Final Engagement

By Aloysius Tanyi, Amina Kitwara, Branden George, Cindy Dieng, Di Gallata, James Williams, Ngazwa Andre- Jay

Attack, Defense & Analysis of a Vulnerable Network

Table of Contents

This document contains the following resources:

Offensive

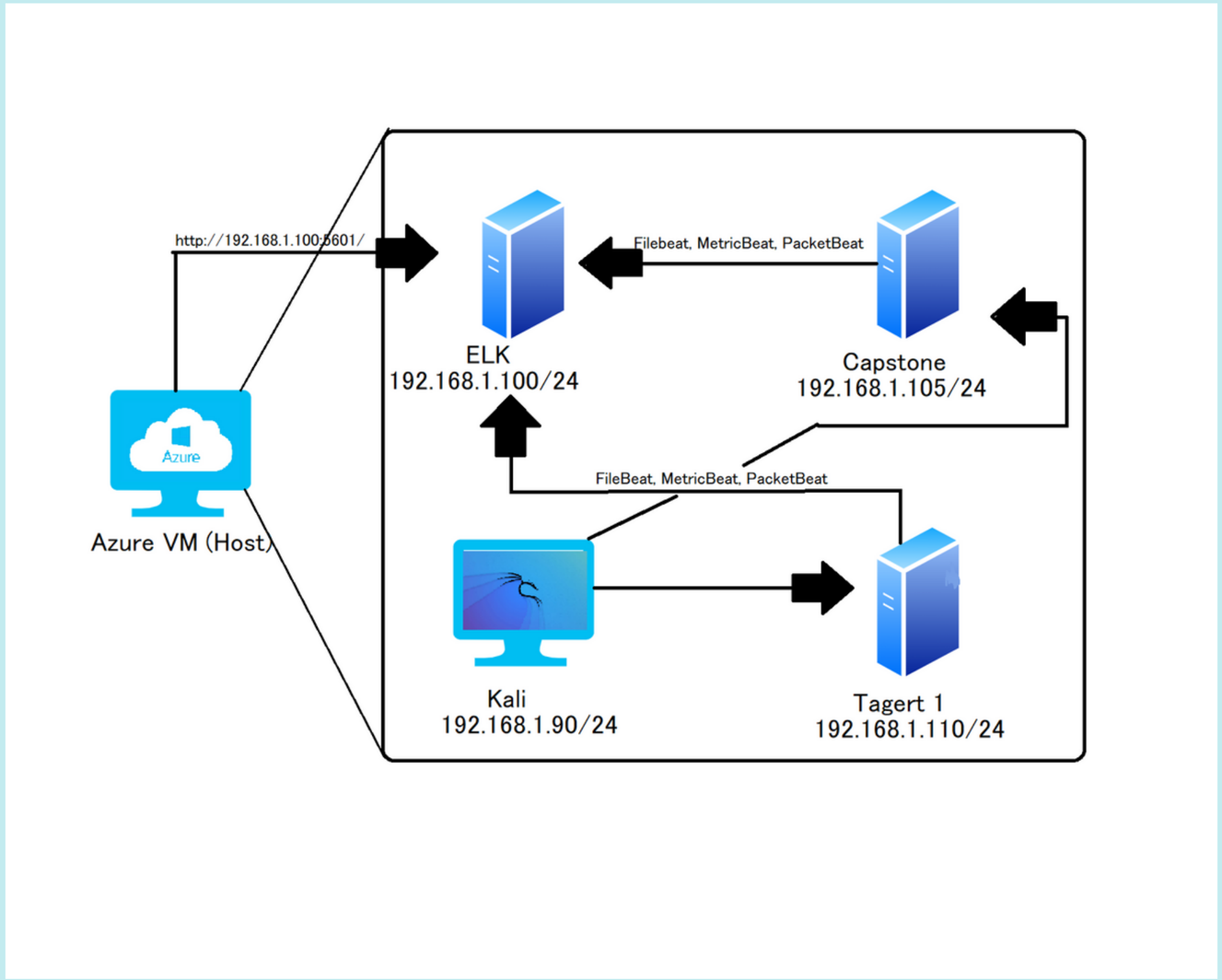
Defensive

Network

Offensive

- Network Topology & Critical Vulnerabilities
- Exploits Used
- Methods Used to Avoid Detection

Network Topology & Critical Vulnerabilities



Network
Address Range: 192.168.1.0/24
Netmask: 255.255.255.0
Gateway: 192.168.1.1

Machines
IPv4: 192.168.1.90
OS: Linux
Hostname: Kali

IPv4: 192.168.1.105
OS: Linux
Hostname: Capstone

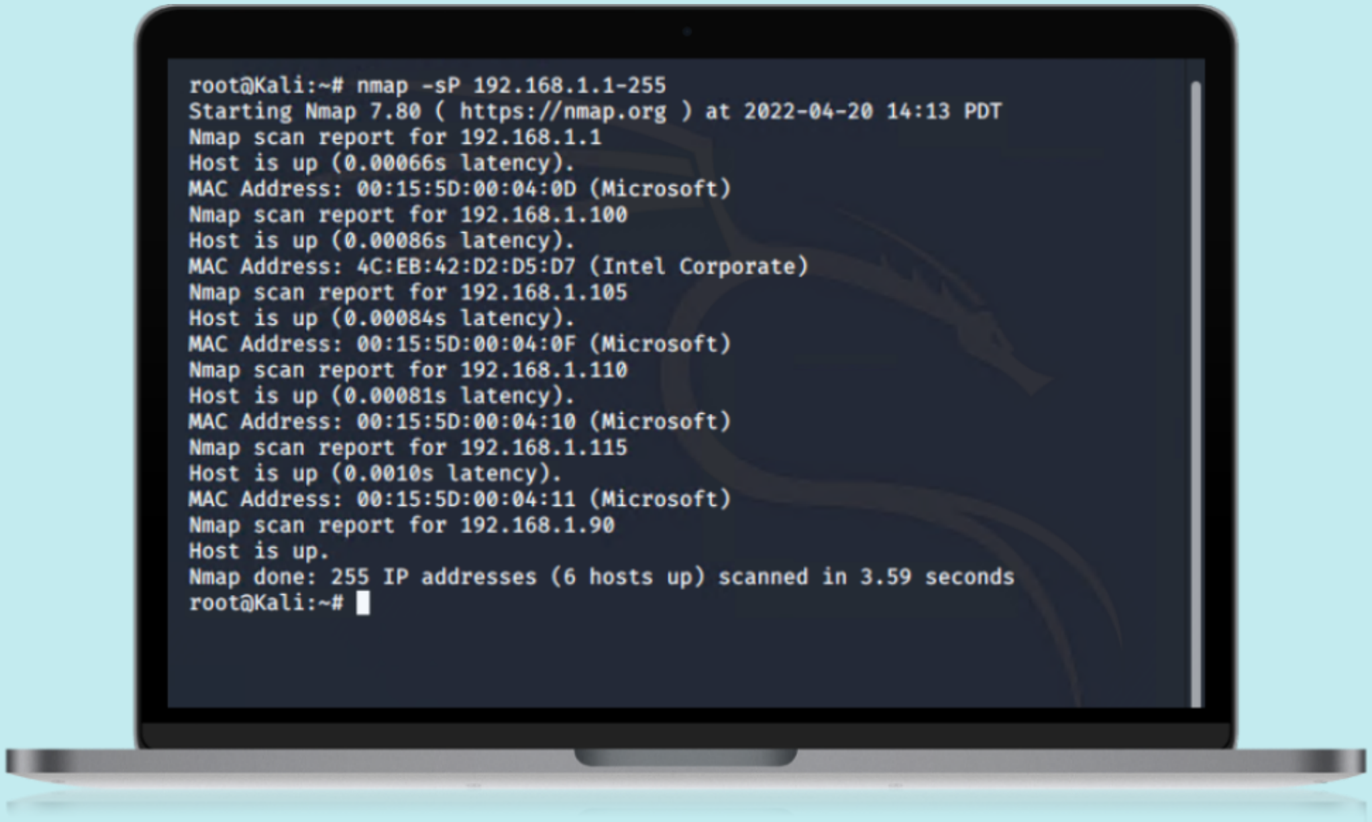
IPv4: 192.168.1.110
OS: Linux
Hostname: Target 1

IPv4: 192.168.1.100
OS: Linux
Hostname: ELK

Critical Vulnerabilities: Target 1

Vulnerability	Description	Impact
Network Vulnerability	Open ports (80,22)	Intrusion into a physical/non-physical environment.
Human Vulnerability	Weak credentials	Ease of access to target 1 machine and databases within it.
Process Vulnerability	Wordpress user enumeration	Found user credentials within database
OS-Based Vulnerability	Privilege escalation	Gained root privileges on system

Exploits Used

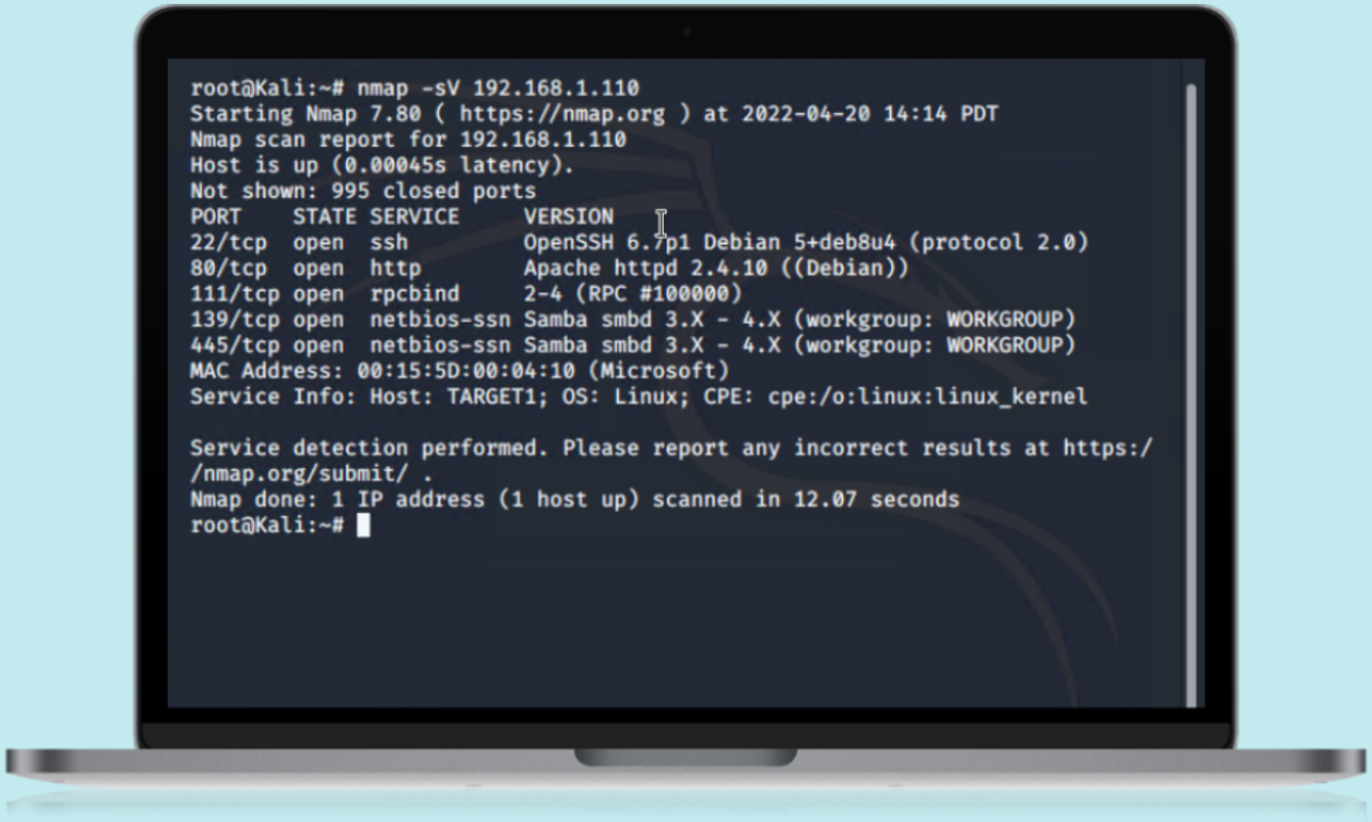


HOW WE EXPLOITED THE VULNERABILITY

- Nmap scan revealed IPs and exposed ports and services
  - $nmap -sP 192.168.1.1-255. (Command Used)

WHAT THE EXPLOIT ACHIEVED

- Identified potential points of entry
  - $nmap -sV 192.168.1.110
  - SSH(Port 22)
  - HTTP(Port 80)



HOW WE EXPLOITED THE VULNERABILITY

- Nmap scan revealed IPs and exposed ports and services
  - $nmap -sP 192.168.1.1-255. (Command Used)

WHAT THE EXPLOIT ACHIEVED

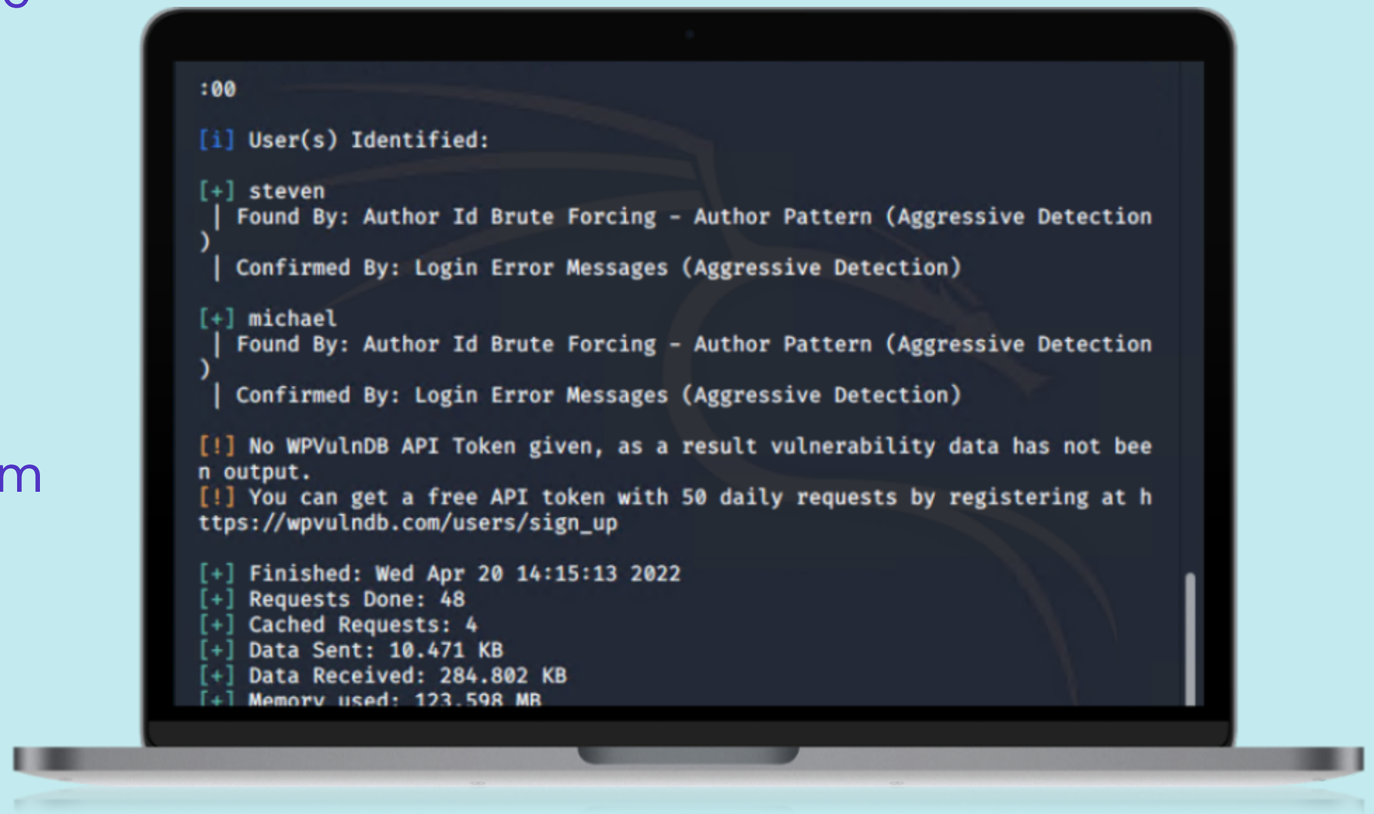
- Identified potential points of entry
  - $nmap -sV 192.168.1.110
  - SSH(Port 22)
  - HTTP(Port 80)

HOW WE EXPLOITED THE VULNERABILITY

- Using Linux we discovered users on the network.
- wpscan ––url http://192.168.1.110/wordpress -eu

WHAT THE EXPLOIT ACHIEVED

- We were able to identify two users from the scan. (michael, steven)
- MySQL server login credentials were also listed in plain text
- We discovered weak login credentials and were able to access michael's user account.

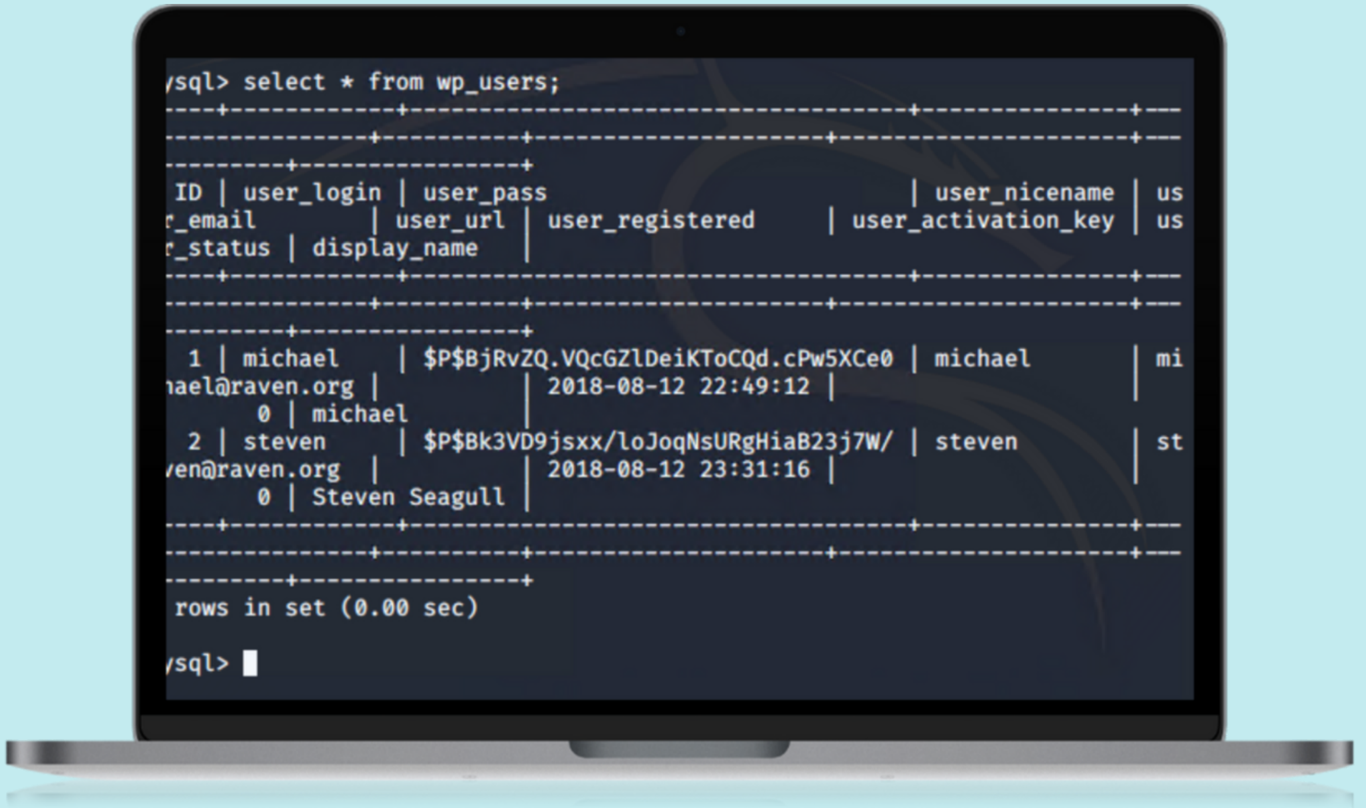


HOW WE EXPLOITED THE VULNERABILITY

Using the login credentials found in wp-config we got the user password hashes.

WHAT THE EXPLOIT ACHIEVED

- We were able to use the hashes we found to log into another account.

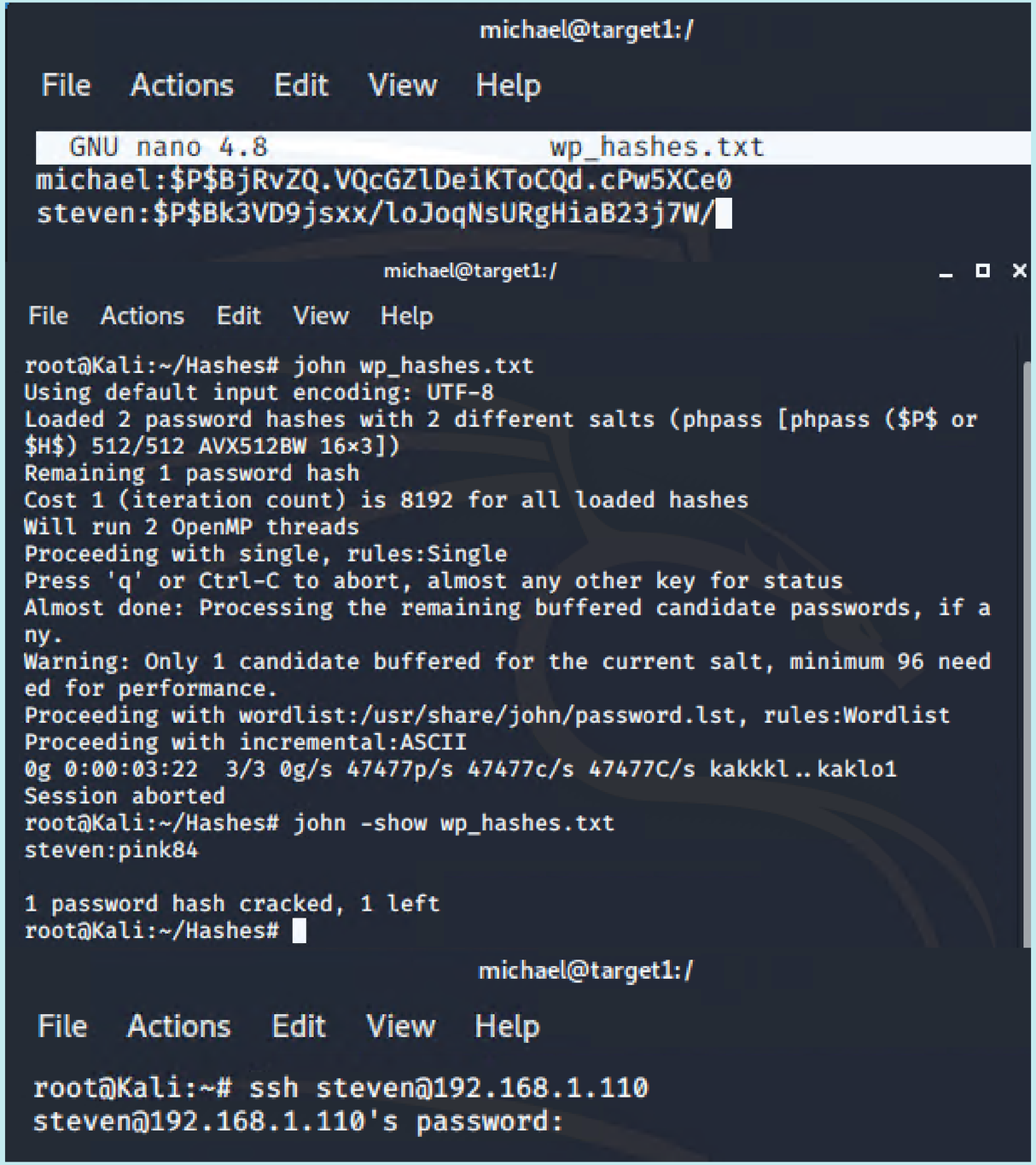


HOW WE EXPLOITED THE VULNERABILITY

- We used john the ripper to exploit the hashes discovered. We got the following
- Steven: pink84

WHAT THE EXPLOIT ACHIEVED

- We used the cracked password and we were able to SSH into steven's account.



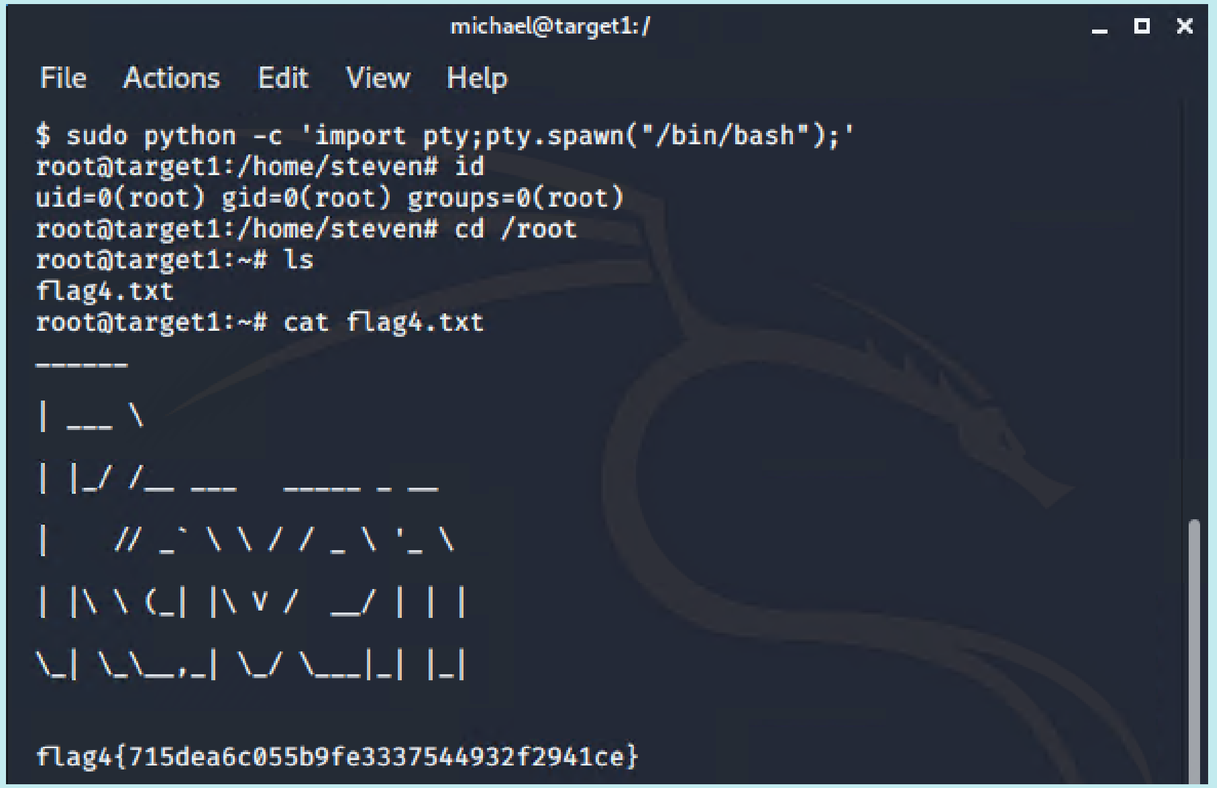
HOW WE EXPLOITED THE VULNERABILITY

- Command used to gain root access:

sudo python -c 'import pty;pty.spawn("bin/bash")'

WHAT THE EXPLOIT ACHIEVED

- User (Steven) had sudo privileges which we exploited through a spawn shell.
- We were able to gain root access allowing us to find the 4th flag.



Methods Used to Avoid Detection

Stealth Exploitation of [Network Vulnerability]

Monitoring Overview

- Which alerts detect this exploit?
- Which metrics do they measure?
- Which thresholds do they fire at?

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
- Are there alternative exploits that may perform better?

Stealth Exploitation of [Network Vulnerability]

Monitoring Overview

- Excessive HTTP Request Alert
- http.response >400
- IS ABOVE 400 FOR THE LAST 5 Minutes

Mitigating Detection

- Backdoor access to the target
- using meterpreter reverse shell

Stealth Exploitation of [Human Vulnerability]

Monitoring Overview

- Which alerts detect this exploit?
- Which metrics do they measure?
- Which thresholds do they fire at?

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
- Are there alternative exploits that may perform better?

Stealth Exploitation of [Human Vulnerability]

Monitoring Overview

- SSH Login Alert
- Triggers when user attempts to access the system over Port 22.
- Any unauthorized attempt

Mitigating Detection

- Using another port that is less obvious.
- Using a reverse shell exploit. Using hydra to exploit weak passwords.

Stealth Exploitation of [Process Vulnerability]

Monitoring Overview

- Which alerts detect this exploit?
- Which metrics do they measure?
- Which thresholds do they fire at?

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
- Are there alternative exploits that may perform better?

Stealth Exploitation of [Process Vulnerability]

Monitoring Overview

- SQL Database alert
- Monitor server traffic for unauthorized attempts to access SQL Database or related directories.
- Alert is triggered when an unauthorized IP connections is made to the SQL Database.

Mitigating Detection

- Spoof our IP address or use metasploit to attack the payload allowing reverse shell and gaining access to system.
- Bind TCP payload and reach the victim's machine on a new port.

Stealth Exploitation of [OS-Based Vulnerability]

Monitoring Overview

- Which alerts detect this exploit?
- Which metrics do they measure?
- Which thresholds do they fire at?

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
- Are there alternative exploits that may perform better?

Stealth Exploitation of [OS-Based Vulnerability]

Monitoring Overview

- Privilege Escalation Alert
- Unauthorized root access attempts
- Triggered when unauthorized sudo command is executed.

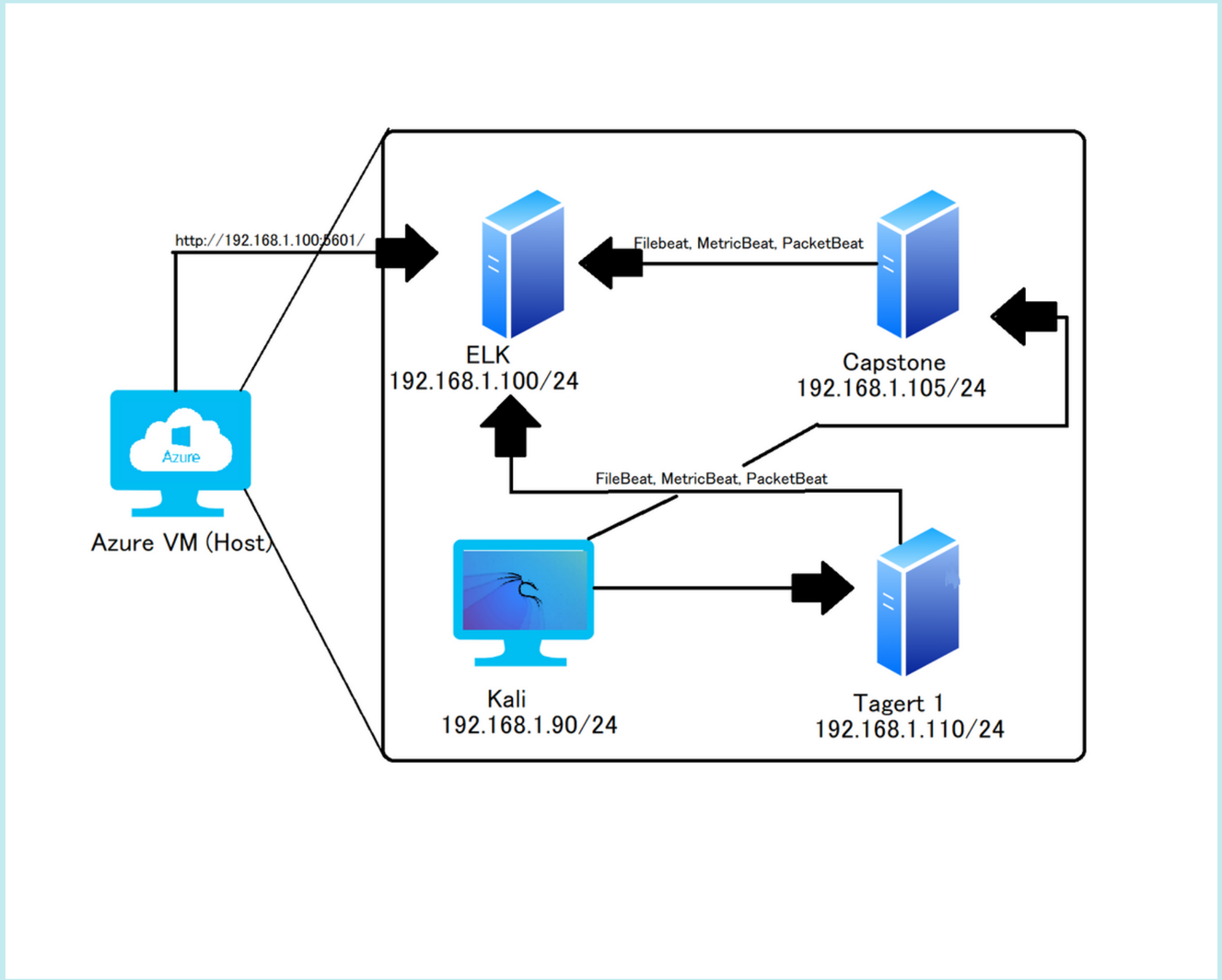
Mitigating Detection

- Using an OS Command Injection
- Finding vulnerabilities in the kernel and exploiting them.

Defensive

- Network Topology & Critical Vulnerabilities
- Alerts Implemented
- Hardening
- Implementing Patches

Network Topology



Network
Address Range: 192.168.1.0/24
Netmask: 255.255.255.0
Gateway: 192.168.1.1

Machines
IPv4: 192.168.1.90
OS: Linux
Hostname: Kali

IPv4: 192.168.1.105
OS: Linux
Hostname: Capstone

IPv4: 192.168.1.110
OS: Linux
Hostname: Target 1

IPv4: 192.168.1.100
OS: Linux
Hostname: ELK

Our assessment uncovered the following critical vulnerabilities in Target 1.

Vulnerability	Description	Impact
Network Vulnerability	Open ports (80,22)	Intrusion into physical/non-physical environment.
Human Vulnerability	Weak credentials	Ease of access to target 1 machine and databases within it.
Process Vulnerability	Wordpress user enumeration	Found user credentials within database
OS-Based Vulnerability	Privilege escalation	Gained root privileges on system

Alerts Implemented

Alert

Summarize the following: Excessive HTTP Error

Which metric does this alert monitor? 'http.response.status_code'

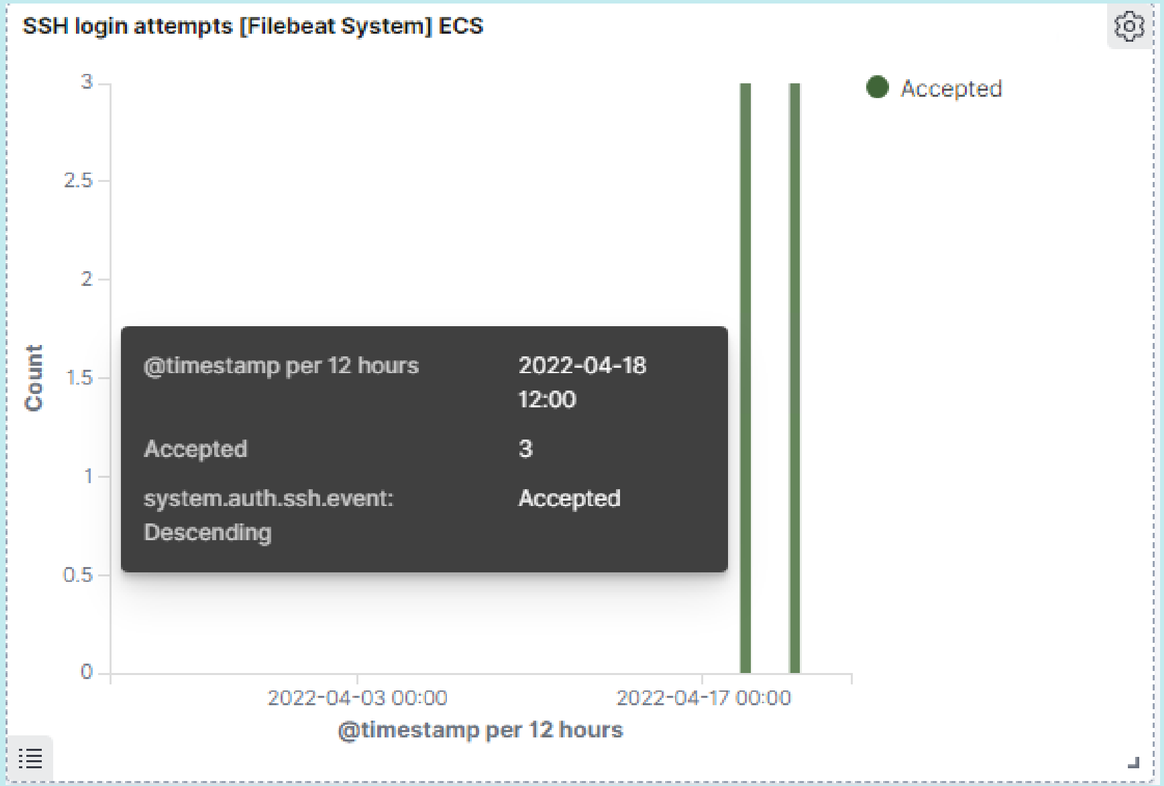
What is the threshold it fires at? IS ABOVE 400 FOR THE LAST 5 minutes

Alert

Summarize the following: SSH Alert

Which metric does this alert monitor? SSH Connections

What is the threshold it fires at? When any conections is made.

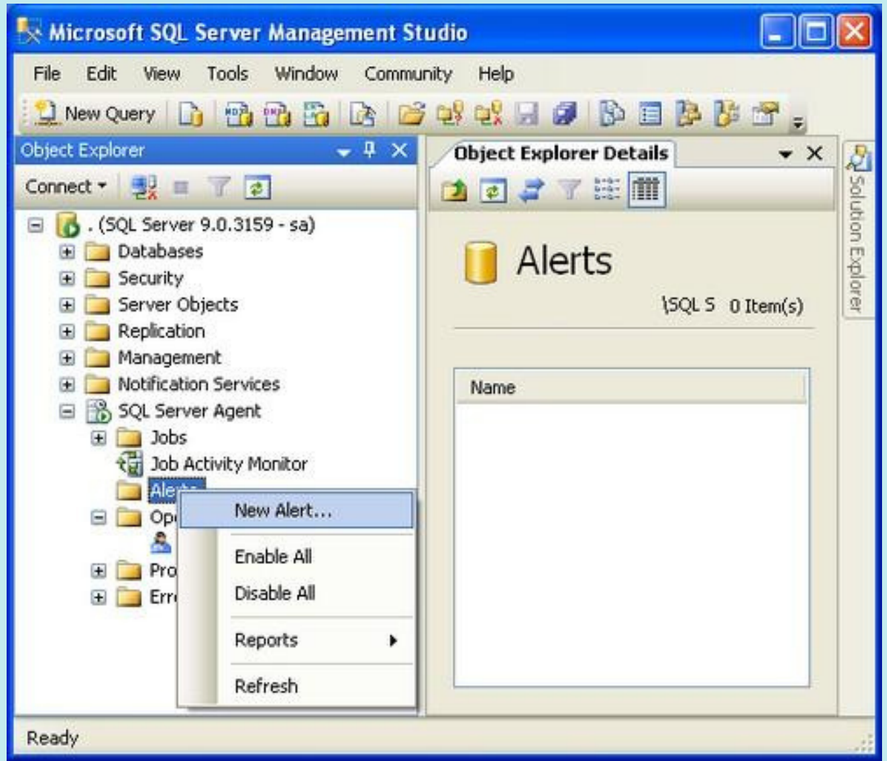


Alert

Summarize the following: SQL Database Alert

Which metric does this alert monitor? Monitor server traffic for unauthorized attempts to access SQL Database or related directories.

What is the threshold it fires at? Alert is triggered when an unauthorized IP connections is made to the SQL Database.

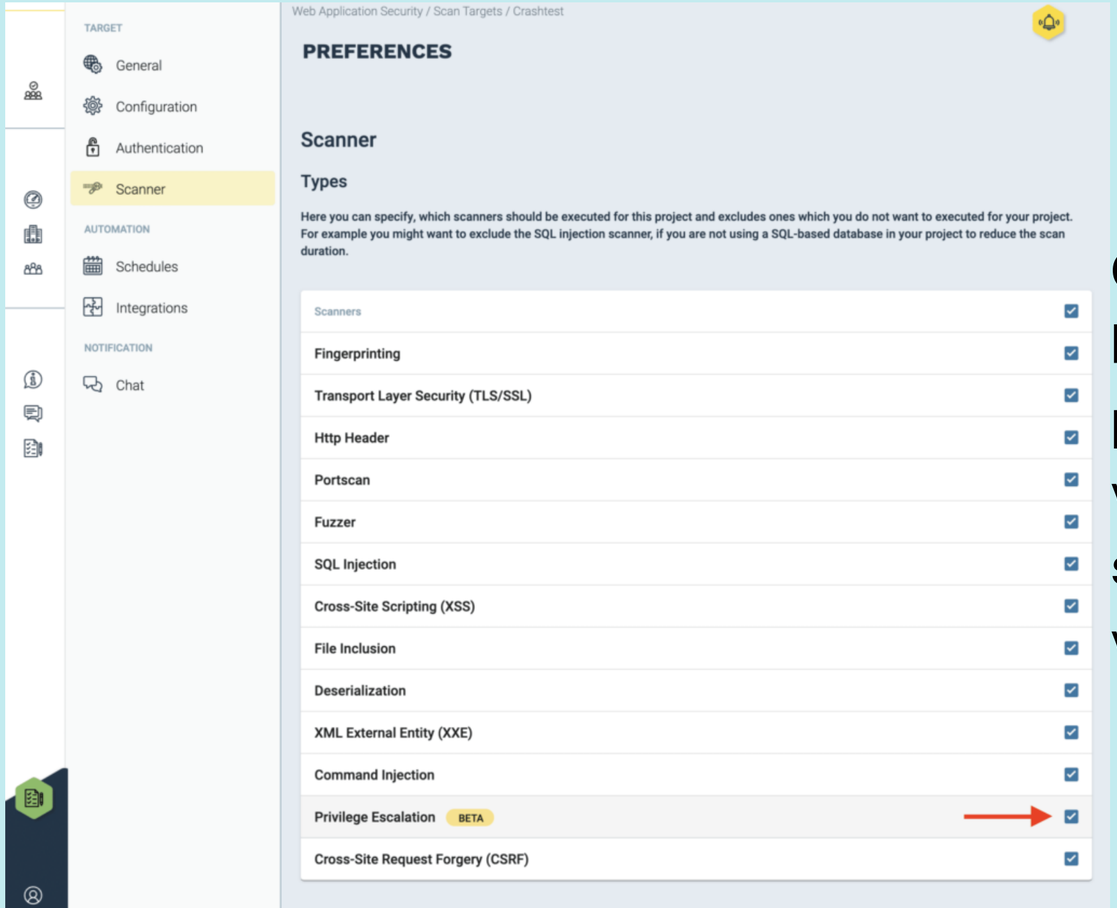


Alert

Summarize the following: Privilege Escalation Alert

Which metric does this alert monitor? Unauthorized root access attempts

What is the threshold it fires at? Triggered when unauthorized sudo command is executed.



Crashtest Security has released a privilege escalation vulnerability scanner in a beta version.

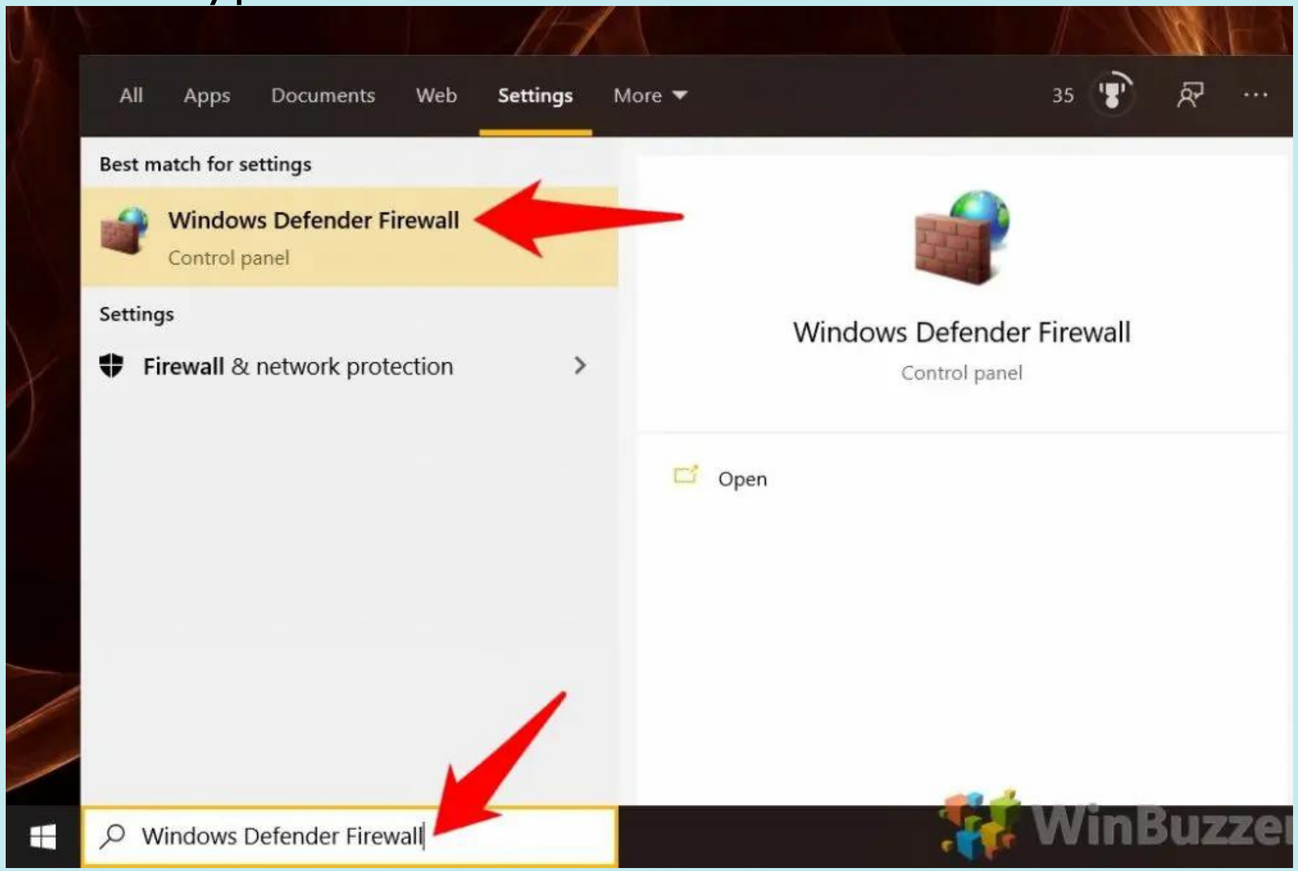
Hardening

Hardening Against Open Ports

Step 1:

Open the Windows Firewall app in Windows 10

Press the Start button and type “Windows Defender Firewall”. Click the top result to open it.

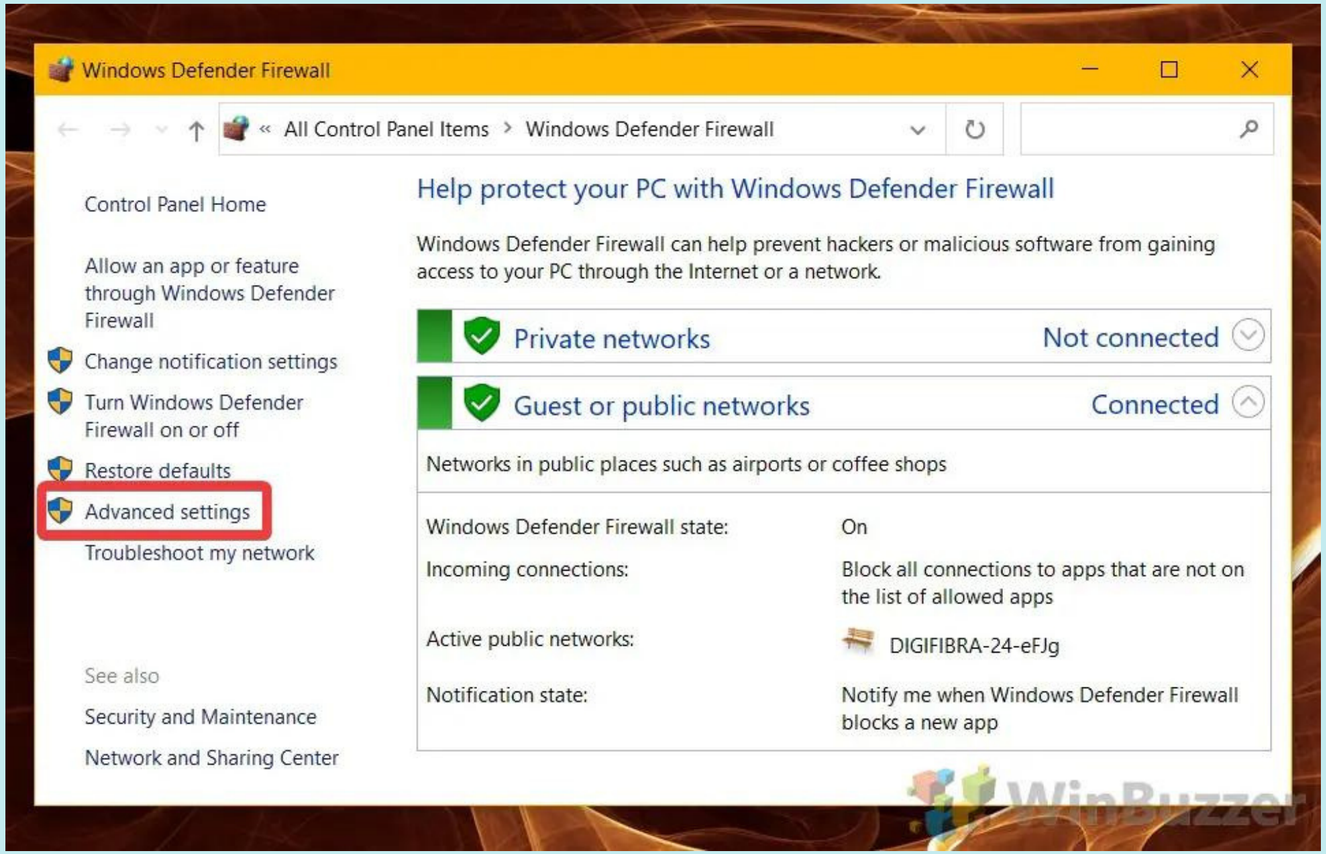


Hardening Against Open Ports

Step 2:

Open advanced iFrewall settings

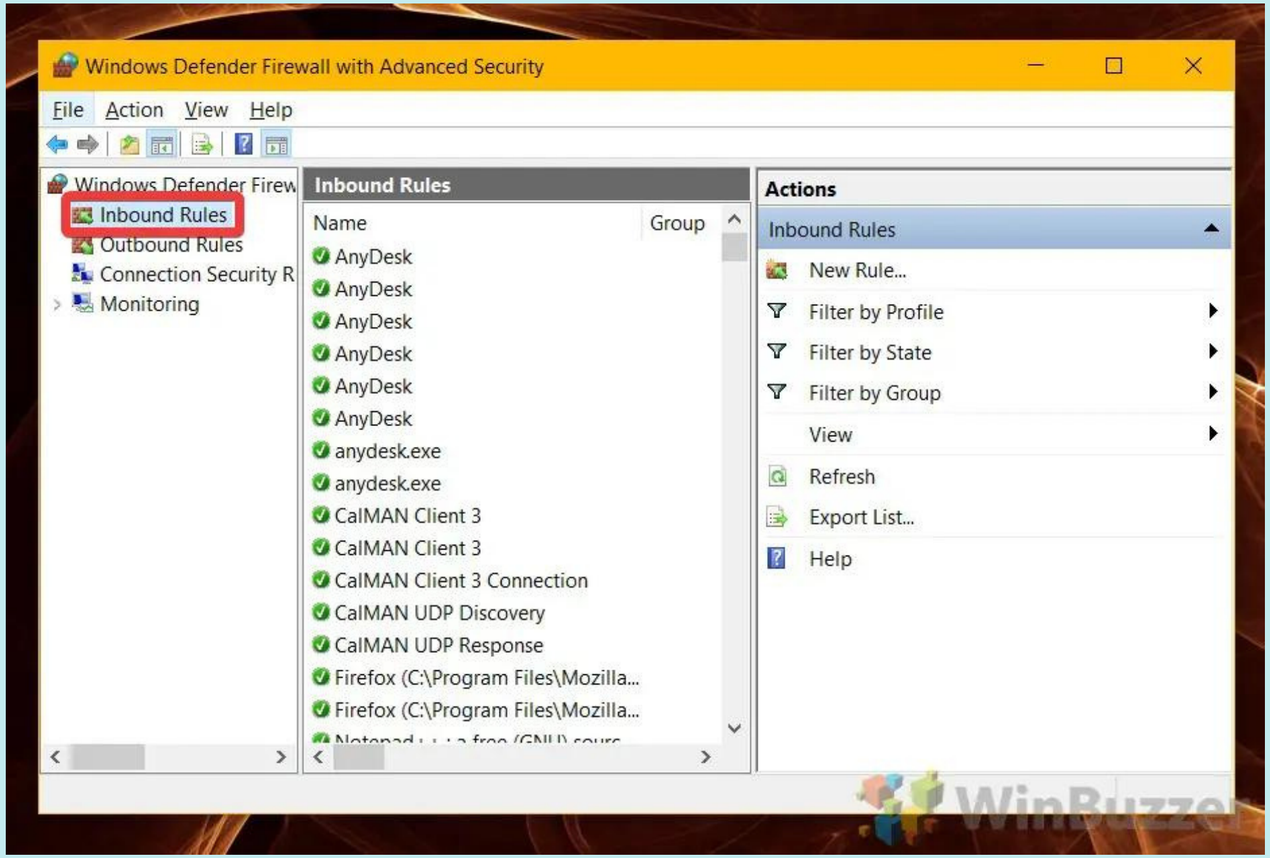
In the left side-bar of the Firewall, click the “Advanced settings” header and press “Yes” to any admin prompts.



Hardening Against Open Ports

Step 3:

Click ‘Inbound Rules’ in the sidebar

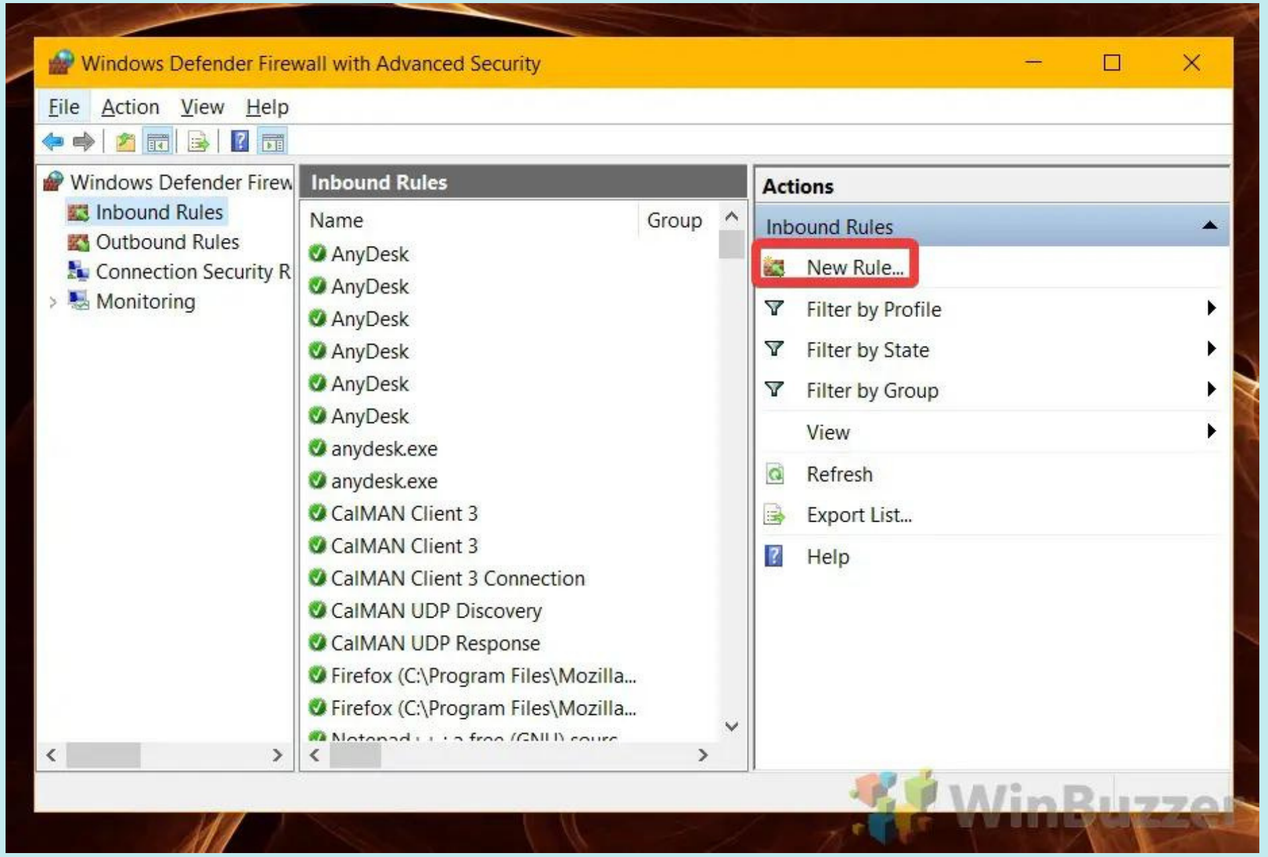


Hardening Against Open Ports

Step 4:

Press ‘New Rule…’ in the right sidebar

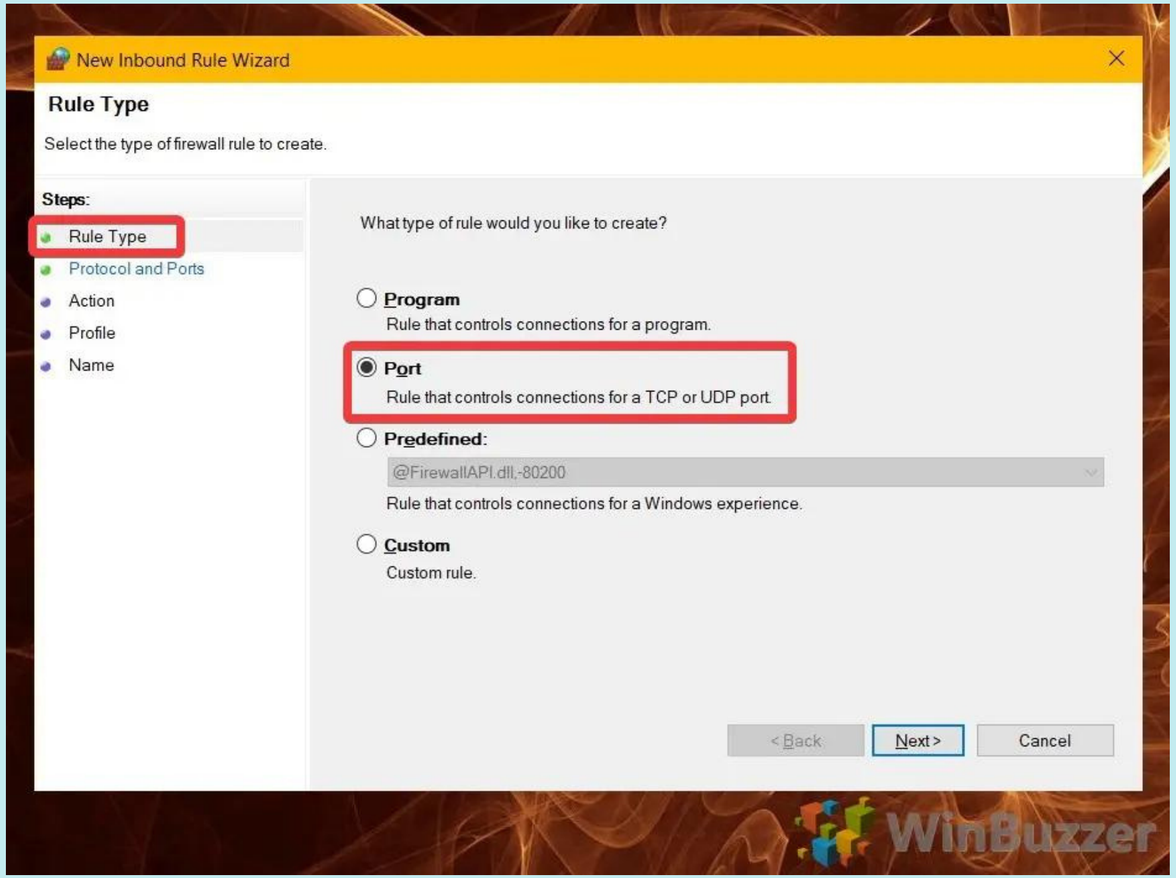
Adding this Windows Firewall Rule will let us control the traffic it does and doesn’t let through.



Hardening Against Open Ports

Step 5:

Select the ‘Port’ rule type and press ‘Next’



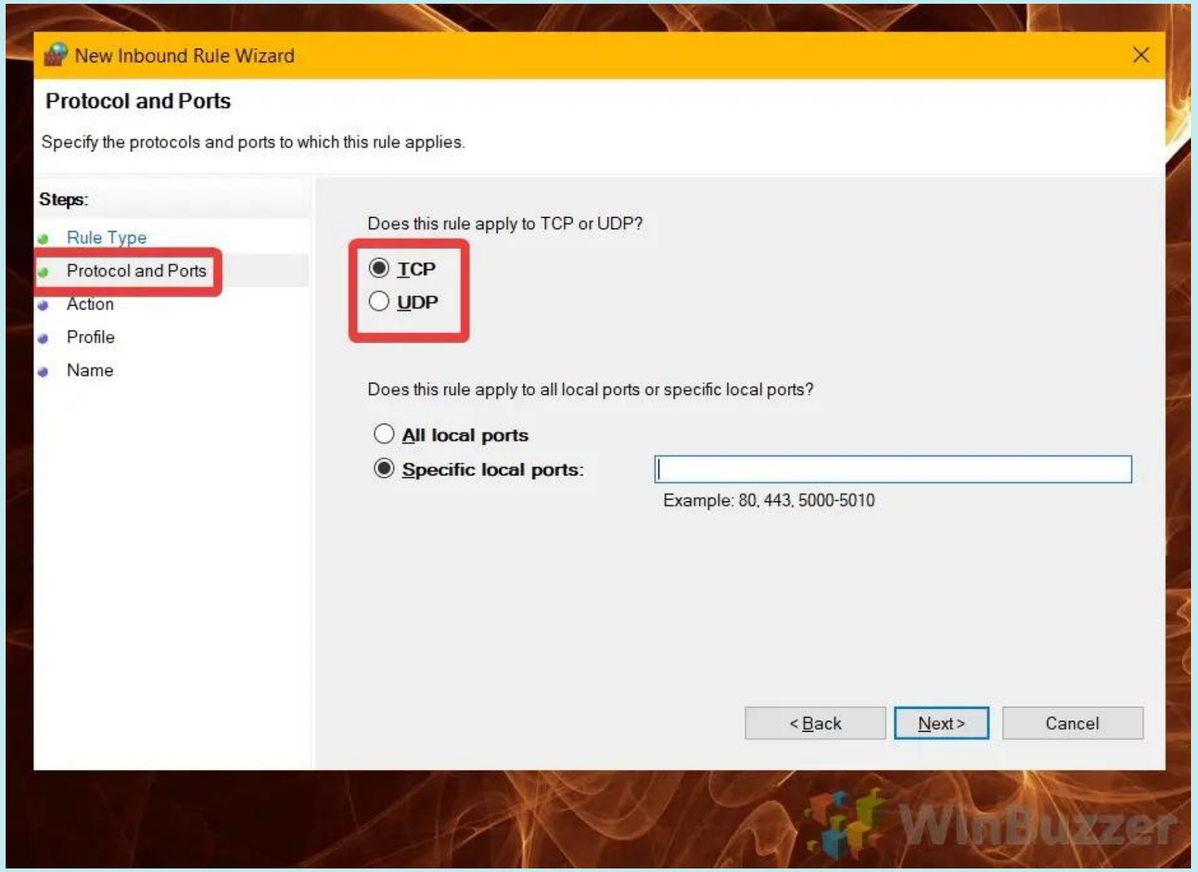
Hardening Against Open Ports

Step 6:

Choose your protocol

There are two choices wehen it comes to Windows 10 ports – TCP and UDP.

TCP is a connection-oriented protocol. Connection-orientation means that the communicating devices should establish a connection before transmitting data and should close the connection after transmitting the data.

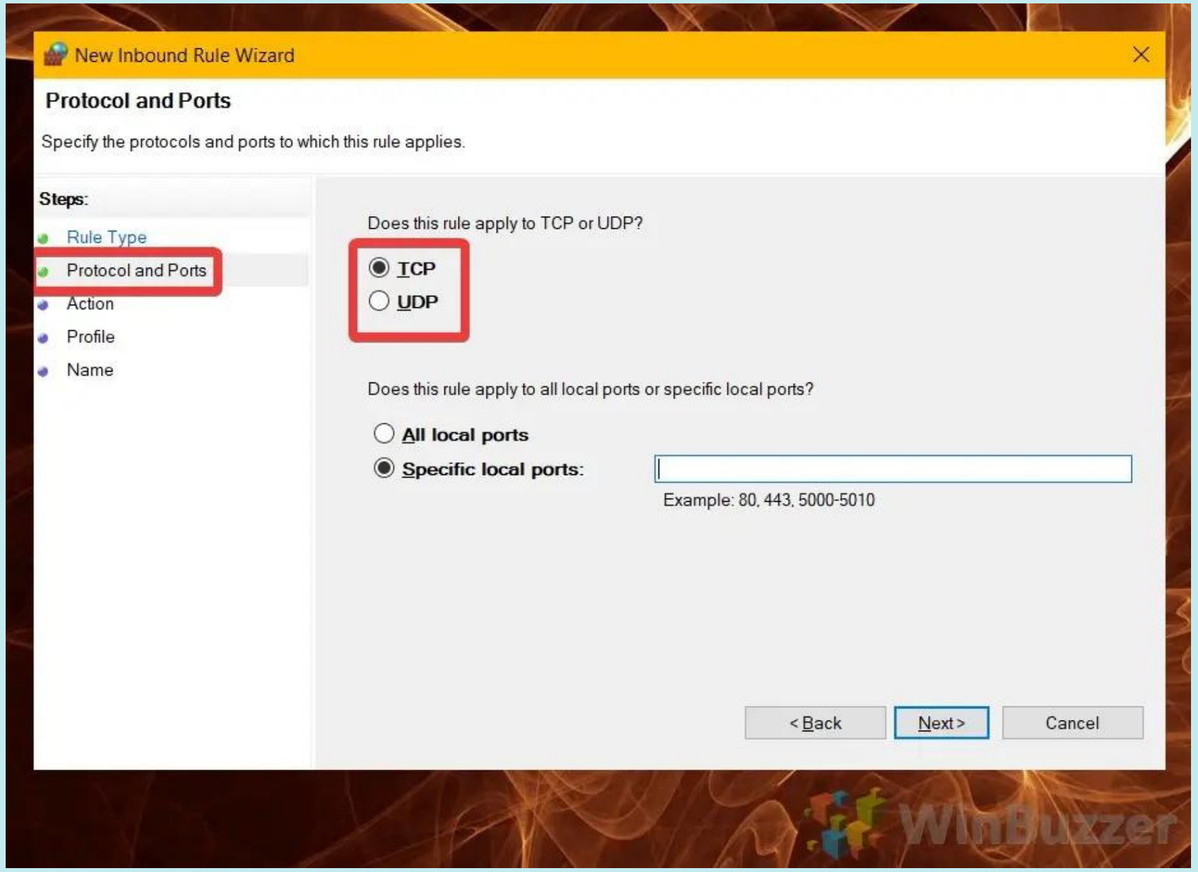


UDP is the Datagram-oriented protocol. This is because there is no overhead for opening a connection, maintaining a connection, and terminating a connection. UDP is efficient for broadcast and multicast types of network transmission.

Hardening Against Open Ports

Step 6:

Press "Next"

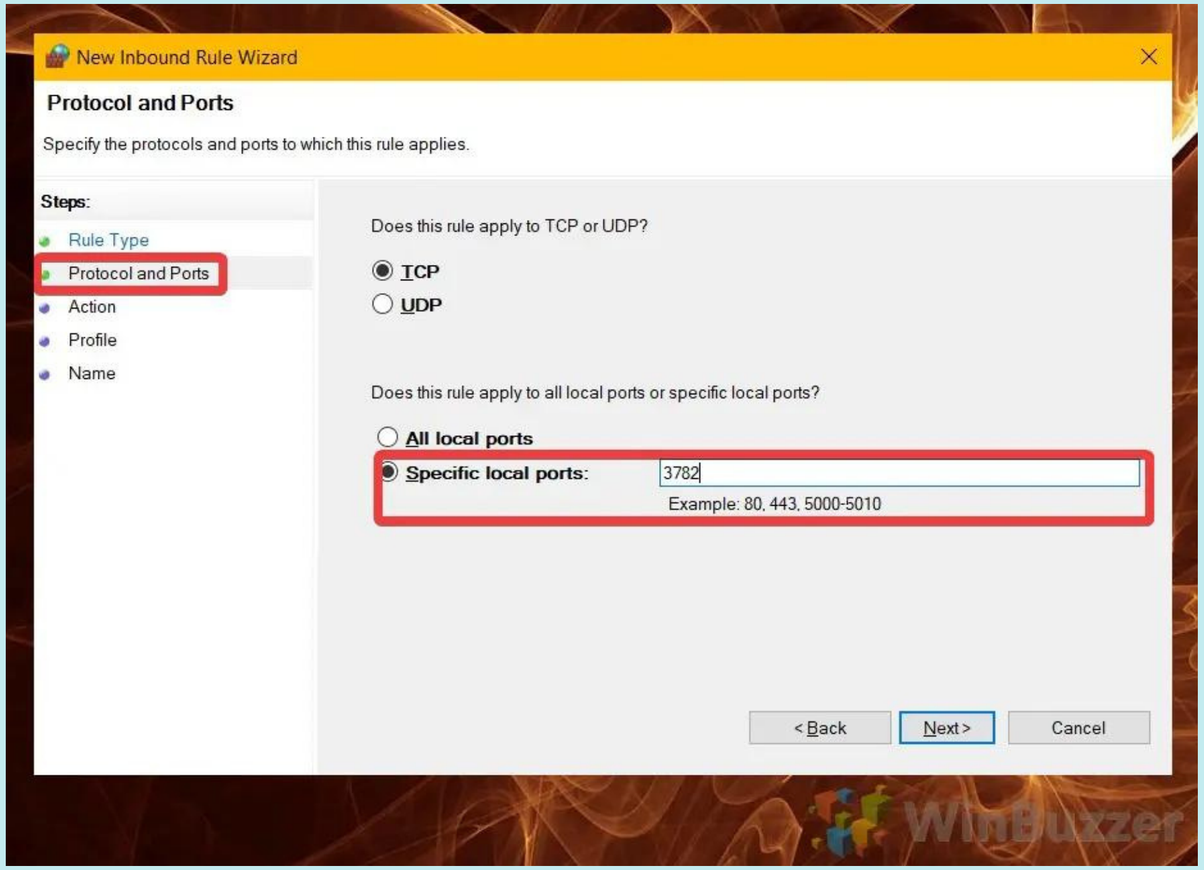


Hardening Against Open Ports

Step 7:

Enter the Windows 10 ports you want to open or close

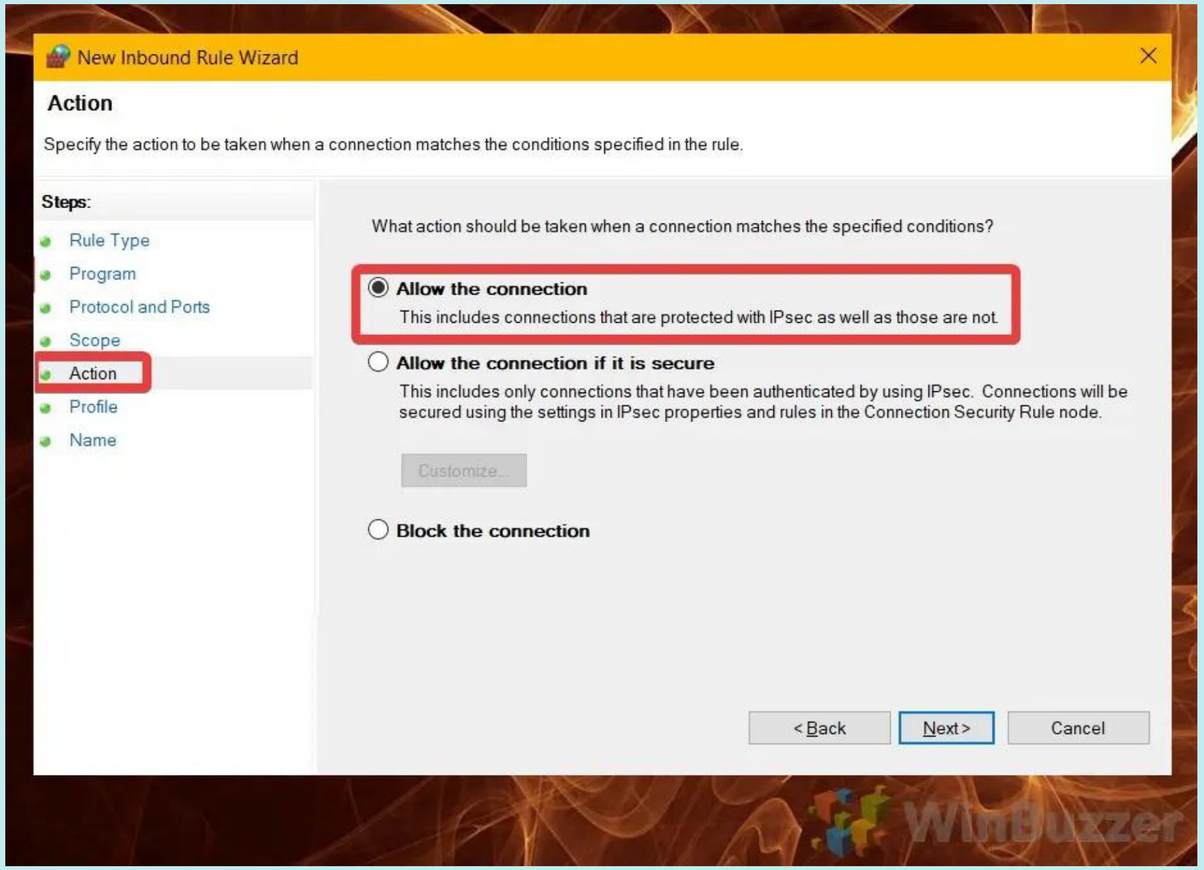
Now that you've selected your protocol, turn your eye to the "Does this rule apply to all local ports or specific local ports" heading. Check "Specific local ports" and enter the Windows 10 ports you want to open or block.



Hardening Against Open Ports

Step 8:

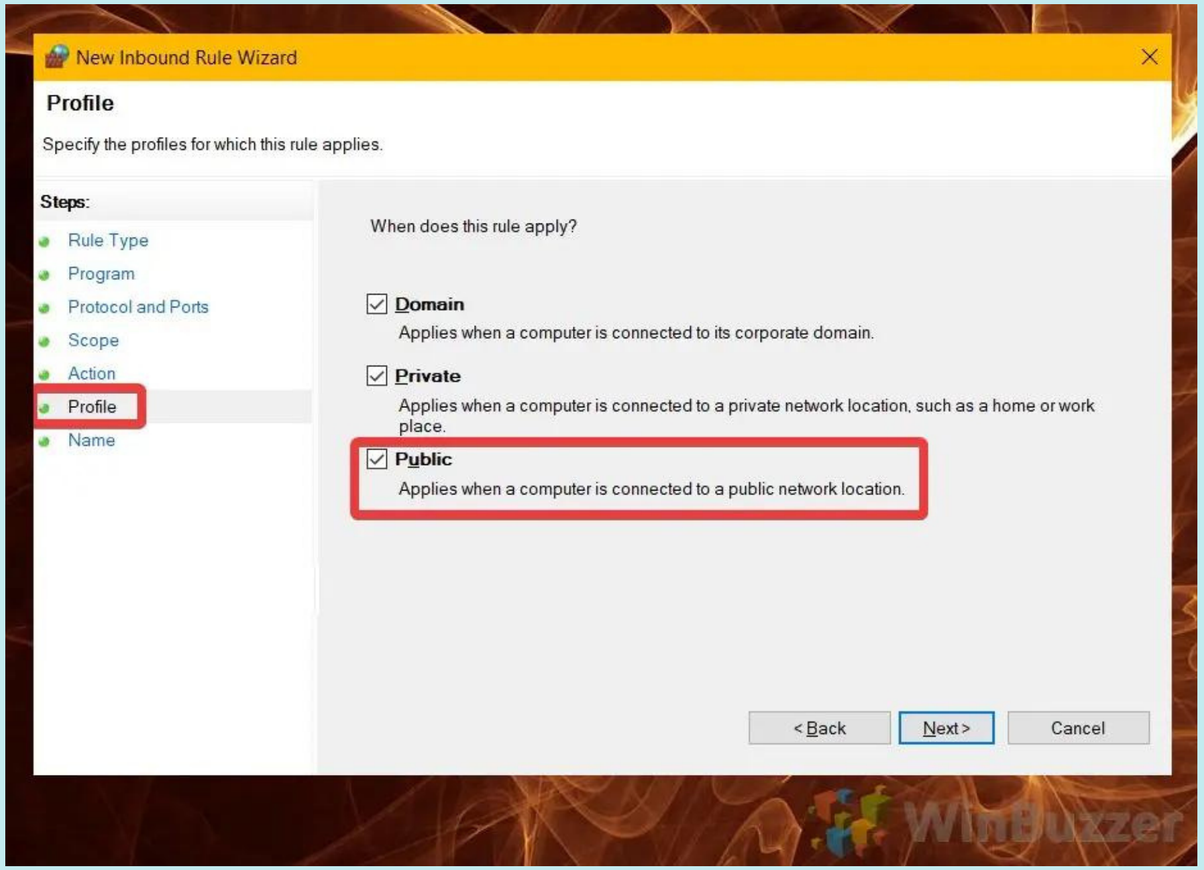
On the “Action” screen, we can choose what we want the the Windows Firewall rule to do. To let Firewall allow a port, check **“Allow the connection”**. Alternatively, block a port by checking **“Block the connection”**. Click **“Next”**.



Hardening Against Open Ports

Step 9:

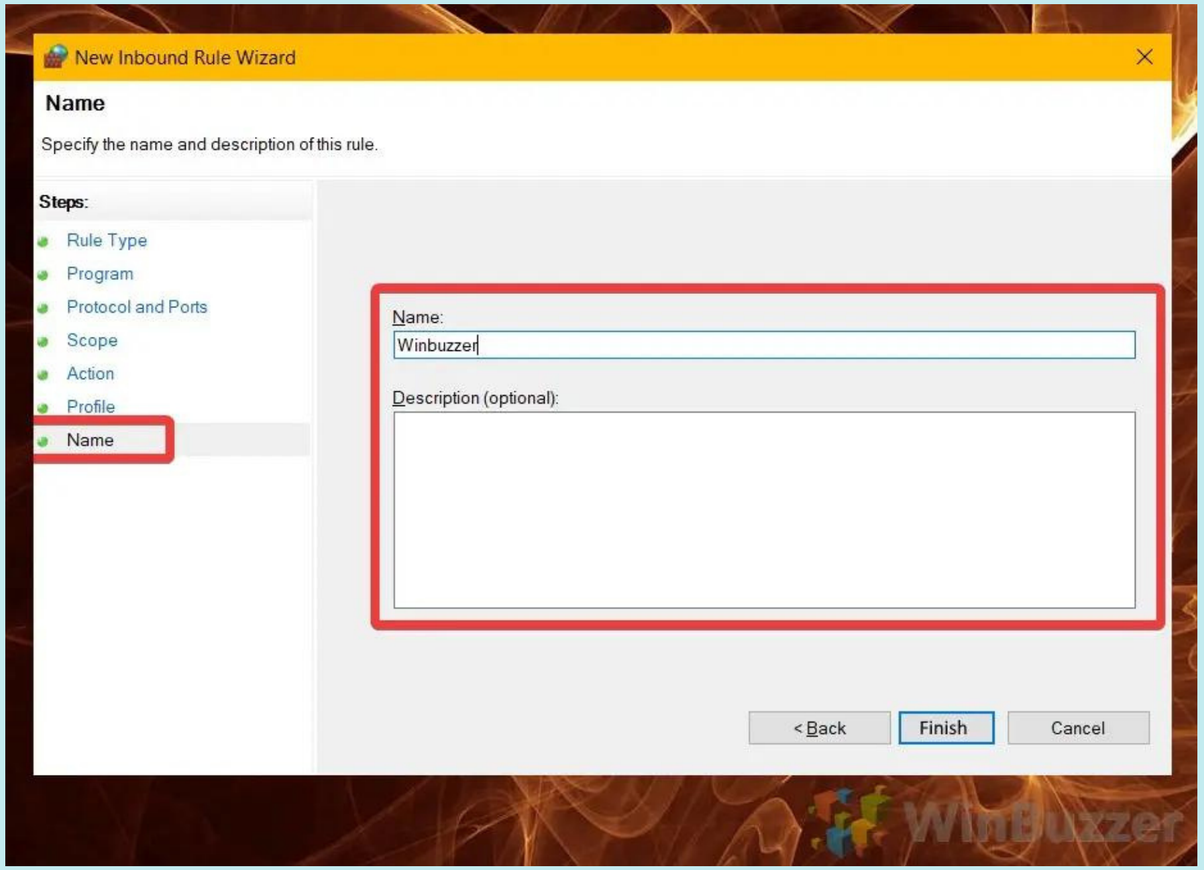
Choose where the Firewall rule applies Tick “Domain” to apply the rule at work, “Private” at home, and “Public” for shared networks. You may want to check what type your current network is by running the Get-NetConnectionProfile command in PowerShell. Click **“Next”** when you’re done..



Hardening Against Open Ports

Step 10:

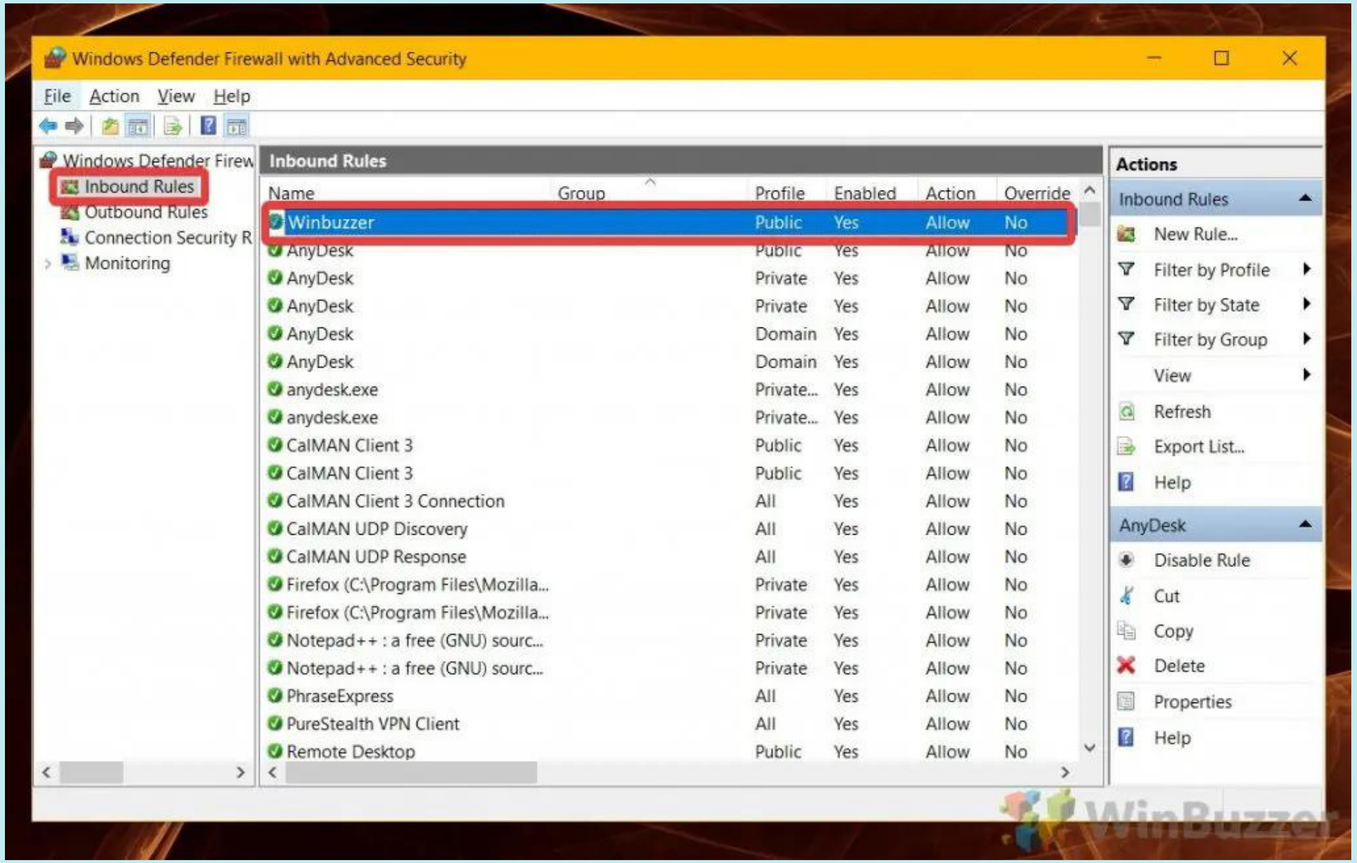
Make sure you're descriptive with this so you can find it later. In the description, you may want to enter why you opened the port. Press "Finish".



Hardening Against Open Ports

Step 11:

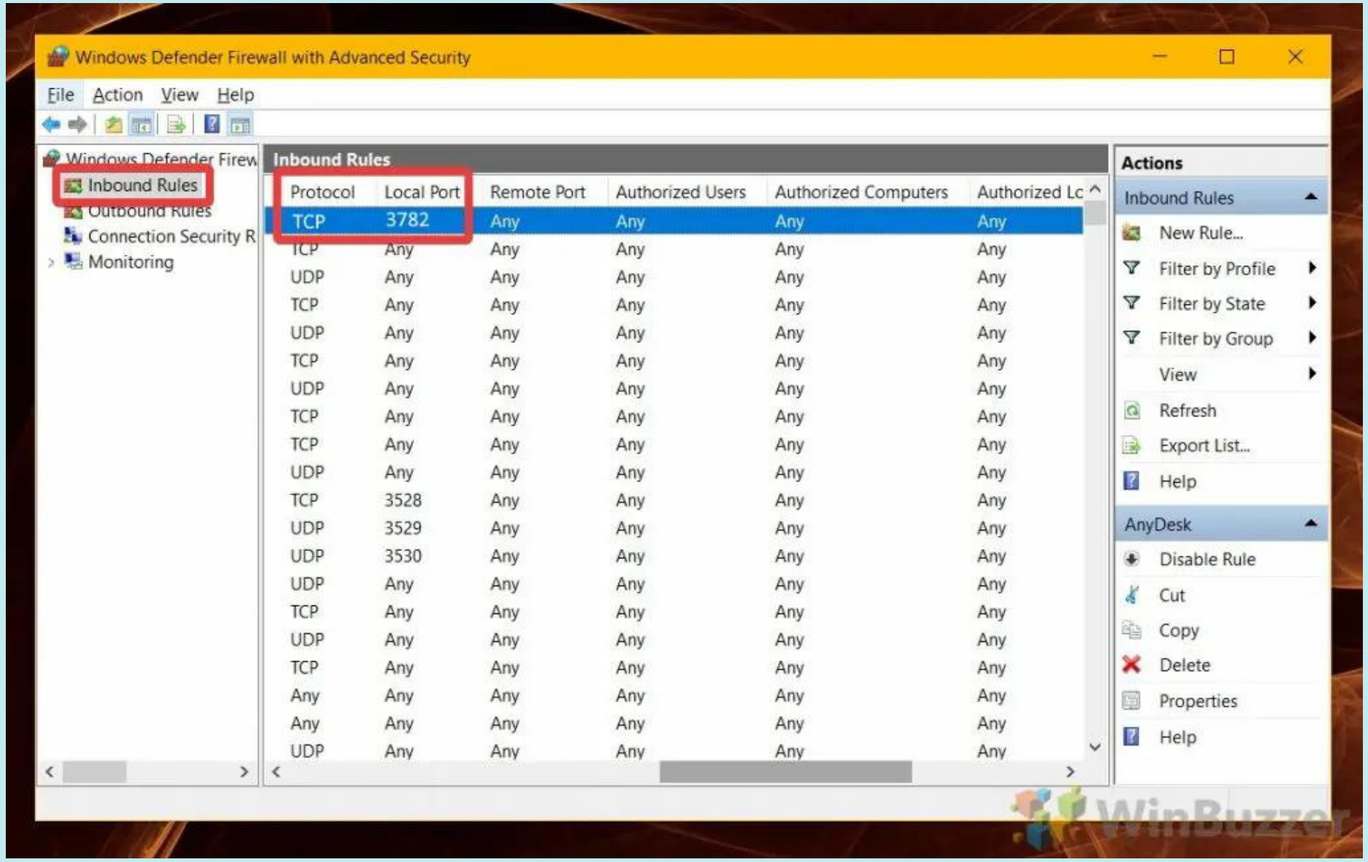
Back in the main Windows Defender Firewall screen, you should now see your new rule at the top of the Inbound Rules page. The “Profile” column will tell you the type of network it’s allowed on and “Action” whether it’s enabled.



Hardening Against Open Ports

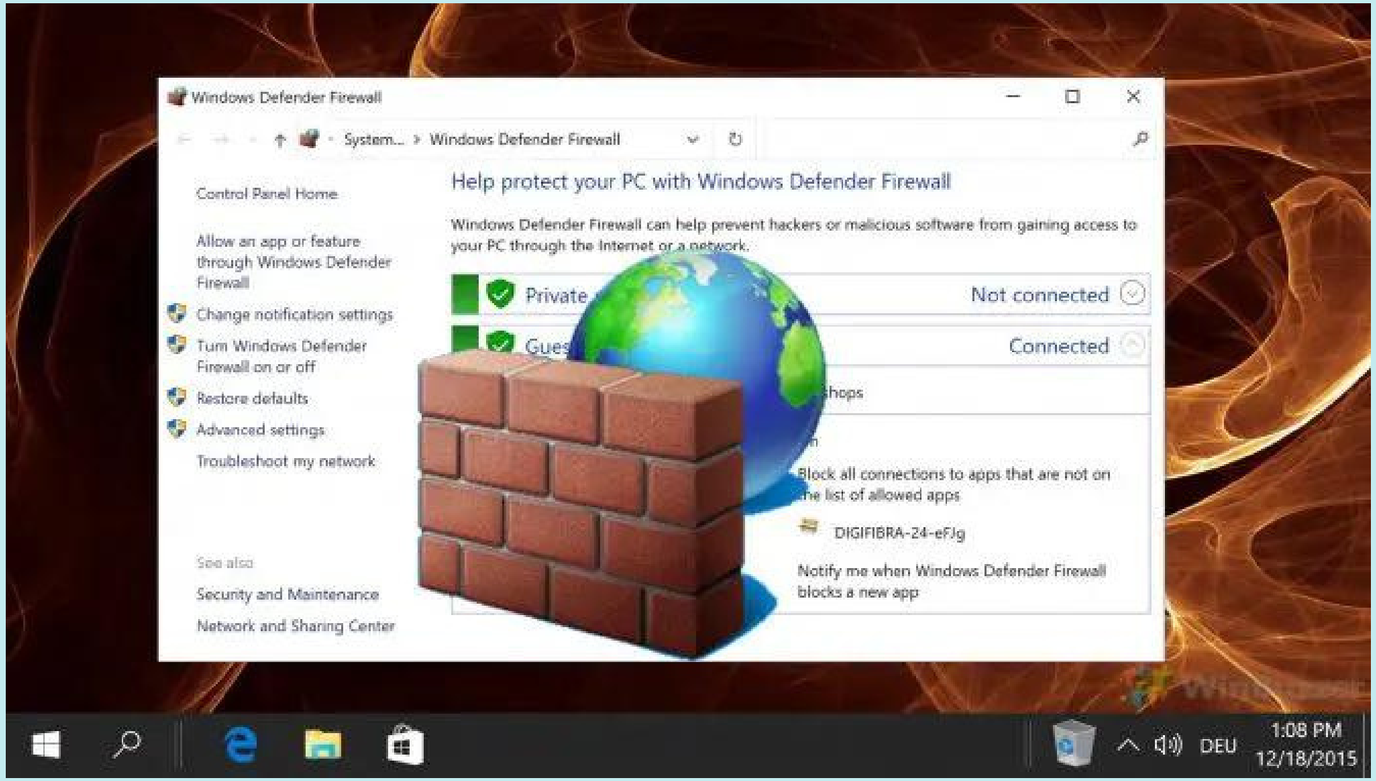
Step 12:

Move the horizontal scroll bar at the bottom of the screen until you see the “Protocol” and “Local Port” columns. Double-check them to ensure you opened or closed the correct port. Adding the wrong port could weaken your security or cause connection issues elsewhere.



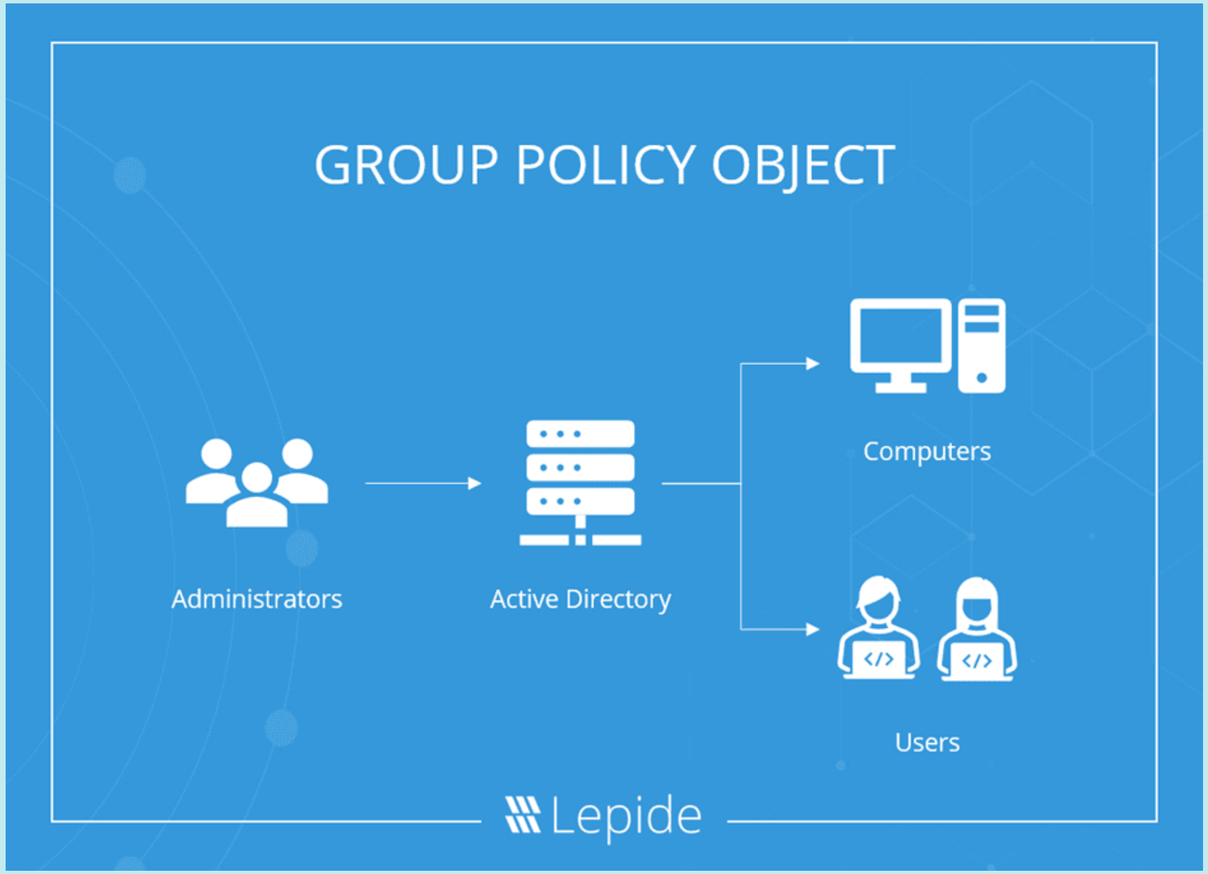
Hardening Against Open Ports

Open ports on a server are a security vulnerability that can potentially allow a hacker to exploit services on your network. If those services are unpatched, a hacker can easily take advantage of the system by running a simple port scan using free software like nmap to discover the open ports.



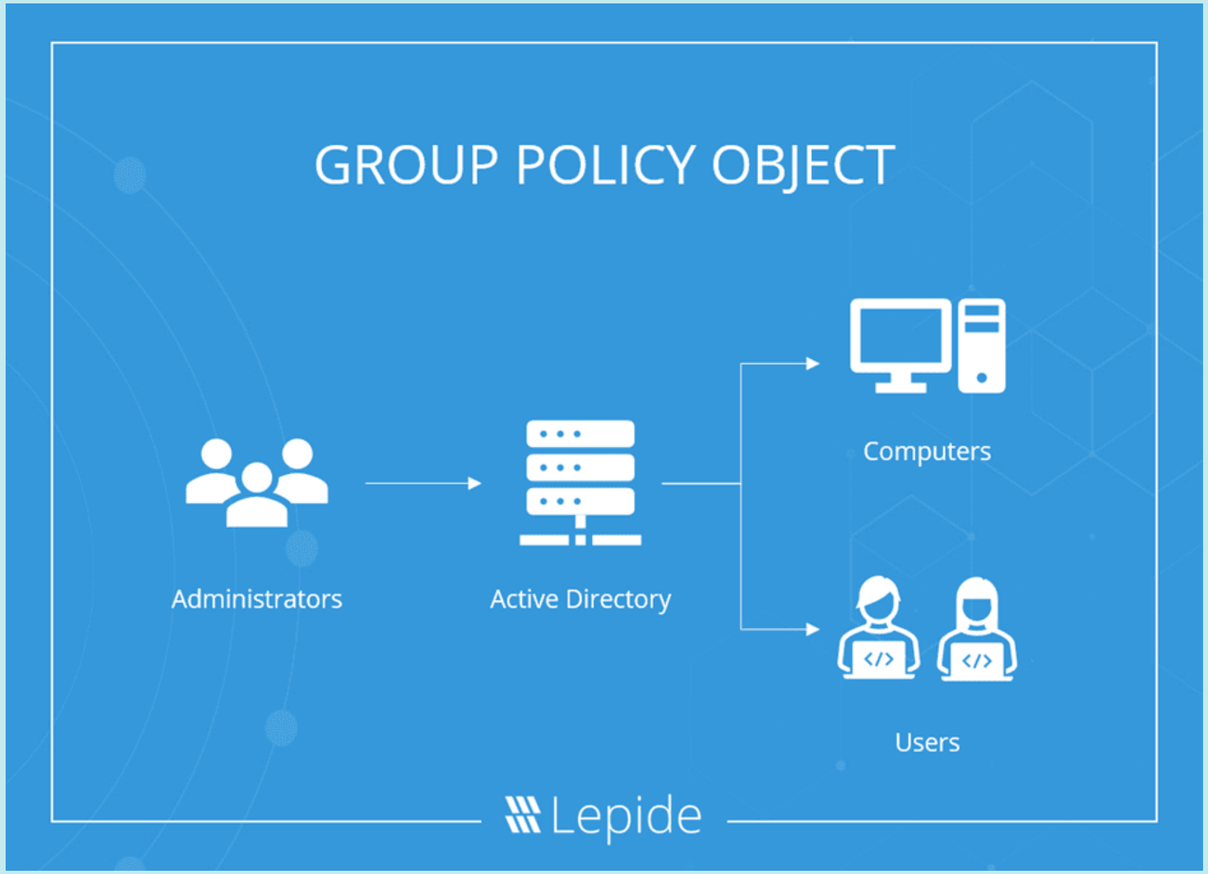
Hardening Against Weak Credentials

Having strong passwords in an Active Directory (AD) network ensures that hackers can’t crack user’s passwords with methods such as brute-force dictionary attacks.



Hardening Against Weak Credentials

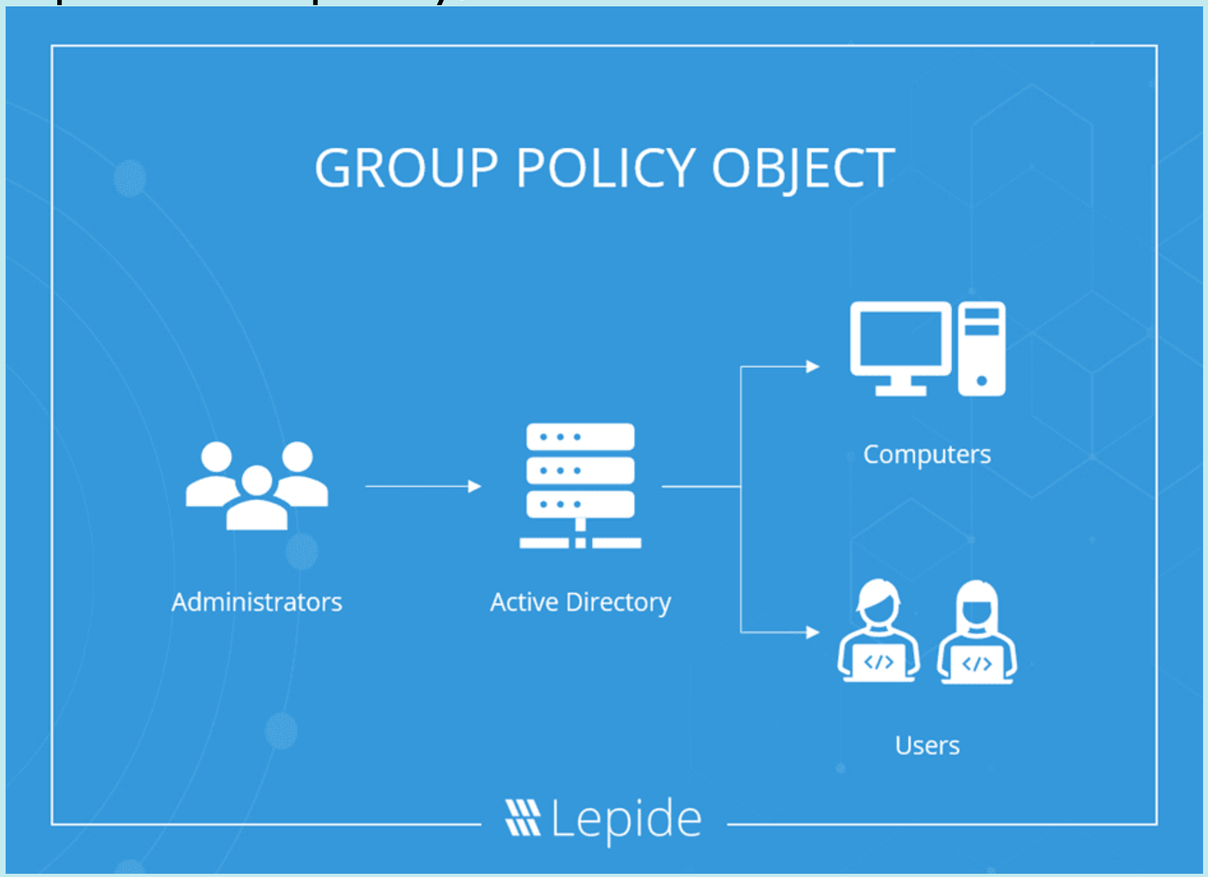
The Active Directory domain comes with the “Default Domain Password Policy,” which helps to improve security through password hardening. The policy is intended to enforce passwords to have enough complexity, to be longer than usual, and to expire after some time.



Hardening Against Weak Credentials

Steps:

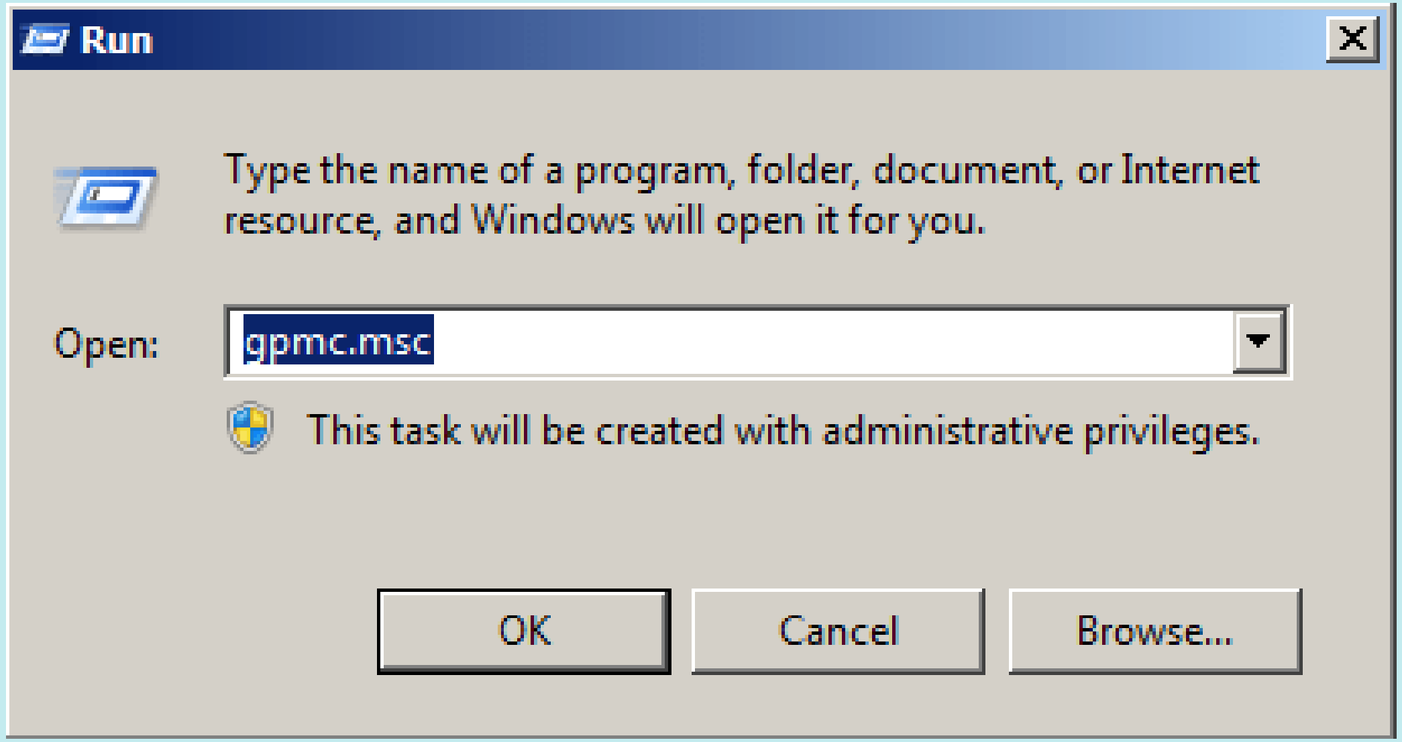
Password policies are associated with the root domain and are configured through a group policy. To view the current AD domain password policy,



Hardening Against Weak Credentials

Step: 1

Open the Group Policy Management console using the “gpmc.msc” command



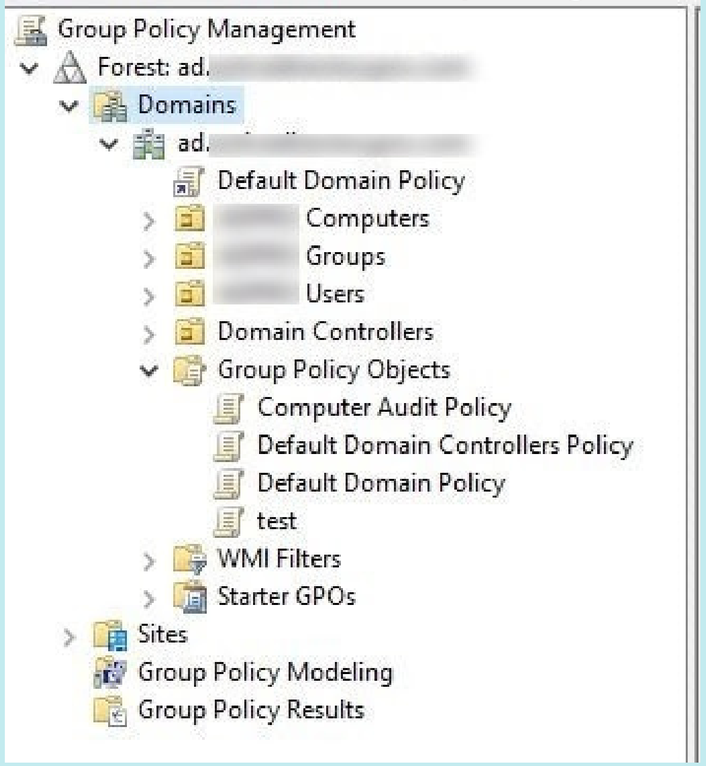
Hardening Against Weak Credentials

Step: 2

The domain password policy is under Group Policy Objects (GPO).

Browse through the right-hand window pane, expand your Domains, and then open the Group Policy Objects.

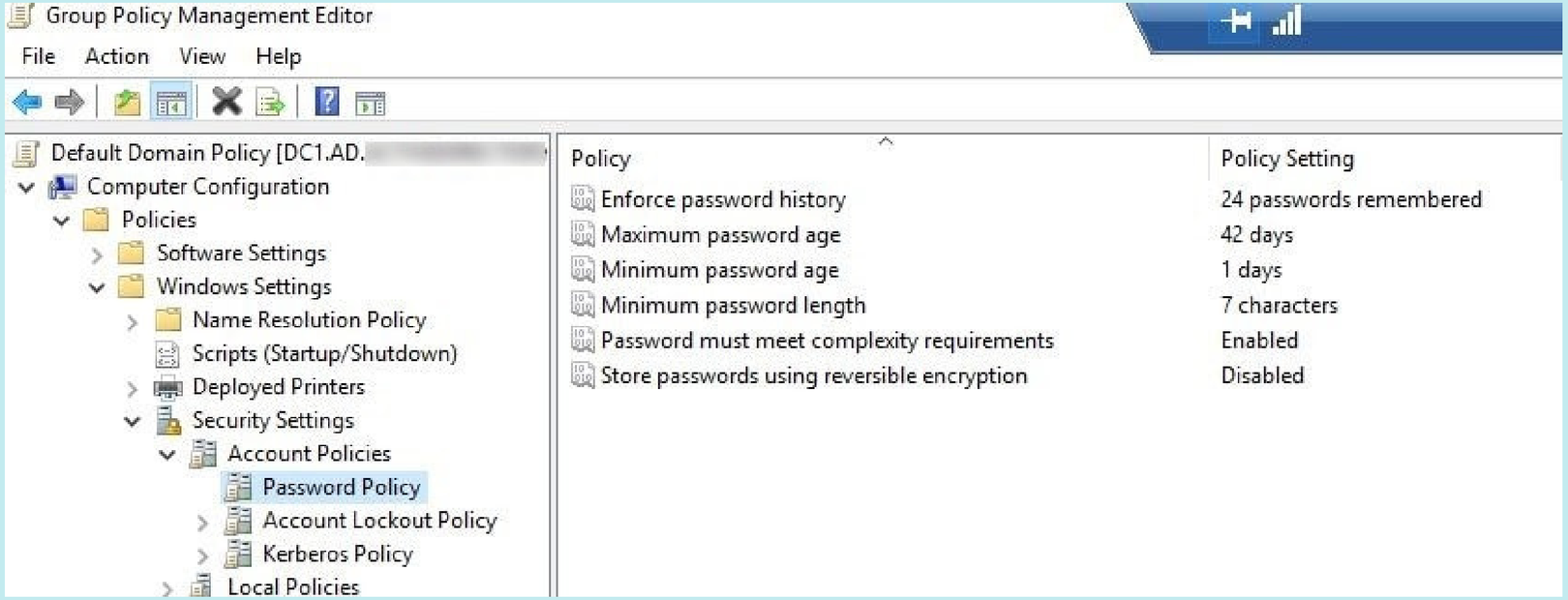
Find the GPO with the name “Default Domain Policy,” right-click it, and select Edit



Hardening Against Weak Credentials

Step: 3

The Group Policy Management Editor will let you view and configure the password requirements. Find the “Password Policy” node, which is under the Computer Configuration\Policies\Windows Settings\Security Settings\Account Policies\Password Policy

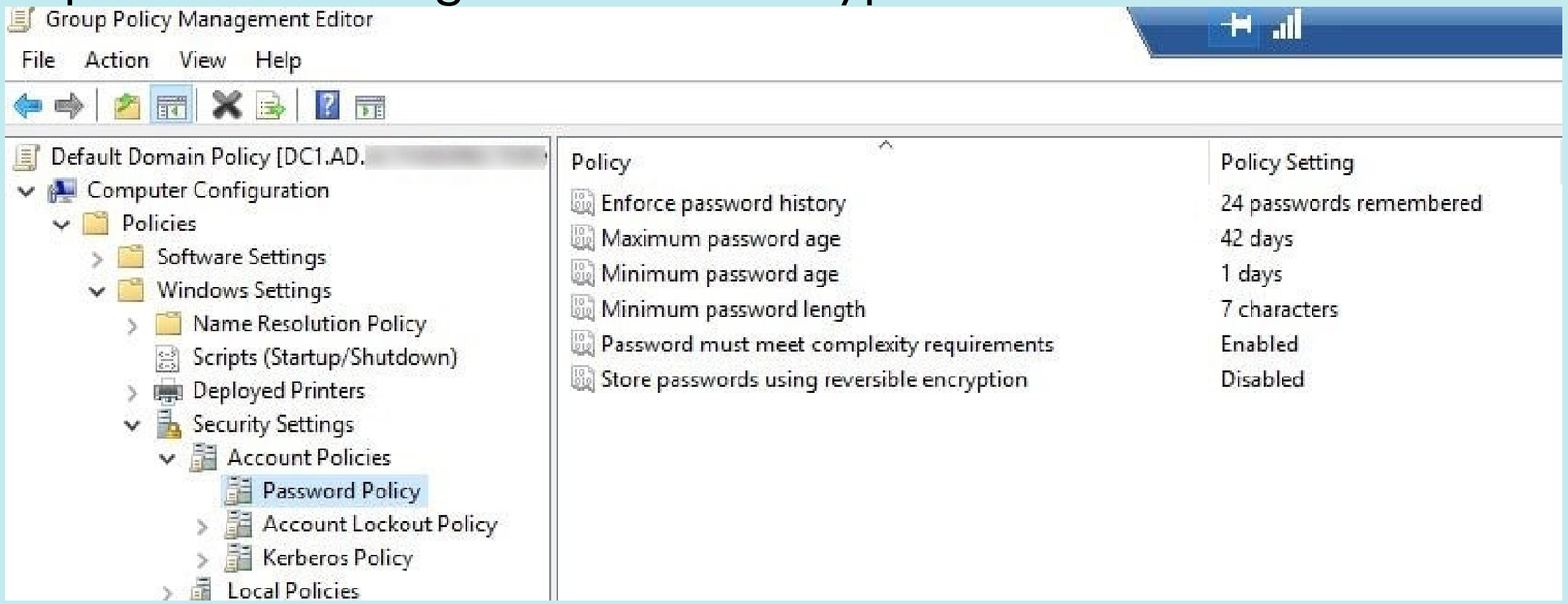


Hardening Against Weak Credentials

Step: 3

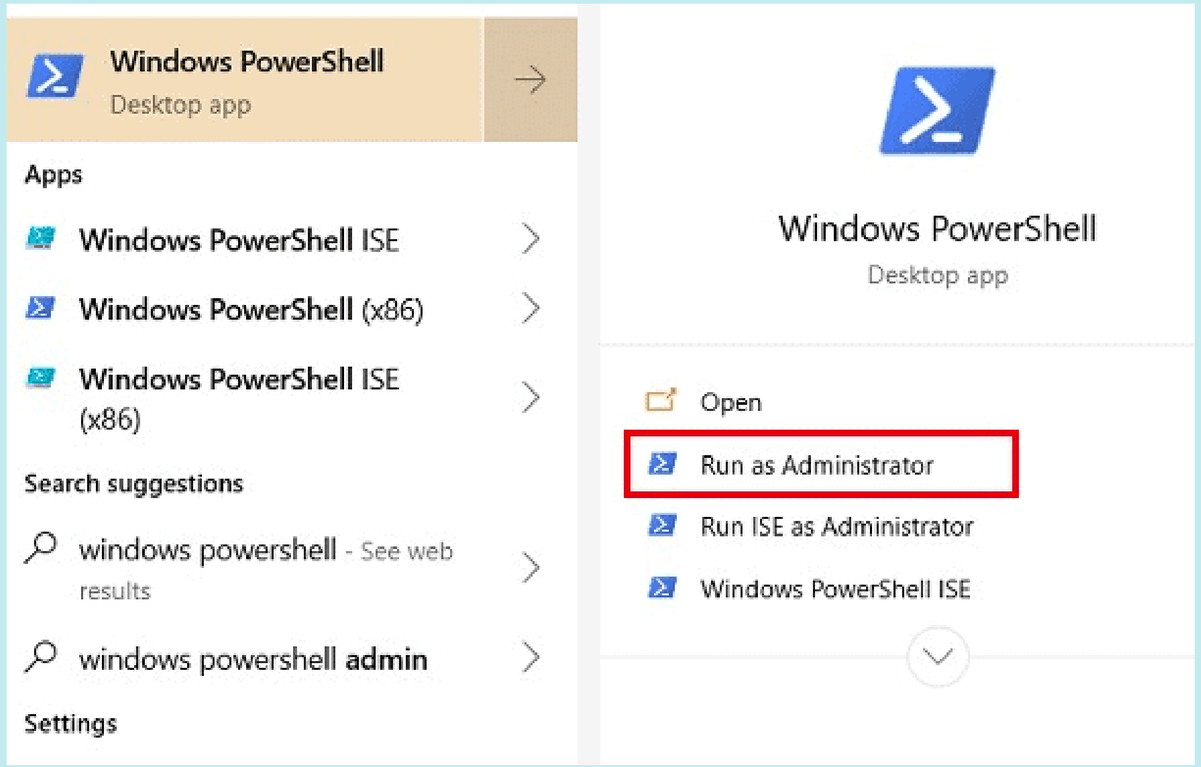
The Password Policy node that is shown here has the default configuration, which is:

- Enforce password history: Remember the last 24 passwords.
- Maximum password age: Expire after 42 days.
- Minimum password age: One day.
- Minimum password length: Seven Characters.
- Password must meet complexity requirements.
- Do not store passwords using reversible encryption.



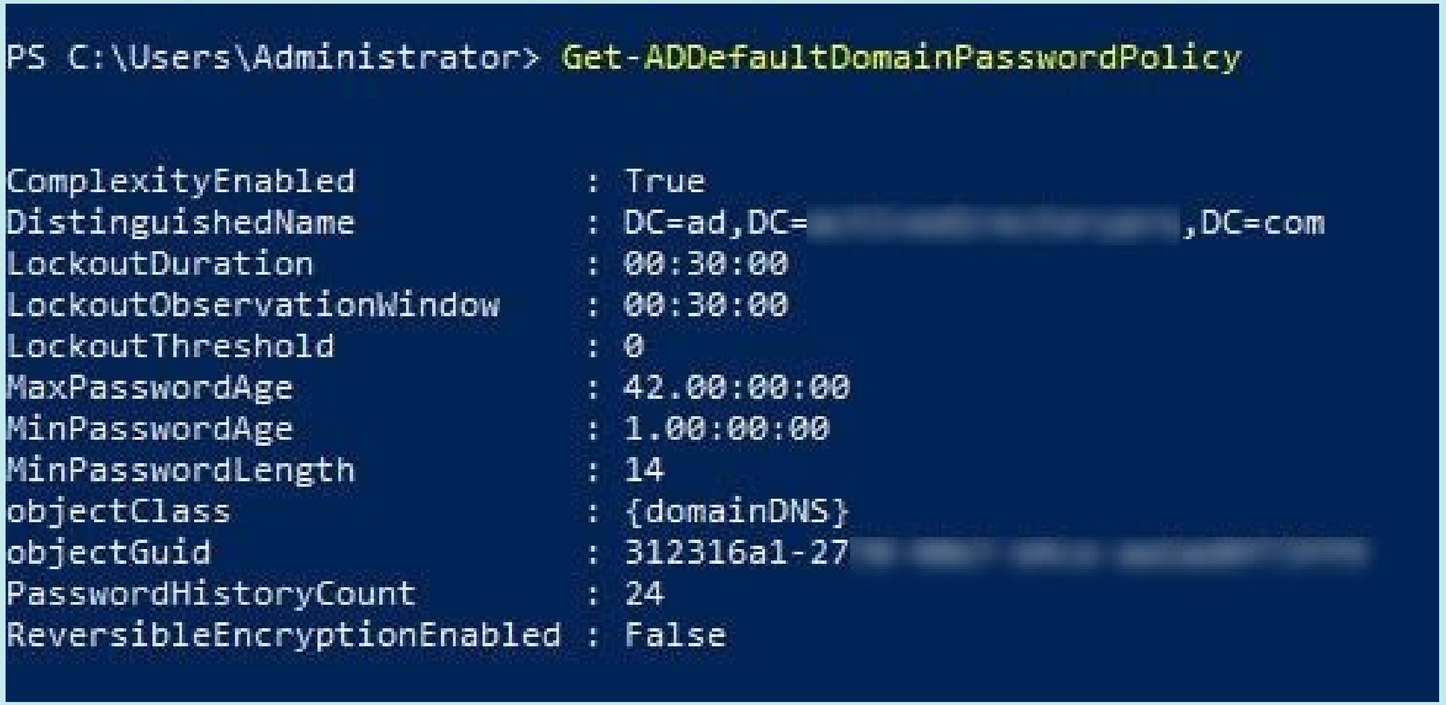
Hardening Against Weak Credentials

There is another way to get the default password policy for the Active Directory domain using the Powershell, command-line shell. Open the Windows PowerShell command line as an administrator.



Hardening Against Weak Credentials

Enter the command: [Get-ADDefaultDomainPasswordPolicy]. The command output will show you the current Password Policy along with the Lockout Policy settings.



Hardening Against Weak Credentials

The Password Policy Settings

There are six different password policies that you can configure.

1. Enforce Password History

The “Enforce Password History” specifies the number of previous passwords stored in Active Directory.

The setting enforces users to create unique and new passwords by preventing them from reusing old passwords too often.

The default value of this setting is 24, which means that the user won’t be able to use the current password only after 24 new passwords have already been used.

Hardening Against Weak Credentials

The Password Policy Settings

There are six different password policies that you can configure.

2. Maximum Password Age

The “Maximum Password Age” defines the number of days that a password can be used before it needs to be renewed.

It gives the password, an expiration date.

Once the password has reached its maximum password age, the system will request a password change.

Hardening Against Weak Credentials

The Password Policy Settings

There are six different password policies that you can configure.

3. Minimum Password Age

The “Minimum Password Age” determines the period in days that the password should be used before users need to change it.

This setting helps the “Enforce Password History” by denying users to change passwords too often and quickly, to get back to an old password.

It is good to configure this value but keep it to a minimum, in case the password gets compromised. The default value is one day.

Hardening Against Weak Credentials

The Password Policy Settings

There are six different password policies that you can configure.

4. Minimum Password Length

The Minimum Password Length determines the number of characters in the password.

The default for this setting is seven, which means that all passwords will have to be created with at least seven characters.

Be careful! If you set this policy with a value of zero, the password will not even be required.

Hardening Against Weak Credentials

The Password Policy Settings

There are six different password policies that you can configure.

5. Password Must Meet Complexity Requirements

You can only Enable or Disable this setting.

When you enable it, you'll require users to create complex passwords based on certain guidelines. This setting is enabled by default.

The Requirements Are:

- Users can't use the account name or name of the user anywhere on the password. It is not allowed to use two characters of the name in a row.
- Users must include three different types of characters from any of the numbers (0–9), uppercase and lowercase letters, and non-alphabetic characters (@, $, &, #).
- The password must be at least six characters in length.

Hardening Against Weak Credentials

The Password Policy Settings

There are six different password policies that you can configure.

6. Store Passwords Using Reversible Encryption

Active Directory encrypts passwords and stores them in the database.

Usually, the encrypted passwords can’t be reversed to “plain-text.” But in some cases, users will need to use their passwords with certain apps to gain access to the domain.

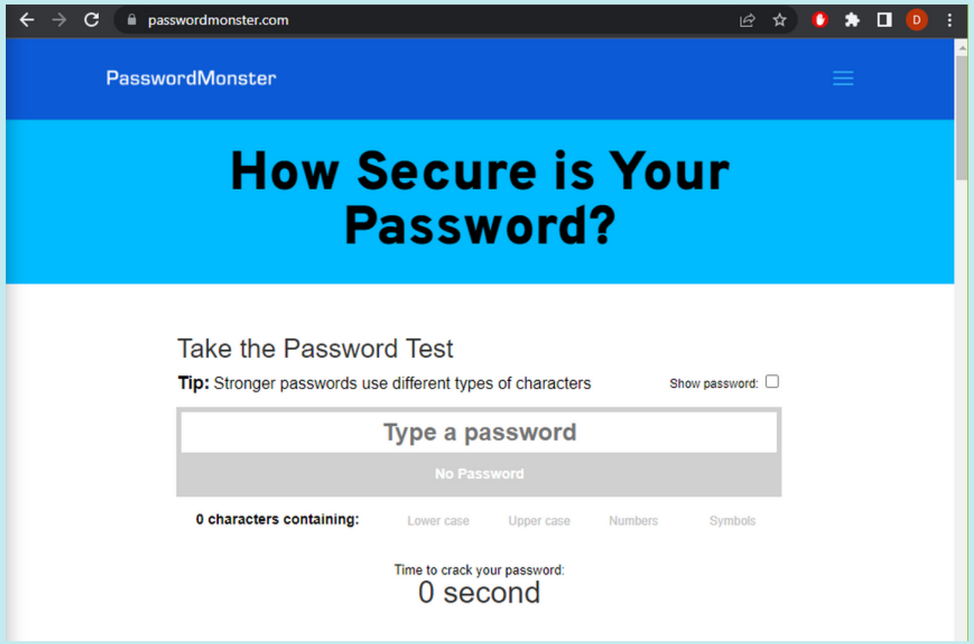
Most likely, the app won’t be able to decrypt the password, so you’ll need to enable the Reversible Encryption.

It is not recommended to enable this setting unless there is a specific need for the application. If you need to enable it, do it on a per-user basis.

Hardening Against Weak Credentials

Users are usually the easiest target within a domain network.
The account username and password might be the only security measures protecting their computers. The username might be easy to guess, but passwords shouldn’t be acceptable to be weak. They should be complex and difficult to guess.

Use this webstie to test your password strength:



Hardening Against SQL Injection

Preventing SQL Injection vulnerabilities is not easy. Specific prevention techniques depend on the subtype of SQLi vulnerability, on the SQL database engine, and on the programming language. However, there are certain general strategic principles that you should follow to keep your web application safe.

Hardening Against SQL Injection

Step 1: Validate Input

Train and maintain awareness

To keep your web application safe, everyone involved in building the web application must be aware of the risks associated with SQL Injections. You should provide suitable security training to all your developers, QA staff, DevOps, and SysAdmins. You can start by referring them to this page.

Hardening Against SQL Injection

Step 2: Prepare a Query

Don't trust any user input

Treat all user input as untrusted. Any user input that is used in an SQL query introduces a risk of an SQL Injection. Treat input from authenticated and/or internal users the same way that you treat public input.

Hardening Against SQL Injection

Step 3: Create prepared statement

Use whitelists, not blacklists

Don’t filter user input based on blacklists. A clever attacker will almost always find a way to circumvent your blacklist. If possible, verify and filter user input using strict whitelists only.

Hardening Against SQL Injection

Step 4: Bind the parameters

Adopt the latest technologies

Older web development technologies don’t have SQLi protection. Use the latest version of the development environment and language and the latest technologies associated with that environment/language. For example, in PHP use PDO instead of MySQLi.

Hardening Against SQL Injection

Step 5: Execute query

Employ verified mechanisms

Don’t try to build SQLi protection from scratch. Most modern development technologies can offer you mechanisms to protect against SQLi. Use such mechanisms instead of trying to reinvent the wheel. For example, use parameterized queries or stored procedures.

Hardening Against SQL Injection

Step 6: Fetch Result

Scan regularly

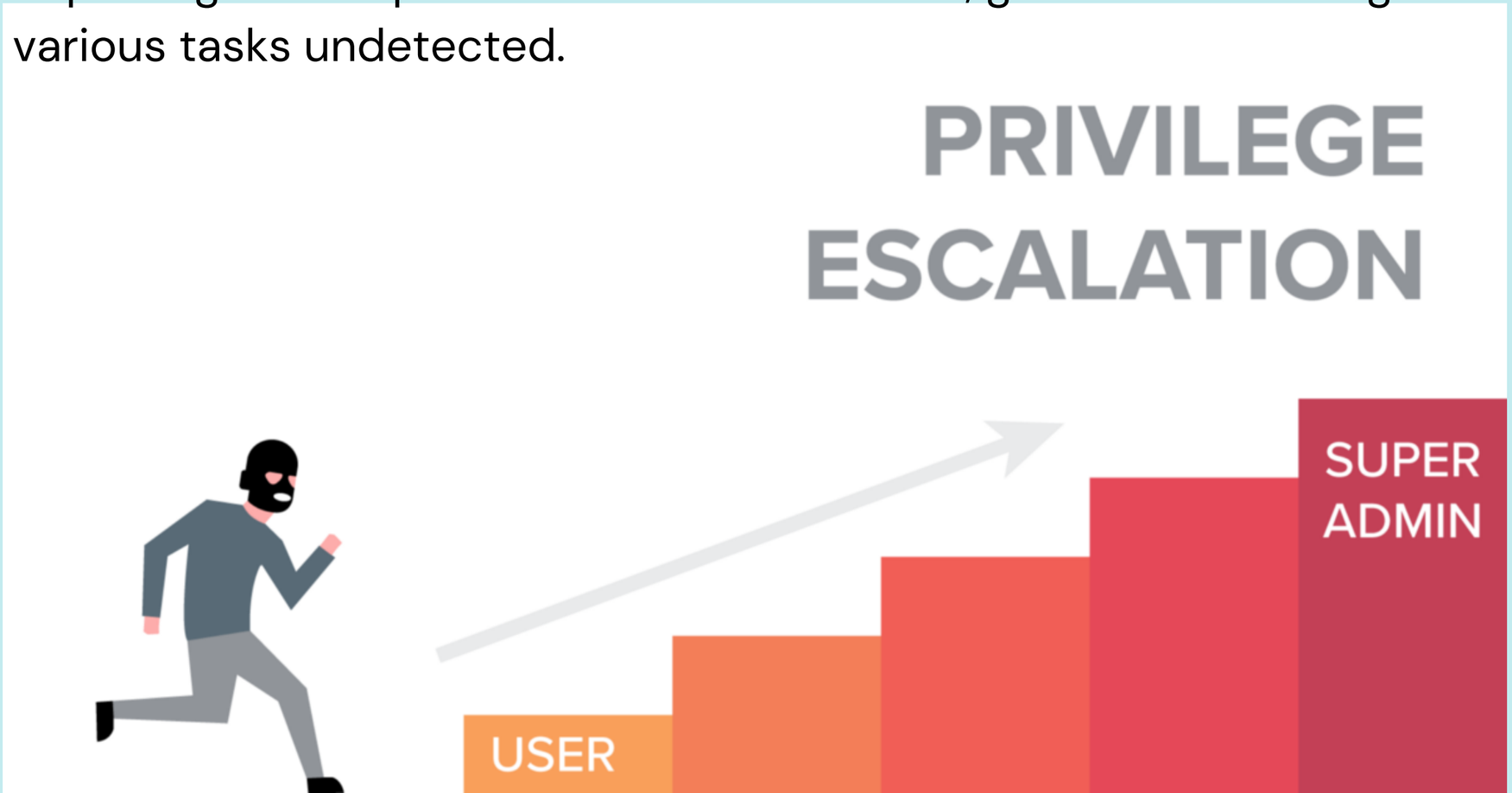
SQL Injections may be introduced by your developers or through external libraries/modules/software. You should regularly scan your web applications using a web vulnerability scanner.

Hardening Against SQL Injection

The best way to prevent SQL Injections is to use safe programming functions that make SQL Injections impossible: parameterized queries (prepared statements) and stored procedures. Every major programming language currently has such safe functions and every developer should only use such safe functions to work with the database.

Hardening Against Privilege Escalation

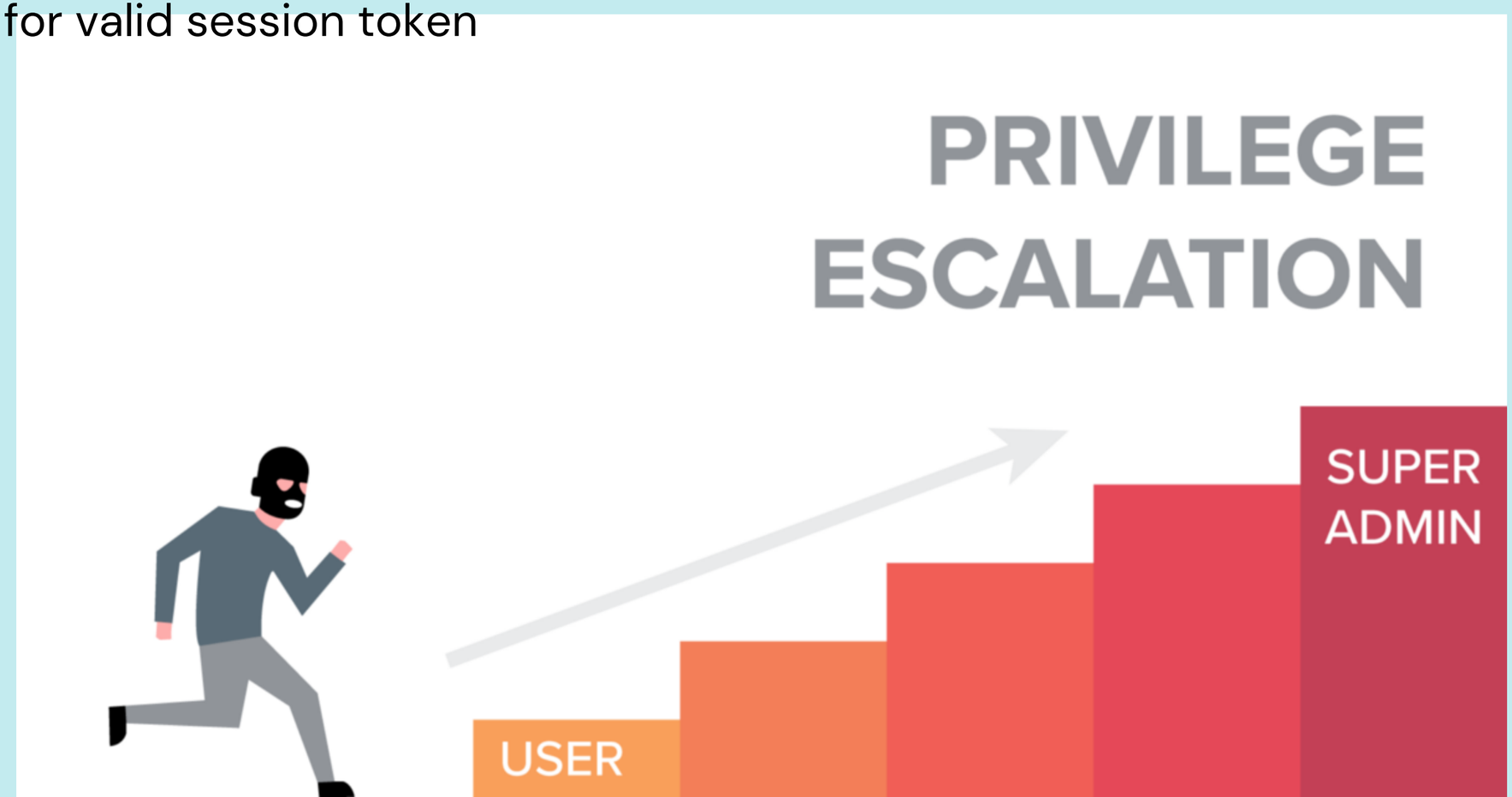
Privilege escalation attacks occur when bad actors exploit misconfigurations, bugs, weak passwords, and other vulnerabilities that allow them to access protected assets.
A typical exploit may start with the attacker first gaining access to a low-level privilege account. Once logged in, attackers will study the system to identify other vulnerabilities they can exploit further. They then use the privileges to impersonate the actual users, gain access to target resources, and perform various tasks undetected.



Hardening Against Privilege Escalation

You can prevent privilege escalation in the following ways -

- User?s privilege role?s checks should be performed on server side
- Sensitive data and session-related data should be kept on server side
- Cookies used for the session should be tamper-proof
- Data sent to the client should be encrypted
- Server should check for valid session token

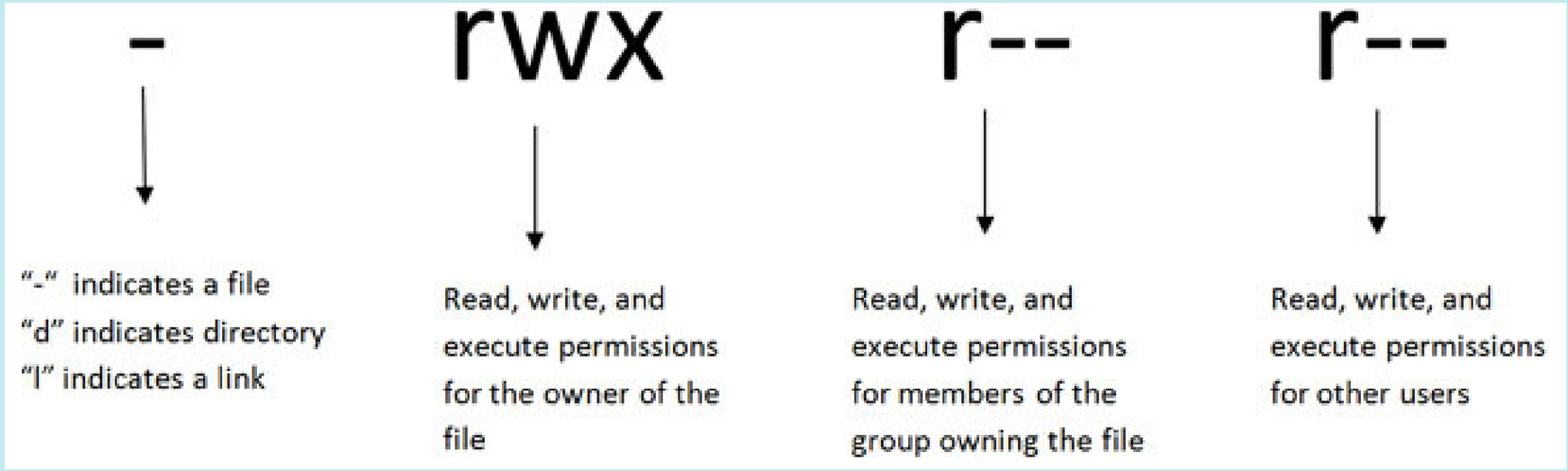


Hardening Against Privilege Escalation

Changing directory permissions in Linux

To change directory permissions in Linux, use the following:

chmod +rwx filename to add permissions.
chmod -rwx directoryname to remove permissions.
chmod +x filename to allow executable permissions.
chmod -wx filename to take out write and executable permissions.

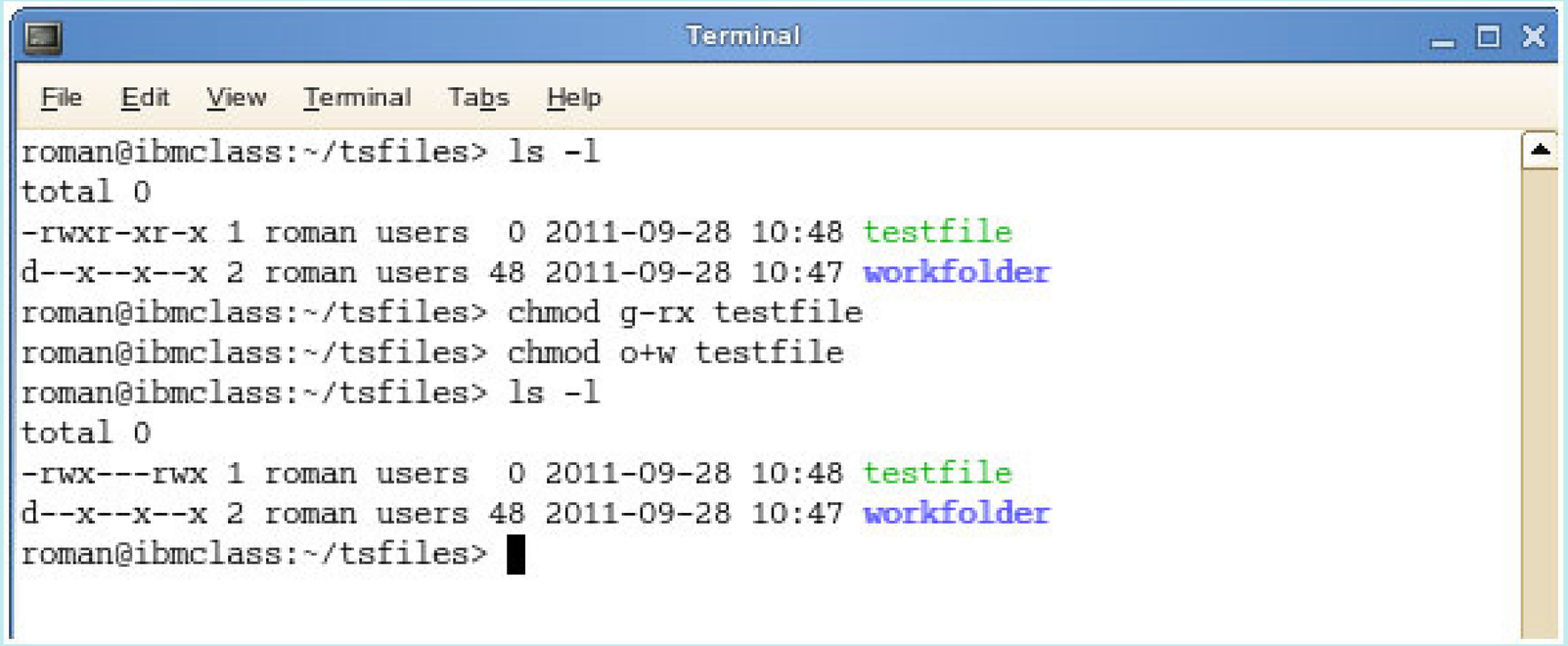


Hardening Against Privilege Escalation

Directory Permissions for the Group Owners and Others

The command for changing directory permissions for group owners is similar, but add a “g” for group or “o” for users:

chmod g+w filename
chmod g-wx filename
chmod o+w filename
chmod o-rwx foldername

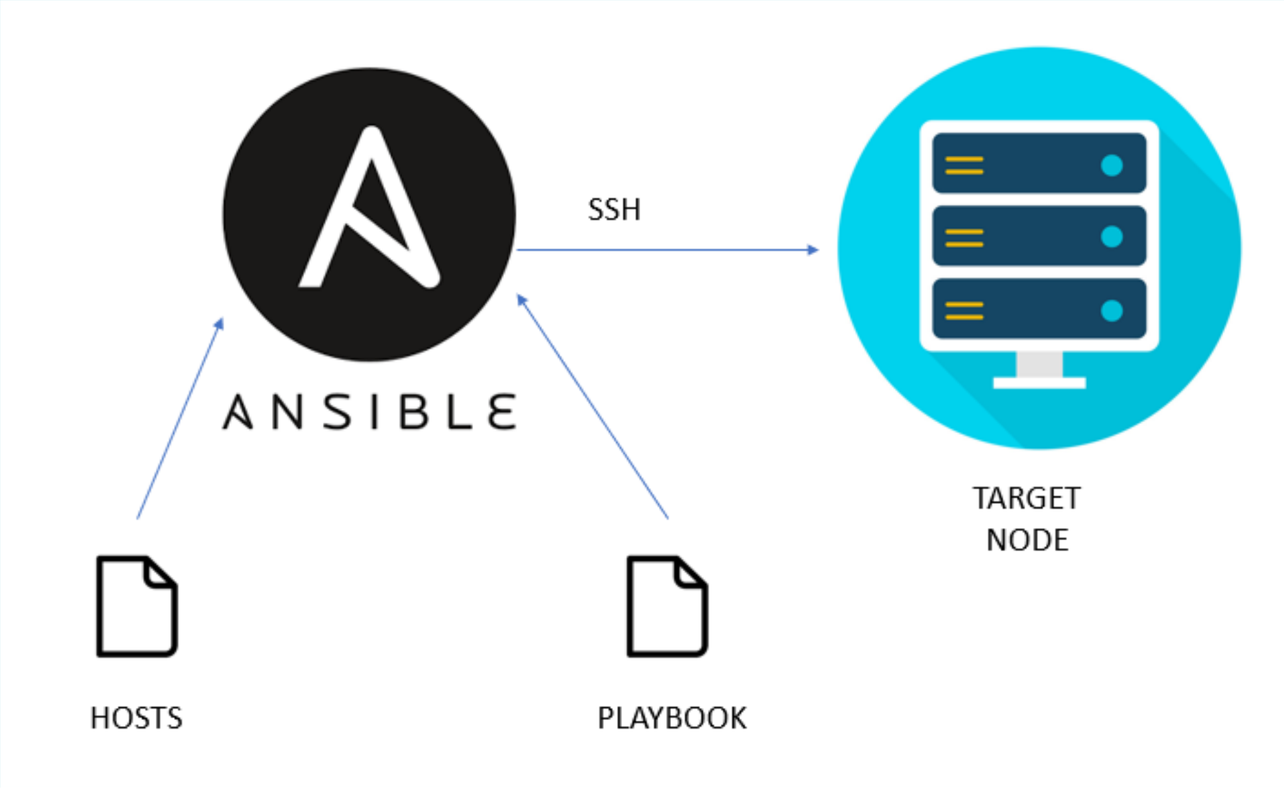


To change directory permissions for everyone, use “u” for users, “g” for group, “o” for others, and “ugo” or “a” (for all).
chmod ugo+rwx foldername to give read, write, and execute to everyone.
chmod a=r foldername to give only read permission for everyone.

Implementing Patches

Ansible-

Ansible is an open-source software provisioning, configuration management, and application-deployment tool enabling infrastructure as code.

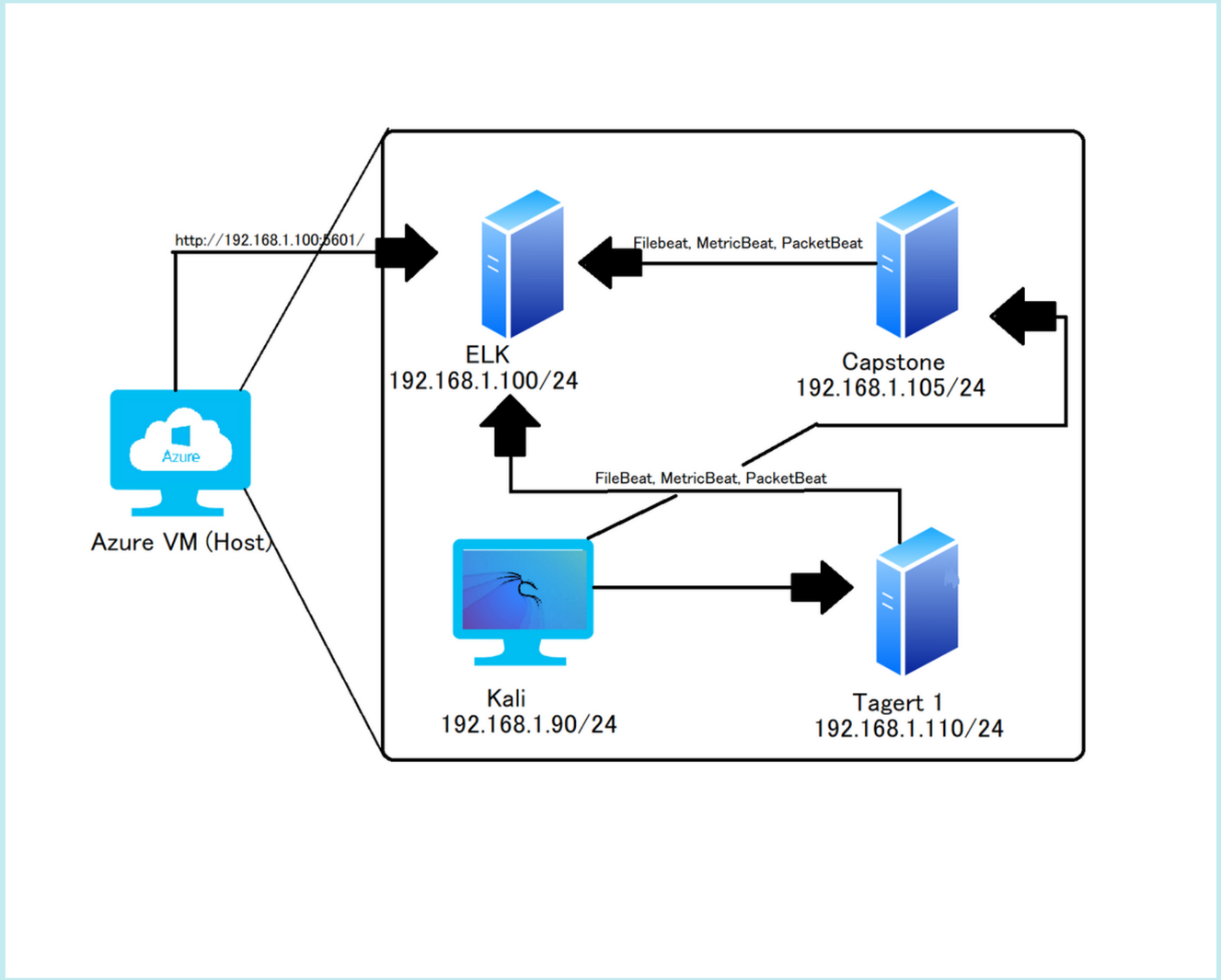



```
--
- name: restart system to reboot to newest kernel
   shell: "sleep 5 && reboot"
   async: 1
   poll: 0

 - name: wait for 10 seconds
   pause:
     seconds: 10

 - name: wait for the system to reboot
   wait_for_connection:
     connect_timeout: 20
     sleep: 5
     delay: 5
     timeout: 60

 - name: install user process
   yum:
     name: user process
     state: latest
```

Network

- Network Topology & Critical Vulnerabilities
- Traffic Profile
- Normal Activity
- Malicious Activity

Network Topology & Critical Vulnerabilities



Network
Address Range: 192.168.1.0/24
Netmask: 255.255.255.0
Gateway: 192.168.1.1

Machines
IPv4: 192.168.1.90
OS: Linux
Hostname: Kali

IPv4: 192.168.1.105
OS: Linux
Hostname: Capstone

IPv4: 192.168.1.110
OS: Linux
Hostname: Target 1

IPv4: 192.168.1.100
OS: Linux
Hostname: ELK

Critical Vulnerabilities

Our assessment uncovered the following critical vulnerabilities in Target 1.

Vulnerability	Description	Impact
OS Vulnerability	Download of trojan	A Trojan is designed to damage, disrupt, steal, or in general inflict some other harmful action on your data or network.
Human Vulnerability	Illegal downloads	Torrents users are liable to face legal consequences. copyright infringement
Network Vulnerability	Infected Windows host on the network	Damaging programs, deleting files, or reformatting the hard drive.

Traffic Profile

Traffic Profile

Our analysis identified the following characteristics of the traffic on the network:

Feature	Value	Description
Top Talkers (IP Addresses)	172.16.4.205	Machines that sent the most traffic.
Most Common Protocols	TCP(1372) UDP(1977)	Three most common protocols on the network.
# of Unique IP Addresses	808	Count of observed IP addresses.
Subnets	0/24	Observed subnet ranges.
# of Malware Species	june.dll	Number of malware binaries identified in traffic.

Behavioral Analysis

Users were observed engaging in the following kinds of activity.

“Normal” Activity
Visiting YouTube
Using a Search Engine
Downloading Windows Updates

Suspicious Activity
Created their own web server on the corporate network.
Downloading torrents
Going to unwarranted websites and downloading malware to infect windows

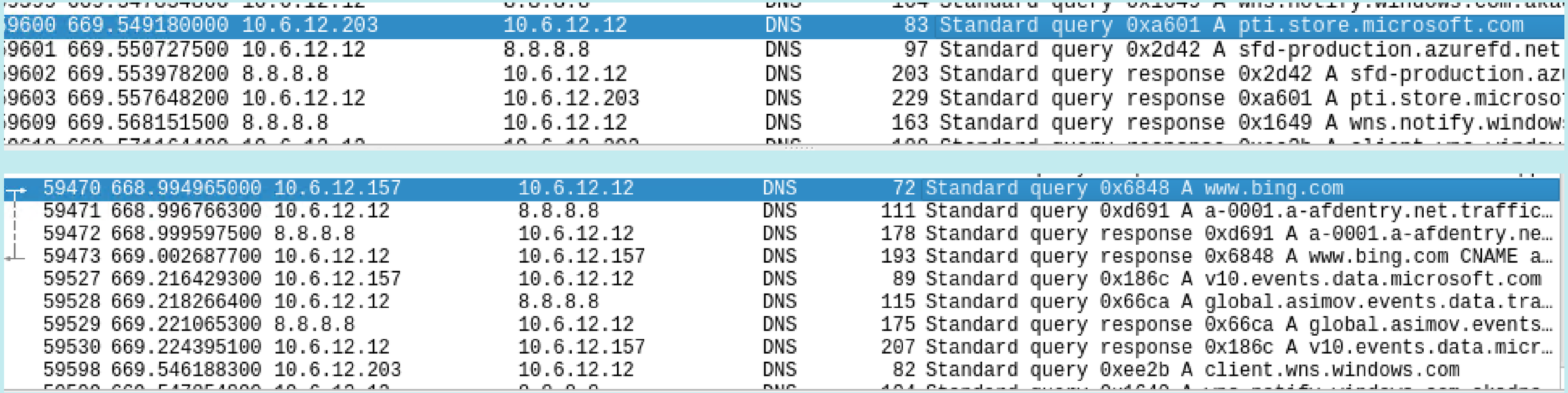
Normal Activity

What kind of traffic did you observe? Web traffic

Which protocol(s)? DNS

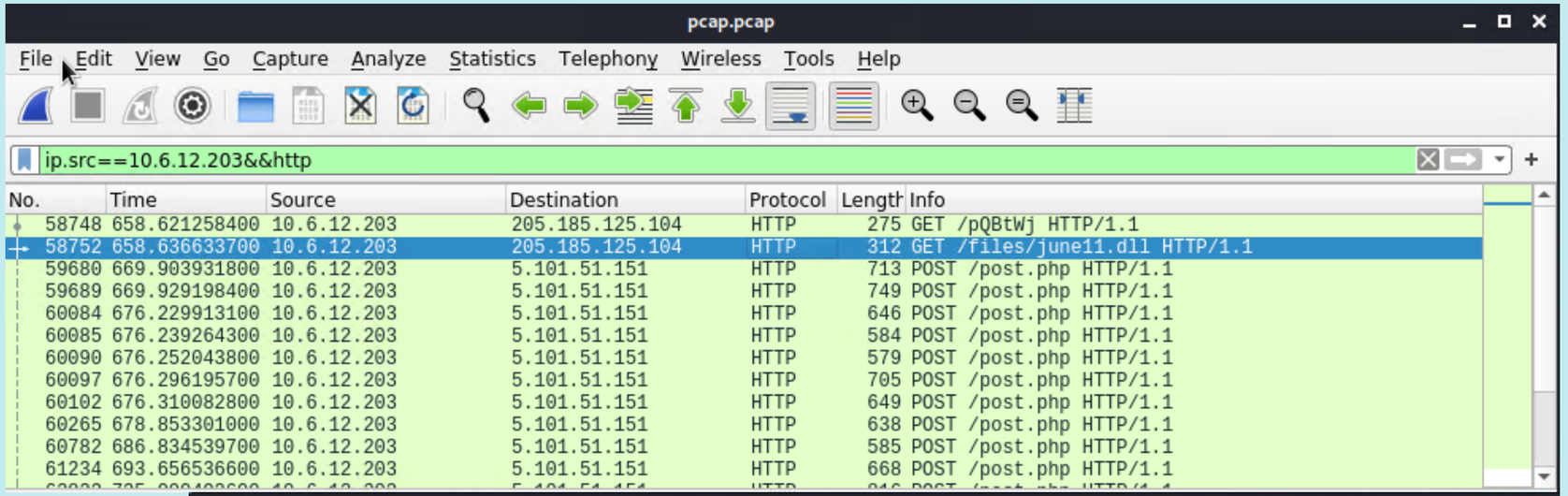
What, specifically, was the user doing? Browsing the web.

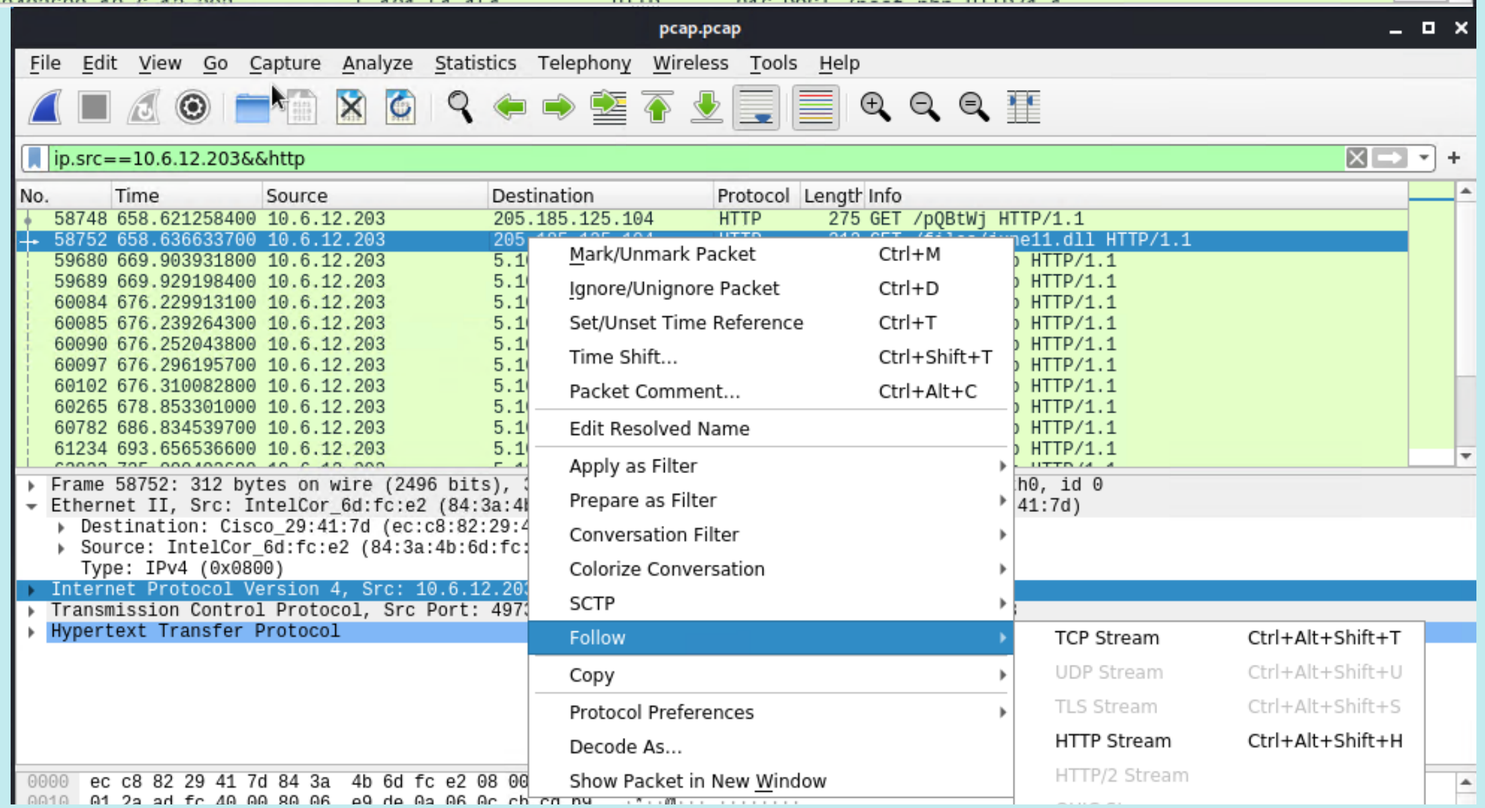
Which site were they browsing? Searching on bing. Exploring the microsoft store.



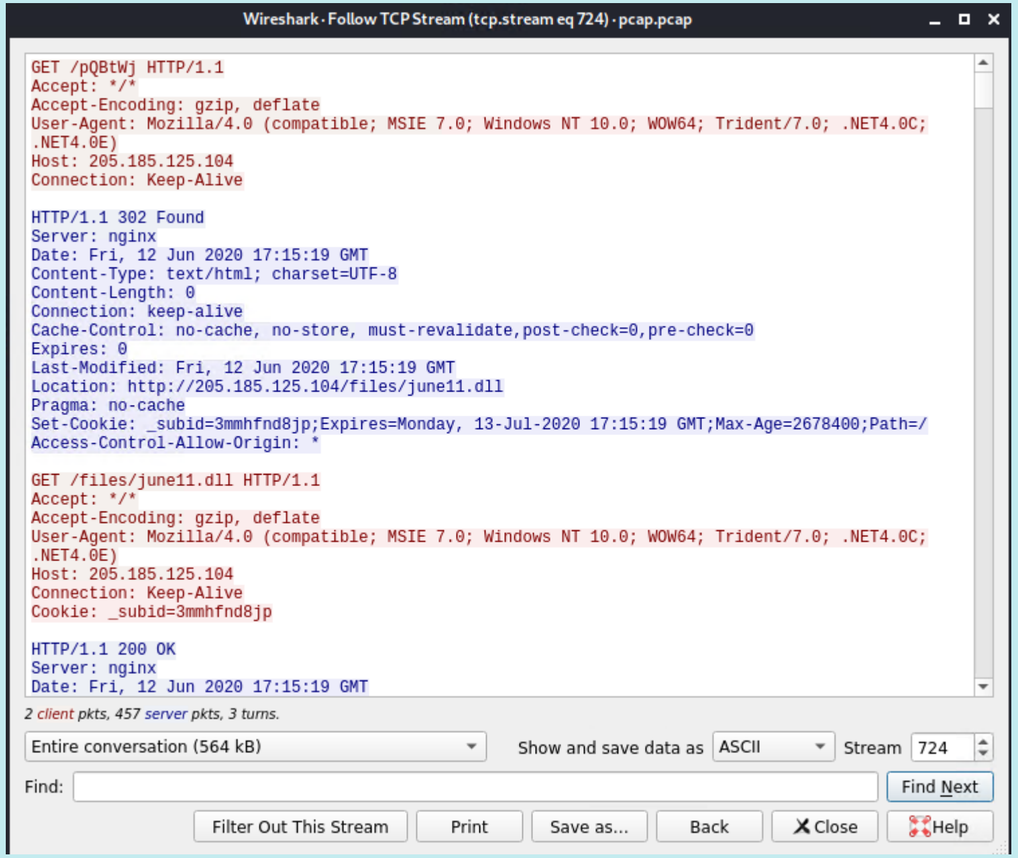
Malicious Activity

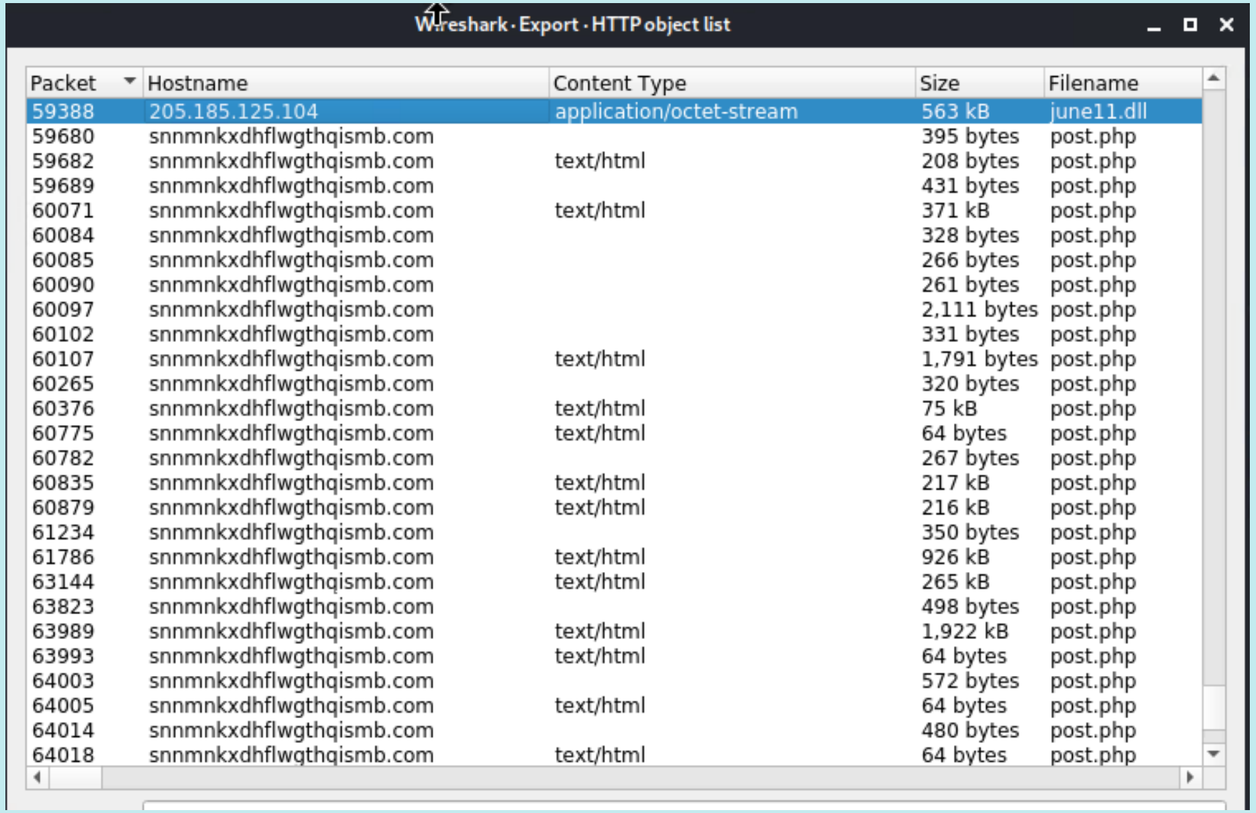
We observed some activity on the http protocol.
We followed a TCP stream to find more information on a GET request.



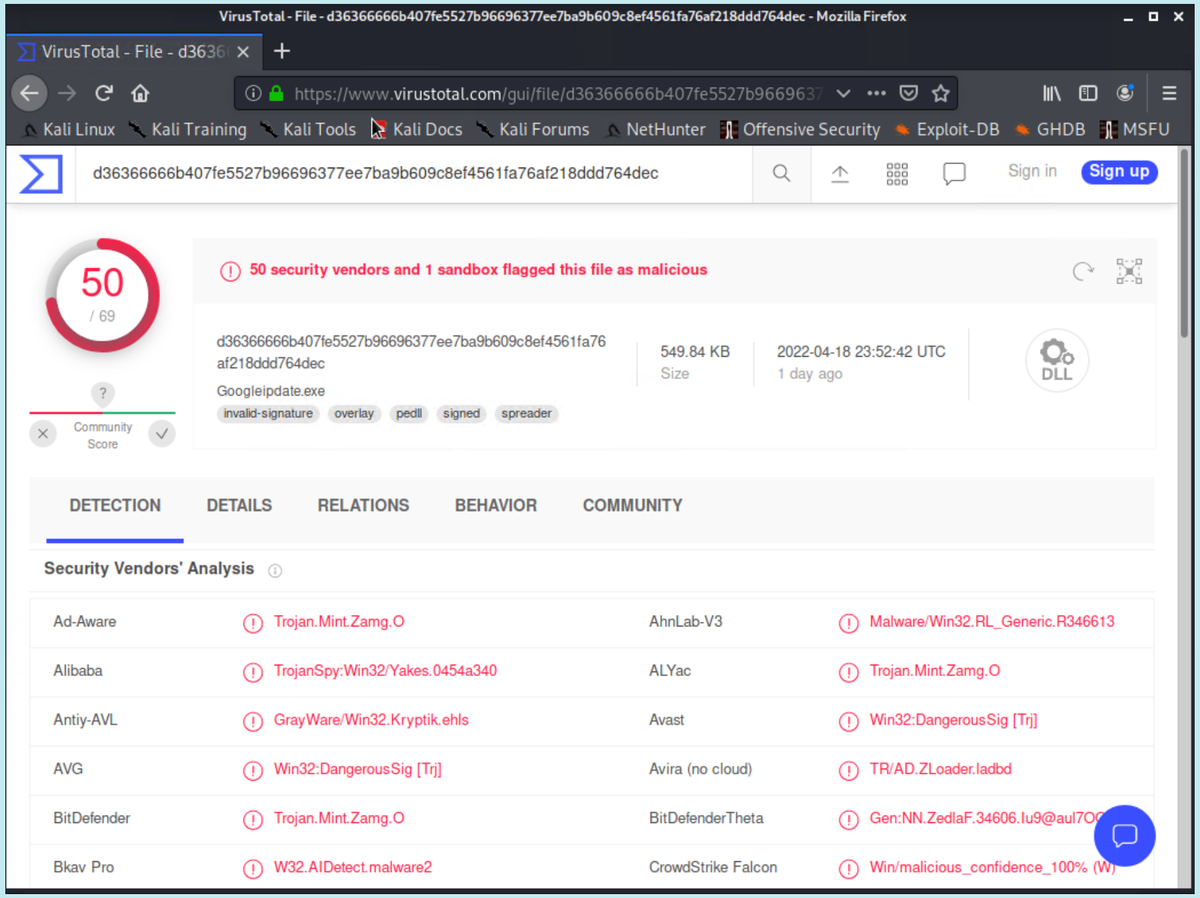


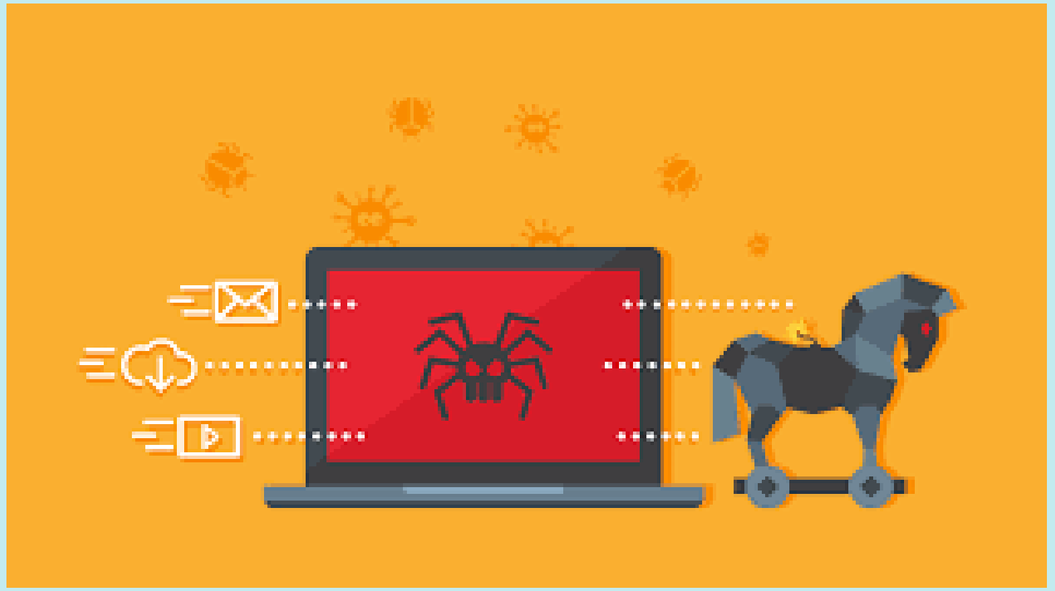
We were able to identify some sort of malicous file from the TCP stream. We saved the packets to get more information on it. We were able to find a file called june11.dll.





After uploading the packets to VirusTotal, we discovered a trojan. A type of malicious code or software that can take control of your computer.





The End

We hope you learned something new.