Extending BloodHound for Red Teamers

Tom Porter (@porterhau5)
MATCH (n:Person {name:'porterhau5'}) RETURN n

- Flow data analytics
- Penetration testing
- Red teaming
- Password analysis & wordlist generation

Tom Porter
@porterhau5
porterhau5.com

SR. SECURITY CONSULTANT
FUSIONX
Extending BloodHound

- Handling Compromised Nodes
- Visualize Deltas in Privilege Gains
- Automating via Neo4j REST API

[Diagram showing network nodes and connections]

- Adding Properties and Relationships
- Custom Queries
- UI Enhancements

https://github.com/porterhau5/BloodHound-Owned
https://github.com/porterhau5/BloodHound (forked)
BloodHound

• Released at DEF CON 24
  – Andy Robbins (@_wald0)
  – Rohan Vazarkar (@CptJesus)
  – Will Schroeder (@harmj0y)
• Attack graphs for Active Directory

https://github.com/BloodHoundAD/BloodHound
BloodHound Stack

LINKURIOUS + ELECTRON = BLOODHOUND

neo4j

REST

CSV

Bolt
SharpHound.ps1/SharpHound.exe

Originally leveraged Powerview (PowerShell)

Rewritten in C#: https://github.com/BloodHoundAD/SharpHound
Neo4j: A Graph Database

Property Graph Model
- Nodes
- Relationships
- Labels
- Relationship types
- Properties
- Paths

Person
Name: ‘Tyrion’

MemberOf

House
Name: ‘Lannister’

Sword
Name: ‘Ice’

ReforgedInto

*Sip Ned

HasItem

Sword
Name: ‘Widows Wail’
## Nodes, Labels, and Properties

### App Icons

<table>
<thead>
<tr>
<th>Icon</th>
<th>Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>🌍</td>
<td>Name: ‘INTERNAL.LOCAL’</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Icon</th>
<th>User</th>
</tr>
</thead>
<tbody>
<tr>
<td>👤</td>
<td>Name: ‘<a href="mailto:JDOE@INTERNAL.LOCAL">JDOE@INTERNAL.LOCAL</a>’</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Icon</th>
<th>Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>🐦</td>
<td>Name: ‘DOMAIN <a href="mailto:ADMINS@INTERNAL.LOCAL">ADMINS@INTERNAL.LOCAL</a>’</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Icon</th>
<th>Computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>🖥</td>
<td>Name: ‘MGMT3.INTERNAL.LOCAL’</td>
</tr>
</tbody>
</table>
More added in 1.3: – The ACL Attack Path Update: https://wald0.com/?p=112
A Path for Escalation

- AdminTo: JDOE
- HasSession: APPDEV
- MemberOf: BSMITH

- AdminTo: SRVADM
- MemberOf: HELPDESK

- HasSession: FILESRV01
- MemberOf: BSMITH-DA
- MemberOf: DOMAIN ADMINS
BloodHound

Pre-Built Analytics Queries

- Find all Domain Admins
- Find Shortest Paths to Domain Admins
- Find logged in admins
- Find Top 10 Users with Most Sessions
- Find Top 10 Computers with Most Sessions
- Find Top 10 Users with Most Local Admin Rights
- Find Top 10 Computers with Most Admins

Graph

Search Container

Menu

Zoom

Raw Query

MATCH (n:Group) WHERE n.name =~ (name) WITH n MATCH (n)<-[r:MemberOf*1..]-[m] RETURN n,r,m
Node Info Tab

User Info
- Name: CBARCLAY@INTERNAL.LOCAL
- Display Name: No Data
- Password Last Changed: No Data
- Last Logon: No Data
- Enabled: No Data
- Email: No Data
- Service Principal Names: None
- Sessions: 2

Group Membership
- First Degree Group Memberships: 4
- Unrolled Group Memberships: 4
- Foreign Group Membership: 1

Local Admin Rights
- First Degree Local Admin: 1
- Group Delegated Local Admin Rights: 20
- Derivative Local Admin Rights: 264

Outbound Object Control
- First Degree Object Control: 0
- Group Delegated Object Control: 467
- Transitive Object Control: 473

Inbound Object Control
- Explicit Object Controllers: 5
- Unrolled Object Controllers: 18
- Transitive Object Controllers: 50
BPT: Query Debug Mode

Learn Cypher as you go!
Cypher using relationship ‘likes’

(a) -[:LIKE\_S]-(b)

https://neo4j.com/developer/cypher-query-language/
Only MATCH a Label

```
MATCH (n:Computer) RETURN (n)
```

Displaying 253 nodes, 12 relationships (completed with 12 additional relationships).
```sql
MATCH (n:Computer) RETURN n.name
```

<table>
<thead>
<tr>
<th>n.name</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATABASE4.INTERNAL.LOCAL</td>
</tr>
<tr>
<td>WEBSERVER3.INTERNAL.LOCAL</td>
</tr>
<tr>
<td>MANAGEMENT7.INTERNAL.LOCAL</td>
</tr>
<tr>
<td>SECRET2.INTERNAL.LOCAL</td>
</tr>
<tr>
<td>CONTRACTOR1.INTERNAL.LOCAL</td>
</tr>
<tr>
<td>LAPTOP11.INTERNAL.LOCAL</td>
</tr>
<tr>
<td>DESKTOP40.EXTERNAL.LOCAL</td>
</tr>
<tr>
<td>DESKTOP32.EXTERNAL.LOCAL</td>
</tr>
<tr>
<td>DESKTOP3.EXTERNAL.LOCAL</td>
</tr>
<tr>
<td>DESKTOP7.EXTERNAL.LOCAL</td>
</tr>
<tr>
<td>DESKTOP39.EXTERNAL.LOCAL</td>
</tr>
<tr>
<td>R_AND_D_13.EXTERNAL.LOCAL</td>
</tr>
<tr>
<td>DESKTOP10.EXTERNAL.LOCAL</td>
</tr>
<tr>
<td>DESKTOP30.EXTERNAL.LOCAL</td>
</tr>
<tr>
<td>LAPTOP1.INTERNAL.LOCAL</td>
</tr>
<tr>
<td>HR10.INTERNAL.LOCAL</td>
</tr>
</tbody>
</table>

Started streaming 253 records after 7 ms and completed after 8 ms.
MATCH (n:User) WHERE n.name='CBARCLAY@INTERNAL.LOCAL' RETURN n
$ MATCH (n:User)-[:AdminTo]->(m) WHERE n.name='CBARCLAY@INTERNAL.LOCAL' RETURN n,m
Find Derivative Local Admin Access

$ MATCH p=(:User {name:'CBARCLAY@INTERNAL.LOCAL'})-[*1..3]->(:Computer) RETURN p

Displaying 25 nodes, 24 relationships.
Extensions

• Owned / Wave
• Password Reuse via SharesPasswordWith
• Blacklisting Nodes & Relationships
• Integrating Network Connections
All Hope Isn’t Lost

Password Reuse
- Same password between accounts
- Shared local administrator password

Kerberoasting
- Abuse Kerberos SPNs linked to domain accounts
- Escalate from low-privileged user to service account

Share Plundering
- Notes.txt
- Configuration files
An Example Engagement

1. LLMNR/NBNS MiTM: Capture and crack NTLMv2 hashes
   - 3 accounts obtained

2. Password spray: Fall2017
   - 4 more accounts obtained

3. Password reuse: user account and admin account
   - 1 more account obtained (admin)
   - 1 computer obtained

4. Secretsdump: dump local account hashes from host
   - 3 more accounts obtained (local)

5. Pass-the-Hash: local administrator hash against other hosts
   - 34 more computers obtained
Let's Use BloodHound to Answer a Question

When a new set of nodes are owned by some method, what other nodes can we now collaterally reach?

“Wave”
‘Owned’ and ‘Wave’ Properties

‘owned’
- Method used to compromise the node:
  • LLMNR wpad
  • Password spray
  • Mimikatz
  • Found on SMB share
  • Phished

‘wave’
- Number representing the order in which the node was compromised:
  • 1
  • 2
  • 3, etc.
SET Properties

1. LLMNR/NBNS poisoning for WPAD: BLOPER@INTERNAL.LOCAL JCARNEAL@INTERNAL.LOCAL

```sql
$ MATCH (n:User {name:'BLOPER@INTERNAL.LOCAL'}) SET n.owned='LLMNR wpad', n.wave=1
```

Set 2 properties, statement completed in 2 ms.

```sql
$ MATCH (n:User {name:'JCARNEAL@INTERNAL.LOCAL'}) SET n.owned='NBNS wpad', n.wave=1
```

Set 2 properties, statement completed in 1 ms.
Node Info Addition

[Image of a user interface showing details about a node named BLOPER@INTERNAL.LOCAL. The interface is divided into sections for Database Info, Node Info, and Queries. The Node Info section includes:

- **Password Last Changed**: No Data
- **Last Logon**: No Data
- **Enabled**: No Data
- **Email**: No Data
- **Service Principal Names**: None
- **Sessions**: 0

**Group Membership**
- First Degree Group Memberships: 1
- Unrolled Group Memberships: 1
- Foreign Group Membership: 0

**Local Admin Rights**
- First Degree Local Admin: 1
- Group Delegated Local Admin Rights: 0
- Derivative Local Admin Rights: 1

**Outbound Object Control**
- First Degree Object Control: 0
- Group Delegated Object Control: 0
- Transitive Object Control: 1

**Inbound Object Control**
- Explicit Object Controllers: 5
- Unrolled Object Controllers: 19
- Transitive Object Controllers: 51

**Owned in Wave**
- No owned in wave

**Owned via Method**
- LLMNR wpad]
$ MATCH (n)-[r*]->(m) WHERE n.wave=1 RETURN n,r,m

Displaying 4 nodes, 3 relationships.
MATCH (n)-[*]->(m) WHERE n.wave=1 RETURN n,r,m
Add Nodes to 1st Wave

\$ MATCH (n)-[r*]->(m) WHERE n.wave=1 SET m.wave=1

Set 3 properties, statement completed in 9 ms.

\$ MATCH (n) WHERE n.wave=1 RETURN n.name

n.name
- DOMAIN USERS@INTERNAL.LOCAL
- BLOPER@INTERNAL.LOCAL
- JCARNEAL@INTERNAL.LOCAL
- SYSTEM38@INTERNAL.LOCAL
2nd Set of Owned Nodes

2. Password spray (Fall 2017):
   ZDEVEN@INTERNAL.LOCAL
   BPICKEREL@INTERNAL.LOCAL

$ MATCH (n:User {name:'ZDEVEN@INTERNAL.LOCAL'}) SET n.owned='Password spray', n.wave=2

Set 2 properties, statement completed in 1 ms.

$ MATCH (n:User {name:'BPICKEREL@INTERNAL.LOCAL'}) SET n.owned='Password spray', n.wave=2

Set 2 properties, statement completed in 1 ms.
Collateral Spread for 2nd Wave

Also in Wave 1
Hiding Nodes from Previous Waves

$ MATCH (n)-[r*]->(m) WHERE n.wave=2 AND NOT EXISTS(m.wave) RETURN n, r, m$
SET ‘wave’ for Nodes

$ MATCH (n)-[r*]->(m) WHERE n.wave=2 AND NOT EXISTS(m.wave) SET m.wave=2

Set 6 properties, statement completed in 9 ms.

Completed after 9 ms.
Delta in Access
Automation

bh-owned.rb – Ruby script to interacts with Neo4j

Usage: ruby bh-owned.rb [options]

Server Details:
- \( \texttt{-u, --username <username>} \) Neo4j database username (default: 'neo4j')
- \( \texttt{-p, --password <password>} \) Neo4j database password (default: 'BloodHound')
- \( \texttt{-U, --url <url>} \) URL of Neo4j RESTful host (default: 'http://127.0.0.1:7474/')

Owned/Wave/SPW:
- \( \texttt{-a, --add <file>} \) add 'owned' and 'wave' property to nodes in <file>
- \( \texttt{-A, --add-no-wave <file>} \) add 'owned' property to nodes in <file> (skip 'wave' property)
- \( \texttt{-w, --wave <num>} \) value to set 'wave' property (override default behavior)
- \( \texttt{-s, --spw <file>} \) add 'SharesPasswordWith' relationship between all nodes in <file>

Blacklisting:
- \( \texttt{-b, --bl-node <file>} \) add 'blacklist' property to nodes in <file>
- \( \texttt{-B, --bl-rel <file>} \) add 'blacklist' property to relationships in <file>
- \( \texttt{-r, --remove-bl-node <file>} \) remove 'blacklist' property from nodes in <file>
- \( \texttt{-R, --remove-bl-rel <file>} \) remove 'blacklist' property from relationships in <file>

Connections:
- \( \texttt{-c, --connections <file>} \) add connection info from netstat <file>
- \( \texttt{-d, --dns <file>} \) contains DNS mapping of IP to computer name (10.2.3.4,srv1.int.local)

Misc Queries:
- \( \texttt{-n, --nodes} \) get all node names
- \( \texttt{-e, --examples} \) reference doc of custom Cypher queries for BloodHound
- \( \texttt{-reset} \) remove all custom properties and SharesPasswordWith relationships
3. SMB Share Plundering:
   SMADDUX@INTERNALLOCAL
   QBULLIS@EXTERNALLOCAL

```
6 [05/17/17 14:31] [porterhaus5@compute:~/BloodHound-Owned] cat 3rd-wave.txt
SMADDUX@INTERNAL.LOCAL,Creds in file on DATABASE5
QBULLIS@EXTERNAL.LOCAL,Creds in file on DATABASE5
```

1. Create Cypher query to SET properties
2. Create Cypher query for wave of compromise
3. Wrap it in JSON
4. POST to RESTful endpoint
5. Parse API response, display to user
Wave 3 in BloodHound
Custom Queries

- Find all owned Domain Admins
- Find Shortest Paths from owned node to Domain Admins
- Show Wave
- Highlight Delta for Wave
- Find Clusters of Password Reuse
- Show Blacklisted Nodes
- Show Blacklisted Relationships
- Show Blacklist

Mac: ~/Library/Application Support/bloodhound/customqueries.json
Windows: %APPDATA%\Roaming\bloodhound\customqueries.json
### Custom Query Syntax

```
{
    "name": "Find all owned Domain Admins",
    "requireNodeSelect": false,
    "query": "MATCH (n:Group) WHERE n.name =~ {name} WITH n MATCH p=(n)<-[r:MemberOf*1..]-(m)
WHERE exists(m.owned) RETURN nodes(p),relationships(p)",
    "allowCollapse": false,
    "props": {"name": "(?i).*DOMAIN ADMIN.*"}
}
```

- **name**: Display name in BloodHound UI
- **requireNodeSelect**: Require input from the user
- **query**: Cypher query to run
- **allowCollapse**: Allow Nodes to be collapsed
- **props**: Variable used in “query” statement, helps with performance
4. Mimikatz, Local accounts:

BGRIFFIN@EXTERNAL.LOCAL
FILESERVER6.INTERNAL.LOCAL

```
[05/17/17 20:45][porterhau5@compute:~/BloodHound-Owned] cat 4th-wave.txt
BGRIFFIN@EXTERNAL.LOCAL, Mimikatz on MANAGEMENT3
FILESERVER6.INTERNAL.LOCAL, Local account password guess
```
Find all owned Domain Admins

Custom Queries:
- Find all owned Domain Admins
- Find Shortest Paths from owned node to Domain Admins
- Show wave
- Show delta for wave

Highlight our owned node
Find Shortest Paths from owned node to Domain Admins
Show Wave

Custom Queries

Find all owned Domain Admins
Find Shortest Paths from owned node to Domain Admins

Show wave
Show delta for wave
Show Wave

Can also run “Show wave” query by clicking wave number in Node Info
Show Delta for Wave
Show Delta for 2nd Wave

Nodes added in wave 2

- ZDEVENS@INTERNAL.LOCAL
- MANAGEMENT3.INTERNAL.LOCAL
- DATABASES.INTERNAL.LOCAL
- BPICKEREL@INTERNAL.LOCAL
- BACKUP_SVC@INTERNAL.LOCAL
- JCARNEAL@INTERNAL.LOCAL
- BLOPER@INTERNAL.LOCAL
- SYSTEM38.INTERNAL.LOCAL
- CONTRACTINGS@INTERNAL.LOCAL
- BACKUP3@INTERNAL.LOCAL
- DOMAIN USERS@INTERNAL.LOCAL
Extensions

- Owned / Wave
- Password Reuse via SharesPasswordWith
- Blacklisting Nodes & Relationships
- Integrating Network Connections
Representing Password Reuse

- MemberOf
- AdminTo
- HasSession
- TrustedBy
- SharesPasswordWith
- SharesPasswordWith
A Path for Escalation

- AdminTo
  - JDOE
  - SRVADM
  - FILESRV01

- HasSession
  - APPDEV
  - BSMITH
  - BSMITH-DA

- MemberOf
  - BSMITH
  - HELPDESK
  - DOMAIN ADMINS
SharesPasswordWith:

- JDOE AdminTo APPDEV
- FILESRV01 SharesPasswordWith SRVADM
- APPDEV HasSession BSMITH
- BSMITH MemberOf BSMITH-DA
- BSMITH-DA HasSession HELPDESK
- HELPDESK MemberOf DOMAIN ADMINS
SharesPasswordWith: Users

AdminTo

HasSession

MemberOf

JDOE

APPDEV

BSMITH

SRVADM

HELPDESK

FILESRV01

BSMITH-DA

DOMAIN ADMINS
SharesPasswordWith: Manual

Accounts with same password:
BREYES@INTERNAL.LOCAL
BREYES.ADMIN@INTERNAL.LOCAL

```
MATCH (a {name:'BREYES@INTERNAL.LOCAL'}),(b {name:'BREYES.ADMIN@INTERNAL.LOCAL'})
WITH a,b MERGE (a)-[:SharesPasswordWith]->(b)
WITH a,b MERGE (a)<-[::SharesPasswordWith]-(b)
```

$ MATCH (a {name:'BREYES@INTERNAL.LOCAL'}),(b {name:'BREYES.ADMIN@INTERNAL.LOCAL'}) WITH a,b MERGE (a)-[:Sh...

Created 2 relationships, completed after 115 ms.
Computers with same local admin password
Find Clusters of Password Reuse
Shortest Path to DA using

BREYES
BREYES.ADMIN
ZDEVENS
DOMAIN ADMINS
BLOPER
BPICKEREL

Start typing to search for a node...
Extensions

• Owned / Wave
• Password Reuse via SharesPasswordWith
• Blacklisting Nodes & Relationships
• Integrating Network Connections
Blacklisting

AdminTo

HasSession

JDOE → APPDEV → BSMITH
Blacklisting

Computer Offline
Blacklisting

Computer Offline

Session Expired
Blacklisting

Computer Offline
Session Expired
User Disabled
Adding to the Blacklist

Via the UI

Via bh-owned.rb

6 [10/26/17 11:51][porterhau5@compute:~/BloodHound-Owned] cat blacklist-nodes.txt
MANAGEMENT3.INTERNAL.LOCAL
BGRiffin@EXTERNAL.LOCAL
BPICKEREL@INTERNAL.LOCAL
DESKTOP20.EXTERNAL.LOCAL

6 [10/26/17 11:51][porterhau5@compute:~/BloodHound-Owned] ruby bh-owned.rb -b blacklist-nodes.txt
[*] Using default username: neo4j
[*] Using default password: BloodHound
[*] Using default URL: http://127.0.0.1:7474/
[+] Success, marked 'MANAGEMENT3.INTERNAL.LOCAL' as blacklisted
[+] Success, marked 'BGRiffin@EXTERNAL.LOCAL' as blacklisted
[+] Success, marked 'BPICKEREL@INTERNAL.LOCAL' as blacklisted
[+] Success, marked 'DESKTOP20.EXTERNAL.LOCAL' as blacklisted
**Blacklist – “Find all Domain Admins”**

**Original Query**

```
MATCH p=(n)-[r:MemberOf*1..]->(m:Group {name:"DOMAIN ADMIN@EXTERNAL.LOCAL"})
RETURN p
```

**Remove paths that contain “blacklist” property**

```
MATCH p=(n)-[r:MemberOf*1..]->(m:Group {name:"DOMAIN ADMIN@EXTERNAL.LOCAL"})
WHERE NONE (x IN nodes(p) WHERE exists(x.blacklist))
AND NONE (x IN relationships(p) WHERE exists(x.blacklist))
RETURN p
```
Viewing the Blacklist

Custom Queries
- Find all owned Domain Admins
- Find Shortest Paths from owned node to Domain Admins
- Show Wave
- Highlight Delta for Wave
- Find Clusters of Password Reuse
- Show Blacklisted Nodes
- Show Blacklisted Relationships

Blacklisted nodes & relationships are filtered out of other Custom Queries

BPICKEREL@INTERNAL.LOCAL

AdminTo

ZDEVENS@INTERNAL.LOCAL

MANAGEMENT3.INTERNAL.LOCAL

BGRIFFIN@EXTERNAL.LOCAL

HasSession

DESKTOP20.EXTERNAL.LOCAL

DESKTOP21.EXTERNAL.LOCAL

JANTHONY@EXTERNAL.LOCAL
Extensions

• Owned / Wave
• Password Reuse via SharesPasswordWith
• Blacklisting Nodes & Relationships
• Integrating Network Connections
## Netstat Connections

<table>
<thead>
<tr>
<th>Proto</th>
<th>Local Address</th>
<th>Foreign Address</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>0.0.0.0:445</td>
<td>0.0.0.0:0</td>
<td>LISTENING</td>
</tr>
<tr>
<td>TCP</td>
<td>10.1.8.83:445</td>
<td>10.1.2.118:53210</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>TCP</td>
<td>10.1.8.83:49188</td>
<td>54.91.1.122:443</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>TCP</td>
<td>10.1.8.83:49230</td>
<td>10.1.17.47:1433</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>TCP</td>
<td>10.1.8.83:49276</td>
<td>10.1.6.115:22</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>TCP</td>
<td>10.1.8.83:49295</td>
<td>10.1.72.47:80</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>TCP</td>
<td>10.1.8.83:49296</td>
<td>52.27.96.23:443</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>TCP</td>
<td>10.1.8.83:49297</td>
<td>10.1.7.18:135</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>TCP</td>
<td>10.1.8.83:49298</td>
<td>10.1.7.18:3389</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>Proto</td>
<td>Local Address</td>
<td>Foreign Address</td>
<td>State</td>
</tr>
<tr>
<td>-------</td>
<td>---------------------</td>
<td>----------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>TCP</td>
<td>0.0.0.0:445</td>
<td>0.0.0.0:0</td>
<td>LISTENING</td>
</tr>
<tr>
<td>TCP</td>
<td>10.1.8.83:445</td>
<td>10.1.2.118:53210</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>TCP</td>
<td><strong>10.1.8.83:49188</strong></td>
<td><strong>54.91.1.122:443</strong></td>
<td><strong>ESTABLISHED</strong></td>
</tr>
<tr>
<td>TCP</td>
<td>10.1.8.83:49230</td>
<td>10.1.17.47:1433</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>TCP</td>
<td>10.1.8.83:49276</td>
<td>10.1.6.115:22</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>TCP</td>
<td><strong>10.1.8.83:49295</strong></td>
<td><strong>10.1.72.47:80</strong></td>
<td><strong>ESTABLISHED</strong></td>
</tr>
<tr>
<td>TCP</td>
<td><strong>10.1.8.83:49296</strong></td>
<td><strong>52.27.96.23:443</strong></td>
<td><strong>ESTABLISHED</strong></td>
</tr>
<tr>
<td>TCP</td>
<td>10.1.8.83:49297</td>
<td>10.1.7.18:135</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>TCP</td>
<td>10.1.8.83:49298</td>
<td>10.1.7.18:3389</td>
<td>ESTABLISHED</td>
</tr>
</tbody>
</table>
## Netstat Connections

<table>
<thead>
<tr>
<th>Proto</th>
<th>Local Address</th>
<th>Foreign Address</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>0.0.0.0:445</td>
<td>0.0.0.0:0</td>
<td>LISTENING</td>
</tr>
<tr>
<td>TCP</td>
<td>10.1.8.83:445</td>
<td>10.1.2.118:53210</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>TCP</td>
<td>10.1.8.83:49188</td>
<td>54.91.1.122:443</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>TCP</td>
<td>10.1.8.83:49230</td>
<td>10.1.17.47:1433</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>TCP</td>
<td><strong>10.1.8.83:49276</strong></td>
<td><strong>10.1.6.115:22</strong></td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>TCP</td>
<td>10.1.8.83:49295</td>
<td>10.1.72.47:80</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>TCP</td>
<td>10.1.8.83:49296</td>
<td>52.27.96.23:443</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>TCP</td>
<td>10.1.8.83:49297</td>
<td>10.1.7.18:135</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>TCP</td>
<td><strong>10.1.8.83:49298</strong></td>
<td><strong>10.1.7.18:3389</strong></td>
<td>ESTABLISHED</td>
</tr>
</tbody>
</table>

### Web Apps

- RDP/SSH
## Netstat Connections

<table>
<thead>
<tr>
<th>Proto</th>
<th>Local Address</th>
<th>Foreign Address</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>0.0.0.0:445</td>
<td>0.0.0.0:0</td>
<td>LISTENING</td>
</tr>
<tr>
<td>TCP</td>
<td>10.1.8.83:445</td>
<td>10.1.2.118:53210</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>TCP</td>
<td>10.1.8.83:49188</td>
<td>54.91.1.122:443</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>TCP</td>
<td>10.1.8.83:49230</td>
<td>10.1.17.47:1433</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>TCP</td>
<td>10.1.8.83:49276</td>
<td>10.1.6.115:22</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>TCP</td>
<td>10.1.8.83:49295</td>
<td>10.1.72.47:80</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>TCP</td>
<td>10.1.8.83:49296</td>
<td>52.27.96.23:443</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>TCP</td>
<td>10.1.8.83:49297</td>
<td>10.1.7.18:135</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>TCP</td>
<td>10.1.8.83:49298</td>
<td>10.1.7.18:3389</td>
<td>ESTABLISHED</td>
</tr>
</tbody>
</table>

- **Web Apps**
- **RDP/SSH**
- **Databases**
# cat dns-mappings.txt
10.1.72.27,"R_AND_D_16.EXTERNAL.LOCAL"
10.1.4.67,"R_AND_D_8.EXTERNAL.LOCAL"
10.1.72.12,"DESKTOP35.EXTERNAL.LOCAL"
10.1.72.13,"SQL4.EXTERNAL.LOCAL"
10.1.4.12,"DESKTOP40.INTERNAL.LOCAL"
10.1.4.13,"SYSTEM33.INTERNAL.LOCAL"
...
TCP 10.1.8.83:49298 10.1.7.18:3389 ESTABLISHED

(s:Computer {name:'<src>'})-[:Connected_3389]->(d:Computer {name:'<dst>'})
Adding Connections (-c, -d)

G [10/26/17 12:02] [porterhau5@compute:~/BloodHound-Owned] head -n5 netstats.txt
TCP 10.1.8.126:49188 54.27.21.64:443 ESTABLISHED
TCP 10.1.8.126:49295 10.1.72.27:80 ESTABLISHED
TCP 10.1.8.126:49297 10.1.4.67:135 ESTABLISHED
TCP 10.1.8.127:51745 10.1.72.12:445 ESTABLISHED
TCP 10.1.8.130:51985 10.1.72.27:445 ESTABLISHED

G [10/26/17 12:02] [porterhau5@compute:~/BloodHound-Owned] head -n5 dns-mappings.txt
10.1.72.27, "R_AND_D_16.EXTERNAL LOCAL"
10.1.4.67, "R_AND_D_8.EXTERNAL LOCAL"
10.1.72.12, "DESKTOP35.EXTERNAL LOCAL"
10.1.72.13, "SQL4.EXTERNAL LOCAL"
10.1.4.12, "DESKTOP40.INTERNAL LOCAL"

G [10/26/17 12:02] [porterhau5@compute:~/BloodHound-Owned] ruby bh-owned.rb -c netstats.txt -d dns-mappings.txt
[*] Using default username: neo4j
[*] Using default password: BloodHound
[*] Using default URL: http://127.0.0.1:7474/
[*] Adding 158 connections
## Connections over 389/tcp

<table>
<thead>
<tr>
<th>Database Info</th>
<th>Node Info</th>
<th>Queries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show Connections over 22/tcp</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Show Connections over 80/tcp</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Show Connections over 135/tcp</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Show Connections over 139/tcp</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Show Connections over 389/tcp</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Show Connections over 443/tcp</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Show Connections over 445/tcp</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Show Connections over 1433/tcp</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Show Connections over 1521/tcp</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Show Connections over 3306/tcp</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Diagram showing network connections over 389/tcp.
Show Web App Connections

MATCH p=((s:Computer)-[Connected_80|Connected_443]->(d:Computer)) RETURN p LIMIT 100
Cypher Query Using Both Datasets

“Simple” derivative local

MATCH p=((s)-[:Admin:Member]*1..)->(m:Computer)-[:Connected_3389*1..]->(c:Computer))
RETURN p

Connected to one or more hosts via RDP
MATCH p=((s)-[AdminTo|MemberOf]-{m:Computer})-[Connected|ConnectedTo]-{c:Computer}) RETURN p
Escalation Paths to RDP Jumpboxes

MATCH p=((s)-[AdminTo|MemberOf*1..]->(m:Computer)-[:Connected_3389*1..]->(d:Computer)) RETURN p
Future Research

- Incorporating more flow data
- Critical Path to Compromise
- More inclusion of local accounts
- Alternative data collection
- Ideas? Reach out!
References

• BloodHound on Slack: https://bloodhoundgang.herokuapp.com/
• BloodHound project on GitHub: https://github.com/BloodHoundAD/BloodHound
• SharpHound project on GitHub: https://github.com/BloodHoundAD/SharpHound
• Cypher Reference Card: https://neo4j.com/docs/cypher-refcard/current/
• Rohan Vazarkar - Intro to Cypher: https://blog.cptjesus.com/posts/introtocypher
• Andy Robbins – BloodHound 1.3 – The ACL Attack Path Update: https://wald0.com/?p=112
• Will Schroeder – Local Group Enumeration: https://www.harmj0y.net/blog/redteaming/local-group-enumeration/
• Tal Be'ery & Marina Simakov – The Enemy Within: Stopping Attacks Against Local Users: https://youtu.be/HE7X7I-k-A4
Thank you!

Twitter/GitHub: @porterhau5
Email: tom [at] porterhau5.com
https://github.com/porterhau5/BloodHound-Owned
https://github.com/porterhau5/BloodHound (forked)

Appropriate Tributes
@godtributes

@porterhau5 BEJEUSES FOR THE BEJESUS
LORD! BLOODHOUNDS FOR THE
BLOODHOUND MONSTER

9:23 AM - 20 Oct 2017