

Abstract

This report presents the analysis of the packages gathered by Wireshark. I did observe several protocols vulnerabilities. They are related to unsafe handling of data. Unsecure protocols such as FTP and HTTP uses plain-text transmission of passwords over the network. Report also covers tools to diagnose network like ping and traceroute. Ping is used to test reachability, on the other hand traceroute is used to obtain messages path.

Introduction

Our task was to investigate network traffic while given tasks were performed. The virtual machine, on which all tasks were executed runs Ubuntu with root privileges. I used Wireshark in order to monitor and sniff local network traffic. Everything else was already preinstalled.

Ping

Ping is a administration utility used to test host reachability. In my cases I pinged to google.com host. In order to obtain IP address of given host DNS query was executed.

No.	Source	Destination	Protocol	Length	Info
1	10.0.2.192	168.1.1	DNS	70	Standard query 0x4aa9 A google.com
2	192.168.10.0	2.15	DNS	326	Standard query response 0x4aa9 A 89.228.4.222 A 89.228.4.251 A 89.228.4.247 A

Figure 1: Sniffed DNS packages

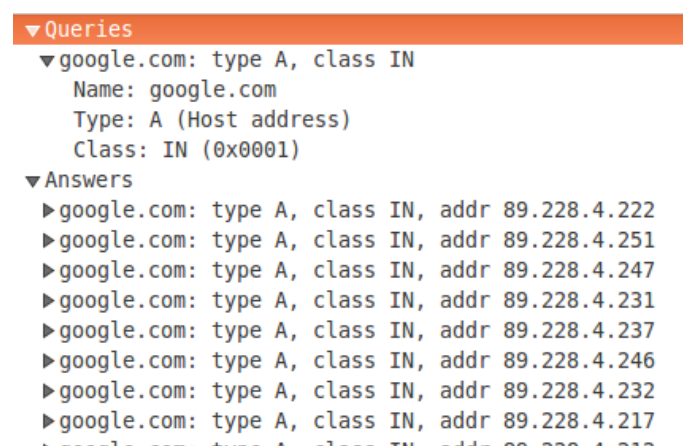


Figure 2: DNS query response

Ping uses ICMP protocol's 'echo' request and reply message types. ICMP is categorized as a layer 3 protocol in the OSI model (Network layer). ICMP packet is an IP packet with ICMP data in it. Ping is limited as a diagnostic tool, information received declares if we can reach the

7	10.0.2.89.228.4.222	ICMP	98	Echo (ping) request	id=0x0d85, seq=2/512, ttl=64 (reply in 8)
8	89.228.10.0.2.15	ICMP	98	Echo (ping) reply	id=0x0d85, seq=2/512, ttl=58 (request in 7)
9	10.0.2.89.228.4.222	ICMP	98	Echo (ping) request	id=0x0d85, seq=3/768, ttl=64 (reply in 10)
10	89.228.10.0.2.15	ICMP	98	Echo (ping) reply	id=0x0d85, seq=3/768, ttl=58 (request in 9)
11	10.0.2.89.228.4.222	ICMP	98	Echo (ping) request	id=0x0d85, seq=4/1024, ttl=64 (reply in 12)
12	89.228.10.0.2.15	ICMP	98	Echo (ping) reply	id=0x0d85, seq=4/1024, ttl=58 (request in 11)
13	10.0.2.89.228.4.222	ICMP	98	Echo (ping) request	id=0x0d85, seq=5/1280, ttl=64 (reply in 14)
14	89.228.10.0.2.15	ICMP	98	Echo (ping) reply	id=0x0d85, seq=5/1280, ttl=58 (request in 13)
15	10.0.2.89.228.4.222	ICMP	98	Echo (ping) request	id=0x0d85, seq=6/1536, ttl=64 (reply in 16)
16	89.228.10.0.2.15	ICMP	98	Echo (ping) reply	id=0x0d85, seq=6/1536, ttl=58 (request in 15)
17	10.0.2.89.228.4.222	ICMP	98	Echo (ping) request	id=0x0d85, seq=7/1792, ttl=64 (reply in 18)
18	89.228.10.0.2.15	ICMP	98	Echo (ping) reply	id=0x0d85, seq=7/1792, ttl=58 (request in 17)
19	10.0.2.89.228.4.222	ICMP	98	Echo (ping) request	id=0x0d85, seq=8/2048, ttl=64 (reply in 20)
20	89.228.10.0.2.15	ICMP	98	Echo (ping) reply	id=0x0d85, seq=8/2048, ttl=58 (request in 19)

Figure 3: Echo request/replay packages

target network element. Moreover it doesn't tell if we can reach target service. It can only tell determine if we can reach the IP layer of the target.

Traceroute

Traceroute is a diagnostic tools used to diagnosis latency in the transit path. This time around I decided to ping google DNS server.

```

traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
 1  192.168.1.1 (192.168.1.1)  1.313 ms  1.270 ms  1.220 ms
 2  10.146.0.1 (10.146.0.1)  9.006 ms  12.808 ms  11.910 ms
 3  host-89-228-14-57.dynamic.mm.pl (89.228.14.57)  27.778 ms  27.749 ms  27.708
ms
 4  gdy-lub-1.gdynia.mm.pl (217.172.225.145)  28.853 ms  28.816 ms  28.781 ms
 5  89.228.2.33 (89.228.2.33)  27.543 ms host-176-221-97-1.dynamic.mm.pl (176.22
1.97.1)  27.718 ms host-176-221-97-233.dynamic.mm.pl (176.221.97.233)  27.670 ms
 6  89.228.6.38 (89.228.6.38)  27.414 ms  22.469 ms  25.769 ms
 7  de-cix20.net.google.com (80.81.193.108)  48.859 ms  44.108 ms  48.327 ms
 8  209.85.253.244 (209.85.253.244)  49.245 ms  51.416 ms  51.369 ms
 9  209.85.251.248 (209.85.251.248)  48.959 ms 209.85.251.178 (209.85.251.178)
48.000 ms 72.14.234.231 (72.14.234.231)  48.951 ms
10 209.85.240.142 (209.85.240.142)  57.813 ms  57.764 ms  57.704 ms
11 209.85.255.75 (209.85.255.75)  57.650 ms 209.85.254.213 (209.85.254.213)  61
.058 ms 209.85.255.75 (209.85.255.75)  56.404 ms
12 216.239.49.28 (216.239.49.28)  61.369 ms 209.85.255.51 (209.85.255.51)  61.3
48 ms 209.85.254.189 (209.85.254.189)  54.075 ms
13 * * *
14 google-public-dns-a.google.com (8.8.8.8)  57.513 ms  61.344 ms  58.564 ms

```

Figure 4: Traceroute response

Traceroute sends ICMP Echo Request packages with TTL (Time-to-live) value starting with 1. Each receiver decrement packet' TTL value by one, when TTL value is equal to zero packet is dropped. As a response router sends an ICMP Time Exceeded message to the source. Thus gives us an ability to reproduce transmit path.

The path (following IP addresses) alongside with round-trip times of packets was displayed in terminal (Figure 4).

36	192.168.1.100	8.8.8.8	ICMP	74 Echo (ping) request id=0x08fc, seq=18/4608, ttl=6
37	217.172.225.145	192.168.1.100	ICMP	182 Time-to-live exceeded (Time to live exceeded in transi
38	217.172.225.145	192.168.1.100	ICMP	182 Time-to-live exceeded (Time to live exceeded in transi
39	192.168.1.100	8.8.8.8	ICMP	74 Echo (ping) request id=0x08fc, seq=19/4864, ttl=7
40	192.168.1.100	192.168.1.1	DNS	83 Standard query 0x5c41 PTR 1.0.146.10.in-addr.arpa
41	192.168.1.1	192.168.1.100	DNS	160 Standard query response 0x5c41 No such name
42	192.168.1.100	8.8.8.8	ICMP	74 Echo (ping) request id=0x08fc, seq=20/5120, ttl=7
43	192.168.1.100	8.8.8.8	ICMP	74 Echo (ping) request id=0x08fc, seq=21/5376, ttl=7
44	192.168.1.100	8.8.8.8	ICMP	74 Echo (ping) request id=0x08fc, seq=22/5632, ttl=8
45	192.168.1.100	192.168.1.1	DNS	85 Standard query 0x13d3 PTR 57.14.228.89.in-addr.arpa
46	80.81.193.108	192.168.1.100	ICMP	70 Time-to-live exceeded (Time to live exceeded in transi
47	192.168.1.1	192.168.1.100	DNS	130 Standard query response 0x13d3 PTR host-89-228-14-57.
48	192.168.1.100	8.8.8.8	ICMP	74 Echo (ping) request id=0x08fc, seq=23/5888, ttl=8
49	192.168.1.100	8.8.8.8	ICMP	74 Echo (ping) request id=0x08fc, seq=24/6144, ttl=8
50	192.168.1.100	8.8.8.8	ICMP	74 Echo (ping) request id=0x08fc, seq=25/6400, ttl=9

Figure 5: ICMP Echo Request TTL values

35	10.146.0.1	192.168.1.100	ICMP	70 Time-to-live exceeded
36	10.146.0.1	192.168.1.100	ICMP	70 Time-to-live exceeded
37	10.146.0.1	192.168.1.100	ICMP	70 Time-to-live exceeded
38	89.228.14.57	192.168.1.100	ICMP	70 Time-to-live exceeded
39	89.228.14.57	192.168.1.100	ICMP	70 Time-to-live exceeded
40	89.228.14.57	192.168.1.100	ICMP	70 Time-to-live exceeded
41	89.228.6.38	192.168.1.100	ICMP	70 Time-to-live exceeded
42	89.228.2.33	192.168.1.100	ICMP	70 Time-to-live exceeded
43	176.221.97.233	192.168.1.100	ICMP	70 Time-to-live exceeded
44	176.221.97.1	192.168.1.100	ICMP	70 Time-to-live exceeded
45	217.172.225.145	192.168.1.100	ICMP	182 Time-to-live exceeded
46	217.172.225.145	192.168.1.100	ICMP	182 Time-to-live exceeded

Figure 6: ICMP with TTL Expired filter

FTP auth

FTP (File Transfer Protocol) is used to transfer files in network. There is multiple methods to secure FTP transfers like FTPS or SFTP. For our task we used standard FTP protocol, without any encryption.

```
Name (ftp.icm.edu.pl:jd): anonymus
331 Please specify the password.
Password:
530 Login incorrect.
Login failed.
ftp> 
```

Figure 7: FTP login - terminal

Our job was simply sniff the password sent through FTP protocol.

Here is the sniffed data in a raw form.

The password and username were sent as a ASCII characters. AN attacker has access to all three things required to lo on to FTP account (destination IP address, username and password).

Same rule applies to file transfer. Files transferred via FTP were also sent as a plain-text. We are able to reconstruct entire file using Follow TCP Stream option.

47	192.168.1.100	193.219.28.2	FTP	80 Request: PASS 1234567
49	193.219.28.2	192.168.1.100	FTP	88 Response: 530 Login incorrect.
51	192.168.1.100	193.219.28.2	FTP	72 Request: SYST
53	193.219.28.2	192.168.1.100	FTP	104 Response: 530 Please login with USER a

▶ Frame 47: 80 bytes on wire (640 bits), 80 bytes captured (640 bits) on interface 0
▶ Ethernet II, Src: CadmusCo_c7:e8:a7 (08:00:27:c7:e8:a7), Dst: Tp-LinkT_fb:9c:6a (54:e6:fc:fb:9c:6a)
▶ Internet Protocol Version 4, Src: 192.168.1.100 (192.168.1.100), Dst: 193.219.28.2 (193.219.28.2)
▶ Transmission Control Protocol, Src Port: 35797 (35797), Dst Port: ftp (21), Seq: 16, Ack: 1952, Len: 14
▼ File Transfer Protocol (FTP)
▼ PASS 1234567\r\n
Request command: PASS
Request arg: 1234567

Figure 8: sniffed packages

HTTP auth

HTTP auth uses a combination of a username and password to authenticate the user. Server as a response sends a message with "Authorization Required" header. User is prompted to enter username and password, entered data is sent in 'Authorization' header. Username and password is encoded using the Base 64. In our case Wireshark (as shown in Figure 9) decrypted it for us (username: test, password: test).

6	176.9.30.169	192.168.1.100	HTTP	703 HTTP/1.1 401 Authorization F
7	192.168.1.100	176.9.30.169	TCP	66 35338 > http [ACK] Seq=287 A
8	192.168.1.100	176.9.30.169	HTTP	381 GET /favicon.ico HTTP/1.1
9	176.9.30.169	192.168.1.100	HTTP	703 HTTP/1.1 401 Authorization F
10	192.168.1.100	176.9.30.169	TCP	66 35338 > http [ACK] Seq=602 A
11	192.168.1.100	176.9.30.169	HTTP	370 GET / HTTP/1.1
12	176.9.30.169	192.168.1.100	HTTP	703 HTTP/1.1 401 Authorization F
13	192.168.1.100	176.9.30.169	TCP	66 35338 > http [ACK] Seq=906 A
14	192.168.1.100	176.9.30.169	HTTP	405 GET / HTTP/1.1
15	176.9.30.169	192.168.1.100	HTTP	703 HTTP/1.1 401 Authorization F
16	192.168.1.100	176.9.30.169	TCP	66 35338 > http [ACK] Seq=1245

▶ Transmission Control Protocol, Src Port: 35338 (35338), Dst Port: http (80), Seq: 906, Ack: 1912
▼ Hypertext Transfer Protocol
▶ GET / HTTP/1.1\r\n
Host: trumpet.nd.s4.stermedia.eu\r\n
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:31.0) Gecko/20100101 Firefox/31.0\r\n
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
Accept-Language: en-US,en;q=0.5\r\n
Accept-Encoding: gzip, deflate\r\n
Connection: keep-alive\r\n
▼ Authorization: Basic dGVzdDp0ZXN0\r\n
Credentials: test:test

Figure 9: sniffed HTTP-Base Auth

References

- [1] <http://www.networksorcery.com/enp/protocol/icmp.htm>
- [2] <http://linux.die.net/man/8/traceroute>

[3] <http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html>