Threat Hunting with Wireshark

# Situation

Your organization recently suffered a data breach from an attacker inside the network. We don’t know how they got in, but they obtained an IP address in the local subnet and have compromised at least one workstation and possibly a server. We need to know:

1. What workstation the attacker compromised
2. The attack vector used and any associated users or credentials.
3. What was taken (if anything)
4. Is the attacker still on the network or have they left a backdoor?
5. Do we expect another attack?
6. How can we prevent this **internal** attack from happening again?

Instructor Note:

The **ANSWER –** and **EXPLAINED –** sections below each question may be only one of the many ways to find the answer. It also may not be the best one, and in fact, it may be the best suited to ***teach*** and not for quick analysis. The answer to this whole scenario can be obtained by extracting HTTP and SMB objects from the capture and jumping to relatively obvious conclusions then following packet stream 131. Information that you have access to that is not an answer or explained section will be highlighted in yellow.

With this in mind, the goal is to learn SMB ***and*** *Wireshark* **and** do a threat-hunting exercise, not simply get the answer. Have fun!

# Information

**Attacker –** 10.10.0.7

**Victim –** 10.10.0.4

**Domain Controller –** 10.10.0.6

**Domain –** EVOLVE.LOCAL or just EVOLVE

**Primary Attack Vector** – SMB for enumeration and initial compromise. The user account “Guest” with no password can enumerate shares, and the user account “test” with the password “test” was used for access. The students don’t know the “Guest” user had no password, nor do they know the “test” user’s password. This is purely background information

**Secondary Attack Vector -** An **aspx** webshell was uploaded to the share and then accessed over **HTTP** via the IIS server running on the victim. The attacker then ran local information-gathering commands.

## Helpful Hints

The attack took place quickly, and we know that there is little extra traffic. It seemed like the attacker knew what to look for and acted quickly.

The system attacked was in dev, and had encryption turned off for most services, meaning we are likely able to read traffic in the clear.

The new developer has not had a chance to harden the dev system and has been out on vacation for a few days.

The plan for the service is to deploy an IIS front-end that serves as a web-based file-share

# Resources

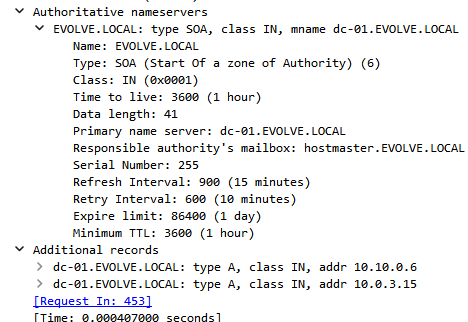
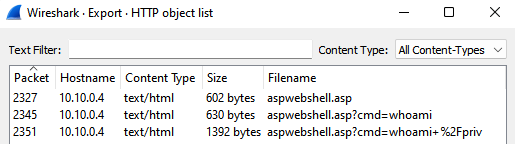
* What is SMB? What IT Decision Makers Need To Know | Visuality Systems - <https://visualitynq.com/resources/articles/what-is-smb-what-it-decision-makers-need-to-know/>
* Microsoft SMB Protocol Authentication - Win32 apps | Microsoft Learn - <https://learn.microsoft.com/en-us/windows/win32/fileio/microsoft-smb-protocol-authentication>
* SMB - <https://wiki.wireshark.org/SMB>
* SMB2 - <https://wiki.wireshark.org/SMB2>

# Sections

## Part 1 – Observations

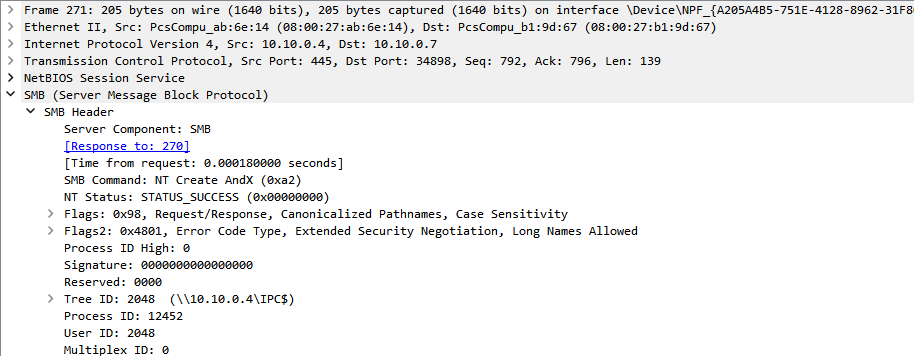
### Questions

1. Text

   Description automatically generatedWhat is the **victim’s** **hostname**?
   1. **ANSWER –** LABTEMP or LABTEMP.EVOLVE.LOCAL for the FQDN at 10.10.0.4.
   2. **EXPLAINED –** The easiest way to determine this is to put DNS in the Wireshark filter window and look for any of the conversations between the victim IP address and the **10.10.0.6** address. The second query/response pair happens at **Packet 453**. The client is asking who is the start of authority for **LABTEMP.EVOLVE.LOCAL.** We can gather this is likely the hostname of our target from this information. Furthermore, in packets 465 and 466 we see a dynamic update. And response. In the dynamic update response, we can see updates containing the hostname and IP address, and DNS records for **LABTEMP**.
2. What type of host is the other endpoint on the local **10.10.0.0/24** subnet?
   1. **ANSWER –** 10.10.0.6 is the Doman Controller.
   2. **EXPLAINED –** Filtering for DNS is another good choice. We can see that the victim queries this IP address for domain name resolution, and it states that it is the start of authority. Expanding the records in the response to the first packet, the very bottom option underneath DN S and additional records provide an IP address that matches and a hostname that states it is the DC.
3. Are there any **objects you can export** right away?
   1. **ANSWER –** Yes. We can export both **HTTP** and **SMB** objects.
   2. Graphical user interface, text, application

      Description automatically generated**EXPLAINED –** Clicking **File > Export Objects** provides a list of possible objects we can export. Selecting either of the two above lists what *Wireshark*can find in the capture. It tells almost the entire story, minus the fine details.
4. What is the most likely **attack vector**?
   1. **ANSWER –** A combination of **HTTP** and **SMB**.
   2. **EXPLAINED –** The exported objects make it pretty clear what happened. It’s evident that they were able to put a shell on the system, and it looks like it happened over SMB. We can’t be sure at this point though. When they got the Academy sequel document, it is likely that it was also over SMB and that HTTP was used to gain a further foothold.
5. What is the *Wireshark* filter to **detect *Write* activity on the** **SMB2** protocol?
   1. **ANSWER –** smb2.cmd == 0x09 ***or*** smb.access.write ***and*** maybe others more specific/generic.
   2. **EXPLAINED –**
   3. **smb2.cmd == 0x09**: Asking google *“filtering for smb commands wireshark”* gives [this link](https://osqa-ask.wireshark.org/questions/59864/how-to-filter-out-a-specific-smb-operation/) at the top as of 1/30/23. The first answer says to…

Try "smb.cmd == codenumber" or "smb2.cmd == codenumber"

* 1. **smb.access.write**: Via this [link](https://www.wireshark.org/docs/dfref/s/smb.html) we can get a “cheat sheet” of SMB Filters. From the question, search the page for “write” and infer from there. Notes
  2. You can also build this with ***Analyze > Display Filter Expression…***and build your own
  3. The specific command number is visible in different locations for **SMB** and **SMB2**
  4. **SMB1** – This is “Create” with the code “0xa2” -- The name and hex-code (OpCode) is after **SMB Command:**
  5. Graphical user interface, text, application

     Description automatically generated**SMB2** – This is a “Write” with the code “0x09” – This is much simpler than **SMB1** and the codes are easier to read. The name is after **Command:** and is a simple name and number. This number is the OpCode is base-ten, but it is written as hex/OpCode in the highlighted “Write Request” section.

1. What is the *Wireshark* filter for **NT Status: STATUS\_SUCCESS** in both **SMB** and **SMB2**
   1. **ANSWER –** smb.nt\_status == 0xc0000022
   2. Graphical user interface, text, application, chat or text message

      Description automatically generated**EXPLAINED –** Searching google *for “wireshark nt status filter”* and finding [this link](https://www.wireshark.org/docs/dfref/s/smb.html) second from the top allows us to search the page for “Status.” Finding **smb.nt\_status** helps but not enough. Heading to **Analyze > Display Filter Expressions…** you can start with that filter and find that *Wireshark*can build the rest for you.

## Part 2 – Scanning and Enumeration

### Questions

1. What indicators exist that this **pcap** starts with a “*nmap”* scan?
   1. **Bonus**: When did the **protocol-specific scan** start and what protocol was queried?
   2. **Bonus**: Would you consider this a **"targeted" scan?** Why or Why Not?
   3. **ANSWER –** Several TCP 3-Way handshakes, followed by **RST,ACK** packets, then a series of **SMB** connections and queries. Yes, it is targeted. Other than an initial port 80 scan, the rest of the interaction is focused on port 445. Filter with **ip.addr == 10.10.0.7** to see this more easily.
   4. **EXPLAINED –** An initial series of duplicated `ack` requests and a TCP-handshake #packet/21 that terminates with a reset is odd. The following series from #packet/25-91 is the `nmap` scan negotiating **SMB** protocols. Specifically, the `NBNS` protocol query using [nbtstat](https://learn.microsoft.com/en-us/windows-server/administration/windows-commands/nbtstat) shows some sort of information-gathering query. The multiple **RST** packets make this look like a quick on/off conversation. Finally, it all takes place over less than 1 second from the first **SMB** protocol request.
   5. A picture containing calendar

      Description automatically generated**Bonus 1** is explained by the series of quick connections immediately followed by an **SMB** protocol connection in **Packet 30**. The trend then continues.
   6. **Bonus 2** is no because it creates an enormous amount of generic traffic quickly, and many of the connections are non-standard (i.e., no data exchanged before the reset, multiple times.
2. What is an indicator that the username that started a **negotiation request** in #packet/127 is successful?
   1. **ANSWER –**
   2. **EXPLAINED –**
3. What **username is used to enumerate** the shares on the target?
   1. **ANSWER –**
   2. **EXPLAINED –**
4. **What shares exist** on the target, and **which were access granted** to?
   1. **ANSWER –**
   2. **EXPLAINED –**
5. **Bonus:** Can the **client** write shares to the server?
   1. **ANSWER –**
   2. **EXPLAINED –**

### Notes

## Part 3 – Information Gathering

### Questions

1. What two **usernames were attempted** in the two negotiation sessions started in **Packet 108** and **Packet 127**?
   1. **ANSWER –**
   2. **EXPLAINED –**
2. Which of these was accepted and how can you tell?
   1. **ANSWER –**
   2. **EXPLAINED –**
3. What is the *Wireshark* display-filter expression to check if the **NT Status** of a packet is equal to **Access Denied**?
   1. **ANSWER –**
   2. **EXPLAINED –**
4. What **username is used to enumerate** the shares on the target and what shares exist?
   1. **Bonus:** To which shares was the attacker granted access?
   2. **ANSWER –**
   3. **EXPLAINED –**
5. **Can the attacker write** **new shares** to the victim host?
   1. **ANSWER –**
   2. **EXPLAINED –**
6. What common type of plant name is used to indicate a client (attempted) to connect to a share in the *Wireshark* “info” column?
   1. **Bonus:** Provide **any or all** of the following:
      1. The “OpCode” for the answer.
         1. **ANSWER –**
         2. **EXPLAINED –**
      2. Display-filter expression to list only packets containing the answer.
         1. **ANSWER –**
         2. **EXPLAINED –**
      3. “Command: Negotiate Protocol” number for the answer.
         1. **ANSWER –**
         2. **EXPLAINED –**

### Notes

## Part 4 – Compromising the Victim

1. How many packets take place between the “Time” 315.861028 and 317? (Round to the nearest hundred)
   1. **ANSWER:** 1500
   2. **EXPLAINED:** Packet 578 is the start and Packet 2039 is the finish. 2039 – 578 = ~1500 rounded up to the nearest hundred
2. What is/are the **most common type of SMB command(s)** in this short period*? (Hint: smb.cmd, smb.cmd == […something])*
   1. **ANSWER –**
   2. **EXPLAINED –**
3. **What type of attack** does this look like?
   1. **ANSWER –**
   2. **EXPLAINED –**
4. **What "Session Setup Response" packet and “stream” number** are part of the successful login attempt and what is the username?
5. **Why can’t we see a password**, even though the session setup is unencrypted?
   1. **ANSWER –**
   2. **EXPLAINED –**

### Questions

### Notes

## Part 5 – Data Exfiltration

### Questions

1. What is the name of the “Tree” the attacker connected to (including the IP and formatting)
   1. **Bonus:** What is the command the attacker could have used with ***smbclient*** to connect to the attacker assuming the same IP as is in the pcap?
   2. **ANSWER –**
   3. **EXPLAINED –**
2. What is the **significance of the “KeepAlive” requests and responses** between the attacker and victim?
   1. ANSWER –
   2. **EXPLAINED –**
3. What is the **name of the file** the attacker reads?
   1. **Bonus:** What does it tell the attacker they can expect to find on the share?
   2. **Bonus:** Can you find and decode the encoded “Data” that represents the text document?
   3. ANSWER –
   4. **EXPLAINED –**
4. What is the **name and the type of file** exfiltrated from the victim?
   1. **Bonus:** What SMB command was used on the client side to get the file onto their system?
   2. **ANSWER –**
   3. **EXPLAINED –**
5. What are the **“OpCodes” for the following SMB2 commands** and use one in an example display-filter
   1. *Find* –
      1. *ANSWER –*
      2. ***EXPLAINED –***
   2. *Create* –
      1. *ANSWER –*
      2. ***EXPLAINED –***
   3. *GetInfo* –
      1. *ANSWER –*
      2. ***EXPLAINED –***
   4. *Read* –
      1. *ANSWER –*
      2. ***EXPLAINED –***
   5. *Close* –
      1. *ANSWER –*
      2. ***EXPLAINED –***
   6. Filter Example –
      1. *ANSWER –*
      2. ***EXPLAINED –***

### Notes

## Part 6 – Second Foothold

### Questions

1. What is the **name of the webshell** uploaded by the attacker?
   1. *ANSWER –*
   2. ***EXPLAINED –***
2. What is the **common command name (e.g., read, create), packet number, and opcode Hex number** for the command used to place the file on the system?
   1. *Common Name* –
      1. *ANSWER –*
      2. ***EXPLAINED –***
   2. *Packet Number* –
      1. *ANSWER –*
      2. ***EXPLAINED –***
   3. *OpCode (in hex)* –
      1. *ANSWER –*
      2. ***EXPLAINED –***
   4. **Bonus:** What was the **SMB command** used to “place” the file on the system?
      1. *ANSWER –*
      2. ***EXPLAINED –***
3. When the attacker next connects, **what port and service are they connecting to, and over what Layer 7 protocol**?
   1. *ANSWER –*
   2. ***EXPLAINED –***
4. What is the **name of the user** the attacker is running commands as in the webshell?
   1. *ANSWER –*
   2. ***EXPLAINED –***
5. What are **the three “enabled” privileges** available to the attacker?
   1. **Bonus:** What is one or more publically available exploit associated with ***SeImpersonatePrivilege***?
   2. *ANSWER –*
   3. ***EXPLAINED –***

### Notes

# Student Summary

To record the engagement in your own words, practice writing a summary, and work through the “key findings” of the engagement, please answer the questions below. Most can also be supplemented with what potential remediations exist.

1. What overall security flaws allowed the attacker to enumerate the shares on the victim?
2. What specific vulnerability allowed the attacker to gain read/write access to the share?
3. Why could we read the exfiltrated data in plain text? What major security flaw does this implicate is present on the system?
4. How did the attacker move to a different service on the system?
5. What would the inevitable consequences of the attacker exploiting the privilege found in question 6.5? What is this type of attack called?