CS177: Computer Security
Prof. Stefano Tessaro

UC Santa Barbara
Fall 2018

Homework 6

Posted: Wednesday, November 28, 2018 – 11:59pm
Due: **Friday**, December 7, 2018 – 11:59pm (on Gradescope)

Task 1 – Denial-of-Service Attacks (5 points)

You want to perform a denial-of-service (DoS) attack against some host with a known IP address. You have become aware that a mis-configured sub-network, corresponding to the range `w.x.y.z/24`, allows for external access to its *broadcast address*, i.e., traffic sent to `w.x.y.255` reaches *all* hosts simultaneously on that sub-network. Note that neither you nor your target are on this sub-network.

Describe as clearly as possible how you can take advantage of the sub-network to perform a denial-of-service attack on the target.

Task 2 – TCP SYN Floods (8 points)

The main idea behind a "SYN flood" DoS-attack is that every TCP connection initiated with a SYN message makes the server allocate some memory – containing some information about the connection – while it waits for the client's final ACK message. Typically, while waiting for the client's ACK, the server stores the connection information (client/server IP and ports), as well as the *maximum segment size* (MSS), the maximal size of the data a TCP packet may contain, which is communicated by the client in its initial SYN's message. We assume here that there are only 8 possible values for the MSS.

Consider now the following choice of the server's sequence number for a connection. The server chooses the 32-bit sequence number seq_s defined as follows:

- The first 5-bit are a time-counter t, increased every minute, and reduced modulo 32 to fit in 5 bits.
- The middle 3 bits (denote them as m) encode 8 possible values for the MSS field.
- The last 24 bits are the output σ of a cryptographic message-authentication code MAC (using a secret key only known to the server) applied to the concatenation of the server IP address and port number, the client IP address and port number, and the values t and m.

In other words, $\mathrm{seq}_s = t \parallel m \parallel \sigma$, where $\parallel$ denotes concatenation.

Explain how this choice of seq_s can be used by the server to prevent SYN flooding!

Task 3 – TCP SYN Floods, Second Take (8 points)

The following countermeasure against SYN flood attacks is widely adopted, and is known as a "SYN cache". For simplicity, we assume that the server only responds on port 80, and only needs to remember the client's sequence number seq_C and its own sequence number seq_S between the second and third message in the TCP handshake. The server maintains a table with exactly B cells (e.g., $B = 2^{16}$), numbered $1, \dots, B$, and each cell in the table can store an entry with form

$$(\text{ClientIP}, \text{ClientPort}, \text{seq}_S, \text{sec}_C) \ .$$

In particular, the handshake is now modified as follows:

- When a client with IP ClientIP initiates the handshake with the server, sending a SYN message from port ClientPort to the server's port 80 with sequence number seq_C, the server generates its initial sequence number seq_S, and stores $T = (\text{ClientIP}, \text{ClientPort}, \text{seq}_S, \text{sec}_C)$ in the table at location i_T, where i_T is a deterministic function of T. (E.g., one applies a MAC to T, and then maps the MAC output to a number between 1 and B.) **If cell i_T already stores something, its contents are replaced by T.** The server then sends back a SYN/ACK message with sequence number seq_S and ACK number $\text{seq}_C + 1$. *The server does not remember anything else in its memory.*
- When the server receives an ACK message from port ClientPort of a client with IP address ClientIP with sequence number seq_C+1 and ACK number seq_S+1, the server checks whether $T = (\text{ClientIP}, \text{ClientPort}, \text{seq}_S, \text{sec}_C)$ is at location i_T (note that i_T can be recomputed as above). If yes, it continues the TCP connection as usual. If not, it simply aborts the handshake.

a) Why does a SYN cache mitigate the effects of SYN flood attacks?

b) What is a possible disadvantage of using a SYN cache?

Task 4 – The Network Time Protocol (5 points)

The network time protocol (NTP) is used to synchronize Internet devices to within a few milliseconds of UTC time. The current version of the protocol, NTPv4, is an evolution of the protocol described in RFC 5905 (`https://tools.ietf.org/html/rfc5905`), and is built on top of UDP, i.e., requests to NTP servers are via UDP packets.

Earlier versions of NTP had servers accept a command called "monlist" for monitoring purposes. It returns the addresses of up to the last 600 machines that the NTP server has interacted with.

Why is "monlist" potentially problematic?

Task 5 – NMAP and Port Scanning (9 points)

a) Use NMAP to find out which ports are open on our CS177 machine (192.35.222.247)!

b) Which services do these ports run? (Give both the names of these services and their versions!)

c) John runs his own Linux machine, connected directly to the Internet with a fixed IP address. He explicitly configured his machine to have SSH use the non-standard port 4242. He claims that this is going to confuse attackers, and he therefore does not need to worry about setting strong passwords.

(1) Give an example of an attack scenario where John's suggestion is useful.
(2) Give an example of an attack scenario where John's suggestion is not useful.

Explain both answers!

Note: You can use NMAP directly from your account on the CS177 machine, using "localhost" as the host to be scanned. Do not scan other machines!