CSCI3403 CyberSecurity
Name: Chen Hao Cheng

Review Question Chapter 7

7.9) Define a reflection attack

The attacker sends the network packet with a spoofed source address to service runs on the network server and the server responds back to this packet by sending it to the spoofed address that belongs to authentic attack target.

7.10) Define an amplification attack

It is used to transmit a packet with spoofed source address to the target system through mediators. This is different than reflection attack.

- It will generate multiple responses for each original packet transmitted and it's achieved by sending the original request to some other network by broadcasting the address after transmission.
- The host on the entire network responds to the request and generates a huge number of responses
- Finally, it used DNS that generates a longer response than original request

Problems Chapter 7:

2) 256 TCP connection requests. The system retries sending the SYN-ACK packet 5 times when it fails receiving ACK in response at 30 seconds intervals.

SYN-ACK packet is 1(initial) + 5(repeat) = 6, and 30 seconds intervals, so it's 6 * 30 = 180 seconds = 3 minutes

Number of requests that attacker needs to send through TCP connection:

$$\frac{Number\ of\ connection\ requests\ in\ table}{Rate\ of\ the\ attacker\ to\ send\ the\ TCP\ requests} = \frac{256}{3} \approx 86\ requests/minute$$

Assuming the TCP SYN packets is 40 bytes in size
40 bytes = 320 bits, I minute = 60 seconds

Bandwidth required or attacker = $\frac{86*320}{60} = 458.66\ bits/seconds$

7.3) Assuming ICMP(Internet Control message Protocol) echo request packets with a size of 500 bytes and an average uplink capacity of 128 kbps.

1 byte = 8 bits, 1 kbps = 10^3 bits
Maximum number of packets sent by single zombie: $\frac{128*10^3}{500*8} = \frac{128000}{4000} = 32$

Zombie systems would need to flood a target organization using 0.5-Mbps link?

1 Mbps = 10^6 bits/second , 1 kbps = 10^3 bits/second
Using 0.5 Mbps: $\frac{0.5*10\hat{}6}{128*10\hat{}3} = \frac{500}{128}$ = 3.906 ≈ 4

Zombie systems would need to flood a target organization using 2-Mbps link?

Using 2 Mbps: $\frac{2*10^6}{128*10^3} = \frac{2000}{128} \approx 16$

Zombie systems would need to flood a target organization using 10-Mbps link?

Using 10 Mbps: $\frac{10*10^6}{128*10^3} = \frac{10000*10^3}{128*10^3} \approx 78$

What can I conclude based on botnet?

It is possible to launch the DoS(denial of service) attacks on multiple systems with a given report of botnets composed of many thousands of zombie systems.
It's an attack on major of the organization with multiple larger network links
For instance: 1000 zombies with 128 kbps links can flood 128 Mbps of network link capacity.

7.4)
Consider a DNS packet using packets with a size of 500 bytes and the intermediary with the size of 60 bytes.
Note: 1 Mbps = 10^6 bites per second and 1 byte = 8 bits

0.5 Mbps = $\frac{0.5*10^6}{500*8} = \frac{500000}{4000} = 125$, so 125 packets will be sent per second

Required bandwidth = $125 * 60 * 8 = 60 * 10^3$ = 60 Kbps

2 Mbps = $\frac{2*10^6}{500*8} = \frac{2000000}{500*8} = 500$, so 500 packets will be sent per second

Required bandwidth = $500 * 60 * 8$ = 240000 = 240 Kbps

10Mbps = $\frac{10*10^6}{500*8} = \frac{10000000}{4000} = 2500$, so 2500 packets will be sent per second

Required bandwidth = 2500 * 60 * 8 = 1200000 = 1.2 * 10^6 = 1.2 Mbps

Amplification of three data rates:

Consider the "number of DNS response packets" as 500, and "size of intermediary DNS packet" as 60

Amplification of three data rates = $\frac{Number\ of\ DNS\ response\ packets}{Size\ of\ intermediary\ DNS\ packet} = \frac{500}{60} = 8.33$, in all three cases the amplification is 8.33 times.

Review Question Chapter 10
10.9) Describe what a NOP sled is and how it is used in a buffer overflow attack?

NOP = No-operation; it is a sequence of no-operation instructions to move the program execution flow to the specific location in the buffer memory.

i) Attacker find the starting address of the program to attack
    (1) Attacker find the size of creating program should be less than the size of free space existing in the buffer
    (2) Attacker can fill the free space with "No-operation" instructions and insert the shellcode into the buffer after the instructions

(a) This action transfer the control of the program to the attacker by returning the actual location of the shellcode

(3) The CPU execute all "No-operation" instructions until reach the actual shellcode in the buffer

(4) The buffer will be overflowed

Problems Chapter 10

10.2) Rewrite this buffer overflow C code:

```
    int main(int argc, char *argv[]) {
        int valid = FALSE;
        char str1[8];
        char str2[8];

        next_tag(str1);
        gets(str2);
        if (strncmp(str1, str2, 8) == 0)
            valid = TRUE;
        printf("buffer1: str1(%s), str2(%s), valid(%d)\n", str1, str2,valid);
    }
```

The following code is correct:

```
int main (argc, char *argv[]){
    int valid = FALSE;
    char str1[8];
    char str2[8];

    next_tag(str1); //fxn to read string value from user and store into str1
    fgets(str2, sizeof(str2), stdin); // Read string value with limited
character size
    if(strncmp(str1, str2, sizeof(str2)) == 0){ //Compare two strings to check
if the strings are valid or not
    valid = TRUE
    printf("buffer1: str1(%s), str2(%s), valid(%d)\n", str1, str2,valid);
    }
}
```

10.3) Review this stack overflow C code:

```
void hello(char *tag)
{
    char inp[16];

    printf("Enter value for %s: ", tag);
    gets(inp);
    printf("Hello your %s is %s\n", tag, inp);
}
```

The following code is correct:

```
void hello(char *tag){
    char inp[16];
    printf("Enter value for %s: ", tag);

    fgets(inp, sizeof(inp), stdin); //read string value with limited size
    printf(“Hello your %s is %s\n”, tag, inp);
}
```