CO2515 Coding for Penetration Testing Worksheet 2

Aim – To introduce students to some basic PT coding using Python3 and Scapy with IPv6. Only basic programming skills are assumed. All code is written to be as simple as possible.

NOTE – The techniques discussed in this worksheet should only be used on networks with the full written permission of the network owner otherwise you may be in breach of the GDPR.

|  |  |
| --- | --- |
| 1. | In this Lab you will complete a TCP three-way hand shake and send a GET HTTP request. |
| 2. | When sending a SYN from the attacker and the SYN-ACK is returned. Because the Linux kernel receives the SYN-ACK but didn't send the SYN it will issue a RST. To prevent this IP6tables can be used, using the syntax below on the attacking machine. |
| 3. | **Sudo IP6tables -A output -p tcp –tcp-flags RST RST -J DROP**  This prevents any RST packets coming out of the attacker machine to the server. |
| 4. | Now we will emulate the TCP three-way handshake below. |
| 5. | Now create a new script called **synack.py**. |
| 6. | Add the following code: |
|  | **#!/usr/local/bin/python**  **from scapy.all import \***  **import sys**  **dest=sys.argv[1]**  **getStr = 'GET / HTTP/1.0\n\n'**  **# SYN**  **SYN=(IPv6(dst=dest)/TCP(sport=333,dport=80, flags='S',seq=1000))** |
|  | At this point we are only creating a simple SYN packet.  You will notice the **getStr** variable. This will be explained later. The seq number is simply to start a TCP sequence for the other packets after this one. |
|  | Now add this underneath it. This simply sends the SYN packet and gets the SYN-ACK response from the server. |
|  | **#Get SYN/ACK**  **SYN\_ACK = sr1(SYN)** |
|  | Start Wireshark and run this script:  **sudo ./synack.py fd80:1234:5678:2::254**  Try **netstat -anp** on Gserver, what do you see? |
|  | The next part of the sequence is to send an ack response to the SYN/ACK from the server as well as a HTTP GET request.  Add this to the code: |
|  | **#ACK/HTTP GET**  **ACK=send(IPv6(dst=dest)/TCP(sport=333, dport=80, flags='A', seq=SYN\_ACK.ack, ack=SYN\_ACK.seq +1)/getStr)** |
|  | For segment of the TCP sequence, a +1 is required or it will be considered the same as the previous statement. This statement, therefore, defines the TCP sequence it is apart of and then states it is the next packet in the sequence. It then issues a Get HTTP request from the get STR statement seen earlier. |
|  | The script overall should look like this: |
|  | **#!/usr/local/bin/python**  **from scapy.all import \***  **import sys**  **dest=sys.argv[1]**  **getStr = 'GET / HTTP/1.0\n\n'**  **# SYN**  **SYN=(IPv6(dst=dest)/TCP(sport=333,dport=80, flags='S',seq=1000))**  **#Get SYN/ACK**  **SYN\_ACK = sr1(SYN)**  **#ACK/HTTP GET**  **ACK=send(IPv6(dst=dest)/TCP(sport=333, dport=80, flags='A', seq=SYN\_ACK.ack, ack=SYN\_ACK.seq +1)/getStr)** |
|  | Start a packet capture and run the script.  Check the analyse the data sent from the Get HTTP packets. |