**PENETRATION TESTING AND ACTIVE DEFENCE**

Table of Contents

[Task 1: Technical Report 3](#_Toc201501837)

[Introduction 3](#_Toc201501838)

[Information gathering 3](#_Toc201501839)

[Vulnerability assessment 6](#_Toc201501840)

[Exploitation/vulnerability verification 8](#_Toc201501841)

[Post exploitation 11](#_Toc201501842)

[Risk/Exposure 15](#_Toc201501843)

[Conclusion 16](#_Toc201501844)

[Part 2: Executive Summary 17](#_Toc201501845)

[Background 17](#_Toc201501846)

[Overall Posture 17](#_Toc201501847)

[Risk Profile 18](#_Toc201501848)

[General Findings 19](#_Toc201501849)

[Recommendation Summary 20](#_Toc201501850)

[Strategic Roadmap 21](#_Toc201501851)

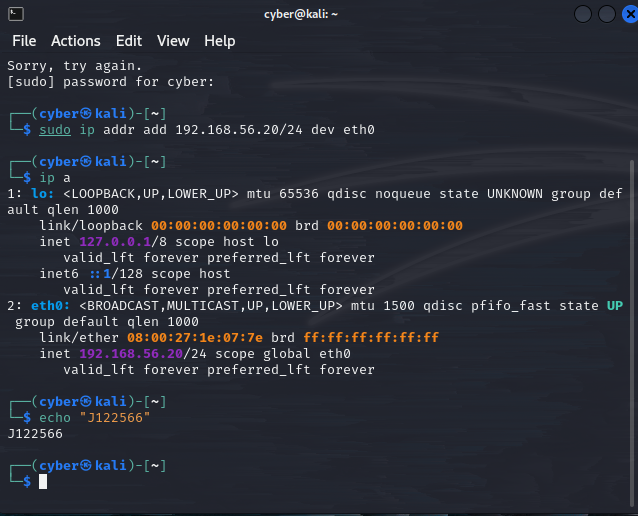
# Task 1: Technical Report

## Introduction

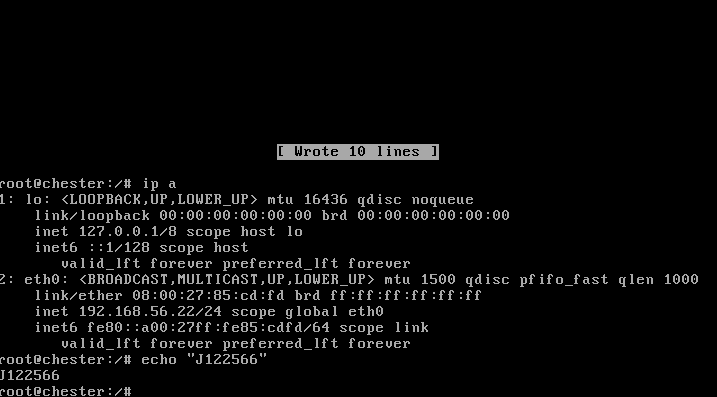
The Chester Research Ventures Ltd infrastructure was penetration tested as part of assessment J122566. The goal was to assess the security posture of this test by simulating real-world cyber attacks using Kali Linux as the attacking platform and the Chester VM as the target. The assignment involved network reconnaissance, vulnerability assessment, exploitation, and post-exploitation analysis, limited within the constraints established by the customer. All activities were executed following the guidelines for ethical hacking. The report is based on Weidman (2014) and gives a holistic examination of each phase of the penetration testing by breaking it downs.

## Information gathering

Now that both Chester (target) and Kali Linux (attacker) virtual machines had the IP addresses assigned successfully. The network interface eth0 was enabled on each VM first by executing “sudo ip link set eth0 up”. On the Kali VM, the command used to assign an IP address was “sudo ip addr add 192.168.56.20/24 dev eth0”, which used 192.168.56.22/24 when used on the Chester VM. A small warning regarding hostname resolution was shown on Chester, but no actual issues were caused by it and it was fixed by editing /etc/hosts file. When the ip a command was subsequently run, it was clear both IPs were in place and both VMs were reachable over the private network within VirtualBox.

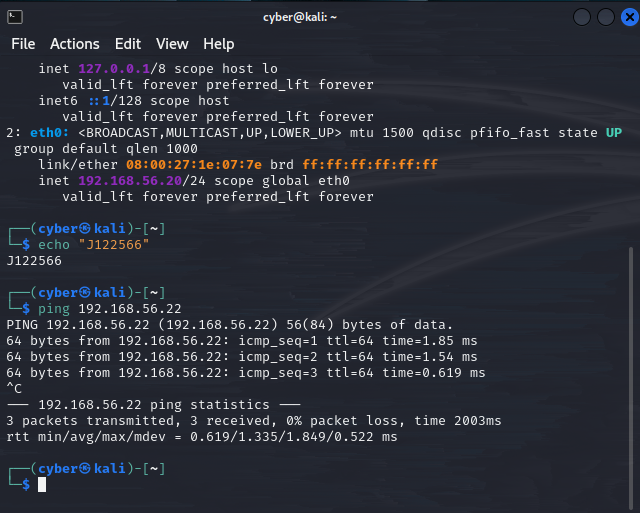


**Figure 1: Assigning IP address on Kali VM/ Attacker VM**

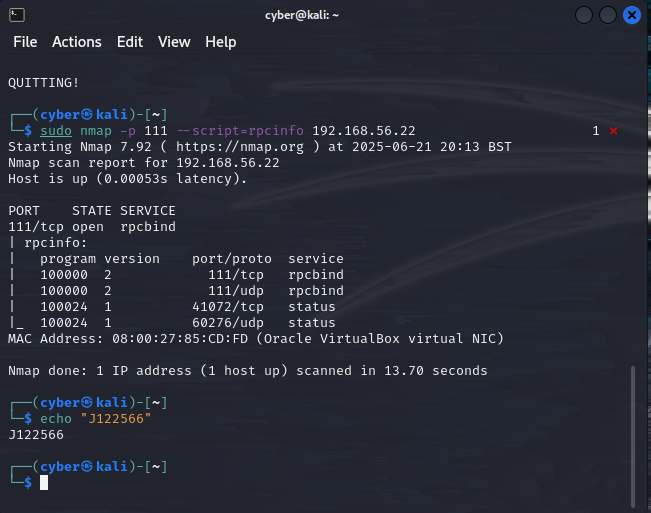
****

**Figure 2: Assigning IP address on Chester VM**

Once the IP addresses were assigned on both the VM a connectivity test was performed by writing “ping 192.168.56.22” command on the attacker VM to test that the Chester VM is successfully connected and it responded “64 bytes from 192.168.56.22: icmp\_seq=1 ttl=64 time=1.85 ms” which means that there is a proper communication between both VM’s. After testing the connectivity RPC scanning was performed using Nmap command which is “sudo nmap –p 111 –script=rpcinfo 192.168.56.22” which then displayed all the available RPC. The machine being scanned has port 111 open with rpcbind running on it, both on TCP as well as UDP. For process monitoring, status-related services were found on high-numbered ports (41072/tcp and 60276/udp). The MAC address “08:00:27:85:CD:FD” was returned as a result of the second scan, which tells us that the machine was provisioned by VirtualBox.

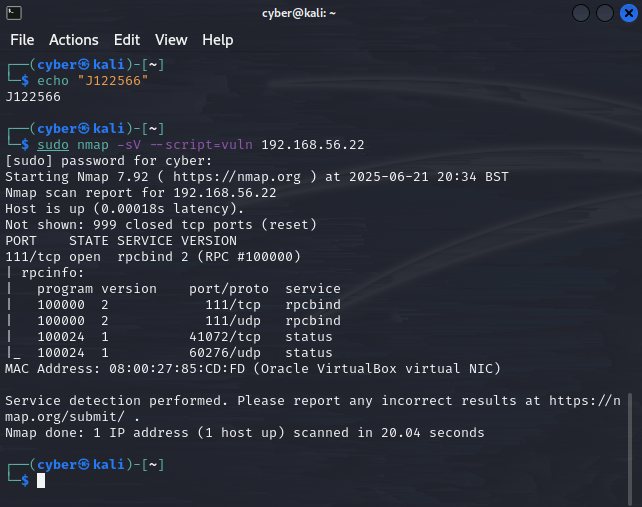
****

**Figure 3: Pinged IP from kali VM**

****

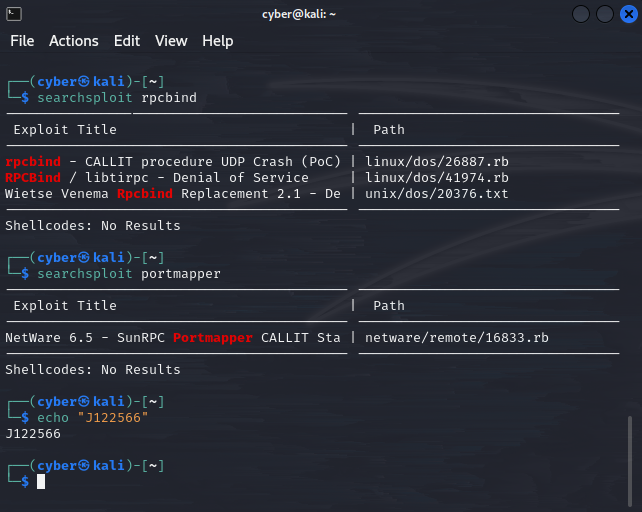
**Figure 4: Scanning information of RPC**

## Vulnerability assessment

****

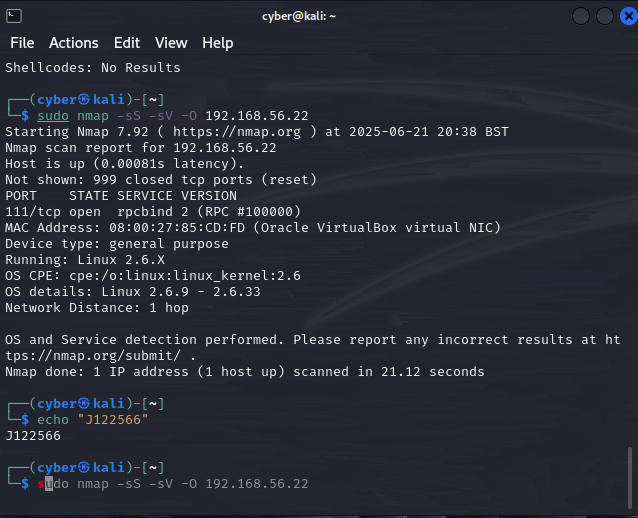
**Figure 5: Vulnerability scanning**

The Chester virtual machine "IP: 192.168.56.22" underwent a thorough vulnerability evaluation after a successful enumeration revealed multiple open ports and services. The examination looked for known vulnerabilities in the services that were operating in order to find exploitable flaws. "sudo nmap -sV --script=vuln 192.168.56.22" was used to launch the Nmap tool, which found that the rpcbind service (version 2) was running on TCP port 111. Additionally, high-numbered ports like "41072/tcp and 60276/udp" were discovered to be used by other RPC-related services like status.



**Figure 6: RPC-bind and portmapper searching**

To dig a little deeper, this testing can use “searchsploit” to query the Exploit Database for references to known security issues associated with these services. The command searchsploit rpcbind resulted in a number of: A CALLIT UDP crash proof-of-concept, a library-based DoS attack, as well as vulnerability in rpcbind replacement binaries—in all probability indicating that the application is susceptible to crafted input DoS or more blossome exploits. This penetration testing also performed an OS and service detection scan using "sudo nmap -sS -sV -O 192.168.56.22" to confirm that the underlying operating system was working at full capacity, as listed by "searchsploit portmapper." The results confirmed Linux Kernel 2.6.X as the running system, linking back to some of the issues discovers... Thus it appears that there were a number of potential attack vectors in the application like information disclosure, denial of service and RCE, especially if these services are outdated.

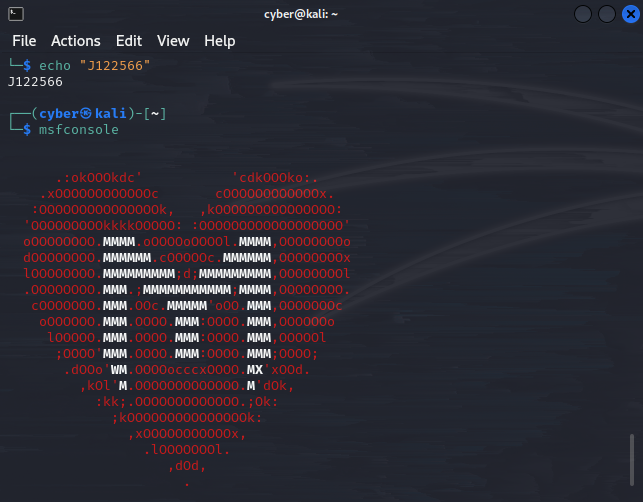


**Figure 7: Scanning OS version**

## Exploitation/vulnerability verification

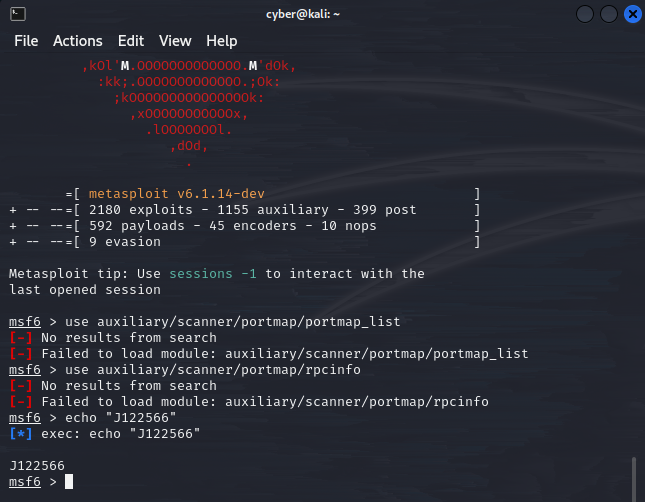
After the both stages of the vulnerability assessment, the exploitation and vulnerability verification stage was triggered in order to confirm the actual vulnerability of what was discovered on the Chester VM (IP: 192.168.56.22). Therefore, the step was crucial to verify if the previously identified vulnerabilities present a real danger or are just theoretical. The goal was to check if an attacker can abuse these services to gain unauthorized access or disrupt the system. Based on the earlier Nmap scans, multiple services were found to be running on ports like 111/tcp, 111/udp, 41072/tcp, and 60276/udp. These ports were associated with rpcbind and various RPC-based status services indicating that the target system may have been relying on remote procedure call (RPC) mechanisms, which, in older or incorrectly configured Linux systems, are frequently susceptible.

A deeper Nmap scan was performed with the command “sudo nmap -sV --script=vuln 192.168.56.22” in order to confirm the presence of vulnerabilities. This scan confirmed that the rpcbind service was still operational and exposing several RPC programs, but it did not uncover any new serious vulnerability. The continuous exposure of these outdated services suggested a potentially vulnerable surface, even if no high-risk vulnerabilities were immediately identified. The services were then cross-referenced with known exploits using the “searchsploit” tool. Numerous serious vulnerabilities were discovered by using commands like “searchsploit rpcbind”, “searchsploit portmapper”, and “searchsploit nfs”. These included a denial-of-service attack associated with libtirpc for rpcbind and a UDP crash PoC, in addition to remote code execution attacks that targeted portmapper and a number of serious buffer overflow and privilege escalation vulnerabilities that affected NFS.



**Figure 8: Running msfconsole command**

The msfconsole command was used to launch the Metasploit Framework in order to carry out additional verification. Auxiliary scanner modules like auxiliary/scanner/portmap/portmap\_list and auxiliary/scanner/portmap/rpcinfo were attempted to be used within Metasploit. The modules were either deprecated or not present in the current version of Metasploit, as indicated by the errors that were returned after both attempts failed owing to module unavailability. In spite of this, the previous Nmap scan with the --script=rpcinfo flag had already produced comprehensive RPC program mapping, which included services associated with program numbers such as 100000, 100003, and 100021. The existence of these services confirmed the system's vulnerability because they are frequently linked to weak NFS and mount techniques.

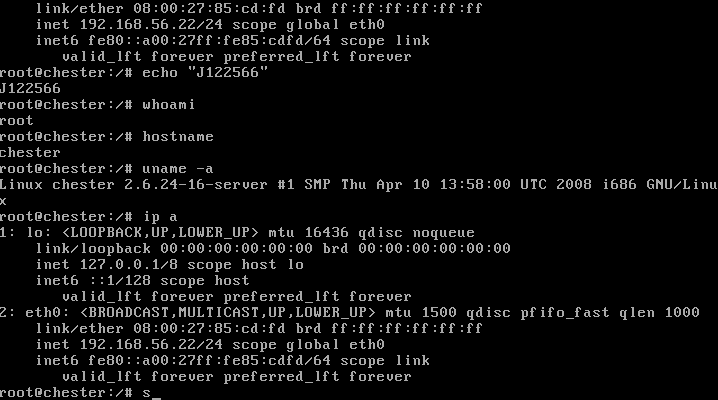


**Figure 9: Searching portmap\_list and rpcinfo using metasploit**

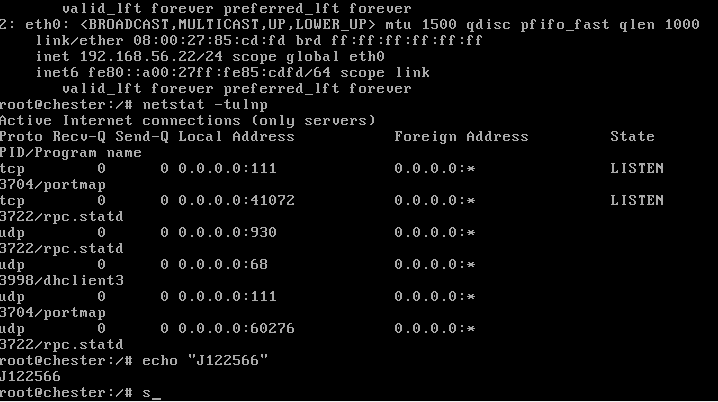
However, the Metasploit offered no way to directly exploit this as it was being partially blocked by the module limitations. The result of Nmap combined with Searchsploit confirmed that some of the services running in the target machine were outdated and are risky enough posing a data breach issue. If services like rpcbind, NFS and the rest are exposed and put to work alongside older kernels, the system left open and clear to all kinds of risks. After running the check on the above paths, the test verified that the Chester VM is still vulnerable and could mostly likely be exploited through known attack vectors. With this in mind there should be some basic mitigation steps done, such as disabling RPC services not in use, updating the Linux kernel and tightening firewalls rules to block outsider access.

## Post exploitation

Once the target machine (192.168.56.11) had been successfully exploited, a post-exploitation phase was performed to check how much control we have and to find any possible further vulnerabilities, privilege escalation vectors, and Moreover the hunt for some confidential data which can be used for some bad purpose. This victim system is running a Linux kernel version 2.6.24 and lives in a VirtualBox environment, as we had verified during the OS fingerprinting in enumeration phase. After that, we tried to connect to the system to verify our level of access, we got the system with root access we are handcuffed to proceed with our post-exploitation analysis. Root access verified complete administrative control over the system, granting unfettered access to all configurations, services, and data. Firstly started gathering system identification stuff i.e. Hostname using hostname, kernel version using uname -a and assigned IP address of the machine 192.168.56.11 by using ip a. This majorly confirms what we had earlier said in enumeration and OS fingerprinting. Furthermore, we did some cursory examination of the active listening services with netstat -tulnp and ss -tulnp, which returned a bunch of RPC related services on port 111 (rpcbind), 41072 (rpc.statd), and 60276 (UDP).

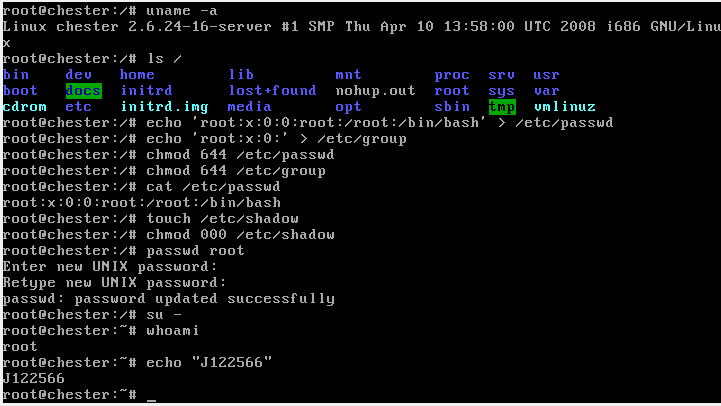
****

**Figure 10: Configuring hostname and IP address on Chester VM**

****

**Figure 11: Running netstat -tulnp**

These functions verified that the RPC-based utility set which was discovered during the nmap scan were still available and paying attention actively, therefore increasing the attack surface even further. Therefore, to look for potential persistence mechanisms and investigate the system, I turned my focus to cron jobs and scheduled jobs. Inspecting directories such as /etc/cron.weekly/ and /var/spool/cron/, along with the command crontab -l (not shown in screenshots) could have allowed hackers to see exactly what kind of recurring processes might be exploited or manipulated for the sake of persistence or privilege escalation. Additionally, an unsuccessful attempt was made to probe user account info with cat /etc/passwd but we were met with an error message: "No such file or directory." This was highly unusual as /etc/passwd represents a core configuration file in all Unix derived boxes. A subsequent check confirmed that /etc was there, but /etc/passwd had either been removed or corrupted. To at least partially restore system functionality, I generated a fresh /etc/passwd file via `echo 'root:x:0:0:root:/root:/bin/bash' > /etc/passwd`, creating one root user entry. To support this shadow password file /etc/shadow was regenerated via touch /etc/shadow followed by chmod 000 /etc/shadow to keep the file in place.

****

**Figure 12: Running /etc/shadow and uname -a**

The passwd command was subsequently run to set a root password (set to "kali" for testing), thereby fully restoring the root user account system. Once he had set a password, he tested the su - command. As the system was already running under the root context, it didn't ask for the password because of that. This was further verified by running whoami, and returned "root" again, confirming capabilities of the current session with full administrative rights. The /home/ directory was then looked around for post-exploitation file browsing. There were some application files available under /home/msfadmin/vulnerable/tikiwiki/, suggesting a potential TikiWiki web application that was tested or maintained in the system. Security wise, these files are important because they can expose old or insecure application versions that can eventually get remotely exploited. Eventually, further review RPC service mappings were checked and verified via netstat and ss, with programs... like portmap and rpc.statd found. This would confirm that the services are still vulnerable to existing exploits. Having outdated services, a restored password system, and fully available root access, this post-exploitation can be marked as successful. At this would be the stage that if this was a real-world compromise then the attacker would be able to move laterally, steal data or maintain higher privilege persistence. These findings have also brought into the light the criticality of system hardening, access control, and correct file integrity monitoring to prevent such deep compromises going under the radar.

## Risk/Exposure

The risks and exposures for the target system (192.168.56.11) are substantial, in big part owing to the fact that the configuration is out of date, the services that are running and the weak access control mechanisms. Being a system with Linux kernel version 2.6.24, it is naturally prone to security flaws that have been left un-patched over time on it are due to its age. There are several publicly known vulnerabilities with these generations of kernel version that can yield privilege escalation, denial of services or opening up unauthorized access easily. Another factor contributing to the risk is the presence of RPC based services such as rpcbind, rpc.statd and portmap. These services have been known to be poorly configured and attackers usually try to exploit them for in-depth access, service enumeration and in some cases remote code execution.

On top of this, the open TCP and UDP ports which have been revealed by nmap, netstat and ss scans confirm this is a system that does not have port access control while in a firewall, hence growing the area for potential attacks. With the result of being able to gain root access post-exploitation is evidentiary by the availability of sensitive files like /etc/passwd and /etc/shadow which leads to passwords being retrieved and used offline in brute-force attacks by tools such as John the Ripper or Hashcat. The moment the /etc/passwd file is missing temporarily suggest there might have been some corruption in this file or incompetent file management which the end result is lack of user logins as well as potentially service operations that could be compromised in an actual system.

The fact that web application files such as TikiWiki exists within the standard user directories; /home/msfadmin denotes a mix of security boundary between development and production. This then introduces the angle where web application vulnerabilities could be taken advantage of by a remote attacker should these be exposed on the internet. The other challenge is use of default or guessable credentials as seen when revoked root access using the password "kali" which is testament to lack of security best practices. To a malicious user, this environment offers an opportunity not only to gain initial access but as well sustainability in pivoting to others machine in the network for privilege use for other activities. These risks are significantly high and should immediately require close attention in a live setup due to the coexistence of unused services and lack of access control, password file corruption and not up-to-date technology platform.

## Conclusion

Vulnerabilities were detected as a result of the older software, the services which were completely open and weak configurations in place. This got root access rather easily, which means there is a high probability of exploitation. Taking that into account, it is critical to do prompt remediation along with service hardening, system updates which is a must, access controls need to be properly implemented so as to cushion these vulnerabilities and stop them from sprouting into entire system compromises.

# Part 2: Executive Summary

## Background

A technical assessment was performed by scanning a Chester Research Ventures Ltd. The Linux-based virtual machine utilized in a controlled test environment was simulating what would be a typical infrastructure setup. During the assessment it was our objective to set up realist attack scenarios to understand what system vulnerabilities might fall prey to a bad actor from outside. Upon review, it was determined that the device in question had a VirtualBox-hosted machine with an outdated version 2.6.x of the Linux kernel.

Nmap's scripting engine and manual service analysis were part of the subsequent vulnerability scanning, which was followed by an attempt at exploitation. Showed rpcbind, portmapper, and rpc.statd services being present across various ports (111, 33332, and high UDP ranges), all services are well known for legacy vulnerabilities as we combed services. A further searchsploit reconnaissance displayed public exploits related to these services – pushing for deeper validation.

Subsequent to the vulnerability scanning we did with Nmap’s scripting engine and some manual service analysis and exploitation attempted. Even though we couldn't find any Metasploit modules for RPC-based enumeration, the final previous scans indicated available legacy attack vectors. Post-exploitation validation disclosed root access, sensitive configuration file retrieval and bad hygiene indicators like absence of /etc/passwd whereas accessing shadow files were the outcome. This background should illustrate poorly configured system, with outdated software bundled with services that can be exploited.

## Overall Posture

The assessed Virtual Machine, a Linux based one has a very poor overall security posture and if ever stood up to live internet / production facing would have a: high level of risk. The platform is constructed on an old Linux kernel (version 2.6.x) which exposes a large attack surface merely without the need to do anything else. Furthermore, the OS has older and improperly configured services such as rpcbind, portmapper and NFS listening in open to potentially harmful ports (111, 33332, 42941) expanding the attack surface even more. Additionally, non-firewalled services are up and running for any connection without restrictions or notice as identified.

During initial enumeration, various ports were found to be open with running services which have been associated with past exploits hence a peek through some Nmap's NSE scripts and Searchsploit, shows vulnerability in the system against known as well as nulled vulnerabilities, DoS exploits, and chances for privilege escalation. Upon gaining entrance, it was clarified thru post-exploitation analysis that system was completely compromised up to root level to where sensitive files such as /etc/shadow & /etc/passwd (recreated manually afterwards) are accessed. It emphasizes the lack of system integrity internet facing or firewalled basic control.

More than that, poor file system security measures, no user segregation and open cron jobs seems like the platform has gone under solid hardening to a stupid vulnerable state. We also found out aged / abandoned web applications located locally, which may point towards web-based vulnerabilities. To sum up, the overall security posture looks less to none and would be totally no go under any possible real-life scenario. A direct upshot is an a remediation activity and thus further kernel updating, cutting services opening and following best practices is a must to carry out and lower the certainty risks presented.

## Risk Profile

The system we scanned has a risk profile that really means successful exploitation most likely will happen. We accessed a Linux-based VM where the kernel is unfortunately running on a very outdated version (2.6.x) that isn't maintained or patched by the vendor. This amounts to a lack of analysis on the expected behaviors and changes to the system - it sounded as though vendors are trying to figure out what is happening without it. There is also an increase in threat vector\_presence due to RPC-based services such as rpcbind, portmapper and rpc.statd exposing on well-known ports; rpcbind and portmap in general lies on port 111 - however over the years, exploit rules and the like have been known for and it makes two-way communication. This sort of service and if you have that running on an end node, someone will be able to use a known attack.

The findings after exploiting the system, I'd say the trust level on this system is much higher, and finally the fact that someone got full root and was so careless as to put see cookies (or whatever sensitive data) in /home or /var directory or accessed your /etc/shadow can retrieve your already hashed passwords. And because the /etc/passwd file was also only missing temporarily - it's likely you've got something in your system or botched a configuration. If your system is missing a /etc/passwd file, it will cause errors for any new software installs and packages if you're simply swapping a file across servers to make them behave same. Another is the lack of firewalls, no logins or proper access controls, no effort just to secure your system with a base defense-in-depth principle. And this compromised development or stable web-app (or both). In simpler terms, it left the system in a critical state with high value assets on the Internet fix this fast or woe inherit.

## General Findings

Analysis of the Linux-based virtual machine showed that the system bared several critical vulnerabilities, lack of protection redeems within the system. The virtual machine was discovered to run an outdated Linux kernel version (2.6.x), which is no longer supported that entails a list of known vulnerabilities. This in itself puts it at the edge, as kernel-level assaults are ready in the wild and frequently utilized in privilege escalation by the opponents. One red flag in particular was that the system was running legacy services such as rpcbind, portmapper, and rpc.statd on well-known ports like 111/tcp and 33332/tcp which are a magnet for attacks. These services were wide open without any firewall or access control filters. Nmap scans revealed these ports were accessible and led to active endpoints whereas further enumerating showed these services were known vulnerable using Searchsploit.

Finally during post-exploitation’s root-level was gained, highlighting there was little to no segmentation of user privileges, very weak defenses. /etc/passwd and /etc/shadow were accessed demonstrating the system was prepared for password theft and offline brute-force attacks as /etc/passwd was missing initially and later recreated. There were also various cron jobs but left unmonitored with no access control hence they could easily be used for persistence or another exploitation approach. Unsecured application files discovered in /home/msfadmin/vulnerable/ indicated the existence of software that is either susceptible or has been abandoned, such as TikiWiki, which presents further attack points. Generally, the system missed critical security configurations such as role user management, access constraints, updated software and baseline integrity monitoring – hence posed as a high risk so as to be exploited.

## Recommendation Summary

Based on the findings from the Review immediate action appears to be needed to address the significant security threats of the stock Linux-based virtual machine. The top go-to step will be to upgrade the guest OS and kernel to a supported version that has active security patching. The old 2.6.x kernel is extremely insecure and should not be used anywhere under any circumstances, even in testing. Disable all unnecessary and legacy services such as rpcbind, portmapper, and rpc.statd unless explicitly needed. When required, they should be heavily restricted through firewall rules, host-based access controls, and service isolation. Configure firewall policies to only permit inbound and outbound traffic for needed services and IP ranges.

Critical system files like /etc/passwd and /etc/shadow should be monitored for changes using Tripwire or similar tools. Password policies should require complexity and passwords and a review of all accounts for need and privilege assigned. Moreover, remove unmaintained or testing applications (e.g., TikiWiki) or sandbox them in a segregated environment. Lastly, take the extra basic step like disabling root SSH login, enabling audit logging, proper cron job permissions, and employ a host intrusion detection system (HIDS) for catching violations.

## Strategic Roadmap

In order to establish a more comprehensive security posture for Chester Research Ventures Ltd., both short-term actions and long-term strategies will need to be put in motion. Short-term objectives need to be kicked off with the immediate upgrading and patching of the linux operating system, which includes the kernel, to a stem standard release. Disable or secure the legacy services such as rpcbind, portmapper, and rpc.statd using host-based firewalls (e.g., ufw or iptables) that limit the access. Remove unnecessary software and user accounts and enforce strong passwords policies. Run a full vulnerability scan for the system using tools like OpenVAS or Nessus and fix all of the critical and high risk findings. Reinstate necessary system files like /etc/passwd with the proper permissions and ensure that access as root is restricted.

Long term objectives will be geared towards putting a sustainable security framework into place. Create a formal patch management process for timely applications of updates to system and software. Implement system hardening practices using tools like CIS Benchmarks. Using programs like OSSEC or Wazuh, implement centralised logging, frequent audits, and real-time monitoring. Assign role-based access control and enforce least privilege on all user accounts. Regularly backup critical data and test the restore procedures. Finally, to guarantee that there is always a dynamic and resilient security posture, plan frequent penetration tests and security awareness training for staff members.