**Session Hijack:**

**Intrusion Detection System**

Alex Amos

3551496

Adam Marciszewski

3525963

**Setup**

Required Applications:

* Vsftpd (FTP server Ubuntu)
* Python 3
* Scapy (Comes packaged with python 3)
* dsniff

Required Hosts:

* Victim (Ubuntu)
* Attacker (Kali, Ubuntu or similar)
* Client (Any OS)

**Vsftpd**

* On Ubuntu:

#sudo apt-get install vsftpd

**Note:** Vsftpd will default the ftp user to the same user installing it, meaning log in credentials will be the same.

**Python 3.7 or later**

* On Ubuntu:

#python3 --version

If version is not >= 3.7 or python 3 not installed:

#sudo apt-get install python3

**Scapy**

* On Ubuntu:
* Check for folder scapy in /usr/lib/python3/dist-packages/
* If it is not there, copy folder /SessionHijackIDS/scapy/ to

/usr/lib/python3/dist-packages/

**dsniff**

* On attacker:

#sudo apt-get install dsniff

**Features**

This application captures packets using python 3 and the scapy library to detect an FTP session Hijack attack using arp spoofing. The script must be run as root. Scapy requires root privileges to sniff packets on the NIC. As packets are captured they will be periodically written to log files in the SessionHijackIDS/log/ folder. The log files are named based on day created and a new file will be created each day as Analyzer.py runs. By default if a packet is captured that doesn’t have a port number, the log will save packet with –1 as port numbers.

Analyzer Detects the attack using 2 Features:

1. Number of arp replies in 1 minute
2. Number of login attempts for a username

**# of Arp Replies**

To capture this feature the application tracks arp replies coming in from each host. If a host is found to have sent 10 or more arp replies in 1 minute, a preliminary arpspoof attack is detected. The application will warn the user of the arpspoof attack in the terminal, as well as print the time that the attack began, and print the MAC address of the attacking host.

**# of Username Attempts**

The second feature is # of username attempts. If a username is attempted 2 or more times the application checks for an arpspoof attack using the above feature. If an arpspoof has been detected then the application matches the MAC address of the attacker the MAC addresses for the username attempts. In this case it will most likely be checking against multiple MAC addresses. If a match is found, meaning the same host conducting the arpspoof also attempted a login by another host’s username, then a session hijack is detected and the user is notified of the attack as well as the credentials stolen.

**Execution**

**Victim**

* Open a terminal:

#sudo service vsftpd start

#sudo service vsftpd status (to ensure it is running)

* Go to SessionHijackIDS/

#sudo ./Analyzer.py

**Attacker**

* Open a terminal:

#sysctl -w net.ipv4.ip\_forward=1

#arpspoof -t [Victim IP] [Client IP]

* Open another terminal:

#arpspoof -t [Client IP] [Victim IP]

* Open another terminal:

#dsniff -t 21/tcp=ftp -n

**Client**

* Open web browser:

ftp://[victim ip]/

* Log in using victim credentials

**Attacker**

* Grab sniffed credentials from dsniff terminal
* Open a web browser:

ftp://[victim ip]/

* Log in using sniffed credentials.

**Victim**

* During arpspoof the application will print Arpspoof detection to terminal then remain idle.
* When attacker logs in with stolen credentials the application will print Session Hijack detection to terminal and warn the user of the stolen credentials.