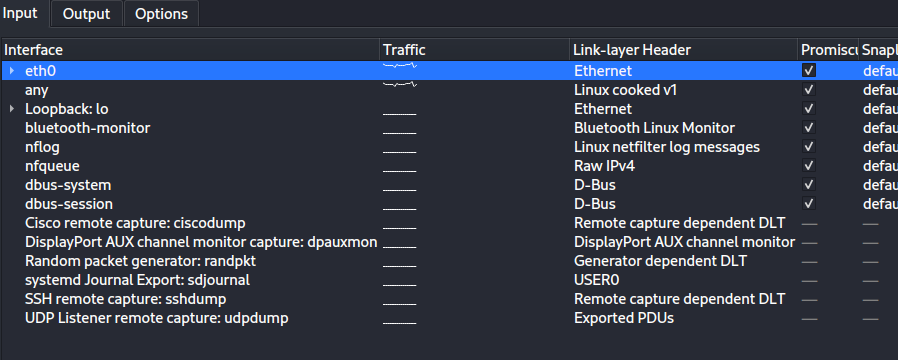
**Lab-05-Part-01 (Sniffing):**

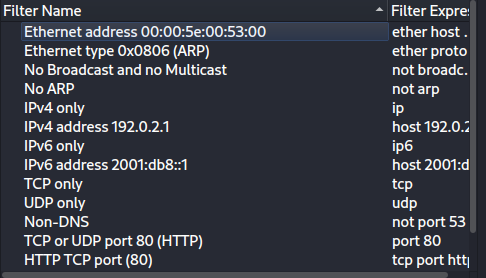
In this lab you are going to sniff some traffic on your test network using Wireshark. For this lab you can either use the Wireshark installed on your system or the one that is part of Kali VM.

This lab is going to ensure that you know how to fire up Wireshark, set up a capture, and do some filtering. Beyond that your learning experiences are up to you. Network traffic analysis is a very deep and broad area.

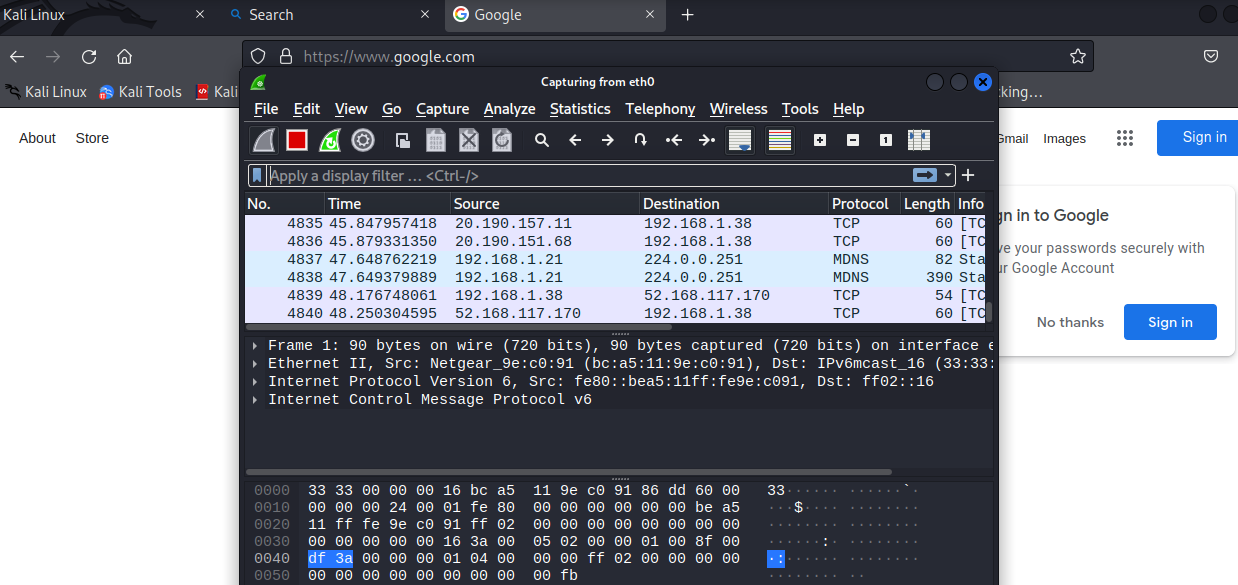
1. Fire up Wireshark
2. After Wireshark loads go to; **Capture > Interfaces**
3. Choose the **Options** on your currently active network connection.



1. Click on the **Capture Filter**button in this Capture Options dialog box.



1. Click on the **filter**: No ARP and no DNS. Do not click **OK**. Take a screenshot of the capture filter dialog when you select that filter.
2. Select **Cancel**. We are not going to specify any capture filters for this lab.
3. On the capture options dialog box, go ahead and select **Start** to start a capture.
   1. *Now you should be seeing traffic flying by. Visit a couple of web pages to generate some traffic.*



1. Scroll back on the interface a bit. Choose a **packet** that is using the TCP protocol.

Graphical user interface, text

Description automatically generated

1. Look through the traffic and see if you can find where your browser made a request to one of the web sites you visited. It should be HTTP and you might notice a GET or POST in the request as well.

In order to find it, I filtered http. I found something that looked like it could’ve been a request to a website, but I’m no expert.

A picture containing text

Description automatically generated

1. Right click on the individual packet and tell Wireshark to Follow TCP Stream.
2. Take a screenshot of the **Follow TCP Stream** window.

Here is the TCP stream.

Text, timeline

Description automatically generated

1. Click **Close** on the Follow TCP Stream window.

(This is the filter before I closed it)

Graphical user interface, text, application

Description automatically generated

1. Click the **Clear** button on the Filter Menu Bar.
2. Use a **display** filter in Wireshark to have it filter out traffic that uses TCP traffic on port 80. Then click **Apply** to apply the filter.
3. Take a screenshot of Wireshark with the filter input and applied.

A picture containing text

Description automatically generated

**Lab-05-Part-02 (PCAP):**

* 1. Navigate to [https://www.malware-traffic-analysis.net/ (Links to an external site.)](https://www.malware-traffic-analysis.net/) or [https://www.netresec.com/?page=PcapFiles (Links to an external site.)](https://www.netresec.com/?page=PcapFiles)
     + (Note: if this site is unavailable there are plenty of other Pcap libraries out there.)
  2. Search for your favorite Malware

I’m trying burnincandle!

* 1. Download the example packet capture and open it in Wireshark
  2. Perform an analysis
  3. What did you find?
  4. Document your procedures and findings.

(I don’t know if any of this is true or correct, but I’m trying my best to understand it.)

I found BURNINCANDLE as well as Ioctl requests

“The SMB2 IOCTL Request packet is sent by a client to issue an implementation-specific [file system control](https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-smb2/b1b7cc8a-4d24-4701-bc3f-220b543ceef8#gt_4ffb96a7-5fad-488e-9438-b7707d2e4226) or device control (FSCTL/[IOCTL](https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-smb2/b1b7cc8a-4d24-4701-bc3f-220b543ceef8#gt_09d6bc87-34ed-48e8-b4d4-962e90543462)) command across the network” (n.a.,2022)

What I’m gathering is something about a remote desktop, a Windows machine, and a ton of remote requests for access. Maybe an attacker is trying to gain access to a Windows machine remotely?

Table

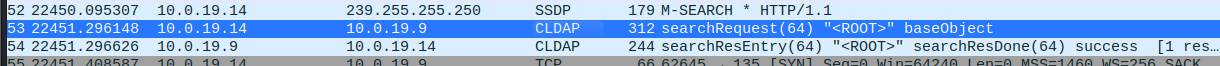
Description automatically generated

“LSARPC is really a set of calls, transmitted with RPC, to a system called the "Local Security Authority". This used in the Microsoft/Windows world to perform management tasks on domain security policies from a remote machine” (Pornin, 2015).

From this I gather that a machine’s security system is being triggered by something?



Looks like something is messing around with root.



Maybe someone tried to log in and that’s why they got the KRB error?

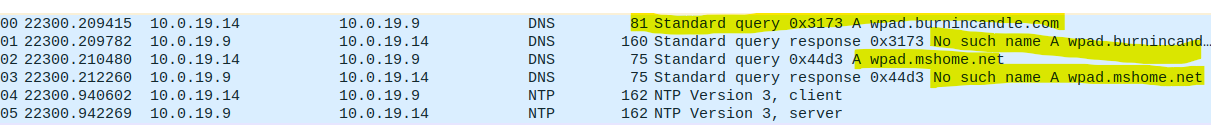


What is otectagain.top?

Graphical user interface, text, application

Description automatically generated

This is interesting too. I know burnincandle is part of the attack. But, what is wpad.mshome.net?



Seaskysafe.com

Text

Description automatically generated

Antnoscience.com



Text

Description automatically generated

A lot of packets from that website.

A picture containing calendar

Description automatically generated

Requests from dilimoretast.com too. Packets from websites aren’t unusual but these packets interest me.

A picture containing calendar

Description automatically generated

And ecs.office.comText

Description automatically generated

There are a TON of packets from several different websites, which isn’t odd if you’re casually searching the web, but these ones are somehow tied to the attack, I think.

Because I sought to understand the exercise, I looked for some answers.

Text

Description automatically generated

A lot of the websites that I found were indeed tied to the attack. I went solely off the weirdness of their names to determine if they were apart of the attack since the DNS packets looked somewhat normal compared to a non-malware packet tracing session (the top is normal and the bottom is the attack):

A computer screen capture

Description automatically generated with medium confidence

The weird part was that some of the DNS requests said “no such name as (insert URL here)” and I didn’t see that in the top Wireshark window.

References

*[MS-SMB2]: SMB2 ioctl request*. (2021, October 4). Developer tools, technical documentation and coding examples | Microsoft Docs. <https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-smb2/5c03c9d6-15de-48a2-9835-8fb37f8a79d8>

Audit My PC. (2010, March 13). *TCP 6023*. <https://www.auditmypc.com/tcp-port-6023.asp>

Congleton, N. (2021, September 21). *What is Port 443?* Lifewire. <https://www.lifewire.com/what-is-port-443-4690657>

GrantMeStrength. (2022, July 27). *DsCrackNamesA function (ntdsapi.h)*. Developer tools, technical documentation and coding examples | Microsoft Docs. <https://docs.microsoft.com/en-us/windows/win32/api/ntdsapi/nf-ntdsapi-dscracknamesa>

IBM. (2022, July 25). *IBM docs*. <https://www.ibm.com/docs/en/was-nd/8.5.5?topic=mechanism-kerberos-krb5-authentication-support-security>

Lastnameholiu. (2021, October 13). *LsaLookupNames function (ntsecapi.h)*. Developer tools, technical documentation and coding examples | Microsoft Docs. <https://docs.microsoft.com/en-us/windows/win32/api/ntsecapi/nf-ntsecapi-lsalookupnames>

*Port 62211 (TCP/udp)*. (n.d.). SpeedGuide. <https://www.speedguide.net/port.php?port=62211>

SyndicatorBBB. (2015, January 27). *How does LSARPC protocol work?* Information Security Stack Exchange. <https://security.stackexchange.com/questions/80167/how-does-lsarpc-protocol-work>

Wireshark. (2020, August 11). *MS-CLDAP*. Wireshark Wiki. <https://wiki.wireshark.org/MS-CLDAP>