Part 2: The Incident Alarm

"Scapy is a Python module created by Philippe Biondi that allows extensive packet manipulation. Scapy allows packet forgery, sniffing, PCAP reading/writing, and real-time interaction with network targets. Scapy can be used interactively from a Python prompt or built into scripts and programs" (from the SANS Institute's Scapy Cheat Sheet (Links to an external site.)).

Scapy and Python 3 are installed on Kali Linux.

We have covered a number of network scanning techniques, and you practiced finding sensitive information in PCAP files in the previous lab. This time, you will apply your knowledge to write a tool that provides notification of incidents via a live stream of network packets or via a set of packets in a PCAP file.

Part 2: Instructions

Using Python and scapy, write a program named alarm.py that provides user the option to analyze a live stream of network packets or a set of PCAPs for incidents. Your tool shall be able to analyze for the following incidents:

- NULL scan
- FIN scan
- Xmas scan

- Usernames and passwords sent in-the-clear via HTTP Basic Authentication or FTP
- Nikto scan
- Someone scanning for Remote Desktop Protocol (RDP)

If an incident is detected, alert must be displayed in the format:

ALERT #{incident_number}: #{incident} is detected from #{source IP address} (#{protocol or port number}) (#{payload})!

Example outputs: ALERT #1: Xmas scan is detected from 192.168.1.3 (TCP)!
ALERT #2: Usernames and passwords sent in-the-clear (HTTP) (username:batman, password:brucewayne)

Your program does not need to support saving the stream of packets to a PCAP file or saving a record of detected incidents.

No credit if you program crashes or if exceptions are not handled properly.

Part 2: Running and Using the Tool

In Kali Linux and assuming you are root, run: python3 alarm.py. By default with no arguments, the tool shall sniff on network interface eth0. The tool must handle three command line arguments:

`-i INTERFACE: Sniff on a specified network interface`
`-r PCAPFILE: Read in a PCAP file`
`-h: Display message on how to use tool`

Example 1: python3 alarm.py -h shall display something of the like:

usage: alarm.py [-h] [-i INTERFACE] [-r PCAPFILE]

A network sniffer that identifies basic vulnerabilities

optional arguments: -h, --help show this help message and exit -i INTERFACE Network interface to sniff on -r PCAPFILE A PCAP file to read

Example 2: python3 alarm.py -r set2.pcap will read the packets from set2.pcap

Example 3: python3 alarm.py -i en0 will sniff packets on a wireless interface en0

When sniffing on a live interface, the tool must keep running. To quit it, press Control-C

Part 2: Getting Started

Here is a working alarm.py (in Python 3): https://gist.github.com/mchow01/f0f498f29d2b3bd095b8c93172c6ecf7 (Links to an external site.)

Feel free to modify the packetcallback function. What has been written for you: the handling and parsing of command line arguments, reading of PCAP file, and sniffing of network. Download and use inside of your Kali VM. You will also need to install pcapy to work in conjunction with scapy on Kali Linux as it is not installed. Run apt-get install python3-pcapy

If you go web browsing in the virtual machine with the alarm running, you will notice the alarm will go off...

Part 2: Testing Your Tool

Your tool must be able to detect the usernames and passwords sent in-the-clear in set1.pcap, set2.pcap, and set3.pcap from the Packet Sleuth lab (Lab 2).

Here are PCAPs you can also use to test your alarm:

1. fin.pcapLinks to an external site.
2. xmas.pcapLinks to an external site.
3. null.pcapLinks to an external site.
4. nikto.pcapLinks to an external site.

Part 2: References

- Scapy documentation: https://scapy.readthedocs.io/en/latest/ (Links to an external site.) (Links to an external site.)
- Scapy Cheat Sheet (SANS Institute): https://blogs.sans.org/pen-testing/files/2016/04/ScapyCheatSheet_v0.2.pdf (Links to an external site.)