Sniffing Compromise: Hunting for Bloodhound activity and building better prevention

Abstract

Join us on a journey as we chase BloodHound through a compromised environment via host and network telemetry. We will dive quickly into detections to become better prepared for next time.

Overview

What is Threat Hunting?

Threat Hunting is the act of searching for anomalous activity using telemetry from hosts and network within a given environment.

What is BloodHound?

BloodHound is a tool widely used by Attackers and Defenders to find miss configured or hidden relationships within a Windows Active Directory. Attackers use this to identify attack paths. Defenders use this to identify misconfigurations what would lead to attacks.

How does BloodHound work?

BloodHound has two components the Collector, This part is what is run against an Active Directory. This can be run on a compromise endpoint, weather is be in memory or from the disk.

It will also conduct all communication with the Domain Controller and Workstations using various network protocols. Getting information from the Domain Controllers it uses LDAPs for communication, it then uses RPC over SMB to send queries to workstations within the network that are awake and connected.

> **Note**
>
> This breakdown of how bloodhound works is a high level overview. For a more indepth view. There will be some refrences for further reading at the end.

The Hunt

With the knowledge we have with how users of BloodHound load these collectors to be run against an active directory. We go to our biggest source of information we have. The SIEM (Security information and event management) platform.

In this case we will be using Splunk, But that does not mean that the things we go over in this walkthrough cannot be applied to other SIEM's.

We can check what sort of data we have available to us by running the the follow query.

Query:

```
| eventcount summarize=false index=* | dedup index | fields index
```

Result:

"/KC3/attachments/Pasted image 20220707134338.png" is not created yet. Click to create.

We get a list of indexes that we can search through.

Our First Query

When it comes to hunting for anomalous behavior, you want to start with a broad search, then iterate to refine the search until you either find what you where looking for which proves the idea you had or something else completely.

Since we are looking for Bloodhound, we are assuming that someone has access to our systems already. That might be from users reporting strange occurrences within their machines, or someone in IT support mentioned some strangeness.

What ever it maybe, we want to see if BloodHound was run. So where do we start? How about with just `BloodHound`?

Query:

```
bloodhound
```

Result:

"/KC3/attachments/Pasted image 20220706222222.png" is not created yet. Click to create.

Well we get a hit! But a singular hit. Unfortunately this results looks to have not been parsed correctly. So looks like we where correct on BloodHound being run.

We also know that there are various ways to run, BloodHound can be run from a file on disk. Taking a look at the GitHub repository for BloodHound. We can see that they have various collectors for gathering data from the active directory.

The collectors are `AzureHound.ps1`, `SharpHound.exe`, `SharpHound.ps1`. We know that we have the windows event logs along with sysmon logs from the index search above. Let's tighten up our scope a bit to those since we know we want to be looking at host specific logs.

We run the following query to see what sort of information these indexes hold.\

Query:

```
index IN (wineventlogs, sysmon)
```

Results:

"/KC3/attachments/Pasted image 20220707135638.png" is not created yet. Click to create.

> **Tip**
>
> Normally you can do something like `index="sysmon" OR index="wineventlogs"` to search various indexes. But using the `IN` operand lets us use an array which is much cleaner and easier to add on more index if needed or remove.

This gives us a large set of data that we can refine to find more evidence of BloodHound.

Well, knowing a bit about how Windows Event logs work, we know that they all have a EventID tied to every kind of log. This is also true with Sysmon events.

Query:

```
index IN (wineventlogs, sysmon)
| stats count by winlog.event_id
```

Results:

"/KC3/attachments/Pasted image 20220707142340.png" is not created yet. Click to create.

Looking at the results we still have a lot of events. Doing some research we find out that Sysmon Event ID `11` is File Creation. Knowing this let us scope our search to this event ID. To do this we can append `winlog.event_id=11` to the first line in our previous query.

Query:

```
index IN (wineventlogs, sysmon) winlog.event_id=11
| stats count by winlog.event_id
```

Results:

"/KC3/attachments/Pasted image 20220707142755.png" is not created yet. Click to create.

The results still show a lot of data. Clicking on the events tab. It should bring us to the events where we can dig a little deeper.

"/KC3/attachments/Pasted image 20220707143007.png" is not created yet. Click to create.

Looking at the events that are available to us. We see that there are various fields that are interesting. We can put these in a table so we can get a better idea as to the kind of data we have.

Query:

```
index IN (wineventlogs, sysmon) winlog.event_id=11
| table host.name file.name file.path
```

Result:

"/KC3/attachments/Pasted image 20220707144449.png" is not created yet. Click to create.

Now that we have a nice table lets see if we can look for just `exe` and `ps1` files being created see if we can catch BloodHound being downloaded. Their happens to be a field that we can use for exactly this `file.extension`.

Query:

```
index IN (wineventlogs, sysmon) winlog.event_id=11 file.extension IN (exe, ps1)
| table host.name file.name file.path
```

Results:

"/KC3/attachments/Pasted image 20220707151929.png" is not created yet. Click to create.

We have a much smaller data set after the last query. We still can refine this to be smaller and more pointed towards our goal. If we look back at the first query we did. There was a hostname attached to that unparsed log. `rdp01.magnumtempus.financial`

Query:

```
index IN (wineventlogs, sysmon) winlog.event_id=11 file.extension IN (exe, ps1) host.name=rdp01.magnumtempus.financial
| table host.name file.name file.path
```

Result:

"/KC3/attachments/Pasted image 20220707162627.png" is not created yet. Click to create.

This gives us a much better set of events, that we can analyze.

We can start to remove things that we know is part of the windows operating system.

After doing some digging we see a couple of interesting things but not exactly anything pointing to BloodHound use.

What Now?

Well if we couldn't find anything in the way of catching the attackers downloading or running bloodhound. There might be another way that we can detect bloodhound being run within an environment.

Networking protocols that BloodHound uses to do it's enumeration.

BloodHound will use LDAPs communication against domain controllers to gather information. Along with that it will also get session information from each computer that is on and connected to AD using RPC over SMB to see what users are logged in during the time of the collection.

We can start just like before with a simple query. Lets see if we see anything happening with `445` or `139` since those are the ports used for SMB

Query: `445 OR 139`

Results:

"/KC3/attachments/first_smb_results.png" is not created yet. Click to create.

Looking at the results we see that there was a spike recently for this type of connection. Well lets see if we can make this data tell us more.

Adding some stats to breakdown what hosts are connecting using 445 or 139.

Query:

```
index="zeek" id.resp_p IN (445,139)
| stats count values(id.resp_p) by hostname
```

Results:

"/KC3/attachments/stats_smb.png" is not created yet. Click to create.

From the results, we observe that host `rdp01` has been making a lot of connections / requests on Port 445. This tells us that we might be on the right path.

Now we want to see exactly which hosts did the `rpd01` machine talk to.

Query:

```
index="zeek" id.resp_p IN (445,139) sourcetype="zeek_ntlm"
| stats count values(hostname) by server_dns_computer_name
```

Results:

"/KC3/attachments/hostnames_by_smb.png" is not created yet. Click to create.

With this search we can see that the `rdp01` machine reached out to almost every single workstation or server within the environment. Now this is something that is anomalous and we can use to detect.

Detections

Now that we have something that we know can be part of bloodhound use. Lets take that and put into a detection. A detection is a sort of "rule" for things that you or your org would like to be notified / tracked.

To start the work with a detection. we need to see what kind of detection or rule do we want or makes sense for what we are looking at. Since we know that BloodHound uses communications via SMB. We can create a rule that triggers every time that any host makes a connection to another host over 445.

This is a great start, but what is our environment actually uses SMB shares for collaboration. Well, we could have a rule where it only triggers when the host `rdp01` makes a 445 connection.. But then what if another host is compromised and that is used to run bloodhound. So we shouldn't restrict this to a single host. So instead of triggering every time a 445 connection happens, it would be less prone for false positives if we did a rule that triggered after a threshold was hit.

To see what threshold we should set this too, we need to understand out environment. How many times per host in a hour does a normal day of work trigger a SMB connection.

Lets say that on any given day, an employee will connect to the SMB share two to three times in a day. So we can have a rule that says.

```
If a host make more than 3 connections via SMB in a hour alert the security team
```

We put this into production and we notice that this still is causing a lot of false positives. Looking back at the last query we had in splunk. We saw that when bloodhound is run it will make connections to all hosts within the given environment. We also know that when using SMB shares in your environment most connections should be directed to the server that holds the SMB shares.

Improving on our previous detection. we can write something like the following.

```
If a host makes a connection via SMB to a host that is not the SMB Server trigger and alert to the security team.
```

Threat Hunting Template

Template:

```
THREAT HUNTER TEMPLATE
Title:
Date Created:
Hypothesis:
Mitre Tactic:
Mitre Sub Technique:
Simulation Details (if any):
Proposed Search Query:
Hunter Limitations/Observation Notes:
Hunt Findings:
Detection Alert Title:
Detection Alert Query:
```

Results:

```
Title: BloodHound Activity
Date Created: 2022-07-11
Hypothesis: Intruders will run bloodhound to enumerate a windows domain
Mitre Tactic:
Mitre Sub Technique:
Simulation Details (if any):
Proposed Search Query:
Hunter Limitations/Observation Notes:
Hunt Findings:
Detection Alert Title:
Detection Alert Query:
```