Playbook

## **Purpose**

This utility toolset is composed of 2 python scripts which can be used to analyze captured packet data or live network traffic.

## **Installation**

The utility can be installed by clone the git repo with (*git clone “repo url”*) or unzip a zip of the git repo.

The dpkt and psutil pip modules are required to run as well.

## **Execution/Usage**

1. slowfile.py takes 2 arguments, IP and FILE. The IP argument is used to identify the local server in the capture file in order to determine the direction of traffic. The FILE argument is the location of the capture file.

usage: slofile.py [-h] --IP IP --FILE FILE

./slofile.py –IP 155.6.2.91 --FILE capture.pcap

1. sloburn.py takes 2 arguments, IP and PORT. The IP argument is used to identify the local address in order to determine direction of the traffic. The PORT argument is used to filter the data on a specific port. Only a single port is supported at this time.

usage: sloburn.py [-h] --IP IP --PORT PORT

./sloburn.py --IP 10.0.2.15 --PORT 80

## 

## 

## 

## 

## 

## **Rule creation**

Rules can be added or changed in both scripts. These rules are used to classify traffic patterns. The format of the rule is as such:

name = direction + tcp\_flags + max\_size + delimiter + direction + tcp\_flags + max\_size + delimiter …

Here is an example rule for a simple sloloris attack. We know that this type of attack will open a connection and keep it alive by sending small headers without finishing the request. We can implement the rule as such:

slowloris\_sig = \

'i,P,200:' + \

'o,A,5:' + \

'i,P,200:' + \

'o,A,5:' + \

'i,P,200:' + \

'o,A,5:'

The rule can then be added to the rules dictionary as follows:

rules = {'NORMAL TRAFFIC' :normalsig1, ‘SLOWLORIS’: slowloris\_sig}

## 

## 

## 

## 

## 

## 

## 

## 

## **Configuration options**

sloburn.py has multiple configurations options for additional functionality. These options can be set inside the script.

1. If this option is enabled, it will look for many simultaneous connections, indicating possible attack:

#rule signature 1: too many concurrent connections from single IP

MAX\_CONNECTIONS = 50

ENABLE\_MAX\_CONNECTIONS = True

1. If this option is enabled, it will look for sockets that have been open for a long time, indicating possible attack:

#rule signature 2: sockets open for too long

MAX\_SOCKET\_TIME = 300

ENABLE\_MAX\_SOCKET\_TIME = False

1. If this option is enabled, it will limit the size of the signature so that very active connections will not use too much memory:

#signature limit size

MAX\_SIGNATURE = 128

1. These are options for mitigation. These options currently impact all rule matches at this time:

#mitigation methods

ENABLE\_LOG = True

ENABLE\_ALERT = False

ENABLE\_KILL = True

ENABLE\_BLOCK = False