Risk Assessment and Vulnerability Remediation Project

I conducted a complete risk assessment on a simulated small business network environment. I built the entire infrastructure from scratch, deploying 9 endpoints (primarily Linux systems with 2 Windows machines), a Linux-based domain controller, an Apache web server hosting a deliberately vulnerable application, and a pfSense firewall. After configuring the domain controller to serve as the network's name server and setting up pfSense to handle DHCP and DNS services, I implemented a structured risk assessment methodology following the NIST SP 800-30 Risk Management Framework and incorporating concepts from NIST SP 800-40 for enterprise patch management.

My risk assessment framework was highly detailed, utilizing a custom-designed risk matrix that rated vulnerabilities on a six-level scale from very low to extreme. I created an extensive risk register in Excel format with multiple tracking categories including Risk ID, Date Raised, Description, Asset Affected, Likelihood Rating, Impact Rating, Risk Score, Risk Action, Risk Treatment, Risk Owner, Status, Residual Risk, and Notes. The risk register utilized formulas to calculate risk scores by multiplying impact and likelihood ratings, which directly informed my prioritization decisions.

1-2 Very Low
3-4 Low
4-9 - Medium
10-12 High
13-16 Very High
16< Extreme

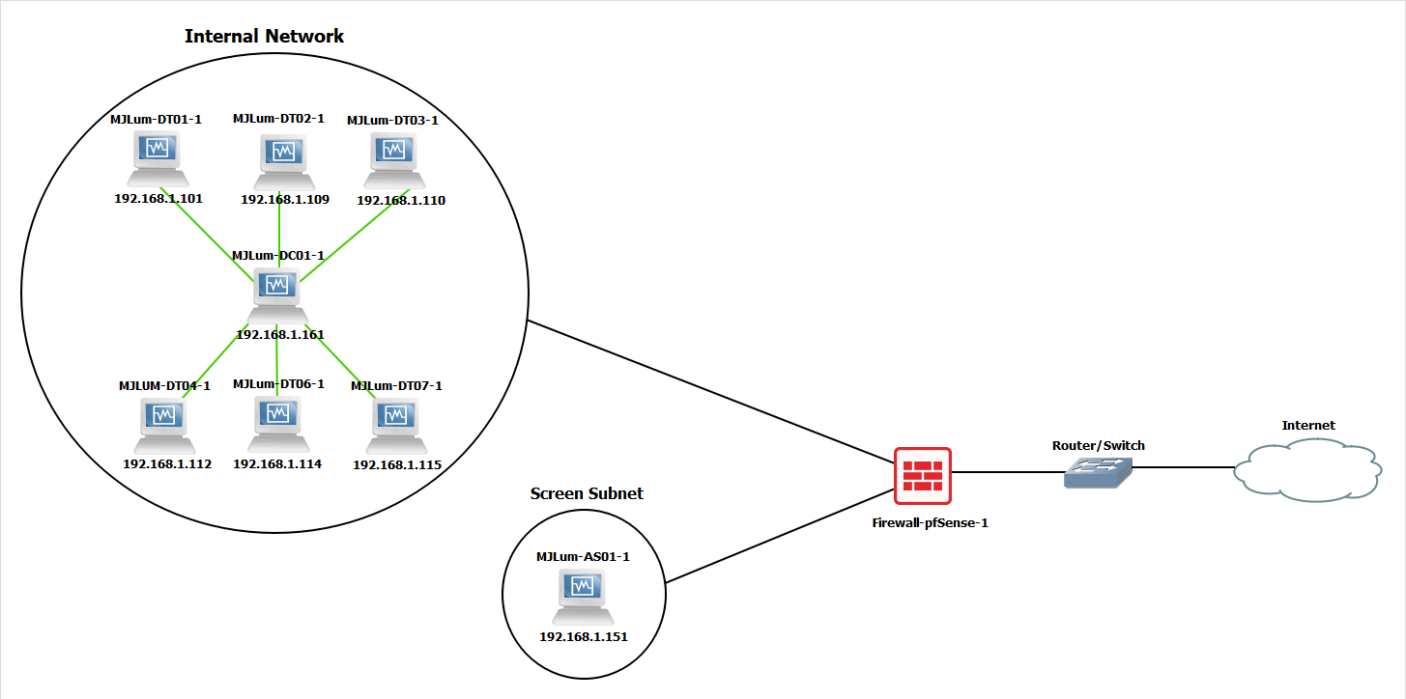
Likelihood \ Impact	1 Insignificant	2 Minor	3 Moderate	4 Major	5 Severe
5 Almost Certain	5 Medium	10 High	15 Very High	20 Extreme	25 Extreme
4 Likely	4 Medium	8 Medium	12 High	16 Very High	20 Extreme
3 Possible	3 Low	6 Medium	9 Medium	12 High	15 Very High
2 Unlikely	2 Very Low	4 Low	6 Medium	8 Medium	10 High
1 Rare	1 Very Low	2 Very Low	3 Low	4 Medium	5 Medium

				Assessment								
Risk ID	Date Raised	Description	Asset Affected	Likelihood Rating	Impact Rating	Risk Score	Risk Action	Risk Treatment	Risk Owner	Status	Notes
R-015	09/28/2024	Arbitrary code execution via outdated Ffmpeg	MJLum-AS01	4	5	20	Mitigate	Apply the latest security patches for Ffmpeg	Mjoseph	Completed	
R-019	09/28/2024	Denial of service vulnerability due to outdated cJSON	MJLum-AS01	3	4	12	Mitigate	Update the cJson package to the latest version	Mjoseph	Completed	
R-020	09/28/2024	Missing Linux Kernel mitigations for 'SSB'	MJLum-AS01	2	4	8	Mitigate	Update to a more recent Linux Kernel, implement compensating control	Mjoseph	Completed	Snort IDS/IPS on firewall, Limit access, Set up Logging, Moved AS01
R-017	09/28/2024	Weak MAC Algorithms Supported in SSH	MJLum-AS01	3	2	6	Mitigate	Disable weak MAC algorithms for SSH	Mjoseph	Completed	
R-016	09/28/2024	System uptime can be disclosed via TCP timestamps	MJLum-AS01	2	1	2	Avoid	Disable TCP timestamps	Mjoseph	Completed	
R-021	09/28/2024	Missing Linux Kernel mitigations for 'SSB'	MJLum-DC01	4	4	16	Mitigate	Apply Package and Firmware updates	Mjoseph	Completed	
R-023	09/28/2024	Cleartext FTP login vulnerable to interception	MJLum-DC01	4	4	16	Avoid	Disable FTP	Mjoseph	Completed	
R-022	09/28/2024	Unauthorized access vulnerability via anonymous FTP login	MJLum-DC01	4	3	12	Avoid	Disable FTP	Mjoseph	Completed	
R-026	09/28/2024	Enumeration of DCE/RPC services	MJLum-DC01	4	3	12	Mitigate	Implement firewall rules to restrict access to port 135.	Mjoseph	Completed	
R-028	09/28/2024	Denial of service vulnerabilitity due to outdated cJSON	MJLum-DC01	3	4	12	Mitigate	Update cJSON to the latest version (False Positive)	Mjoseph	Completed	
R-025	09/28/2024	Weak SSH encryption algorithms	MJLum-DC01	3	3	9	Mitigate	Disable weak MAC algorithms in the SSH configuration.	Mjoseph	Completed	
R-027	09/28/2024	Arbitrary code execution via outdated FFmpeg	MJLum-DC01	1	4	4	Avoid	Remove the package entirely to reduce attack surface.	Mjoseph	Completed	
R-024	09/28/2024	System uptime can be disclosed via TCP timestamps	MJLum-DC01	2	1	2	Avoid	Disable TCP timestamps in /etc/sysctl.conf	Mjoseph	Completed	
R-029	09/28/2024	Missing Linux Kernel mitigations for 'SSB'	MJLum-DT01	3	3	9	Mitigate	Apply Package and Firmware updates	Mjoseph	Completed	
R-032	09/28/2024	Weak MAC Algorithms Supported in SSH	MJLum-DT01	2	2	4	Mitigate	Disable weak MAC algorithms.	Mjoseph	Completed	
R-030	09/28/2024	System uptime can be disclosed via TCP timestamps	MJLum-DT01	2	1	2	Avoid	Disable TCP timestamps in /etc/sysctl.conf	Mjoseph	Completed	
R-031	09/28/2024	ICMP Timestamp Reply Information Disclosure	MJLum-DT01	2	1	2	Avoid	Disable timestamp replies	Mjoseph	Completed	
R-033	09/28/2024	Missing Linux Kernel mitigations for 'SSB'	MJLum-DT03	3	3	9	Mitigate	Update to a more recent Linux Kernel, implement compensating control	Mjoseph	Completed	Snort IDS/IPS on firewall, Limit access, Set up Logging
R-036	09/28/2024	Weak MAC Algorithms Supported in SSH	MJLum-DT03	2	2	4	Mitigate	Disable weak MAC algorithms.	Mjoseph	Completed	
R-034	09/28/2024	System uptime can be disclosed via TCP timestamps	MJLum-DT03	2	1	2	Avoid	Disable TCP timestamps in /etc/sysctl.conf	Mjoseph	Completed	
R-035	09/28/2024	ICMP Timestamp Reply Information Disclosure	MJLum-DT03	2	1	2	Avoid	Disable timestamp replies	Mjoseph	Completed	
R-053	09/28/2024	Outdated Firefox ESR package	MJLum-DT07	4	4	16	Mitigate	Update Firefox ESR to the latest version	Mjoseph	Completed	
R-054	09/28/2024	Outdated Ghostscript package	MJLum-DT07	4	4	16	Mitigate	Update Ghostscript to the latest version	Mjoseph	Completed	
R-055	09/28/2024	Outdated WebKit2GTK package	MJLum-DT07	4	4	16	Mitigate	Update WebKit2GTK package.	Mjoseph	Completed	
R-047	09/28/2024	Missing Linux Kernel mitigations for 'SSB'	MJLum-DT07	3	3	9	Mitigate	Update to a more recent Linux Kernel, implement compensating control	Mjoseph	Completed	Snort IDS/IPS on firewall, Limit access, Set up Logging
R-048	09/28/2024	Unauthorized access via anonymous FTP login	MJLum-DT07	3	3	9	Avoid	Disable FTP	Mjoseph	Completed	
R-049	09/28/2024	Cleartext FTP login vulnerable to interception	MJLum-DT07	3	3	9	Avoid	Disable FTP	Mjoseph	Completed	
R-052	09/28/2024	Weak MAC Algorithms Supported in SSH	MJLum-DT07	2	2	4	Mitigate	Disable weak MAC algorithms.	Mjoseph	Completed	

Using Kali Linux, I set up and configured OpenVAS through the terminal and accessed its web GUI via Firefox. I developed custom scanning profiles tailored to different system types, including specific profiles for the Apache server, domain controller, Linux machines, and Windows systems. After running comprehensive vulnerability scans, I meticulously analyzed each report, documented findings in my risk register, and verified systems for false positives, noting these in my documentation. For the vulnerable web application, I conducted additional specialized assessment using OWASP ZAP to identify web-specific security issues.

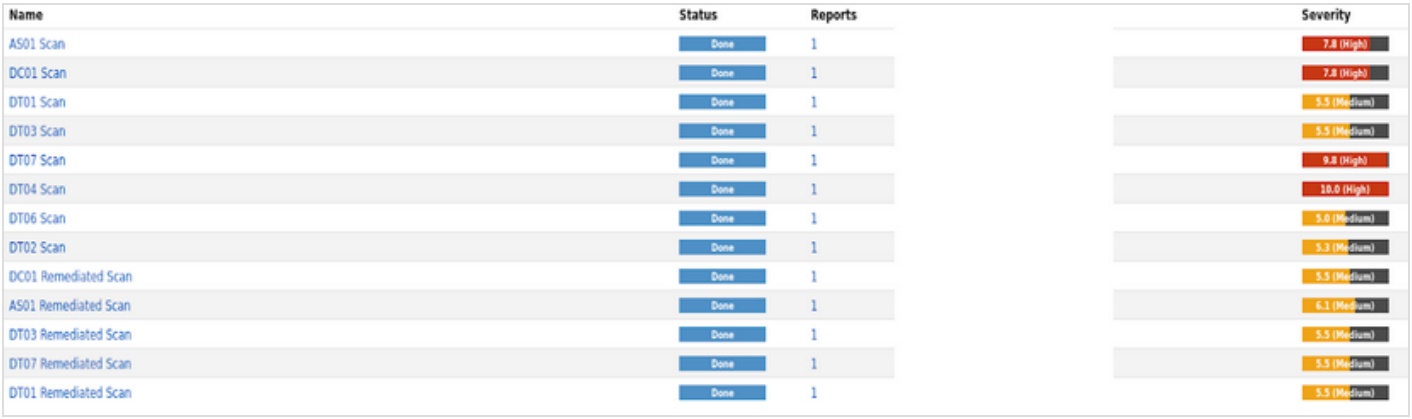
What separated my approach from a standard assessment was the implementation of sophisticated risk mitigation strategies. For the vulnerable Apache web server, I designed and implemented a screened subnet (DMZ) configuration that effectively isolated it from the internal network. I created specific firewall rules that allowed only necessary external traffic to reach the web server while preventing any lateral movement into the internal network. This network

segmentation demonstrated my ability to apply defense-in-depth principles to protect critical assets.



I also deployed and configured Snort IDS on the pfSense firewall as a compensating control for a specific Linux vulnerability that couldn't be directly remediated. This implementation required me to create custom rules to detect and block the malicious traffic patterns that could exploit the vulnerability, providing real-time visibility into network activity and adding an additional security layer. For systems approaching end-of-life, I documented avoidance strategies including OS update requirements and interim protection measures.

After implementing all security controls, I conducted follow-up vulnerability scans to validate the effectiveness of my remediations. I meticulously updated the risk register with post-remediation findings, adjusted risk scores based on the implemented controls, and documented residual risk levels. Through this comprehensive approach, I successfully reduced the network's overall vulnerability exposure while establishing a sustainable, documented risk management process that balanced technical security requirements with simulated business operations.



Name	Status	Reports	Severity
AS01 Scan	Done	1	7.8 (High)
DC01 Scan	Done	1	7.8 (High)
DT01 Scan	Done	1	5.5 (Medium)
DT03 Scan	Done	1	5.5 (Medium)
DT07 Scan	Done	1	9.8 (High)
DT04 Scan	Done	1	10.0 (High)
DT06 Scan	Done	1	5.0 (Medium)
DT02 Scan	Done	1	5.3 (Medium)
DC01 Remediated Scan	Done	1	5.5 (Medium)
AS01 Remediated Scan	Done	1	6.1 (Medium)
DT03 Remediated Scan	Done	1	5.5 (Medium)
DT07 Remediated Scan	Done	1	5.5 (Medium)
DT01 Remediated Scan	Done	1	5.5 (Medium)

The remaining medium severity findings are primarily related to the SSB store vulnerability on Linux machines, which was addressed by implementing a compensating control where pfSense firewall rules and permission adjustments were made to mitigate this vulnerability.

It is also evident that follow-up scans were performed on only five machines as opposed to the eight initially scanned machines. Due to the lack of support for Windows 8, Windows XP, and, more recently, Windows 10, these machines were moved to the disposal and replacement stage in the asset management lifecycle as they posed a significant risk due to the lack of support from Microsoft.

Artifacts

Web Server Vulnerability Scan

Web Application Vulnerability Scan

Linux Ubuntu Machine Vulnerability Scan

Windows 10 Machine Vulnerability Scan

Risk Matrix & Risk Register