Information Security - Network Traffic Analysis

Sandesh Katta

PUID: 0034782137

Question 1

I am using `pyshark` package in python. this package allows me to get filtered trace directly into python without hazel.

1.1

List of all HTTP servers that were involved in a valid HTTP response:
['131.94.67.127', '139.154.214.254', '139.236.101.253', '139.94.185.156', '159.94.233.173', '159.94.233.253', '163.156.221.126', '163.188.121.191', '163.220.241.174', '163.222.117.155', '163.254.99.189', '167.126.51.175', '167.126.51.253', '167.126.75.253', '167.158.213.222', '167.220.77.109', '167.220.77.159', '167.220.77.175', '167.94.213.124', '171.188.209.111', '171.188.225.159', '171.188.94.237', '171.190.65.127', '171.220.116.107', '171.220.116.239', '223.252.50.173', '227.155.92.255', '231.125.83.191', '231.255.59.237', '231.255.91.159', '231.88.223.222', '235.253.79.95', '235.93.205.126', '235.93.73.125', '239.219.203.191', '239.219.203.237']
Total number of such servers: 36

- Filter: `http and http.response.code != 0`

http response code 0 refers to unreachable server. Hence all exchanges with response code other than 0 is considered a valid. Next iterate through all packets and extracted the ip source address(http server).

1.2

- GET /icsc/index.php?p=../../../../../../../../../../../../../etc/passwd%00 HTTP/1.1
- GET /index.php?p=../../../../../../../../../../../../../etc/passwd%00 HTTP/1.1
- GET /icsc/ICSC09_Advance_Program.pdf/index.php?p=../../../../../../../../../../../../../etc/passwd%00 HTTP/1.1

In all these cases, we have a malicious host with IP address 171.252.215.189 is attempting to perform a directory traversal in the response.

- Filter `http && http.request.uri contains "../"`

this will filter all packets with http request with directory traversal. Then we iterate all the packets and collect the set of ip source address.

1.3

Potentially malicious Host(s):
163.252.63.191

- Filter `ftp and ftp.response.code == 530`

This filter return all ftp responses that have result not logged in may be due to incorrect creds.   Iterate through all packets and keep track of all destination ip address (host ip address) and no of times it appears. If this value is higher than some threshold we can probably say that host is malicious host and they are purposely trying a brute force attack.

1.4

USER calrules
PASS thisissosecure

Ftp doesn’t use encryption. Hence passwords and username are in plain text. ftp response code 230 refers to successful logins.

- Filter : `ftp and ftp.response.code == 230`

This filters all ftp packets with succesful logins, Then i iterated over all packets and made set of tcp streams. Then i extracted each tcp stream indiviually and checked for username and password since ftp is unencrypted. To automate this process we can write script to extract username and password from each stream data.

1.5

Oldest version: 1.3.28. HTTP server IP address with this version: {'171.188.225.159'}.

A http response contains information about the “http server” (in our case Apache) and its current version in field http.server .

Filter : `http && http.server contains "Apache/"`
This filter returns http response packets from the server which run Apache server back to the client.

we iterate through these packets extract source IP address(server’s ip address) and Apache version. we keep track of oldest version and its ip address(there can be more than one such servers) and return oldest version.

1.6

we can identify easily identified by looking at the dns request packets for dns and checking the udp source port in them.

['171.188.225.157', '171.188.225.159']

Filter: `dns && dns.flags.response == 0`

This gives us only those packets that use dns as the application level protocol and we are interested in the query request packets (not query responses from the dns server) since we want to check for clients that use static port numbers for dns queries. dns.flags.response 0 refers to query request packets.

we iterate through all packets and keep set of all different port used for each udp source IP address in python dictionary. Then we check this python dictionary and identify hosts that use same port number all the times.

1.7

- 163.220.224.95,
- 171.188.225.157

Filter: `tcp and tcp.flags.syn == 1 and tcp.flags.ack == 0`

When a host tries to connect to server via tcp, it initiates connection by sending SYN. Hence using above filter we get all tcp packets with only SYN flag. Then i iterate through packets and keep track of all ISN for each source IP address(host). Finally i take difference between max and min in this list for each source IP address and sort them.

1.8

Traceroute utility uses UDP protocol to send a stream of udp packets with increasing ttl values for a given a destination/target. We start with a ttl=1, this packet gets dropped at the first router from the source and returns back a response with destination unreachable message and round trip time. Next the host sends a UDP packet with ttl=2 and this is dropped at the 2nd router along the path. This process continues until we actually reach the desired destination.

In our example, we need to identify a host running “traceroute” by checking for udp packets that do not involve any application layer protocol (like dns) so we accept ‘pure’ udp packets and remove other application layer protocol packets (like dns) that also use udp for transport. This is done using the following filter:

```
udp and not dns and not browser and not mdns and not cups and not nbns and not auto_rp and
```

`not smb_netlogon`

Next, we iterate through udp packets for a each udp endpoint (src,dest) we store list of the unique ttl values . We can find one host in the trace which sends udp packets to the destination with ttl values ranging from ttl=1 to 64.

The host and the target destination are : '163.222.204.223', '167.220.69.239'.

1.9

source IP or host IP : '163.220.224.95'

destinatio IP or server IP : '167.220.77.109'

Filter : `http and http.request.uri contains "<script>" and http.request.uri contains "</script>"`

This filters http all requests that contain that contain `<script>` and `</script>` in their URI.

use filter `(ip.src == 163.220.224.95) && (ip.dst == 167.220.77.109)` and follow http streams to see evidence of attack

- GET <script>document.cookie=%22testhzlg=9267;%22</script> HTTP/1.1
- GET /v9j2h7a7.cgi?<script>document.cookie=%22testhzlg=9267;%22</script>
- GET ?username="<script>foo</script HTTP/1.1

Question 2

1. Multiple hosts sent packets on the local network. What are their MAC and IP addresses?

- 10.0.2.1 -- 00:26:08:e5:66:07 Local NS
- 10.0.2.2 -- 04:0c:ce:d8:0f:fa   User

- 10.0.2.3 -- 8c:a9:82:50:f0:a6 User

2. What type of network does this appear to be (e.g., a large corporation, an ISP backbone, etc.)? Point to evidence from the trace that supports this. It appears to be a small personal network. It is comprised of 3 devices: 2 computers and 1 name server.

3. One of the clients connects to an FTP server during the trace.

(a) What is the DNS hostname of the server it connects to?

dl.xs4all.nl

(b) Is the connection using Active or Passive FTP?

Active FTP - the client uses a PORT command and not a PASV command. In active FTP the user specifies the port on their machine that is awaiting the connection and the server connects to it. In passive FTP the user specifies PASV and the server send the user a port that they should look for data from.

(c) Based on the packet capture, what’s one major vulnerability of the FTP protocol?

Information transferred with FTP is not encrypted, so usernames and passwords sent can be seen in plaintext.

(d) Name at least two network protocols that can be used in place of FTP to provide secure file transfer.

- SFTP
- HTTPS

4. One of the clients makes a number of requests to Facebook.

(a) Even though logins are processed over HTTPS, what is insecure about the way the browser is authenticated to Facebook? The browser uses a cookie to authenticate to

Facebook, and this cookie is visible when the user sends other requests through HTTP.

(b) How would this let an attacker impersonate the user on Facebook?

An attacker could grab the user's cookie and use it as their own.

(c) How can users protect themselves against this type of attack?

Users can't protect themselves from this type of attack, but Facebook can protect their users by using HTTPS for all their requests. Users can use browser that doesn't store cookies or use extensions like https://en.wikipedia.org/wiki/HTTPS_Everywhere to protect themselves

(d) What did the user do while on the Facebook site?

Filter : `http.cookie && http.host contains " facebook.com "` and follow http stream.

The user goes to the main page and searches for someone's name (Zakir Durum). The user goes to Zakir's page and sends him a message with an attachment.

Question 3

Part A

1. What is the full command you used to run the port scan (including arguments)?

   ```
   nmap -T4 -p- -sS -A scanme.nmap.org
   ```

2. What is the IP address of scanme.nmap.org?

   45.33.32.156

3. What operating system is the target server running? What version number?

   Aggressive OS guesses: Linux 5.0 - 5.4 (96%), Linux 5.4 (95%), Linux 4.15 - 5.6 (94%), Linux 5.0 - 5.3 (93%), Linux 2.6.32 - 3.13 (93%), Ubiquiti AirMax NanoStation WAP (Linux 2.6.32) (93%), Linux 5.1 (92%), Linux 2.6.22 - 2.6.36 (91%), Linux 3.10 - 4.11 (91%), Linux 5.0 (91%)

4. What ports are open on the target server? What applications are running on those ports? (For this part, you only need to report the service name printed by nmap.)

   22 — ssh, 80 — http, 9929 — nping-echo, 31337 — Elite

5. The target machine is running an SSH server. What SSH software and version is being used?

   OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)

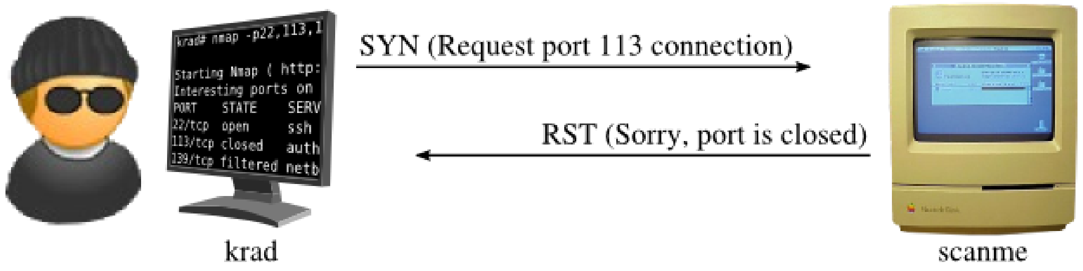
6. The target machine is also running a webserver. What webserver software and version is being used? What ports does it run on?

   Apache httpd 2.4.7 ((Ubuntu)), 80/tcp

Part B

1. what does it mean for a port on scanme.nmap.org to be “closed?” More specifically, what is the TCP packet type, if any, the server gives in response to a SYN packet sent to port that is “closed?”

The first step is a Nmap sends the SYN probe to scanme.nmap.org . But instead of receiving a SYN/ACK back, a RST is returned. Thhis mean the port is closed.



```
krad# nmap -p22,113,1

Starting Nmap ( http:
Interesting ports on
PORT     STATE    SERV
22/tcp   open     ssh
113/tcp  closed   auth
139/tcp  filtered netb
```

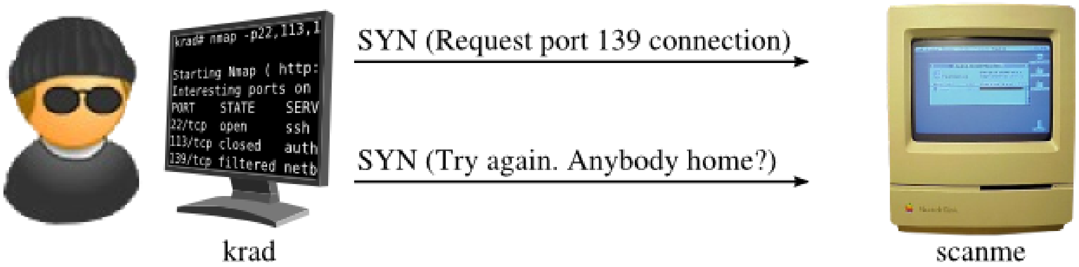
SYN (Request port 113 connection)
RST (Sorry, port is closed)
krad
scanme

2. What does it mean for a port on scanme.nmap.org to be “filtered?” More specifically, what is the TCP packet type, if any, the server gives in response to a SYN packet sent to port that is “filtered?”

The initially nmap sends a SYN, as usual, but sees no reply. The response could simply be slow. From previous responses (or timing defaults), Nmap knows how long to wait and eventually gives up on receiving one. A non-responsive port is usually filtered (blocked by a firewall device, or perhaps the host is down), but this one test is not conclusive. Perhaps the port is open but the probe or response were simply dropped.

Networks can be flaky. So Nmap tries again by resending the SYN probe. After yet another timeout period, Nmap gives up and marks the port `filtered`.

In this case, only one retransmission was attempted. As described in the section called “Scan Code and Algorithms”
, Nmap keeps careful packet loss statistics and will attempt more retransmissions when scanning less reliable networks.



3. In addition to performing an HTTP GET request to the webserver, what other http request types does nmap send?

The additional requests that nmap sends are

- POST
- OPTIONS
- PROPFIND