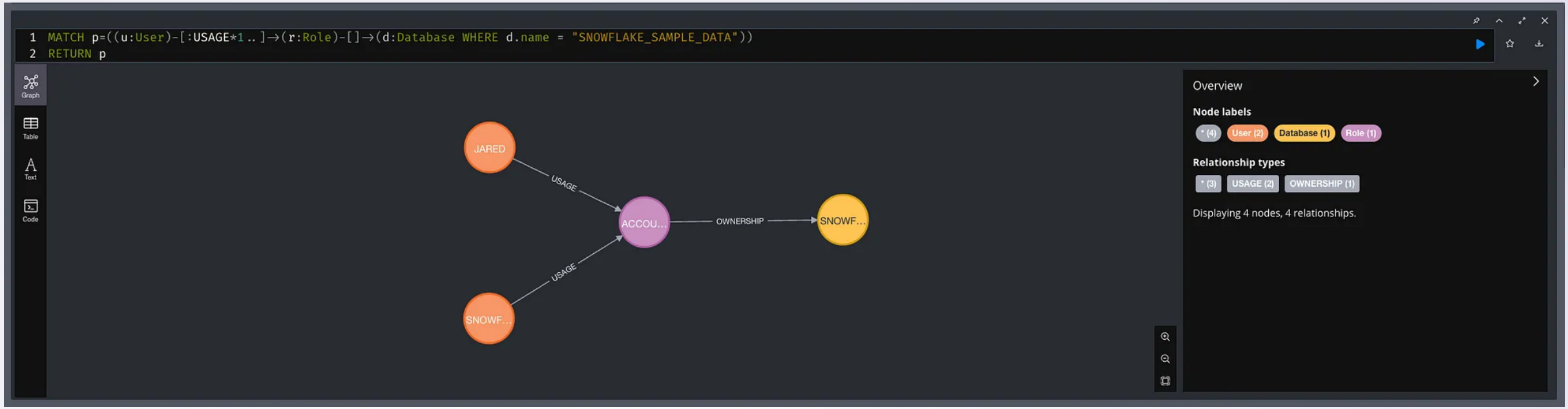
Snowflake Case Study – In Action

Users that can create new users



Snowflake Case Study – In Action

Who has control of a specified database?



Snowflake Case Study

Recommended Interventions for Snowflake Customers

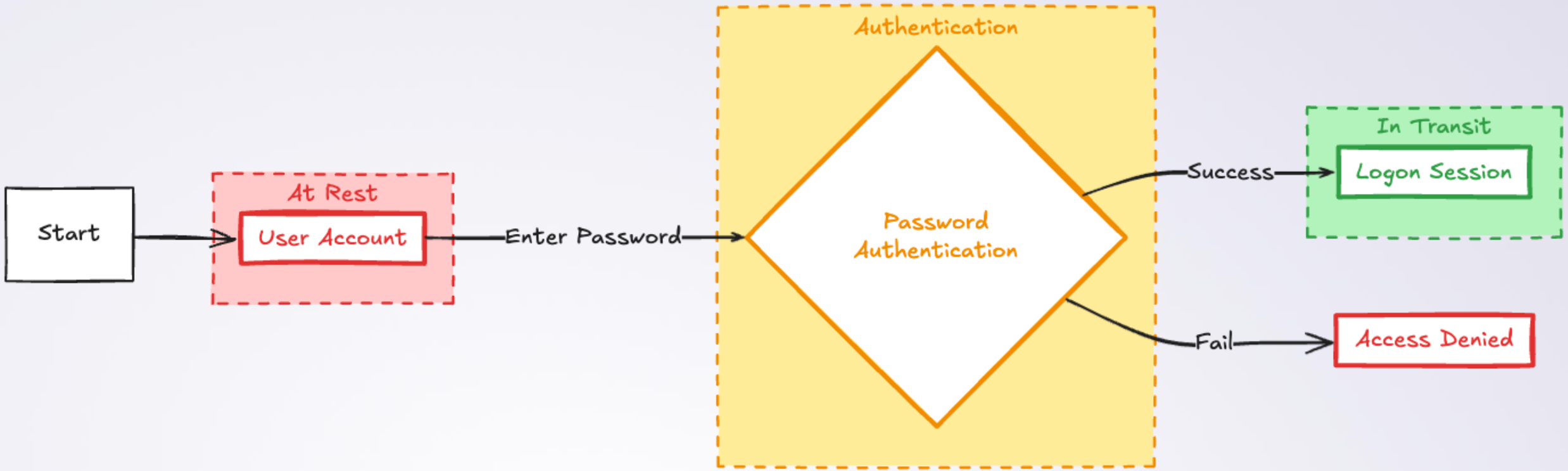
1. Enforce Multi-Factor Authentication on all accounts
2. Set up Network Policy Rules to only allow authorized users or only allow traffic from trusted locations (VPN, Cloud workload NAT, etc.)
3. Impacted organizations should reset and rotate Snowflake credentials

What do all of these recommendations have in common?

*They're solely focused on preventing additional **authentication**!*

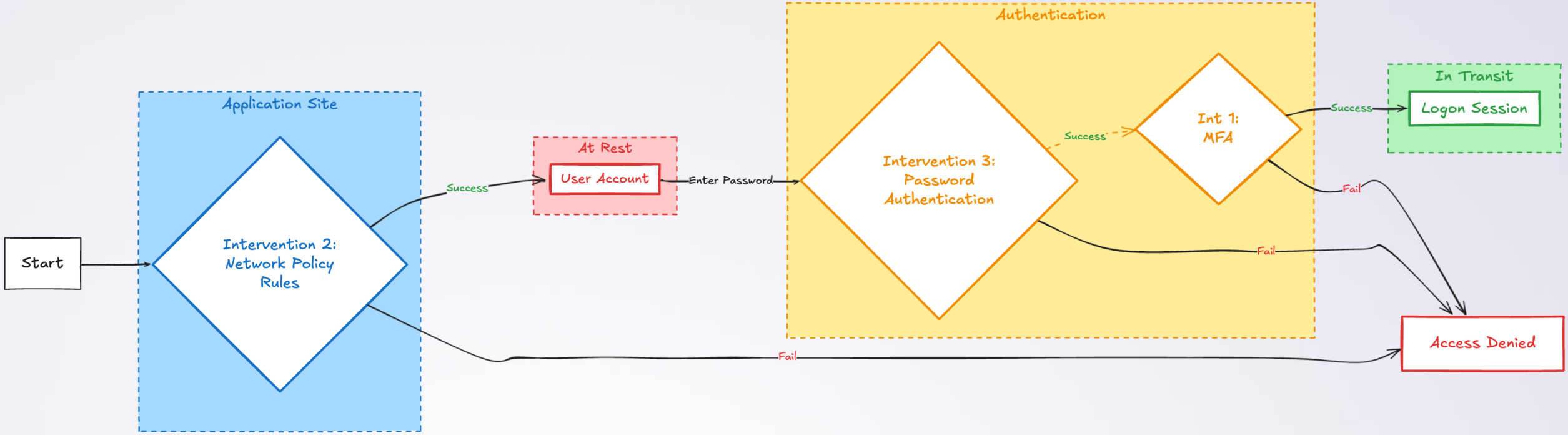
Authentication Flow

Pre Intervention



Authentication Flow

Post Intervention



Do the interventions stop attackers?

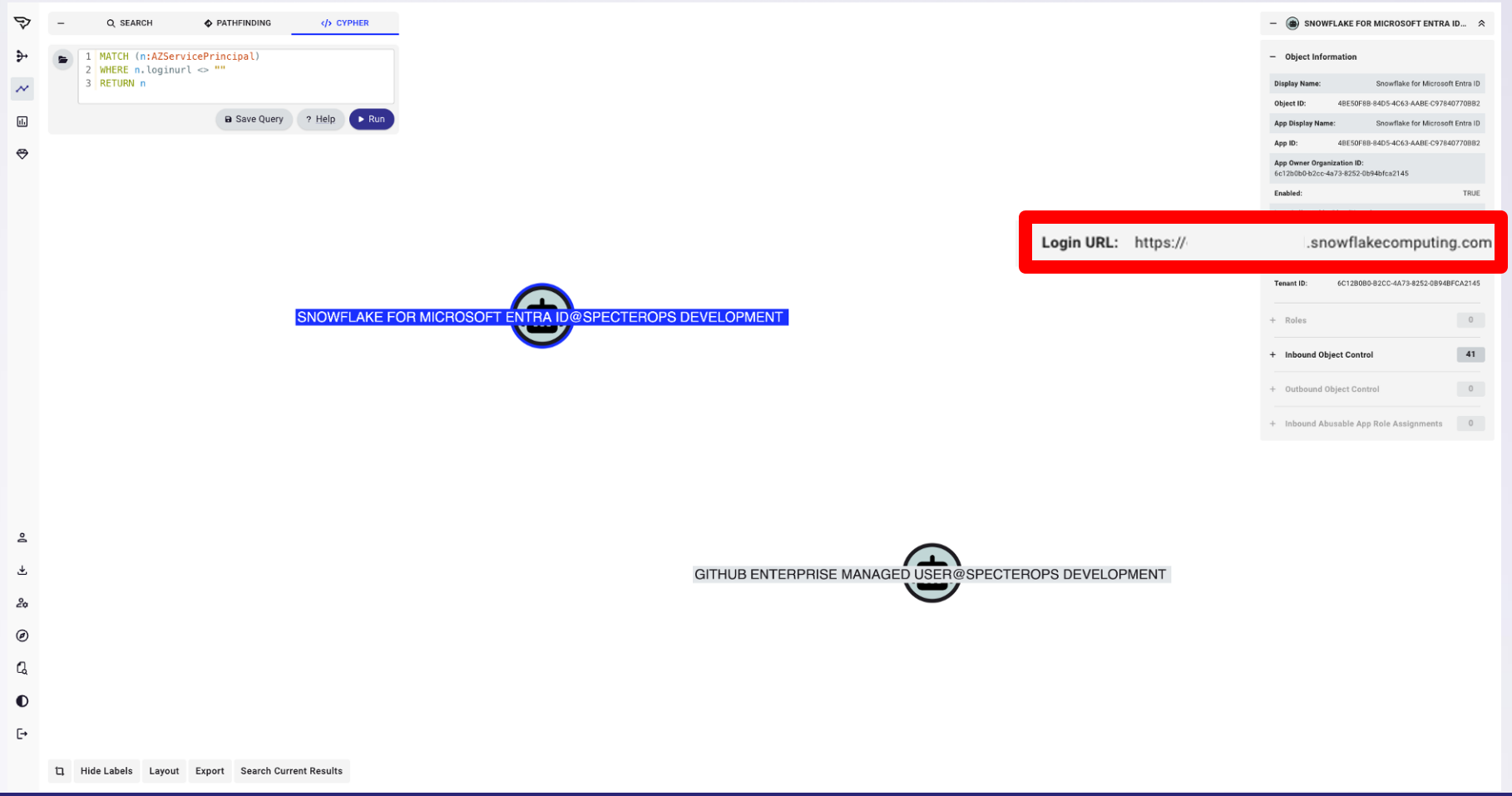
Dump the In-transit Identity

Select Administrator: Windows PowerShell

```
PS C:\Labs\Lab12> .\SharpChrome.exe cookies /statekey:7F426F14C8FAEB454CCC8FDFFE8F4E532D4AC69AA9D9DE3929077266E40706A1 /target:"C:\Labs\Lab12\Cookies"

  v1.11.1

[*] Action: Chrome Saved Cookies Triage

[*] Using AES State Key: 7F426F14C8FAEB454CCC8FDFFE8F4E532D4AC69AA9D9DE3929077266E40706A1

[*] Triaging non-expired cookies. Use '/showall' to display ALL cookies.
[*] Target 'Cookies' File: C:\Labs\Lab12\Cookies

--- Cookies (Path: C:\Labs\Lab12\Cookies) ---

file_path,host,path,name,value,creation_utc,expires_utc,last_access_utc
C:\Labs\Lab12\Cookies,.google.com,/,AEC,AQTF6Hw9SI42zEM2MI484klWHn4IGdFGpiJoX6SrNzJF8agat5i2cTjNCzQ,6/13/2024 2:35:42 AM,12/10/2024 2:35:42 AM,6/13/2024 10
:02:44 PM
C:\Labs\Lab12\Cookies,.google.com,/verify,SNID,AK0uo1RtPiBx4M6qqmcc8TcnWQktqCTxTPUVD_KAHyr7cqGACoeM5fnL1Q5-sCEDywwVgVAryCc9f9gdn9Hj8d0jDA0zIC3dddc,6/13/202
4 2:35:43 AM,12/13/2024 2:35:17 AM,6/13/2024 2:35:43 AM
C:\Labs\Lab12\Cookies,.github.com,/,_octo,GH1.1.1456454573.1718246144,6/13/2024 2:35:44 AM,6/13/2025 2:35:44 AM,6/13/2024 2:40:33 AM
C:\Labs\Lab12\Cookies,.github.com,/,logged_in,no,6/13/2024 2:35:44 AM,6/13/2025 2:35:44 AM,6/13/2024 2:40:33 AM
C:\Labs\Lab12\Cookies,.google.com,/,NID,514=bcMVj_aLFUIvIzVMva_n7-Yd3WeK9qEeu4NNyNUd-NMeBNYRsEE-MMmNFVfk3fketPSTyawE-K7IcuKOh9My8E98drBmnBdy4OW-qh43dy3Zs1f
-8XtuwgmPokfQ4MEdNiL7f2u9ROajTX31YNmR3PMkIYtz7Yu8n-h46ZC_AmBEj_sTALdWlIVjBa4pysRr6j-,6/13/2024 9:53:23 PM,12/13/2024 9:53:23 PM,6/13/2024 10:02:44 PM
C:\Labs\Lab12\Cookies,app.snowflake.com,/,_dd_s,rum=1&id=39eda42e-a3d4-4429-949b-ef7fdf0d9a15&created=1718316206229&expire=1718318340728,6/13/2024 10:24:07
 PM,6/13/2024 10:39:07 PM,6/13/2024 10:24:07 PM


SharpChrome completed in 00:00:01.2803113
```

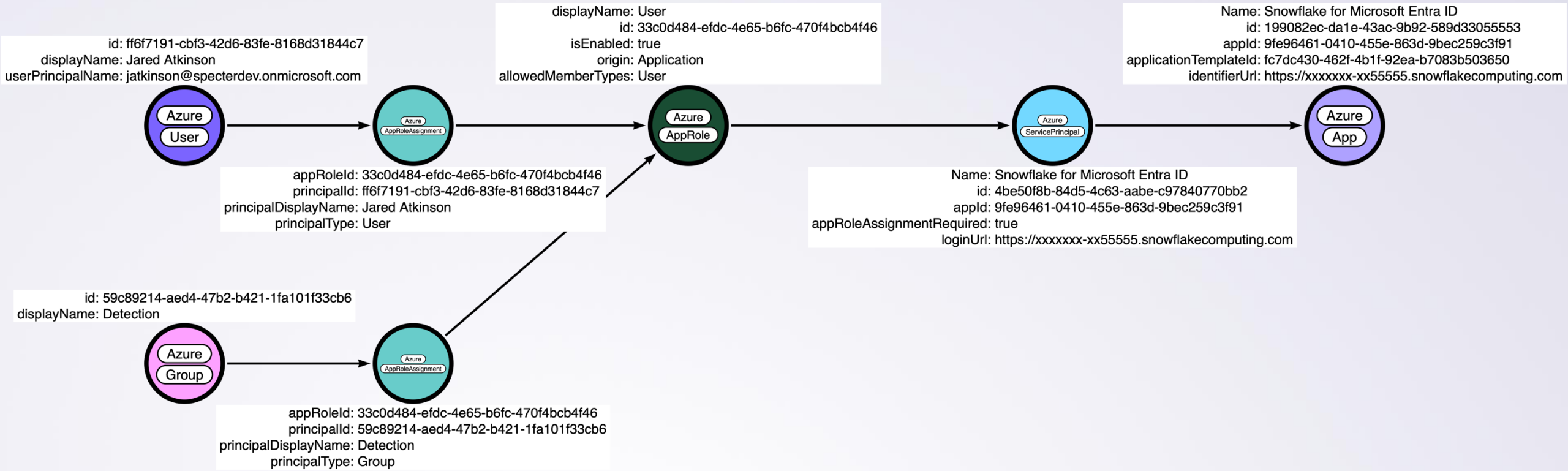
Entra SSO Applications

Identifying Apps that Use Entra SSO



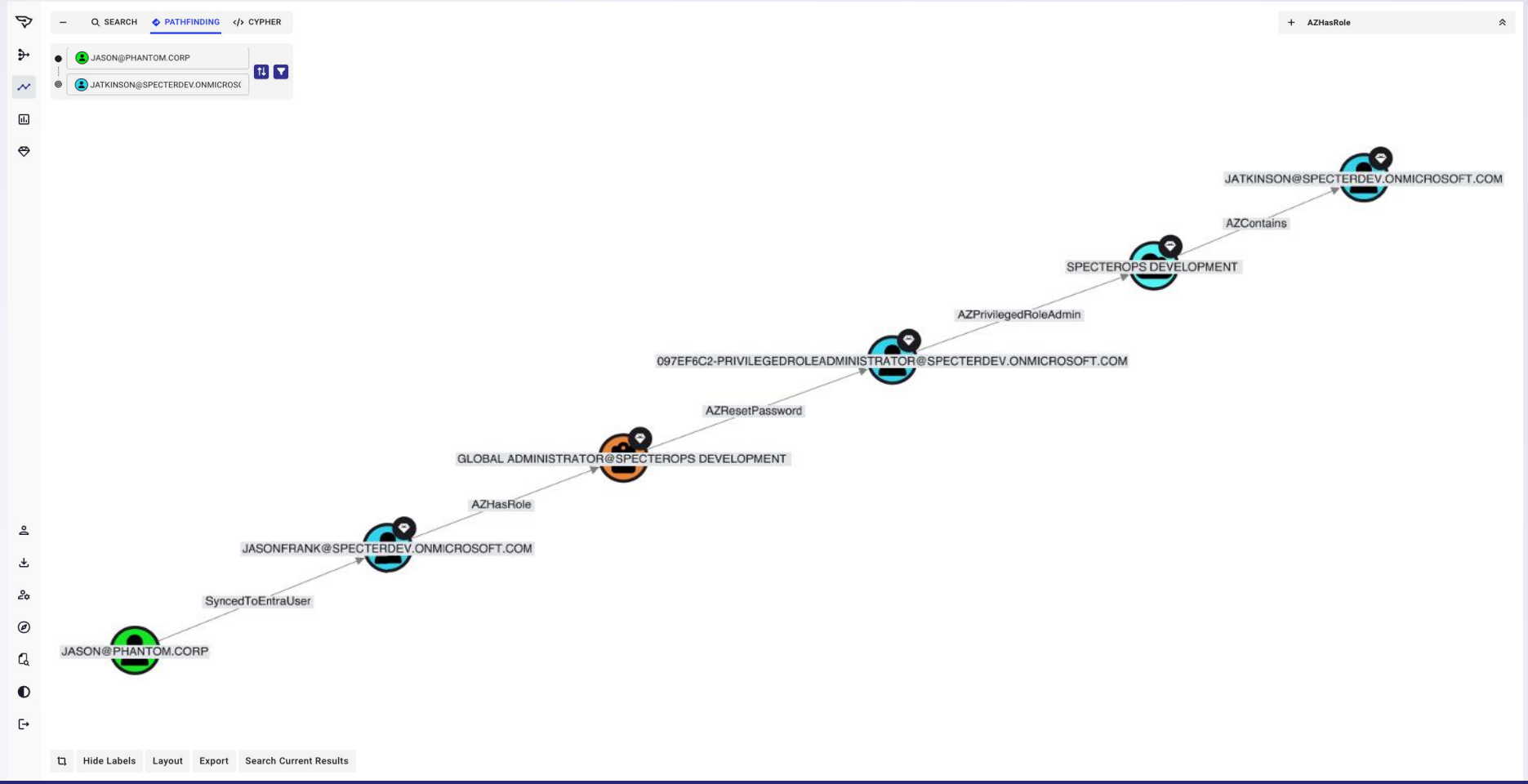
Entra SSO Applications

Granting Entra Users Access to Snowflake



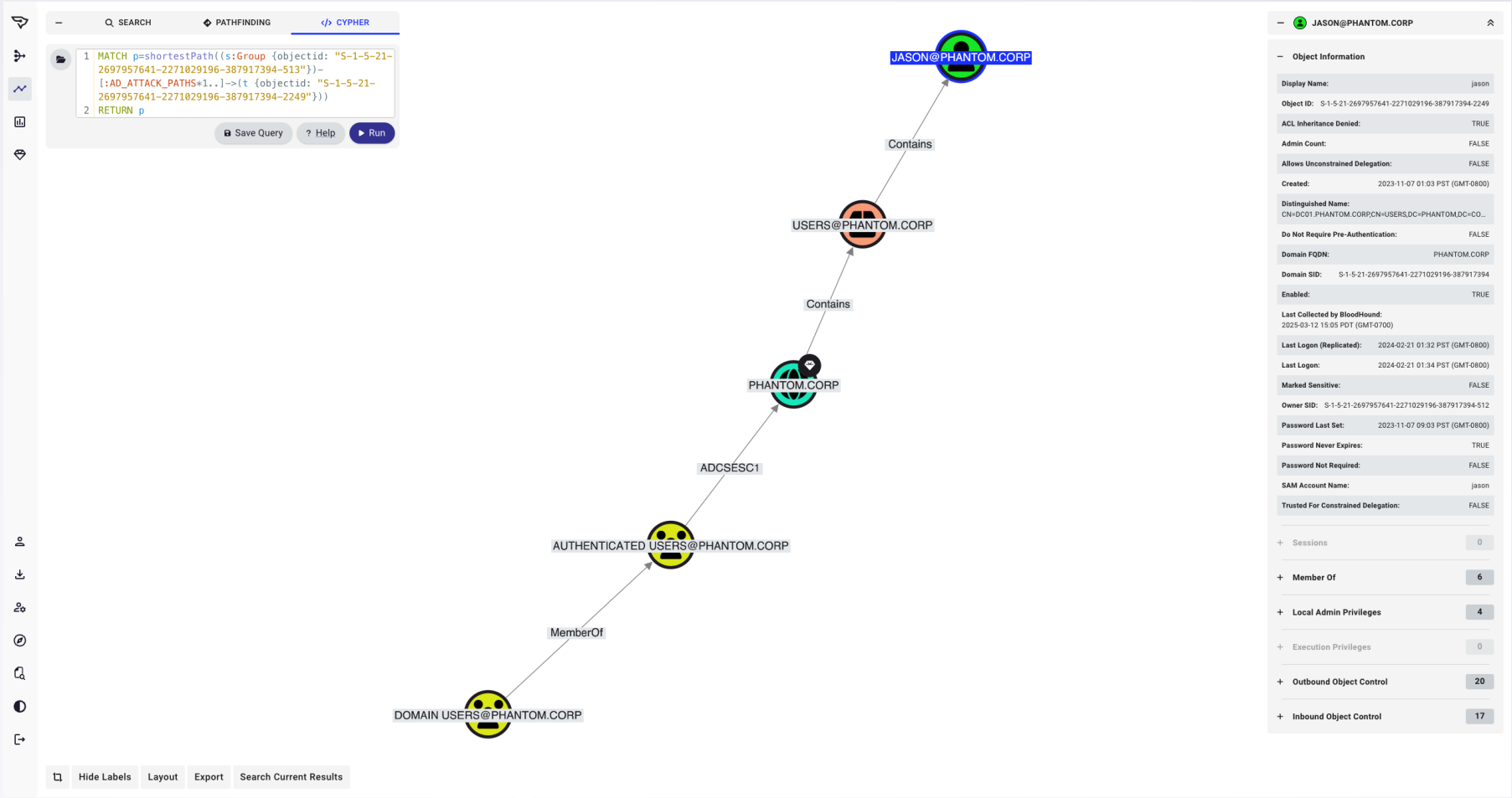
Taking Control of Snowflake

Path from AD User to Entra Account Linked to Snowflake



Taking Control of Snowflake

Phishing Target has Path to Hybrid User



Assume Breach ++

Protecting SaaS Identities

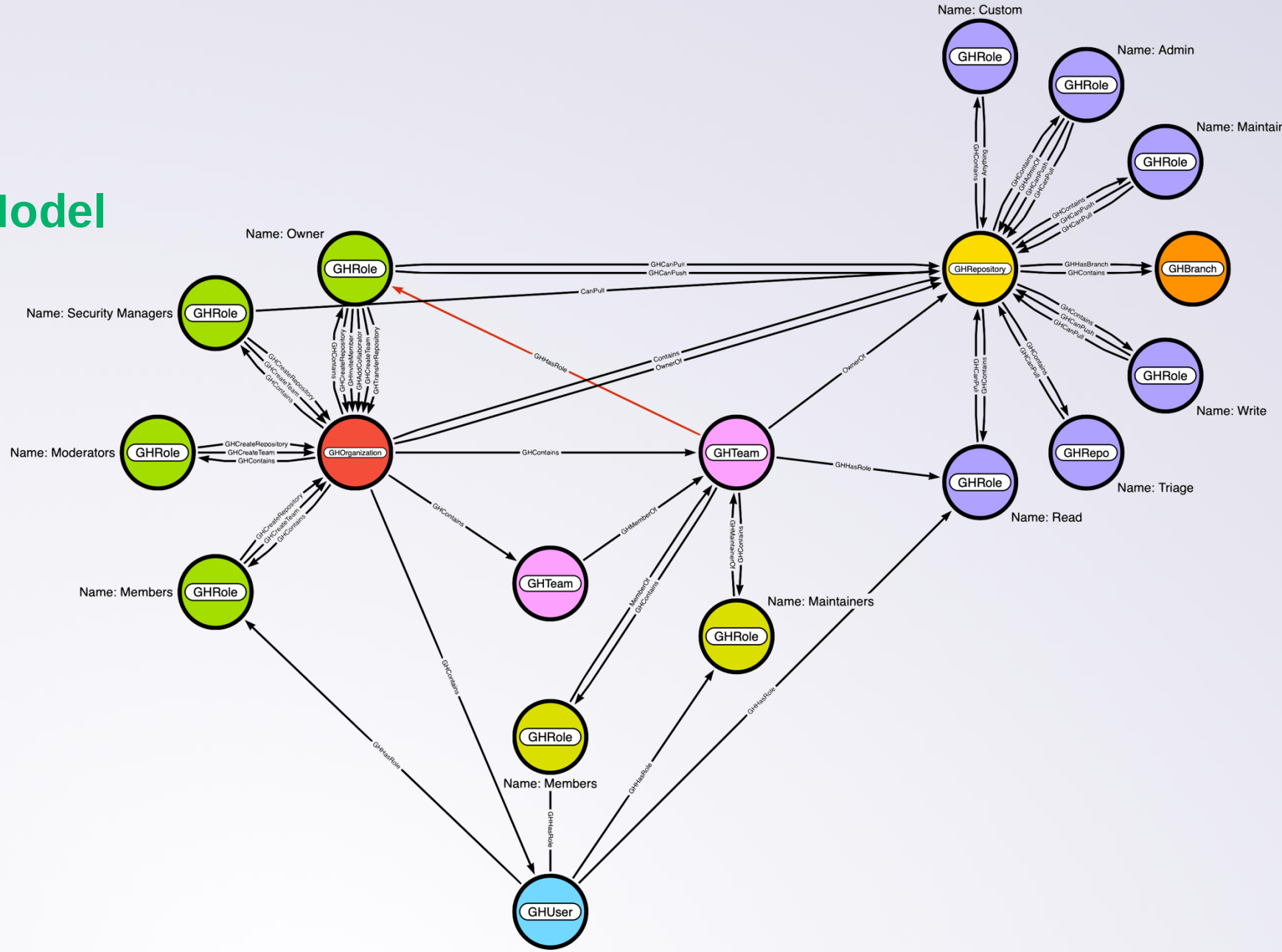
- SaaS users frequently access apps from their on-prem computer
- The administrators of our third-party applications (i.e., Slack, Salesforce, Snowflake, etc.) are not necessarily the same as our on-prem tier 0 administrators.
- As a result, the in-transit form of these identities are rarely as protected as traditional T0 identities.
- **Question:** *How many people use a privileged-access-workstation (PAW) to access GitHub, or Snowflake, or other sensitive third-party apps?*

Case Study

GitHub

GitHub

Permission Model



GitHub

Sidenote: “Classic” Branch Protections

- GitHub branches can have several “classic” protections enabled:
  - Require status checks to pass before merging
  - Require conversation resolution before merging
  - Require signed commits
  - Require linear history
  - Require deployments to succeed before merging
  - ***Lock the branch***
  - ***Restrict who can push to matching branches***
  - ***Require a pull request review before merging***

The ones that we care about

GitHub

Sidenote: “Classic” Branch Protections (ノ°□°)ノ ⌢ ┻━┻

- By default, branch protections don’t apply to people with admin permissions or the permission `bypass_branch_protection`
  - ***UNLESS*** “Do not allow bypassing the above settings” (enforce_admins) is set
- A locked branch makes the branch read-only
- Specific principals can be set to bypass pull request requirements
- The `edit_repo_protections` right and Admin/Owner rights allow you to edit protections

This introduces some complexity for effective-branch-commit rights!

GitHub

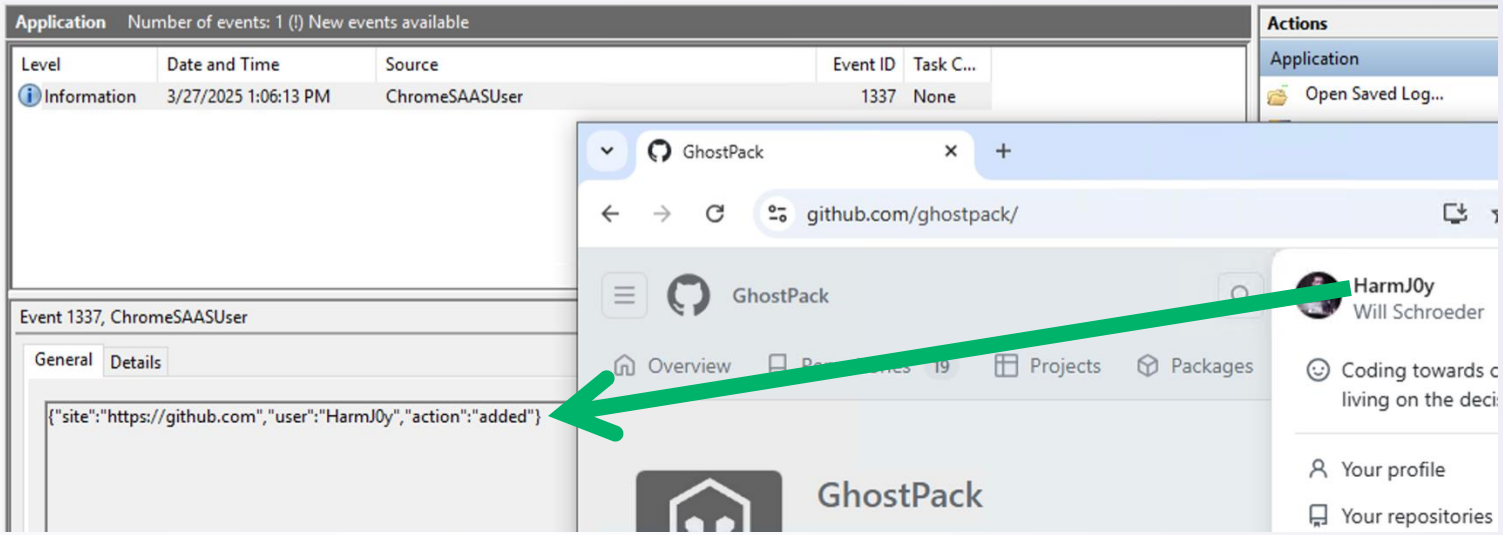
Inbound Access Paths

- Authentication with the browser (cookies)
  - Logging directly into github.com -> **HasGitHubSession**
- Personal access tokens
- GitHub “Apps”
- SSH keys
- OAuth
  - Entra ID/other SSO

GitHub

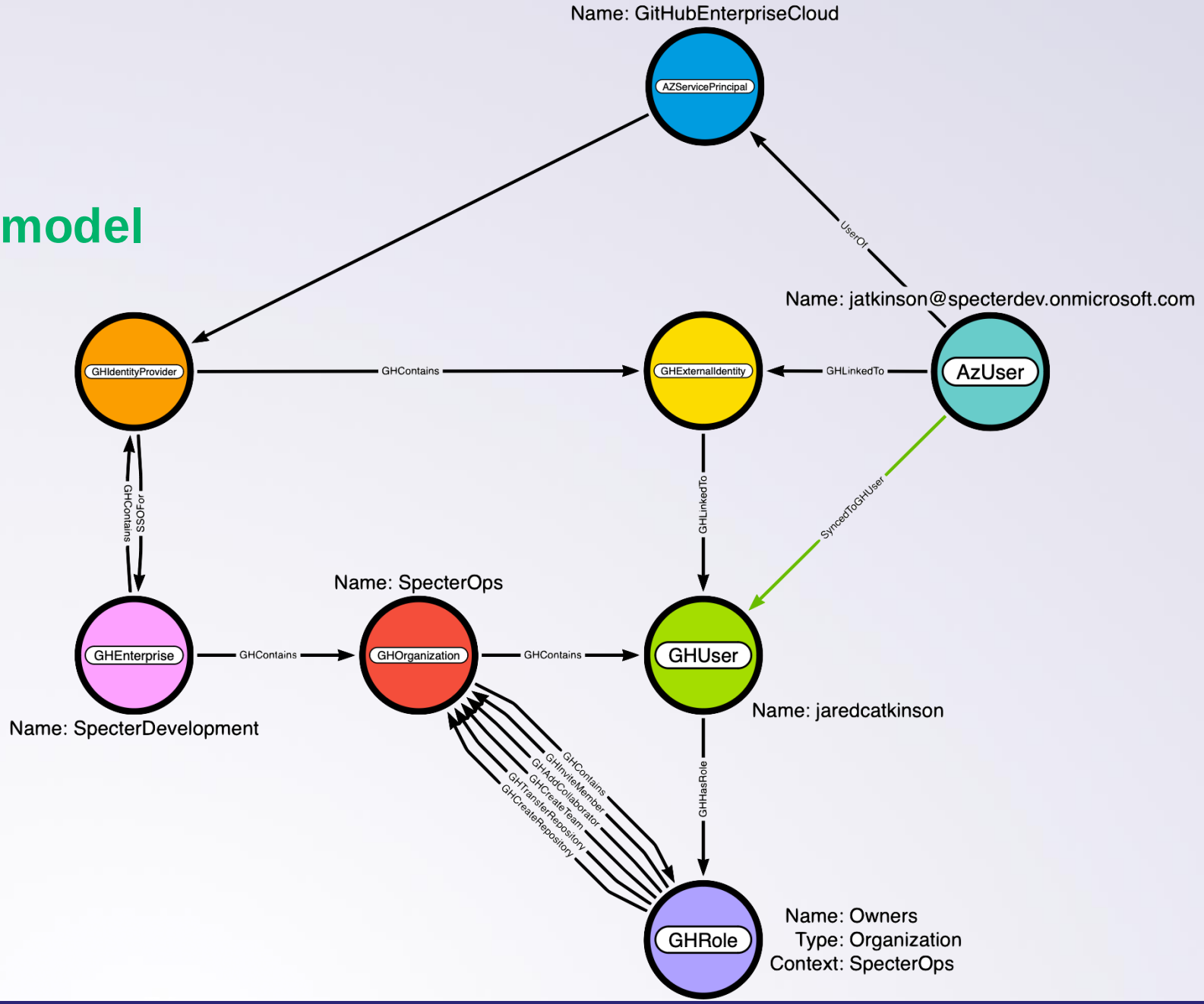
Linking the Local Graph

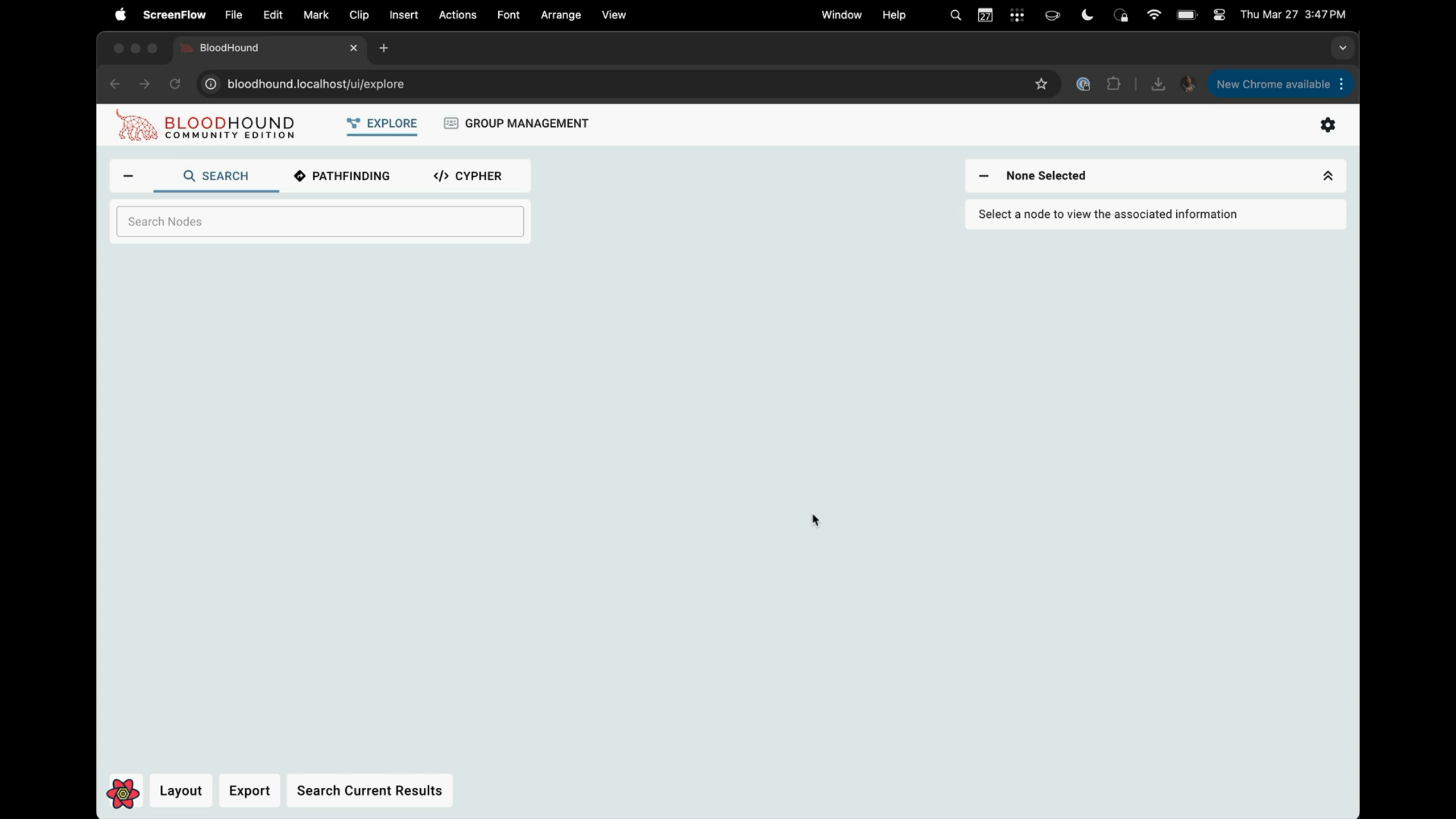
- We developed a proof-of-concept Chromium “Cookie Logger” utilizing Chrome’s Native Messaging for extensions
  - This extension will log SnowFlake, Entra, and GitHub cookie values to the event log
  - We can forward these off to a connector to dynamically build our new edges



GitHub

Enterprise SSO Submodel




ScreenFlow File Edit Mark Clip Insert Actions Font Arrange View Window Help Thu Mar 27 3:47 PM
BloodHound
bloodhound.localhost/ui/explore
New Chrome available
BLOODHOUND COMMUNITY EDITION
EXPLORE
GROUP MANAGEMENT
SEARCH
PATHFINDING
CYPHER
Search Nodes
None Selected
Select a node to view the associated information
Layout
Export
Search Current Results

Conclusions

Conclusions

- BloodHound is a graph-of-graphs of multiple authorization systems
- When the local graph of a SaaS systems is connected to the global graph containing Active Directory, the risk exposure for the SaaS' local graph increases significantly!
- Protecting identities in transit is a different than protecting them at rest
  - Lots of defensive advice is still focused on at-rest (preventing reauthentication)

Thank you!

Any Questions?

Jared Atkinson | jatkinson@specterops.io

Will Schroeder | wschroeder@specterops.io