



Swinburne University of Technology
Faculty of Science, Engineering and Technology

ASSIGNMENT AND PROJECT COVER SHEET

Unit Code: COS30015

Unit Title: IT Security

Assignment number and title: Practical Project

Due date: 5 September

Lab group:

Tutor: Fusen Guo

Lecturer: Dr. Rory Coulter

Family name: Giacoppo

Identity no: 104071453

Other names: Marco

To be completed if this is an INDIVIDUAL ASSIGNMENT

I declare that this assignment is my individual work. I have not worked collaboratively, nor have I copied from any other student's work or from any other source except where due acknowledgment is made explicitly in the text, nor has any part been written for me by another person.

Signature: MarcoG

To be completed if this is a GROUP ASSIGNMENT

We declare that this is a group assignment and that no part of this submission has been copied from any other student's work or from any other source except where due acknowledgment is made explicitly in the text, nor has any part been written for us by another person.

ID Number

Name

Signature

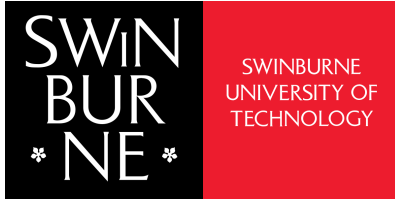
Marker's comments:

Total Mark: _____

Extension certification:

This assignment has been given an extension and is now due on _____

Signature of Convener: _____ Date: _____ / 2024



Swinburne University of Technology

IT Security COS30015

Practical Project (Assignment 1)

Malicious Software / Activity

Marco Giacoppo (104071453)

Submission Date: 5th September
Due Date: 5th September at 23:59pm

Introduction

In today's globalized world, cyber threats are a constant concern for both individuals and companies. Among these dangers, malicious software, or malware as it's commonly known, poses a significant risk. The damage caused by malware can range from simple data loss to total system failure, often happening without the victim's knowledge. Understanding and dealing with malware is a fundamental skill for anyone in cybersecurity, as it's crucial not only to know how to use it but also how to defend against it.

This report offers an in-depth look at a simulated cybersecurity scenario centred around malware. The focus is on using a basic malware sample as an offensive tool and implementing system hardening as a defensive measure. The goal is to recreate a realistic cybersecurity situation within a controlled virtual environment, demonstrating the malware's effects and assessing the effectiveness of the defensive strategies used. Through this case study, we aim to enhance our understanding of cybersecurity practices, particularly in protecting against malicious software.

Case Study: Malicious Software/Activity

Background of the Threat

Malware refers to any type of software designed with the intent to cause harm or gain unauthorized access to computers, networks, or systems. While there are different kinds of malware, their primary goal is often criminal—whether it's to exploit vulnerabilities for financial gain, steal sensitive information, or disrupt operations.

Enumeration is a technique used by attackers in the early stages of a cyber attack. It's gathering detailed information about a target system such as open ports, active services, installed software and user accounts. This is like a burglar scouting a neighbourhood, noting which houses have security, which windows are open and when the homeowners are away. For cyber attackers, enumeration is invaluable because it gives them a roadmap of potential weaknesses that can be exploited in later stages of the attack.

What makes enumeration particularly dangerous is that it's often conducted stealthily, making it difficult for many security tools to detect. By learning the system's architecture and pinpointing its vulnerabilities, attackers can significantly increase their chances of a successful breach, all while staying under the radar. This phase is crucial, as the information gathered can greatly enhance the likelihood of a successful attack.

Typical Adversary Tradecraft

Enumerations are usually done by the attackers using automated tools. These tools can run a set of commands which will test the target system and gather a lot of information. The

information gathered can show such issues as outdated software, misconfigured services or open network ports, all of which are popular targets.

For example, an attacker might run an enumeration script and discover that a system is using an outdated version of SSH (Secure Shell). This is a service known to have vulnerabilities. Armed with this knowledge, the attacker could craft a specific exploit that targets this vulnerability, potentially gaining unauthorized access to the system.

The ease of use makes attackers be able to obtain information and makes enumeration a common and effective strategy. Even normal people like us with minimal technical skills can execute enumeration scripts to gather critical information about a target system, making it a tactic that organizations must be prepared to defend against.

Justification for the Chosen Threat

Understanding enumeration is essential for anyone involved in cybersecurity. It is often the first step in a cyber-attack, making it a crucial area to study and defend against. By delving into how enumeration works and how it can be countered, organizations can significantly strengthen their defences against a wide array of threats.

In this case study, we focus on script-based enumeration because it clearly demonstrates how attackers gather crucial information and more importantly, how defenders can avoid this. Enumeration is a common step in real-world attacks, often executed using simple tools that are easily accessible. By examining how to detect and mitigate enumeration through methods like system hardening, this case study sheds light on both the offensive and defensive sides of cybersecurity.

Tool Selection

Offensive Tool: Enumeration Script

Tool Description:

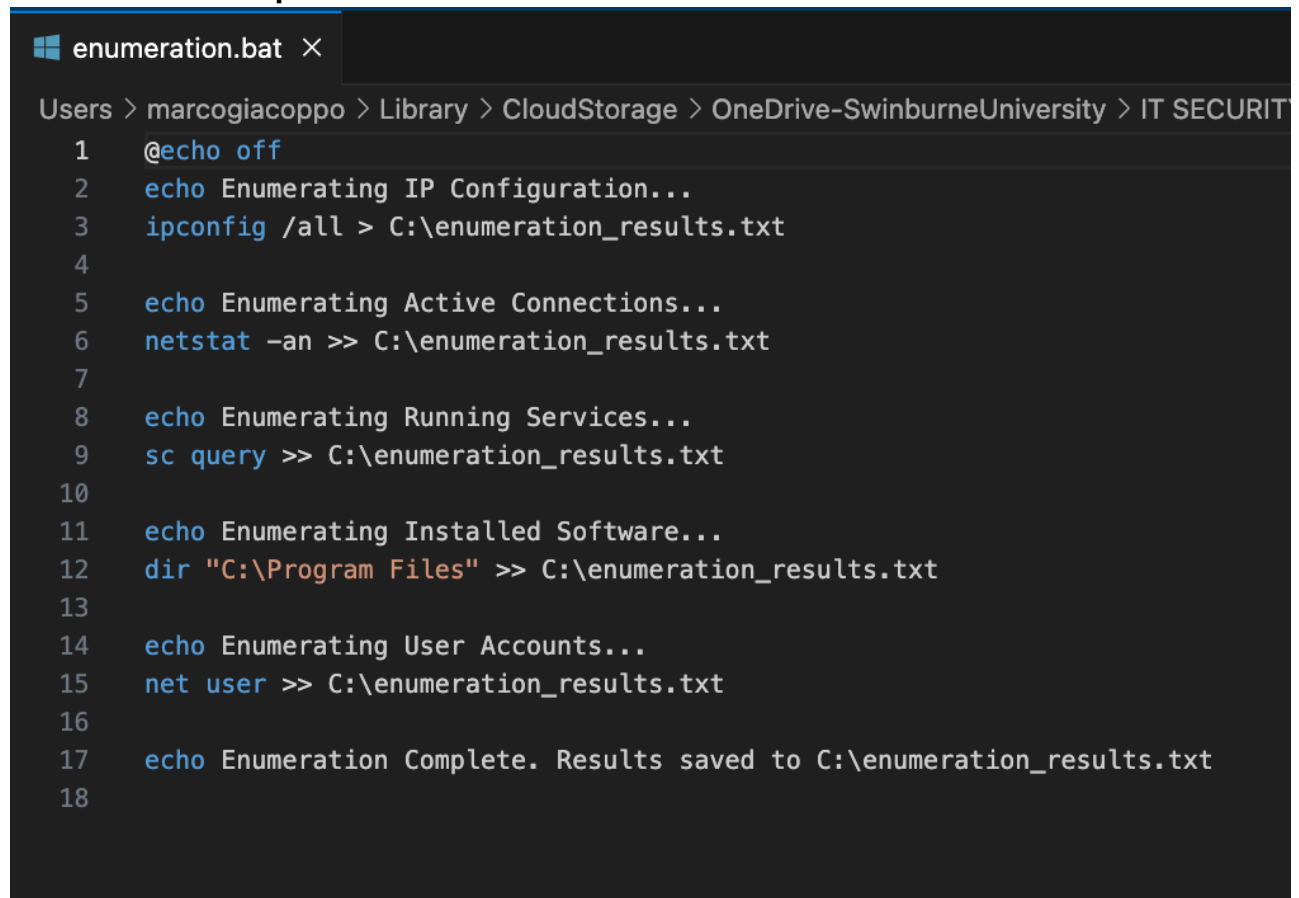
The offensive tool selected for this study is a script-based enumeration tool. This script is designed to automate the collection of detailed information about a target system, including open ports, running services, installed software, and user accounts. Essentially, enumeration scripts run a series of commands that dig into the system and return valuable data, which attackers can then use to pinpoint vulnerabilities and plan their next move.

Justification:

Enumeration scripts are a favourite tool among attackers because they are both simple to use and highly effective. They require minimal setup but can quickly gather a lot of information about a target system with very little effort. This makes them an excellent choice for our case study, as they clearly demonstrate how attackers perform reconnaissance before moving on to more sophisticated attacks.

By selecting this enumeration script, my goal was to simulate a realistic attack scenario that emphasizes the risks of leaving systems vulnerable to unauthorized access. Unlike more complex malware, enumeration scripts are easy to run and straightforward to analyse, making them particularly valuable for educational purposes. The information these scripts uncover can reveal critical weaknesses in a system's configuration, underscoring the importance of implementing strong security measures.

Enumeration Script that I built:



```
enumeration.bat X
Users > marcogiacoppo > Library > CloudStorage > OneDrive-SwinburneUniversity > IT SECURITY
1  @echo off
2  echo Enumerating IP Configuration...
3  ipconfig /all > C:\enumeration_results.txt
4
5  echo Enumerating Active Connections...
6  netstat -an >> C:\enumeration_results.txt
7
8  echo Enumerating Running Services...
9  sc query >> C:\enumeration_results.txt
10
11 echo Enumerating Installed Software...
12 dir "C:\Program Files" >> C:\enumeration_results.txt
13
14 echo Enumerating User Accounts...
15 net user >> C:\enumeration_results.txt
16
17 echo Enumeration Complete. Results saved to C:\enumeration_results.txt
18
```

Defensive Tool: System Hardening

Tool Description:

In this study, the defensive strategy I chose was system hardening. This approach involves adjusting the operating system and applications to minimize their exposure to potential attacks. System hardening can include steps such as disabling unnecessary services, setting up firewall rules, applying security patches, and limiting user privileges to only what is essential.

For this case study, the specific hardening techniques implemented include:

- **Firewall Configuration:** Setting up firewall rules to block unauthorized access to critical ports and services, reducing the number of attack vectors.

- **Service Management:** Identifying and disabling services that are not needed, preventing attackers from exploiting these potential vulnerabilities.

Justification:

System hardening is one of the most effective ways to defend against enumeration and other dangerous techniques. By reducing the system's attack surface, essentially shrinking the number of possible entries makes it become much more difficult for an attacker to gather valuable information about the system. The decision to focus on system hardening is grounded in its essential role in cybersecurity. Hardening a system is often the first line of defense in protecting against various attack types, including enumeration. By showing how these techniques can prevent attackers from obtaining critical information, this study highlights the practical importance of maintaining a well-secured system configuration.

Testing Scenario

To effectively showcase the interaction between the offensive enumeration tool and the defensive system hardening techniques, the following testing scenario was conducted:

Initial Setup:

1. **Virtual Environment:** The test will be conducted on a single virtual machine (VM) configured to mimic a standard operating system setup, similar to what you might find in a typical real-world environment.
2. **Enumeration Script Execution:** The chosen enumeration script will be run on this VM to collect information like open ports, running services, installed software, and system users. This simulates the inspection phase of an attack, where an attacker gathers data about the target system to plan their next move.

System Hardening:

1. **Firewall Configuration:** The VM's firewall will be adjusted to block unauthorized access to critical ports and services. Specific rules were set up to reduce the system's exposure to potential network-based threats.
2. **Service Management:** Unnecessary services on the VM were identified and disabled. For example, services that aren't essential to the VM's primary function but could be exploited by attackers were turned off.

Rerun the Enumeration Script:

After the system hardening measures have been applied, the enumeration script will be executed again on the VM. The aim is to observe any differences in the amount and type of information the script can gather after the system has been secured.

Execution and Results

Initial Execution (Without Defense)

Setup and Execution:

The initial test was conducted on a virtual machine (VM) without any defensive measures in place. The enumeration script was executed to gather information about the system, running a series of commands designed to probe the VM for critical data. This included details about open ports, active services, installed software, and user accounts.

Results:

The script successfully collected a significant amount of information about the target system, revealing several potential vulnerabilities:

- **Open Ports:** The script identified multiple open ports, including those associated with essential services like SSH and HTTP, as well as other commonly used ports.
- **Running Services:** It listed all active services, including several that weren't necessary for the system's intended function. These unnecessary services could potentially serve as additional attack surfaces.
- **Installed Software:** The script retrieved a detailed list of installed software, some of which were outdated versions that might be susceptible to known exploits.
- **User Accounts:** It also identified all active user accounts on the system, including those with administrative privileges, which are particularly valuable targets for attackers.

These findings underscore the vulnerabilities of a system when no defensive measures are in place. The information gathered by the script could easily be leveraged by an attacker to plan more sophisticated attacks, such as targeting specific services or exploiting known vulnerabilities in outdated software.

Defensive Tool Application: System Hardening

Following the initial enumeration, system hardening measures were implemented on the VM to enhance its security:

1. **Firewall Configuration:** The firewall was set up to block access to unnecessary ports and restrict network traffic to only the essential services.
2. **Service Management:** Unnecessary services were identified and disabled, reducing the number of active services that could potentially be targeted by attackers.

Rerun of the Enumeration Script (With Defence)

After hardening the system, the enumeration script was executed again to assess the effectiveness of the defensive measures.

Results:

The results from the second execution of the script showed a marked improvement in the system's security:

- **Open Ports:** The number of open ports detected by the script was reduced. Only essential services, like SSH, remained accessible, while other ports were effectively blocked by the firewall.
- **Running Services:** The script was no longer able to list many of the services that were previously active, as they had been disabled during the hardening process.
- **Installed Software:** Although the script still identified installed software, the list now consisted solely of up-to-date versions, minimizing the risk of exploitation through known vulnerabilities.
- **User Accounts:** The script's ability to enumerate user accounts was also limited due to the enhanced security settings, particularly those that improved protection against privilege escalation.

These results clearly demonstrate the effectiveness of system hardening in reducing the attack surface and limiting the amount of information an attacker can gather through enumeration. By blocking unnecessary ports, disabling non-essential services, and applying critical security patches, the overall security posture of the system was significantly strengthened.

Visual Analysis of Testing Results

To further illustrate the impact of system hardening on the security firewall, this section presents visual evidence from the testing scenarios I conducted.

- Netstat Command Output:

Before implementing any defensive measures, the `netstat` command was used to identify active connections and listening ports in the system. Specifically, port 1020 was the one I chose to demonstrate this.

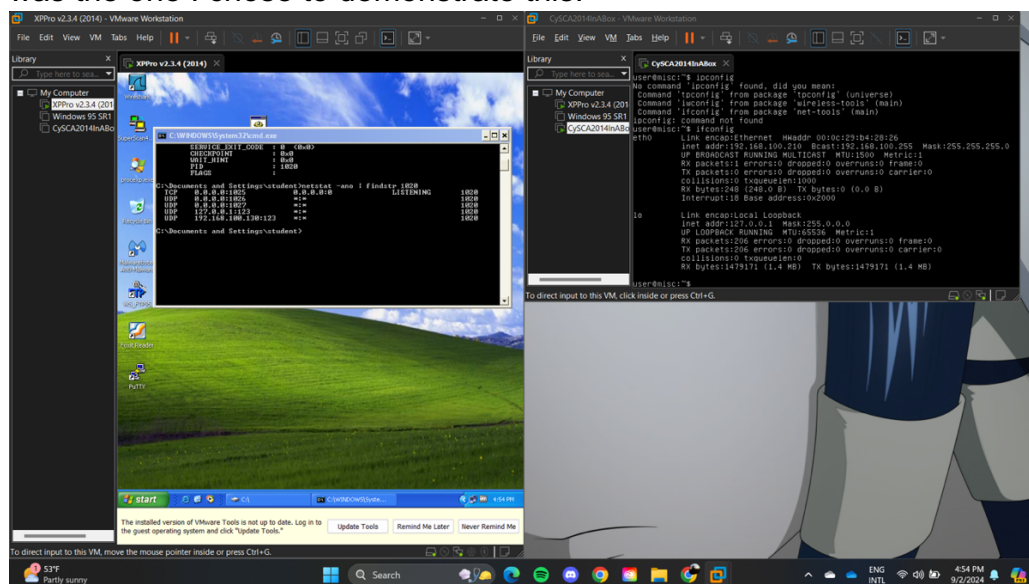


Figure 1: Netstat Command Output

Figure 1 above shows the output, confirming that port 1025 was open and listening for connections.

- **Firewall Configuration:**

The firewall settings were adjusted to block unauthorized access to critical ports and services. Figure 2 below shows the specific configuration used, which contributed to the overall security improvements observed in the post-hardening tests.

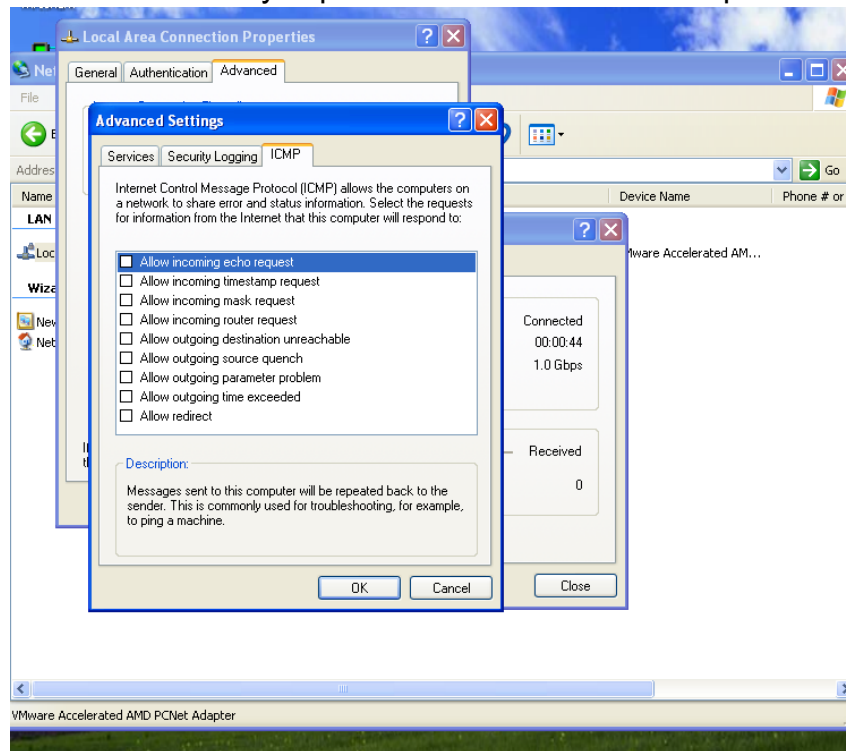
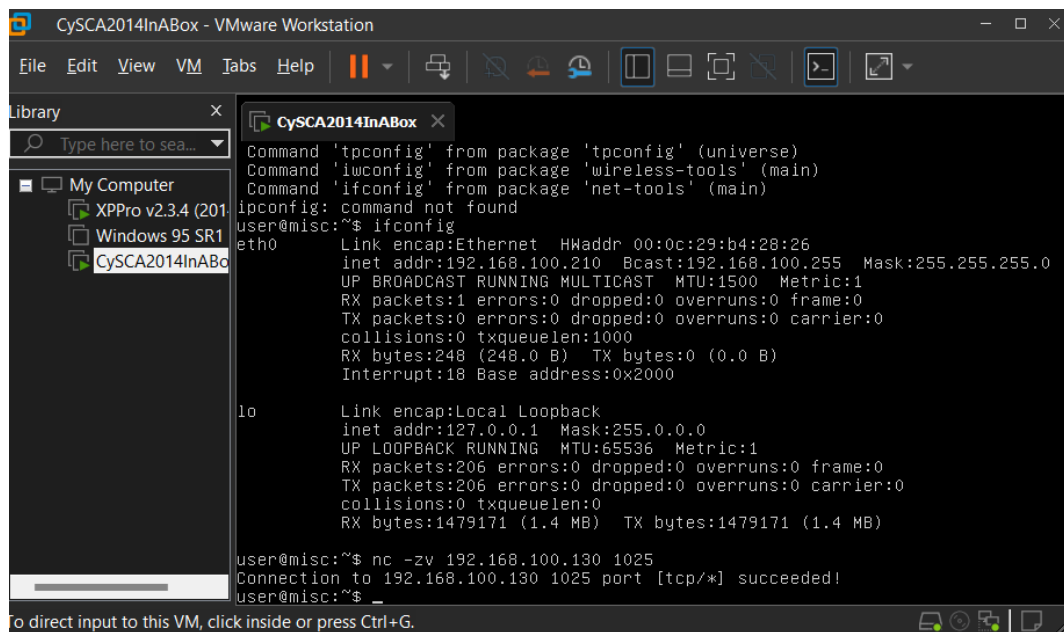


Figure 2: Firewall Settings

- **Netcat Testing Results:**

To validate the presence of open ports, I ran the NC command both before and after the system hardening measures were applied.

Before Firewall Activation: In Figure 3, we can see that the connection to port 1025 was successful. This indicates that the port was open and unprotected, making it vulnerable.



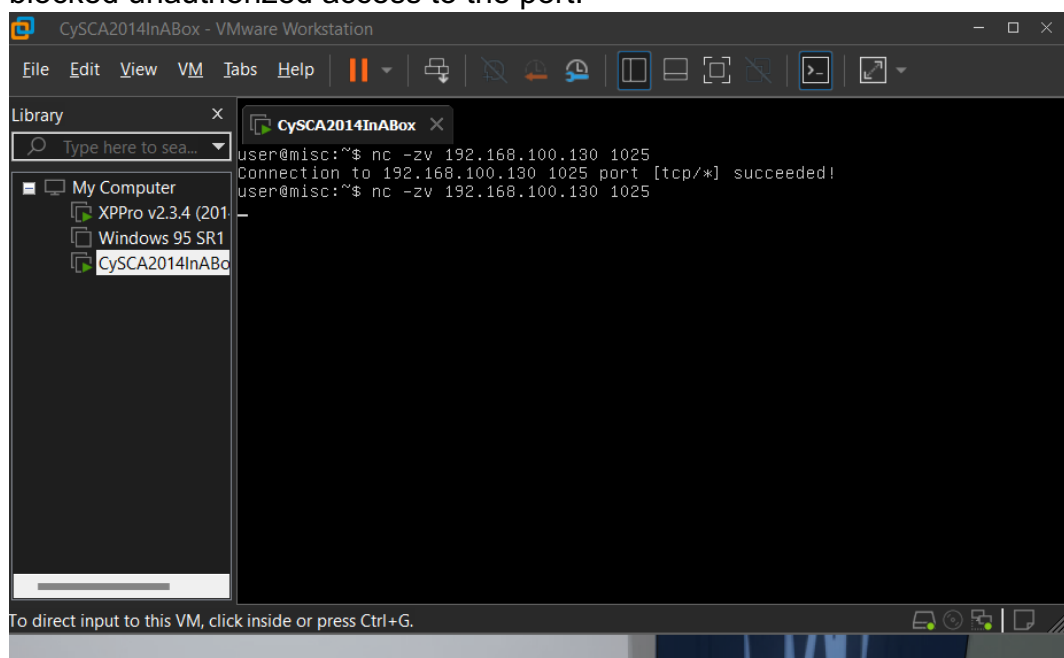
```
Command 'tpconfig' from package 'tpconfig' (universe)
Command 'iwconfig' from package 'wireless-tools' (main)
Command 'ifconfig' from package 'net-tools' (main)
ipconfig: command not found
user@misc:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:b4:28:26
          inet addr:192.168.100.210  Bcast:192.168.100.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:248 (248.0 B)  TX bytes:0 (0.0 B)
          Interrupt:18 Base address:0x2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:206 errors:0 dropped:0 overruns:0 frame:0
          TX packets:206 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1479171 (1.4 MB)  TX bytes:1479171 (1.4 MB)

user@misc:~$ nc -zv 192.168.100.130 1025
Connection to 192.168.100.130 1025 port [tcp/*] succeeded!
user@misc:~$ _
```

Figure 3: Netcat Before Firewall

After Firewall Activation: Figure 4 shows the result of running Netcat after the firewall was enabled. The connection attempt fails, demonstrating that the firewall effectively blocked unauthorized access to the port.



```
user@misc:~$ nc -zv 192.168.100.130 1025
Connection to 192.168.100.130 1025 port [tcp/*] succeeded!
user@misc:~$ nc -zv 192.168.100.130 1025
```

Figure 4: Netcat After Firewall

- **Telnet Testing Results:**

Similar to the Netcat tests, I also used Telnet to verify accessibility of port 1025.

Before Firewall Activation: As shown in Figure 5, Telnet was able to establish a connection to port 1025.

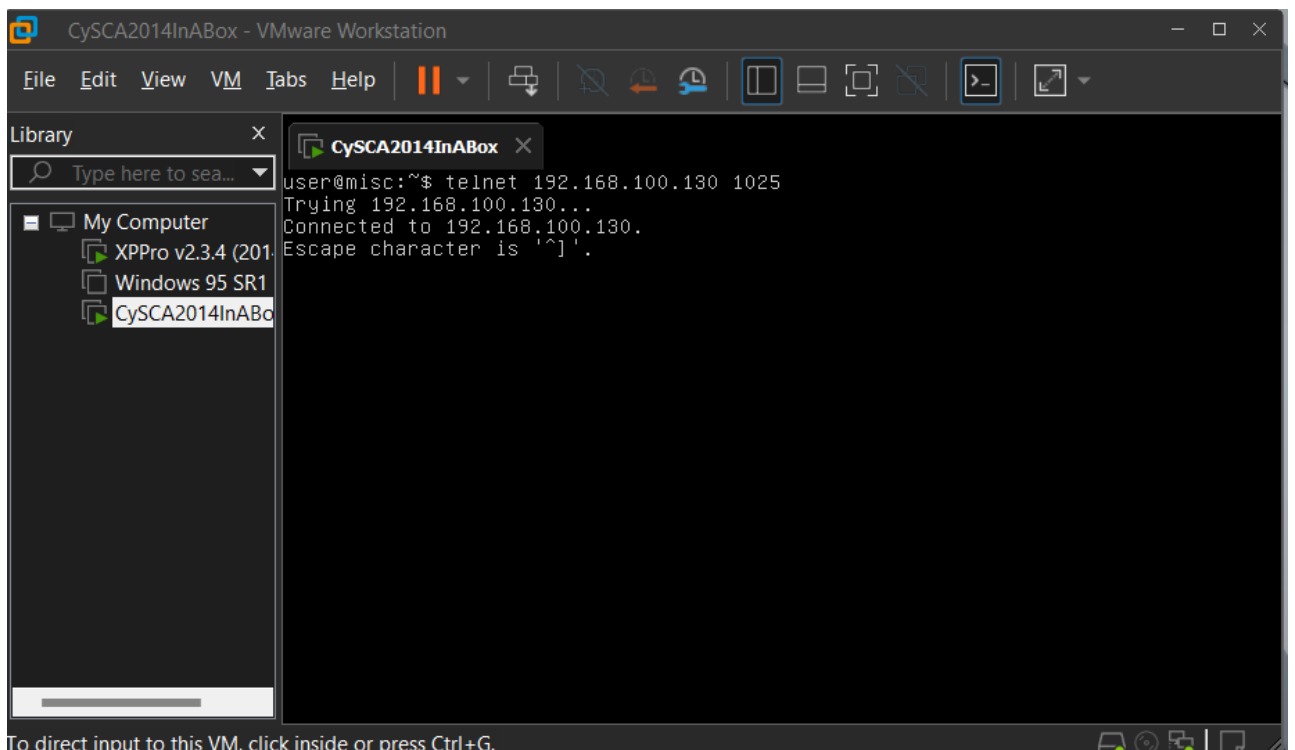


Figure 5: Telnet Before Firewall

After Firewall Activation: Figure 6 shows that connection was unable to be established after putting up the firewall.

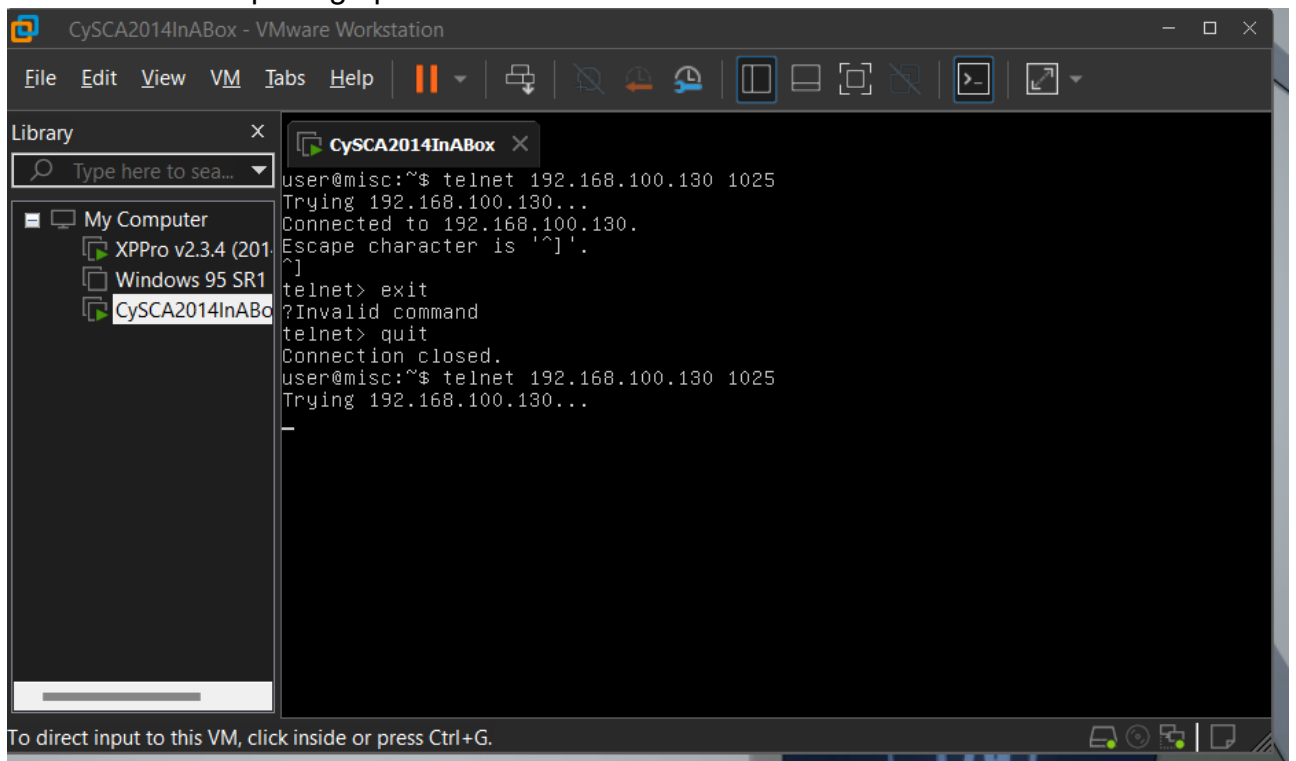


Figure 6: Telnet After Firewall

The visual evidence provided in this section demonstrates the effectiveness of system hardening in enhancing the security of the virtual machine. The transition from an

unprotected system with open ports to a hardened system with blocked connections underscore the importance of implementing strong defensive measures.

MITRE TTPs Mapping

This testing scenario is closely aligned with several techniques from the MITRE ATT&CK framework, which categorizes the tactics, techniques, and procedures (TTPs) commonly used by attackers. The following TTPs are particularly relevant to this scenario:

1. *T1049 - System Network Connections Discovery:*
 - The enumeration script will attempt to identify active network connections, focusing on open ports and running services on the target system. This technique is a staple in an attacker's toolkit, as it helps them map out the network infrastructure and pinpoint potential entry points for further exploitation.
2. *T1016 - System Network Configuration Discovery:*
 - The script will also collect detailed information about the system's network configuration, such as IP addresses, subnet masks, and routing details. Attackers use this data to understand how the system is connected to other devices and networks, which can inform their strategy for lateral movement or network-based attacks.
3. *T1082 - System Information Discovery:*
 - Additionally, the enumeration script will gather general system information, including the operating system version, hardware specifications, and installed software. This information is crucial for attackers as it helps them tailor their exploits to the specific environment they're targeting.

By mapping these TTPs to the testing scenario, we can clearly see how the enumeration script functions within the broader context of established attack techniques. Furthermore, this mapping underscores how system hardening can effectively disrupt these reconnaissance activities, making it much harder for attackers to gather the data they need to plan their attacks.

Analysis

Comparative Analysis

The comparison between the system's initial state and its condition after hardening reveals noticeable differences in the amount and type of information accessible to the enumeration script. Initially, the system was highly vulnerable, with the script successfully extracting detailed data on open ports, running services, installed software, and user accounts. This level of exposure could easily be exploited by an attacker to launch targeted attacks, such as exploiting outdated software or gaining unauthorized access through unprotected ports.

However, after implementing system hardening measures, the effectiveness of the enumeration script was significantly reduced. The number of open ports was minimized to only those essential for the system's operation, with all other ports securely blocked by the firewall. Additionally, some services that were previously active were disabled, further reducing the attack surface and limiting potential entry points for an attacker.

This comparison clearly illustrates the effectiveness of system hardening in defending against enumeration-based attacks. By proactively securing the system, the amount of exploitable information available to an attacker is drastically reduced, thereby strengthening the overall security of the environment.

Impact Evaluation

The impact of the defensive measures is profound. In a real-world scenario, if an attacker successfully ran an enumeration script against an unhardened system, they could likely access a wealth of information that could be used to compromise the system. For instance, open ports could be exploited to attack vulnerable services directly, and outdated software could be targeted using known vulnerabilities. The exposure of user accounts, especially those with administrative privileges, would further increase the risk of a full system breach.

On the other hand, a hardened system offers a much smaller attack surface, making it significantly harder for attackers to gather useful information. The firewall configuration restricts access to only essential services and disabling unnecessary services further limits potential entry points.

Overall, the system hardening measures applied in this scenario not only reduce the immediate risk of exploitation but also provide a strong defense against future attacks. This underscores the importance of ongoing security practices, such as regular patching and careful configuration of system services, in maintaining a secure environment.

Evaluation

Tool Effectiveness

- **Offensive Tool: Enumeration Script**

The enumeration script proved to be a highly effective offensive tool in its initial run, successfully gathering critical system information that could be exploited by an attacker. The script's ability to identify open ports, active services, and installed software highlights its utility in the reconnaissance phase of a cyber-attack. However, its effectiveness was significantly diminished once defensive measures were applied. The reduced amount of accessible information following system hardening demonstrates that while enumeration scripts are powerful tools in unprotected environments, their utility is greatly curtailed when faced with well-implemented defences.

- **Defensive Tool: System Hardening**

The system hardening techniques employed in this scenario—firewall configuration, service management, and software patching—were highly effective in reducing the attack surface and limiting the information that could be gathered by the enumeration script. By blocking unnecessary ports, disabling unneeded services, and ensuring that software was up to date, the defensive measures significantly hindered the script's ability to perform its intended function. This underscores the importance of system hardening as a foundational defense against reconnaissance and other pre-attack activities.

Essential 8 Mitigations

The defensive measures implemented in this scenario align well with several strategies outlined in the Essential 8 framework, which provides a set of prioritized mitigation strategies to help organizations protect against cyber threats:

1. **Application Control:** By disabling unnecessary services and limiting the software that can run on the system, the attack surface is reduced, making it harder for attackers to exploit vulnerabilities.
2. **Patch Applications and Operating Systems:** Regularly applying security patches, as demonstrated in this scenario, is crucial for mitigating known vulnerabilities that could be exploited by attackers.
3. **Configure Microsoft Office Macro Settings:** Although not directly relevant to this scenario, the principle of restricting potentially harmful features or services aligns with the broader system hardening approach.
4. **Restrict Administrative Privileges:** Limiting the exposure of user accounts with administrative privileges, as seen in the post-hardening enumeration results, helps prevent attackers from gaining elevated access.

These mitigations collectively enhance the system's resilience against attacks, particularly those involving reconnaissance and the exploitation of known vulnerabilities.

Comparison to Similar Threats

Enumeration is a common technique used in the early stages of many cyber-attacks, especially those involving targeted exploitation or lateral movement within a network. Attackers often begin by gathering as much information as possible about the target system, like the scenario presented here. The defensive measures applied in this scenario are also relevant to other threats, such as unauthorized access attempts and lateral movement.

For example, the techniques used to harden the system against enumeration could also be applied to protect against brute-force attacks, where an attacker systematically guesses passwords to gain access. In both cases, reducing the attack surface and minimizing the exposure of sensitive information are key defensive strategies. Additionally, the importance of regular patching and careful configuration of services is a common theme across various types of threats, reinforcing the need for proactive and continuous security practices.

Conclusion

In this case study, we explored the dynamics between offensive and defensive cybersecurity tools through the lens of a common exploitation technique: enumeration. The study demonstrated how a simple script-based enumeration tool could effectively gather critical system information in an unprotected environment, highlighting the vulnerabilities that can be exposed during the early stages of a cyber-attack.

However, the application of system hardening techniques, firewall configuration and service management significantly reduced the effectiveness of the enumeration script. By proactively securing the system, the amount of exploitable information accessible to an attacker was quite diminished, underscoring the importance of continuous and comprehensive defensive measures in cybersecurity.

The analysis revealed that while enumeration is a powerful tool in the hands of an attacker, its impact can be effectively mitigated through robust security practices aligned with frameworks such as the Essential 8. This study also highlighted the broader applicability of these defensive strategies to a wide range of cyber threats, reinforcing the need for organizations to maintain a vigilant and proactive approach to security.

In conclusion, the findings of this study emphasize the critical role of system hardening in defending against early-stage cyber-attacks. By reducing the attack surface and limiting the exposure of sensitive information, organizations can better protect themselves against potential exploits and ensure the security of their systems. This case study serves as a practical reminder that even simple security measures can have a great impact on an organization's overall cybersecurity posture.