Habu

Information Security Inc.

Contents

- About Habu
- Dependencies
- Techniques implemented
- Demo Setup
- Installing Habu
- Using Habu
- References

About Habu

- Habu is a Network Hacking Toolkit
- It provides basic functions that help with some tasks for Ethical Hacking and Penetration Testing

Dependencies

- Click
- Delegator.py
- Python (3.x),
- Scapy-Python3
- Matplotlib (Optional, only needed if you want to make some graphs)

```
apt-get install python3-click
pip3 install delegator.py

apt-get install python3-scapy
apt-get install python3-matplotlib
```

Techniques implemented

- ARP Poisoning
- ARP Sniffing
- DHCP Discover
- DHCP Starvation
- LAND Attack
- SNMP Cracking
- SYN Flooding
- TCP Flags Analysis
- TCP ISN Analysis
- TCP Port Scan

Demo Setup

- Setup > Kali Linux 2017

```
root@LUCKY64:/opt3# cat /etc/*rel*
DISTRIB_ID=Kali
DISTRIB_RELEASE=kali-rolling
DISTRIB_CODENAME=kali-rolling
DISTRIB_DESCRIPTION="Kali GNU/Linux Rolling"
PRETTY_NAME="Kali GNU/Linux Rolling"
NAME="Kali GNU/Linux"
ID=kali
VERSION="2017.1"
VERSION_ID="2017.1"
ID_LIKE=debian
ANSI_COLOR="1;31"
HOME_URL="http://www.kali.org/"
SUPPORT_URL="http://forums.kali.org/"
BUG_REPORT_URL="http://bugs.kali.org/"
```

Installing Habu

- Install using pip3

```
root@LUCKY64:/opt3# pip3 search habu
habu (0.0.35)       - Network Hacking Toolkit
phabulous (0.1.2)   - Pythonic abstraction for python-habricator library.
root@LUCKY64:/opt3# pip3 install habu
Collecting habu
  Downloading habu-0.0.35.tar.gz (585kB)
    100% |                                | 593kB 2.7MB/s
Requirement already satisfied: click in /usr/lib/python3/dist-packages (from habu)
Requirement already satisfied: delegator.py in /usr/local/lib/python3.5/dist-packages (from habu)
Requirement already satisfied: requests in /usr/lib/python3/dist-packages (from habu)
Requirement already satisfied: scapy-python3 in /usr/lib/python3/dist-packages (from habu)
Requirement already satisfied: pexpect>=4.1.0 in /usr/local/lib/python3.5/dist-packages (from delegator.py->habu)
Requirement already satisfied: ptyprocess>=0.5 in /usr/local/lib/python3.5/dist-packages (from pexpect>=4.1.0->delegator.p
y->habu)
Building wheels for collected packages: habu
  Running setup.py bdist_wheel for habu ... done
  Stored in directory: /root/.cache/pip/wheels/1f/ce/ed/d41e51a89f046adea3df871b857f813af2258c896afce31fd6
Successfully built habu
Installing collected packages: habu
Successfully installed habu-0.0.35
```

Using Habu

- Habu help

```
root@LUCKY64:/opt3# habu.help
* = Need root privileges
habu.arpoison       ARP cache poisoner *
habu.arpsniff       ARP sniffer *
habu.contest        Check internet connection capabilities
habu.eicar          Print EICAR antimalware test data
habu.forkbomb       Show various forkbomb examples
habu.hasher         Calculate hashes using various algorithms
habu.help           Print this help message
habu.ip             Show your public IP
habu.update         Update data files
habu.xor            XOR cipher
```

Using Habu

- habu.eicar => Prints the EICAR test string

```
root@LUCKY64:/opt3# habu.eicar
X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*
```

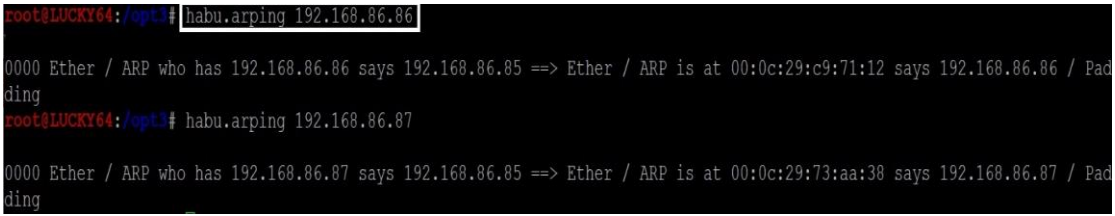
Using Habu

- habu.ping IP –v => “-v” Pinging with verbose information

```
root@LUCKY64:/opt3# habu.ping 8.8.8.8 -v

###[ IP ]###
  version   = 4
  ihl       = 5
  tos       = 0x0
  len       = 28
  id        = 20114
  flags     =
  frag      = 0
  ttl       = 54
  proto     = icmp
  chksum    = 0xf42
  src       = 8.8.8.8
  dst       = 192.168.86.85
  \options   \
###[ ICMP ]###
     type      = echo-reply
     code      = 0
     chksum    = 0x0
     id        = 0x0
     seq       = 0x0
###[ Padding ]###
        load      = '\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00'
```

Using Habu

- habu.arping IP => arping



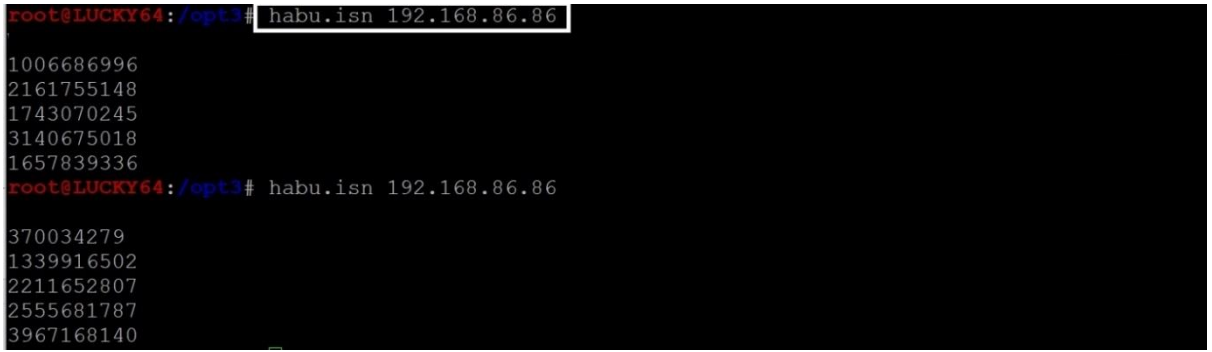
Using Habu

- habu.forkbomb => Show various forkbomb examples

```
root@LUCKY64:/opt3# habu.forkbomb
Usage: habu.forkbomb [OPTIONS] BOMB

Error: Missing argument "bomb".  Choose from bash, batch, c, haskell, perl, php, python, ruby.
root@LUCKY64:/opt3# habu.forkbomb python
import os

while True:
    os.fork()

root@LUCKY64:/opt3# habu.forkbomb php
while(pcntl_fork()|1);

root@LUCKY64:/opt3# habu.forkbomb c
#include <unistd.h>

int main()
{
    while(1)
    {
        fork();
    }
    return 0;
}
```

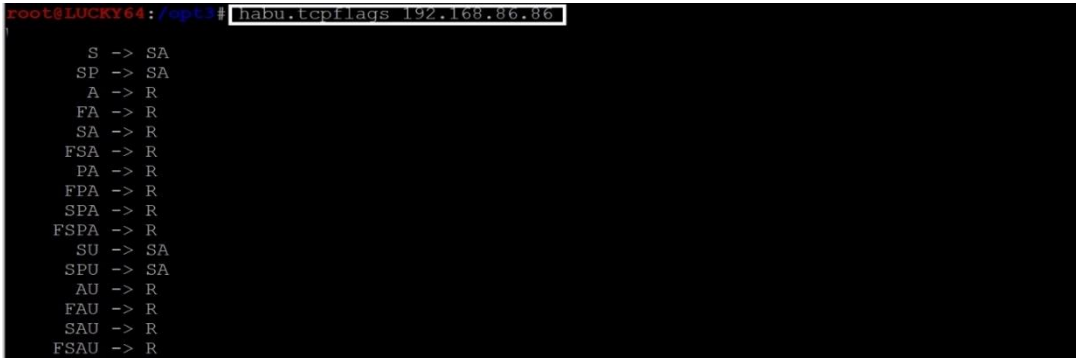
Using Habu

- habu.isn => This command creates TCP connections and prints the TCP initial sequence numbers for each connections



Using Habu

- habu.tcpflags => This command send TCP packets with different flags and tells what responses receives
- It can be used to analyze how the different TCP/IP stack implementations and configurations responds to packet with various flag combinations



References

- Kitploit
http://www.kitploit.com/2017/10/habu-network-hacking-toolkit.html

- Kali Linux
https://www.kali.org/downloads/

- LAND attack
https://security.radware.com/ddos-knowledge-center/ddospedia/land-attack/

- ARP poisoning (spoofing)
https://en.wikipedia.org/wiki/ARP_spoofing

- SYN flood
https://en.wikipedia.org/wiki/SYN_flood

- Forkbomb
https://en.wiktionary.org/wiki/fork_bomb