The Pcap Hunt

Marco Favilli

July 11, 2022

Contents

1 What is PcapHunt?

Pcaphunt is a python program to analyze and discover possible threats in a pcap file. It mainly uses the python module pyshark to interact with the pcap file and examine it through specific *filters*.

2 Installation and usage

To install the program on a linux distribution just copy the repository with a *git clone* and install the required python modules.[1]

```
$ pip3 install -r requirements.txt
```

The third party's python modules that are used in this project are:

- Pyshark
- nest_asyncio
- Scapy

To use this program, once you are in the folder of the project just type:

```
$ python3 pcaphunt.py <pathToPcapFile>
```

or if you want to use the implemented scapy function over the pyshark one:

```
$ python3 pcaphunt.py -s <pathToPcapFile>
```

Furtheremore if you forget to enter the $< pathToPcapFile >$ or the specified path does not belong to a real file the program will automatically prompt you to enter the file's path you want to analyze.

Note: The function implemented with scapy regards the *arp scan*. This function has been implemented in two ways using both the Pyshark and Scapy modules. The need of implementing the function with Scapy has been driven by the difficulty for the Pyshark module to properly recognize the operations in the *ARP* layer of a packet.
Nevertheless the Pyshark implemented arp scan function will work fine overall, although it is important to point out that the Scapy implemented arp scan has a more selective behaviour (could reduce false positives and false negatives).

Warnings: The usage of the "*Scapy arp spoofing function*" *will* require the input capture file to have the *.pcap* extension, while using the functions made with pyshark you can use files with both *.pcap* and *.pcapng* extensions. Also the usage of pcaps with *annotation* or *comments* could **break the application**.

[1] In order to run this program you also need as prerequisites: Wireshark, *Tshark* and a Python 3.x and Pip3

Supported threats:

- *Network Attacks*:
  - ARP Spoofing
  - Ping of Death
  - ICMP Flood
  - TCP Syn Flood
  - DNS Request Flood
  - Unexpected Packet Loss
  - Vlan Hopping
- *Host Scanning*
  - ARP Scan
  - IP Protocol Scan
  - ICMP Ping Scan
  - TCP Syn Ping Scan
  - TCP Ack Ping Scan
  - UDP Ping Scan

3 Theory Behind the Attacks

ARP Spoofing : Is an attack against Ethernet or WiFi networks to get between the router and the target user. In an ARP-spoofing attack an attacker will send *spoofed* messages attempting to get nearby devices to associate his/her MAC address with the IP address of the target. An attacker can poison the ARP cache of other hostst on the network in order to:

1. spy on traffic.
2. intercept and modify the packets in a man-in-the-middle *(aka MitM)* attack.
3. drop the packets meant for the target to create a denial-of-service *(aka DoS)* attack.

The draw backs of an *arp spoofing* attack are of course its limitations. The major one is that this attack only works if the attacker is connected to the same network of the victim. An other limitation is the fact that this attack could fail due to firewalls or monitoring systems on the network detecting this sort of behaviour.

Wireshark Detection:

During this attack you will see occurrence of a single IP address being claimed by more than one MAC address *(duplicated address detetcted or duplicated address frame)*.

Ping Of Death: This attack is a DoS attack, in which the attacker aims to disrupt a targeted machine by sending a packet *larger* than the maximum allowable size, causing the target machine to freeze or crash.
It is important to notice that a correctly formed ping packet is typically 56B in size or 64B when the ICMP header is considered and 84B including the IP version 4 header, however pings packets may be as large as 65535 bytes.

Wireshark Detection:

Searching for ICMP requests which size is larger than the standard correctly formed ICMP packet.

ICMP Flood: This *Distributed Denial of Service (aka DDoS)* attack aimed at consuming computing power and saturing the bandwidth. ICMP flood, if not dropped by DDOS mitigation devices on the perimeter of the network, may overwhelm the internal network architecture. It is important to be aware that this type of flood may also generate outgoing traffic due to answers for the echo requests. This flood is widely used as a basic but yet effective way to bring down perimeter devices or saturate bandwidth.

Wireshark Detection:

To detect this attack using wireshark you can simply filter the traffic with an ICMP filter and check if there is a high volume of ICMP echo packets going from one or multiple IPs to one host. You can also filter the specific ICMP echo requests with a ICMP type 8.

TCP Syn Flood: This is one of the first typology of DDOS attacks ever documented. This attack exploits the *3-way handshake* of TCP where a client first sends a TCP segment with the SYN flag setted, the server then in response to the received SYN allocates and initalizes connection variables and buffers and then sends a SYN/ACK back to the client, and awaits on ACK segment from the client. If the client does not send an ACK to complete the handshake, eventually (generally after a minute or more) the server will terminate the *half-open* connection and reclaim the allocated resources.
In a TCP SYN flood the attacker(s) send a large number of TCP SYN segments, without completing the third handshake step. With this deluge of SYN segments, the server's connection resources become exhausted as they are allocated and never used for the half-open connections.

Wireshark Detection:

A typical SYN flood running against an host will generally see a high rate of SYN packets and a lesser rate of SYN-ACK packets coming from the targeted server.

DNS Request Flood: DNS Request flood is a DDoS attack which sends DNS request packets to a DNS server in an attempt to overwhelm the server's ability to respond to legitimate DNS requests.

Wireshark Detection:

A very basic filter could be UDP. Although since DNS requests generally use the UDP protocol with a *DEST*ination port of 53 it would be better make a filter that does include these informations. Naturally if you see a single host making a lots of these DNS requests in a short period of time it could be an attack, since this kind of traffic more often than not is not that intense.

Unexpected Packet Loss: Even tho *packet loss* is not an attack itself, if we see many packet re-transmissions and gaps in the network communications *(missing packets)* it may indicate that there is a *severe problem* in the network. In addition if there is an unexpected packet loss combined with possible traces of DOS/DDOS attack is a sign that bolster a lot the option of a successfully executed DOS/DDOS attack.

Wireshark Detection:

To find packets loss we have to search for TCP *lost packets* or TCP *retransmission*.

Vlan Hopping: VLAN hopping is a technique for bypassing NAC *(network access controls)* often used by attackers trying to access different VLANs by exploiting misconfigurations of switches. A fact worth mentioning is that VLAN Hopping attack is unidirectional, which means it could be accomplished in one way with no returning traffic. This can still be harmful, because it could easily be used in a DOS/DDOS attack and bring down a target host. VLAN Hopping attack can be accomplished in two ways:

1. *Switch Spoofing* where an attacker imitates a trunking switch by speaking the tagging and trunking protocols used in maintaining a VLAN.

2. *Double Tagging* where an attacker connected to an 802.1*Q* enabled port prepends two VLAN tags to a frame that it transmits. The frame is forwarded without the first tag because it is the native VLAN of a trunk interface. The second tag is then visible to the second switch that the frame encounters. This second VLAN tag indicates that the frame is destined for a target host on a second switch. The frame is then sent to the target host as though it originated on the target VLAN, effectively bypassing the network mechanisms that logically isolate VLANs from one another.

Wireshark Detection:

A solid indicator of VLAN Hopping is the presence of DTP packets (*Dynamic Truncking Protocol*) or packets tagged with multiple VLAN tags.

ARP Scan: This type of scan is an host discovery technique on layer 2. ARP scans are considered one if not the most effective way of detecting hosts in LAN networks. During ARP scanning an attacker is typically sending a large number

of ARP requests on the broadcast *(ff:ff:ff:ff:ff:ff)* destined to the MAC address 00 : 00 : 00 : 00 : 00 : 00 in order to discovery alive IP addresses on the LAN.

Wireshark Detection:

Seeing may ARP requests in a short period of time asking for *many* different IP addresses means that someone is probably trying to discover alive IPs on the network.

Implementation Notes:

While the ARP scan function implemented with the Pyshark module checks only if there are a lot of ARP requests in a short time, the ARP scan function implemented with the Scapy module also checks if these requests are meant for different hosts.

IP Protocol Scan: This scan technique allows an attacker to discover which network protocols are supported by the target OS.

Wireshark Detection:

During this attack, we will see many ICMP type 3 *(Destination unreachable)* and code 2 *(Protocol unreachable)* messages. The reason being that the attacker is sending a large number of packets with different *protocol numbers*.

ICMP Ping Scan: Ping scans are used to determine whether a host is online. ICMP echo request messages were designed specifically for this task and therefore ICMP ping scans use these packets to reliably detect the status of a host. The scan is rather straightforword, an attacker can determine that a host is online if he/she receives an ICMP echo reply to his/her probe.

Wireshark Detection:

If we see too many ICMP echo requests *(type 8)*, or ICMP echo replies *(type 0)* in a short period of time targeting many different IP addresses, then we are probably witnessing an ICMP ping scan.

TCP Syn Ping scan: Sometimes most used and basic scans like TCP SYN, TCP ACK, and ICMP can be blocked by a firewall, so the scanned hosts will be treated as offline. TCP SYN Ping is a very handy scanning technique for determine whether a host is online or at least has more permissive filtering rules.The gist of this scan is:

1. The attacker sends a TCP SYN packet to port 80.
2. If the port is closed, the host responds with an RST packet.
3. If the port is open, the host responds with a TCP SYN/ACK) packet, indicating that a connection can be established.

4. Afterward, an RST packet is sent to reset this connection.

Wireshark Detection:

If we see TCP SYN packets sended to a lot of different hosts on the port 80 coming from the same host, the network is probably under a TCP SYN Ping Scan.

TCP Ack Ping scan: Similar to the TCP SYN Ping scan, the TCP ACK Ping scan determines whether a host is responding. Although this technique will most likely be blocked by modern firewalls that can track connection states *(because this scan sends bogus TCP ACK packets associated with non-existing connections)*, it still works on older systems or not updated firewalls. Moreover it can be used to detect hosts that block SYN packets and ICMP echo requests.
A TCP ACK ping scan works in the following way:

- The attacker sends an empty TCP packet with the ACK flag set to port 80 *(this is the default port but it can be easily assigned an alternate port list, in this case modifying the destination port is common practice)*.
- If the host is offline, it should not respond to this request. Otherwise, it will return an RST packet and will be treated as online.

  Note: The RST packets are sent because the TCP ACK packet sent by the attacker is not associated with an existing valid connection. A peculiar aspect of this scanning technique is that the TCP ACK ping scan requires *root* or *administrator* privileges due to the fact that the attacker has to *forge* raw packets.

Wireshark Detection:

One way to recognize this scan is to check for TCP ACK packets of length 0 delivered from one host to many of hosts *(the TCP ACK packet can contain payload since for the client the connection has already been established, so even tho finding a lot of empty TCP ACK packets is not impossible it can be a suspicious sign that can be added to the cluster)*. Once again relaying on the standard port 80 is not a best practice in this case.
An other way we have to recognize this type of scan is by checking for previous packets in the 3-way handshake, namely the TCP SYN and TCP SYN/ACK and see if the sending host has previously sent the TCP SYN packet. If there is no sign of previous TCP SYN packets then is a clear sign that someone is *forging* TCP ACK packets, most probably for a TCP ACK Ping Scan .

UDP Ping Scan: Ping scans are used to determine whether a host is responding and can be considered online. UDP ping scans have the advantage of being capable of detecting systems behind firewalls with strict TCP filtering but that have left

UDP exposed.
UDP Ping scanning works as follow:

1. The attacker sends an empty UDP packet to the target port.
2. If the host is online, it should return an ICMP port unreachable error.
3. If the host is offline, various ICMP error messages could be returned.

   Note: Open ports that do not respond to empty UDP packets will generate false positives when probed.

Wireshark Detection:
More often then not the port used for this attack will be 40125 *(which is the default port)* therefore filtring UDP packets on that port will help. Additionally we have to consider that the payload of the UDP packet will be 0, therefore the size of the UDP packets that we are looking for will be 8 bytes *(which is the size of the UDP header)*. So if we see packets like the one described above going from one host to many hosts, is a clear sign that someone is scannig the network with an UDP Ping scan.

4 Conctact Info and External Link

- AUTHOR: *Marco Favilli*
- EMAIL: *m.favilli10@studenti.unipi.it*
- GITHUB: *https://github.com/markfvl*
- *PcapHunt repository*: https://github.com/markfvl/Pcaphunt