Lab 5

NetFilterQueue & Password Cracking

NOTE: Make sure that your Kali and Ubuntu VMs are running fine and network is set as “NatNetwork”.

1. NetFilterQueue

   We may want to do more with the MITM attack. A tool for doing is *Scapy* and *NetFilterQueue*, which will enable us to analyze and manipulate the packets on live traffic.

   What we want to do now is to capture incoming and outgoing packets from my local (Kali) machine and put them in the queue, inspect them as Scapy packets and release them to the destination.

   First, we need to install the netfilterqueue package. There are a few ways to install netfilterqueue, but I found that the following method works well at the moment:

```
$ sudo apt-get update
$ python –version
$ sudo apt-get install -y python3
$ sudo apt-get install -y python3-pip
$ sudo apt-get install -y git libnfnetlink-dev libnetfilter-queue-dev
$ pip3 install -U git+https://github.com/kti/python-netfilterqueue
```

   Then, we configure the iptables so that we assign queue number for incoming and outgoing packets. Run the following commands consecutively on the terminal.

```
$iptables -L
$iptables -I INPUT -j NFQUEUE --queue-num 1
$iptables -I OUTPUT -j NFQUEUE --queue-num 1
$iptables -L
```

Now, create a python source file nfq.py (or any file name you like):

```
#! /usr/bin/env python

from scapy.all import *
from netfilterqueue import NetfilterQueue
import os

##What is the effect of this command?
os.system("sysctl net.ipv4.ip_forward=1")

def modify(packet):
    ip_pkt = IP(packet.get_payload())
    print(ip_pkt.show())

    ##Release the packet
    packet.accept()

    ##Drops the packet
    #packet.drop()

nfqueue = NetfilterQueue()

#1 is the queue number
nfqueue.bind(1, modify)
try:
    nfqueue.run()
except KeyboardInterrupt:
    nfqueue.unbind()
```

Run the above program and open a web browser and navigate. What can you see? Modify the above program to display Scapy packets in various ways. (Hint: Refer to Lab 2 note.)

Flush the iptables once you're done:

```
$iptables --flush
```

Basically, what we want do, as a MITM attacker, is to capture packets from the victim's machine and put them in the queue, inspect them and release them to the destination.

First, configure the iptables so that we assign queue number for packets *being forwarded to and from the victim machine only* (not all the packets coming in and going out from this local machine). To do this, run the following command on the terminal.

```
$iptables -I FORWARD -j NFQUEUE --queue-num 1
```

The next step is to run the caplet arpspf.cap we created last week to perform MITM on Ubuntu (the target machine):

On the first terminal window, issue:

```
      $bettercap -iface eth0 -caplet arpspf.cap
```

Then, run the `nfq.py` again and see what happens. In ubuntu, open a web browser and navigate to some websites.

Finally refresh the iptables:

```
      $iptables --flush
```

2. Make your own word list for password cracking using crunch

   When we use password cracking tools like hydra and john-the-ripper, we need to provide them with a password list. There exist ready-made lists but we can create our own using crunch.

   The basic syntax for crunch is as follow:

   For displaying password permutation on your terminal screen.

```
crunch[min len] [max len] [character set][options]
```

   To save crunch output into a specified file

```
crunch[min len] [max len] [character set][options] –o file
```

   On Kali Terminal, issue `crunch 3 3 abc` and run `crunch 3 3 abcd` (Did you get the idea how crunch works?) It will generate all possible words with repetitions (such as `bbb`) using characters `a`, `b` and `c`.

```
  $crunch 6 8 0123456789 –o numword.lst
```

   This will create all the possible words of length 6 to 8, all of which consist of numbers between `0` and `9`. The file size will be big – Nearly 1 GB.

   If you want to use special characters, use backward slash. For example, `\&`, `\*`, `\%` and etc.

One of the useful options is `-t`. You can specify a pattern you're searching.

Suppose that someone uses a password of 8 characters and his birthday is `0829`. An attacker might want to try all the possible combinations ending with 0829. In this case, we can run

```
  $crunch 8 8 -t @@%^0829 -o birthday.txt
```

Here, @ is a wildcard for lowercase alphabetical characters. `,` is a wildcard for uppercase alphabetical characters % is a wildcard for numeric and ^ is for special characters.

You can use pre-defined character set used by rainbowcrack. Download http://project-rainbowcrack.com/rainbowcrack-1.8-linux64.zip and extract the file "charset.txt"

So you can run something like this using –f option:

```
        $crunch 4 4 -f charset.txt mixalpha -o example.txt
```

3. Cracking password using hydra online

This exercise needs to access Metasploitable VM. Run it under the NatNetwork.

First, create a user named "alice" at Metasploitable. Login to Metasploitable and type and run:

```
      $sudo useradd -m alice -G users -s /bin/bash
```

Then, set a password for victim:

```
      $sudo passwd alice
```

(Let us set up an easy password that consists of only 5 numbers. Even it may take quite a while to take find a 5 digit password. So choose a little bit short (and obvious) password for testing.)

Then, go to Kali VM and create words list of 5 numbers using crunch. Can you do it using crunch command? Name your file `myword.txt`

Now run hydra using the words list you have just created:

```
   $hydra -t 64 -l alice -P myword.txt -vV <Meta IP> ftp
```

Have you found the password? (Note that 64 is a maximum number of concurrent connections to the target, and ftp is a protocol that hydra makes use of to perform brute-force.)

4. Cracking password using john-the-ripper

First, create a user `steve` for testing **on Kali**:

```
$(sudo) useradd –m steve –G sudo –s /bin/bash
```

Next, set password for victim on Kali :

```
$(sudo) passwd steve
```

Combine entries of `/etc/passwd` and `/etc/shadow` by unshadowing:

```
$unshadow /etc/passwd /etc/shadow > ./steve_pwd
```

Run John the Ripper using the password list:

```
john –format=crypt --wordlist=/usr/share/john/password.lst steve_pwd
```

Important!
Once the john-the-ripper has cracked the password, it will not do it again.
It will save the cracked passwords. To view it, run

```
$john ––show steve_pwd
```

5. Extracting passwords using a Python program

You have learned how to create a password list. Suppose that you want to filter out passwords having a specific pattern from the existing password list (such as password dictionary).

One useful technique is to use a regular expression filter. Write a python program to find possible passwords containing 0825 or 0827 using re library (https://docs.python.org/2/library/re.html). You may want to refer to https://www.w3schools.com/python/python_regex.asp for quick reference.

To create a password list, use crunch as follows.

```
crunch 6 6 -t @@082% -o birthday2.txt
```

We have written a code that you can start with:

```
#to use regular expression package
import re

#to open password list file
dictionary = open("birthday2.txt","r")
#to write extracted passwords into a file
extracted = open("passwd_ext.txt","w")

#Insert your regular expression inside the quotation marks
rex = re.compile(" ")
passwords = filter(rex.search, dictionary)

for line in passwords:
        extracted.write(line)

dictionary.close()
extracted.close()
```