0xdf hacks stuff

Home About Me Tags Cheatsheets YouTube Gitlab feed

HTB Sherlock: Reaper

ctf htb-sherlock hackthebox forensics sherlock-reaper dfir ntml net-ntlmv2 ntlmrelayx ntlm-relay win-event-4624 win-event-5140 pcap wireshark llmnr jq evtx-dump

Aug 22, 2024

Reaper is the investigation of an NTLM relay attack. The attacker works from within the network to poison an LLMNR response when a victim has a typo in the host in a share path. This results in the victim authenticating to the attacker, who relays the authentication to another workstation to get access there. I'll show how this all happened using the given PCAP and Windows Security Log.

Challenge Info

Name	Reaper Play on HackTheBox
Release Date	15 August 2024
Retire Date	15 August 2024
Difficulty	Very Easy
Category	DFIR
Creator	CyberJunkie Moderator 2 ★ 613 hackthebox.com

Background

Scenario

> *Our SIEM alerted us to a suspicious logon event which needs to be looked at immediately. The alert details were that the IP Address and the Source Workstation name were a mismatch. You are provided a network capture and event logs from the surrounding time around the incident timeframe. Corelate the given evidence and report back to your SOC Manager.*

Notes from the scenario:

- There's suspicious login event.
- The artifacts include both a PCAP and event logs.

Questions

To solve this challenge, I'll need to answer the following 10 questions:

1. What is the IP Address for Forela-Wkstn001?
2. What is the IP Address for Forela-Wkstn002?
3. Which user account's hash was stolen by attacker?
4. What is the IP Address of Unknown Device used by the attacker to intercept credentials?
5. What was the fileshare navigated by the victim user account?
6. What is the source port used to logon to target workstation using the compromised account?
7. What is the Logon ID for the malicious session?

8. The detection was based on the mismatch of hostname and the assigned IP Address.What is the workstation name and the source IP Address from which the malicious logon occur?
9. When did the malicious logon happened. Please make sure the timestamp is in UTC?
10. What is the share Name accessed as part of the authentication process by the malicious tool used by the attacker?

Data

The download includes two files as described above:

```
oxdf@hacky$ unzip -l Reaper.zip
Archive:  Reaper.zip
  Length      Date    Time    Name
---------  ---------- -----   ----
        0  2024-08-05 10:32   Reaper/
   315336  2024-07-31 09:56   Reaper/ntlmrelay.pcapng
  1118208  2024-07-31 10:10   Reaper/Security.evtx
---------                     -------
  1433544                     3 files
```

The name of the PCAP implies that I am looking for an NTLM relay attack.

Artifact Background

Event logs are becoming relatively routine with Sherlocks investigating Windows systems. This file represents a single event log, the Security log. Security has things like logon / logoff, authentication, account management, etc.

Packet capture data (PCAPs) are files showing network traffic. I'll want to try to identify what hosts belong to Forela, and which are interacting with them. In real life, this is typically very noisy data, but in CTF events sometimes it's just interesting traffic.

Tools

Event Logs

I'll use `evtx_dump` (from omerbenamram's evtx repo, download the binary from the Releases page) to convert the event logs to a JSON format and `jq` to query them from a Linux command line.

```
oxdf@hacky$ file Security.evtx
Security.evtx: MS Windows Vista Event Log, 3 chunks (no. 2 in use), next record no.
52
oxdf@hacky$ evtx_dump -o jsonl -t 1 -f Security.json Security.evtx
oxdf@hacky$ wc -l Security.json
51 Security.json
```

There are 51 logs in this file, which matches the `file` output saying the next log would be ID 52.

PCAP

I'll mostly work with Wireshark to view and interact with the PCAP.

Analysis

Data Overview

PCAP Overview

After opening the PCAP in Wireshark, at the bottom right of the window it shows there are 1654 packets in this capture:

Under Statistics –> Endpoints, it shows there are 19 IPv4 addresses:

Nine of these fall in the 172.17.79.0/24 range, which is likely the private network of Forela here. On the TCP tab, the low ports observed are 80, 88, 135, 389, 443, and 445:

From this I'll label 172.17.79.4 as the domain controller because of Kerberos (TCP 88) as well as other standard Windows ports like LDAP (389), RPC (135), and SMB (445). It is unusual / interesting to see a domain controller listening on 80, but it's only 5 packets, no data.

172.17.79.135 could also be some kind of webserver, as it is showing bidirectional traffic on port 80.

Event Log Overview

All 51 event logs come from the WKSTN001 computer:

```
oxdf@hacky$ cat Security.json | jq '.Event.System.Computer' -r | sort | uniq -c |
sort -nr
     51 Forela-Wkstn001.forela.local
```

This suggests these logs were captured there.

I'll use `jq` to get the event ID for each log:

```
oxdf@hacky$ cat Security.json | jq '.Event.System.EventID' | sort | uniq -c | sort -
nr
     38 4702
     11 4624
      1 5140
      1 4662
```

There are four types:

- 4702 - A scheduled task was updated
- 4624 - An account was successfully logged on
- 5140 - A network share object was accessed
- 4662 - An operation was performed on an object

Host Identification

The first two questions involve identifying the IPs of two workstations. There are NetBIOS (NBNS) refresh packets in the PCAP. Adding a filter for "nbns" shows all the NetBIOS traffic:

The refresh packets come from a host letting the network know the name of the workstation. So Forela-Wkstn001 is 172.17.79.129 (Task 1), and Forela-Wkstn002 is 172.17.79.136 (Task 2).

NTLM Relay Attack

Background

An NTLM relay attack happens when an attacker can get a target to authenticate to a host they control. They can capture the hash (typically a NetNTLMv2), or relay it to another host.

The NetNTLMv2 hash is not really a hash, but really a cryptographic challenge response. The server asks the client to encrypt some nonce (dummy value, never reused) with their NTLM hash, and then the client does so. When an attacker captures this, they can brute force passwords, for each password generating the NTLM hash, and then trying to decrypt the nonce. If it works, they found the correct password.

Relaying eliminates the need to crack the challenge. Instead, the attacker waits for the victim to attempt to authenticate. Then they start their own authentication to another server. When that server returns a nonce to be encrypted, the attacker passes that on to the victim, who thinks they are authenticating to the attacker. The result is returned to the attacker who returns it to the target server, and the attacker is now authenticated as the relayed user.

Name Resolution

A common technique is to poison LLMNR. When a host on a Windows domain tries to visit a host by DNS name, it first queries DNS, but if that fails, it tries link-local multicast name resolution (LLMNR).

In this example, 172.17.79.136 tries to access the “D” computer. First there’s a DNS query (packet number 1149:

The response is that there’s no such name. Then the host sends out LLMNR queries on both IPv4 and IPv6 (packet 1162):

172.17.79.135 jumps in to respond with it’s own IP as the answer, which is suspect (Task 4).

At this point, the victim has likely typoed some kind of UNC path for the domain controller as `\\D\` instead of `\\DC\`, and the attacker has responded to say that it is their IP.

Relay

Just after this, there’s a really interesting set of SMB exchanges:

At 1, WKSTN002 (.136) is sending out LLMNR requests which the attacker machine (.135) is poisoning to say it’s their machine. This leads to WKSTN002 setting at a TCP connection to the attacker on 445 at 2, and then starting the SMB authenticating and requesting access to `\\D\IPC$` at 3. At 4, the attacker starts the TCP connection on 445 to WKSTN001 (.129), and begins the SMB connection at 5.

6 is where the relaying happens. The attacker tells WKSTN002 the session is expired, triggering WKSNT002 to start authentication again. Whatever packets WKSTN002 sends to the attacker, it relays on to WKSTN001, autnenticating as arthur.kyle (Task 3). By the end, WKSTN002 is trying to request `\\D\\IPC$` (when SMB is starting, it typically tries to read from `ICP$` before going to the actual share), and the attacker is requesting access to that same share on the target victim. That string `\\172.17.19.129\ICP$` seems like it should be the answer to Task 10, but it is not.

Share Access Event

There is one 5140 share access event.

```
oxdf@hacky$ cat Security.json | jq 'select(.Event.System.EventID==5140)'
{
  "Event": {
...[snip]...
    "EventData": {
      "AccessList": "%%4416\r\n\t\t\t\t",
      "AccessMask": "0x1",
      "IpAddress": "172.17.79.135",
      "IpPort": "40252",
      "ObjectType": "File",
      "ShareLocalPath": "",
      "ShareName": "\\\\*\\IPC$",
      "SubjectDomainName": "FORELA",
      "SubjectLogonId": "0x64a799",
      "SubjectUserName": "arthur.kyle",
      "SubjectUserSid": "S-1-5-21-3239415629-1862073780-2394361899-1601"
    },
    "System": {
      "Channel": "Security",
      "Computer": "Forela-Wkstn001.forela.local",
...[snip]...
      "TimeCreated": {
        "#attributes": {
          "SystemTime": "2024-07-31T04:55:16.243325Z"
..[.[snip]...
}
```

This shows the arthur.kyle user accessing `\\*\IPC$` (Task 10) on WKSTN001 from 172.17.79.135. It’s not clear to me why it’s logged this way when it shows in the PCAP as the IP instead of `*`, but it is the same event.

Logon Events

Event 4624 is a successful logon event. I’ll take a closer look at these to identify the activity above. I’ll use `jq` to filter out only those events, and then select the `IpAddress` field:

```
oxdf@hacky$ cat Security.json | jq 'select(.Event.System.EventID==4624) |
.Event.EventData.IpAddress' -r
::1
::1
172.17.79.135
-
-
-
-
-
-
-
-
```

There’s only one from the attacker machine. I’ll get that event, using `jq` to select out the interesting data:

```
oxdf@hacky$ cat Security.json | jq 'select(.Event.System.EventID==4624 and
.Event.EventData.IpAddress=="172.17.79.135") | .Event | {"Time":
.System.TimeCreated["#attributes"].SystemTime, "SrcIP": .EventData.IpAddress,
"SrcPort": .EventData.IpPort, "SrcHost": .EventData.WorkstationName, "Username":
.EventData.TargetUserName, "LogonID": .EventData.TargetLogonId, "TargetHost":
.System.Computer, "AuthPackage": .EventData.LmPackageName}'
{
  "Time": "2024-07-31T04:55:16.240589Z",
  "SrcIP": "172.17.79.135",
  "SrcPort": "40252",
  "SrcHost": "FORELA-WKSTN002",
  "Username": "arthur.kyle",
  "LogonID": "0x64a799",
  "TargetHost": "Forela-Wkstn001.forela.local",
  "AuthPackage": "NTLM V2"
}
```

This is the authentication attempt at WKSTN001 from WKSTN002 but from the attacker’s IP as arthur.kyle. The output contains the answer to the source port (Task 6), the logon ID (Task 7), and the timestamp (Task 9). This also shows the workstation name and IP that don’t match what I found in the earlier tasks (Task 8).

Share Access

About 15 seconds after the relaying issue where the user mistyped the name of the server, the user on WKSTN002 makes a successful connection to the domain controller, `DC01` (172.17.79.4) requesting the share `\\DC01\Trip` (Task 5). This is likely the share that the user was trying to connect to when they entered `D` as the hostname instead of `DC01`:

This also doesn’t work, as while the user is connected to the right host this time, the share doesn’t exist. I’m not sure why this user is so bad at navigating network shares, but there’s no negative consequence to the use this time.

Results - Question Answers

1. What is the IP Address for Forela-Wkstn001?

   172.17.79.129

2. What is the IP Address for Forela-Wkstn002?

   172.17.79.136

3. Which user account’s hash was stolen by attacker?

   Arthur Kyle

4. What is the IP Address of Unknown Device used by the attacker to intercept credentials?

   172.17.79.135

5. What was the fileshare navigated by the victim user account?

   `\\DC01\Trip`

6. What is the source port used to logon to target workstation using the compromised account?

   40252

7. What is the Logon ID for the malicious session?

   0x64A799

8. The detection was based on the mismatch of hostname and the assigned IP Address.What is the workstation name and the source IP Address from which the malicious logon occur?

   FORELA-WKSTN002, 172.17.79.135

9. When did the malicious logon happened. Please make sure the timestamp is in UTC?

   2024-07-31 04:55:16

10. What is the share Name accessed as part of the authentication process by the malicious tool used by the attacker?

   `\\*\IPC`
