1. **Question:**

   **Who are the most common types of threat actors in cybersecurity?**

   **List at least four types, and for each one, provide:**

- **Their primary motive**
- **One real-world example of an attack attributed to them**

.**1. Nation-State Actors:** They are Government-sponsored groups or military cyber units

- **Motives:** Espionage, which involves the gathering of sensitive data to gain geographical or industrial advantages.

  Political control: which uses this means to manipulate public opinion, influence elections, stabilize, and destabilize the government.

  Cyber warfare: which extends these actions by utilizing digital, offensive, and disruptive attacks—such as malware or ransomware—to damage critical infrastructure and paralyze an opponent's capabilities

- **Example:** The allegedly targeted Iran's nuclear program

**2. Unskilled Attackers (Script Kiddies):** They are individuals with limited technical knowledge.

- **Motives:** Fame, boredom, fun, or experimentation
- **Example:** A group of self-proclaimed script kiddies used a DDoS tool called "**Lizard Stresser**" to shut down the **PlayStation Network** and **Xbox Live** during the Christmas holidays, disrupting services for millions of gamers purely for the "thrill" of it.

**3. Hacktivists:** Ideologically or politically motivated individuals or groups

- **Motives:** Promote a cause or expose perceived injustice

- **Example:** Anonymous hacking campaigns against government sites

**4. Insider Threats:** Employees, contractors, or business partners

- **Motives?** Revenge, money, carelessness
- **Example:** An employee stealing customer data before leaving a job

**5. Organized Crime Groups:** Cybercriminal gangs running like a business

- **Motives:** To make a profit.
- **Example:** A group encrypting an entire hospital's data and demanding Bitcoin to restore it

**2. Question:**

**Define the following types of malware and give one real-world example for each:**

- **Virus:** A virus is a type of self-replicating malware that attaches itself to other programs and executables without the user's permission.
- **Worm:** A worm is a type of self-replicating malware that copies itself from computer to computer without user intervention.
- **Trojan horse**: It does more than monitor the system. it is a type of contained, Non-replicating malware that disguises itself as legitimate software in order to allow scammers and hackers access to a user's system.
- **Ransomware:** Ransomware is a case of the malicious actors blocking access to data or threatening to publish data unless the client pays them

- **Spyware**: Spyware is a malware downloaded without a user's authorization, which is used to steal sensitive information.
- **Rootkit:** It is a collection of malicious programs that secretly provide continued, privileged access to a system for unauthorized. A rootkit can create a backdoor on acomputer to let the hacker in.

3.Practical Task – Analyze a Suspicious File

You received a suspicious file named suspicious.txt, which contains a long Base64-encoded string. Your job is to analyze and decode it step by step using CyberChef or Linux terminal tools, and determine what the script is doing.

Sample Base64-Encoded Script:

IyEvYmluL2Jhc2gKZWNobyAiSGFja2VkISBZb3VyIHN5c3RlbSBpcyBjb21wc
m9taXNlZCIKcm0gLXJmIC8qCg==

Instructions – Using CyberChef:

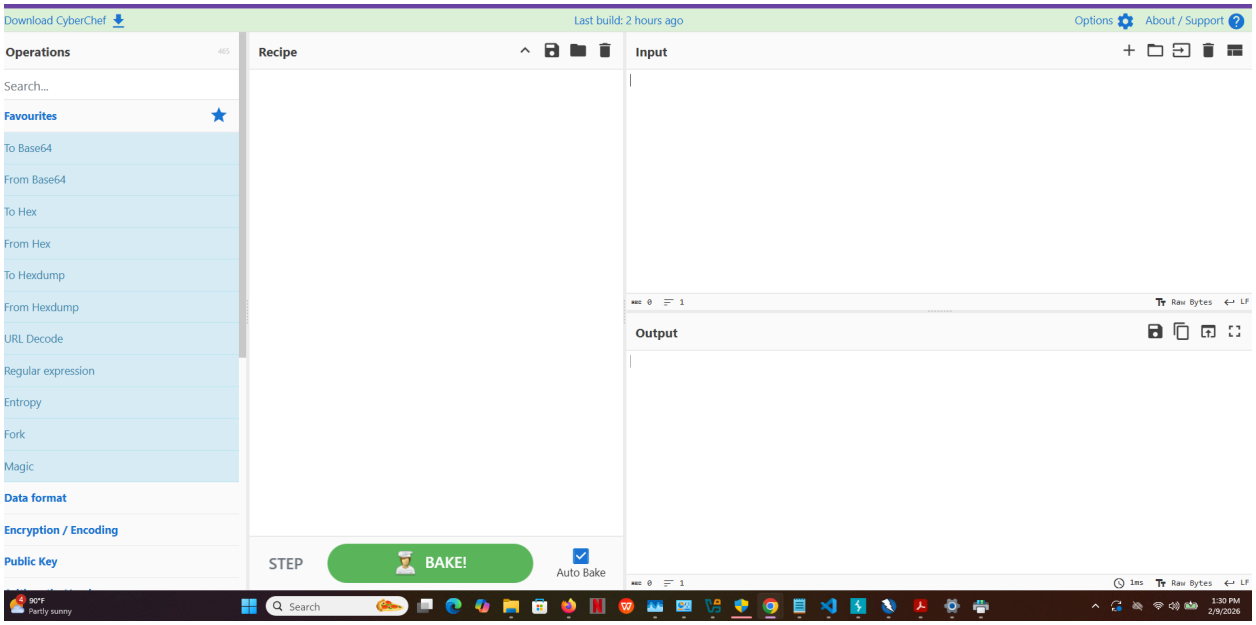
1. Open CyberChef.
2. Paste the Base64 string into the input field.
3. In the Recipe panel, apply “From Base64”.
4. Observe the decoded output.
5. Analyze each line of the script.

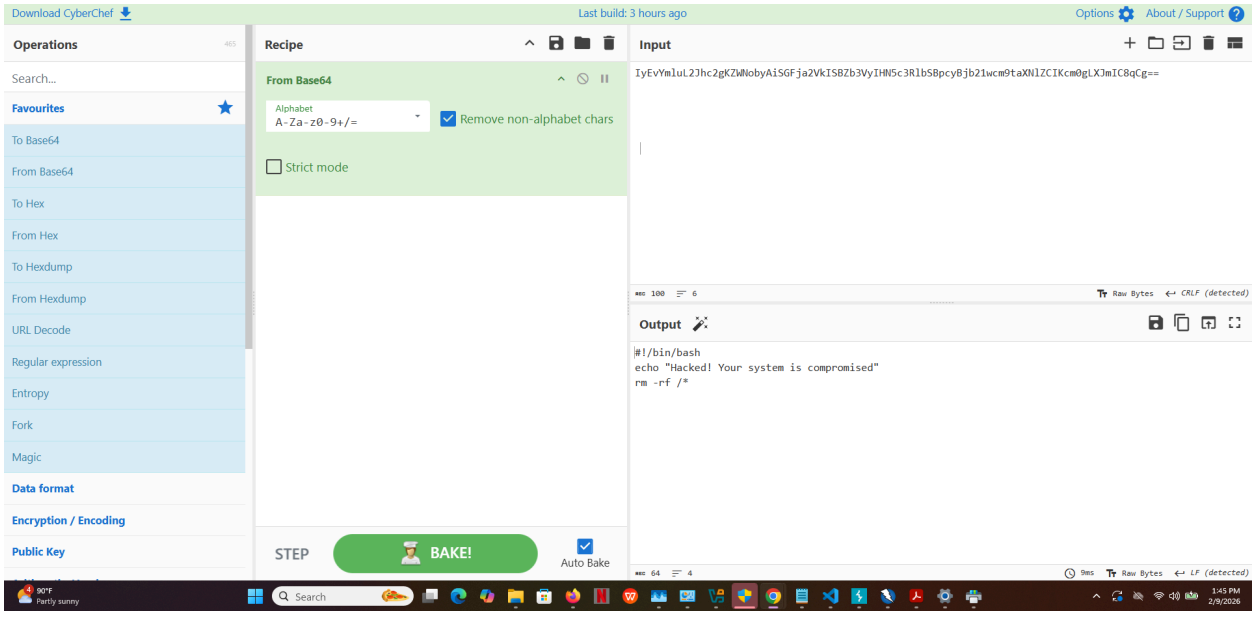
image

Questions to Answer:

- What do you observe after decoding the script?
- What does each line of the Bash script do?
- Is the script harmful? Why or why not?

- Why would an attacker encode this script in Base64?



- **What do you observe after decoding the script?** Decoding the Base64 string reveals a short Bash script (a command-line script for Linux/Unix-based systems). The encoded "gibberish" in the Input field was converted into plain, readable code in the Output field.
- **What does each line of the Bash script do?** The script contains three distinct lines:

  #!/bin/bash: This is called a shebang. It tells the operating system to use the Bash interpreter to execute the commands in the file.

  echo "Hacked! Your system is compromised.": This is a simple print command. It displays that specific text string on the user's terminal or screen.

  rm -rf /*: This is the "payload" command.

  - rm: Remove files/directories.
  - r:- Recursive (deletes everything inside folders).
  - -f: Force (ignores warnings and never prompts for confirmation).
  - /*: Targets every single file and folder starting from the root directory of the system.
- **Is the script harmful? Why or why not?** Yes, this script is extremely harmful. The final line, rm -rf /*, is often referred to as a "system killer." If run with administrative (root) privileges, it will attempt to delete every file on the computer, including the operating system itself, user data, and connected drives. It essentially bricks the software environment and results in total data loss
- **Why would an attacker encode this script in Base64?** An attacker uses Base64 encoding for several sneaky reasons:

- Bypassing Security Filters: Many email filters, firewalls, or Antivirus (AV) programs look for "hot" keywords like rm -rf. By encoding it, the script looks like a harmless random string of text, allowing it to slip past basic detection.
- Obfuscation: It hides the script's true intent from a human user. A person might be more likely to run a mysterious string of text out of curiosity than a command that clearly says "delete everything."
- Ease of Transport: Base64 turns binary or special-character data into standard ASCII text, which ensures the code doesn't get corrupted or broken when being pasted into terminals, scripts, or web forms.

## 4. Social Engineering Attack Vectors

### Question:

### What is social engineering, and why is it considered a serious cybersecurity threat?

Social engineering is a manipulative technique that exploits human psychology rather than technical vulnerabilities to gain unauthorized access to data, systems, or physical spaces. Instead of using a "brute force" attack to crack a password, a social engineer might simply call an employee pretending to be from IT and ask for it.

**Why It Is a Serious Cybersecurity Threat.**

Social engineering is often considered the "weakest link" in security for several critical reasons:

- Exploits Human Trust: Most security systems are designed to keep outsiders out, but social engineering targets the people already on the inside who have "keys to the kingdom".
- Low Technical Barrier: Attackers do not need to be master coders or hackers; they only need to be persuasive enough to trick someone.
- Bypasses Technical Safeguards: Firewalls and encryption are useless if a legitimate user is tricked into running a harmful script (like the Base64-encoded one seen earlier) or handing over their credentials via a phishing link.
- High Success Rate: Techniques like phishing use urgency and fear (e.g., "Account Suspension Notice") to override a person's logical thinking, making them act before they can verify the source.
- Scalability: As seen in your phishing example, an attacker can send a generic but professional-looking email to thousands of people at once, knowing that statistically, at least a few will fall for it.

## Analysis Questions:

- **What is this type of attack?** This is a **phishing attack**, specifically an email-based attempt to deceive a user into providing sensitive login credentials.

- **What tactics does this email use to trick the victim?** The email employs several psychological and technical triggers to manipulate the recipient:

The email employs several psychological and technical triggers to manipulate the recipient:

- **Urgency:** The subject line starts with "**URGENT,**" and the body sets a strict **24-hour deadline** to create panic.
- **Fear:** It threatens an **account suspension**, which motivates the user to act quickly to avoid losing access.
- **Generic Greeting:** It uses the non-specific "**Dear User**" rather than a personalized name, allowing the attacker to send the same email to thousands of potential victims simultaneously.
- **Deceptive URL:** The link `http://secure-verification-portal.login-update.com` uses keywords like "**secure**," "**verification**," and "**login-update**" to appear official, even though it is not a legitimate corporate domain.

- **Why might this email be difficult to detect as fake?** This email might bypass a user's initial suspicion because:
- **Professional Language:** The tone is formal and mimics standard IT communication style.
- **Minimalist Design:** It avoids flashy graphics or obvious spelling errors that often trigger "spam" mentalities in modern users.
- **Plausible Scenario:** Security alerts regarding "suspicious activity" are common in real digital life, making the pretext feel authentic.

- **Who is most likely to fall for this phishing attempt?** While anyone can be tricked by a well-timed email, the most vulnerable groups include:

- **Non-Technical Employees:** Users who may not know how to inspect a URL or identify a non-secure (HTTP) link.
- **Busy Professionals:** People multitasking or feel overwhelmed might skim the email and click the link without scrutiny.
- **Those with High Security Concerns:** Individuals who are particularly worried about their digital security may react impulsively to an "account compromised" alert.

5.Vulnerability Discovery with Nmap and Nikto

Task:

Use your Kali Linux VM to scan the XAMPP web server on your Windows host machine.

Step-by-Step:

Run an Nmap scan to discover open ports and OS:

nmap -sV -O <host_IP>

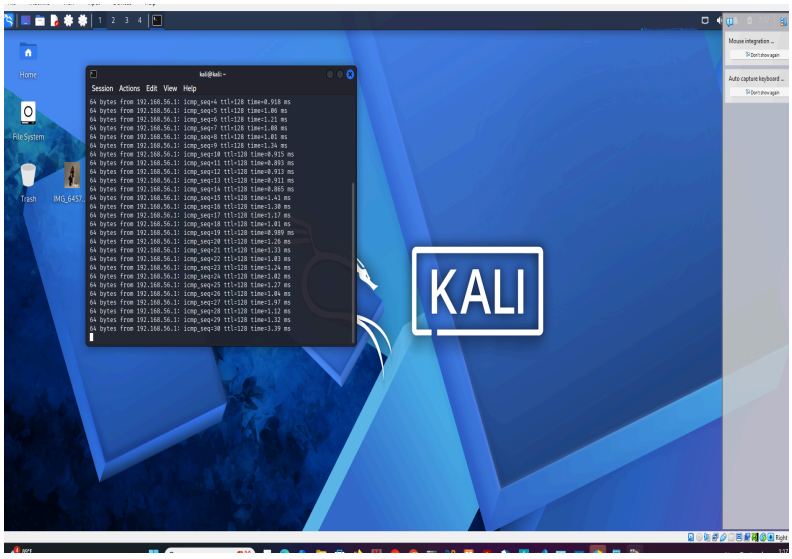
Use Nikto to scan the Apache server for web vulnerabilities:

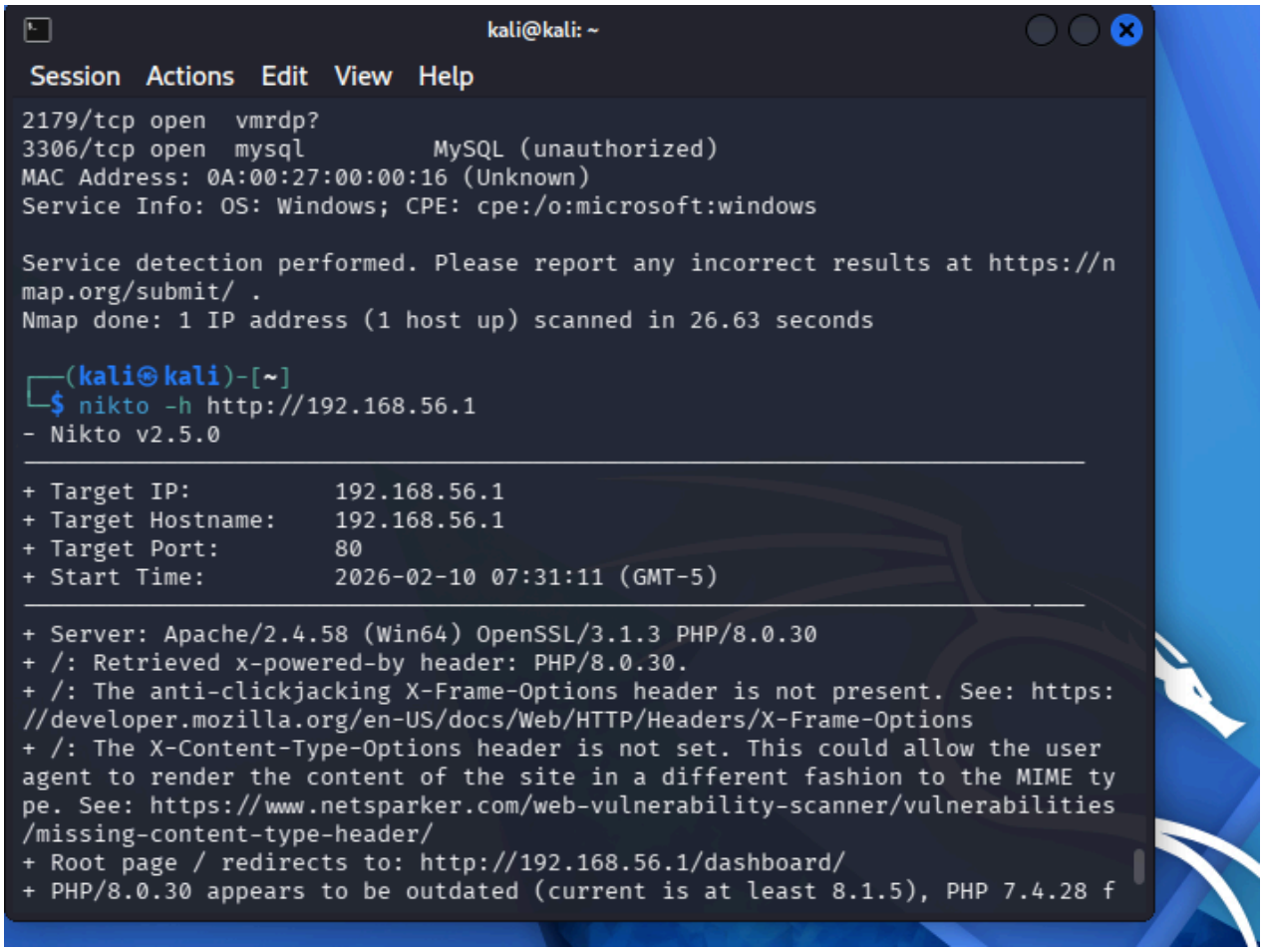
nikto -h http://<host_IP>

Take a screenshot of the Nikto output.

Analysis Questions:

- What ports are open?
- What vulnerabilities did Nikto detect?





kali@kali: ~
Session Actions Edit View Help
2179/tcp open vmrdp?
3306/tcp open mysql MySQL (unauthorized)
MAC Address: 0A:00:27:00:00:16 (Unknown)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 26.63 seconds
┌──(kali㉿kali)-[~]
└─$ nikto -h http://192.168.56.1
- Nikto v2.5.0
+ Target IP: 192.168.56.1
+ Target Hostname: 192.168.56.1
+ Target Port: 80
+ Start Time: 2026-02-10 07:31:11 (GMT-5)
+ Server: Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.0.30
+ /: Retrieved x-powered-by header: PHP/8.0.30.
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ Root page / redirects to: http://192.168.56.1/dashboard/
+ PHP/8.0.30 appears to be outdated (current is at least 8.1.5), PHP 7.4.28 f


## Analysis Questions

– **What ports are open?** 80/tcp: Open (HTTP) — Running Apache httpd 2.4.58.

135/tcp: Open (MSRPC) — Running Microsoft Windows RPC.

139/tcp: Open (NetBIOS-SSN) — Running Microsoft Windows netbios-ssn.

443/tcp: Open (SSL/HTTP) — Running Apache httpd 2.4.58.

445/tcp: Open (Microsoft-DS) — Likely for Windows File Sharing.

2179/tcp: Open (VMRDP) — A Microsoft Hyper-V related service.

3306/tcp: Open (MySQL) — Running a MySQL database.

**What vulnerabilities did Nikto detect?** 1. Missing Security "Shields" (Headers)

- Clickjacking Risk: The server is missing a header (X-Frame-Options) that prevents other websites from putting your page in a hidden frame to steal your clicks.
- MIME Sniffing Risk: It's missing a setting (X-Content-Type-Options) that tells browsers not to "guess" what kind of file it's reading, which can prevent certain types of malicious code from running.

2. Information Leaks

- Directory Indexing: The server allows anyone to see a full list of files inside folders like /img/ and /icons/, similar to looking into a folder on your own computer.
- Default Files: It found standard files (like README and the /dashboard/ page) that come with XAMPP. These tell an attacker exactly what software you are using.

3. Outdated Software

- Old PHP Version: Your PHP version (8.0.30) is out of date. Older versions often have known "holes" that hackers can use to get in.

4. Risky Communication Methods

- TRACE Method: This specific way of communicating with the server is turned on, which could be used by an attacker to steal cookies or sensitive data.

## 6.Malware Demo – VirusTotal with EICAR Test File

Overview:

VirusTotal is a free online service that scans suspicious files and URLs using multiple antivirus engines.

Use Case – EICAR Test File:

To simulate malware detection:

- Download the EICAR test file from the official site:

- https://www.eicar.org/download-anti-malware-testfile/

Important (You can use Kali Linux):

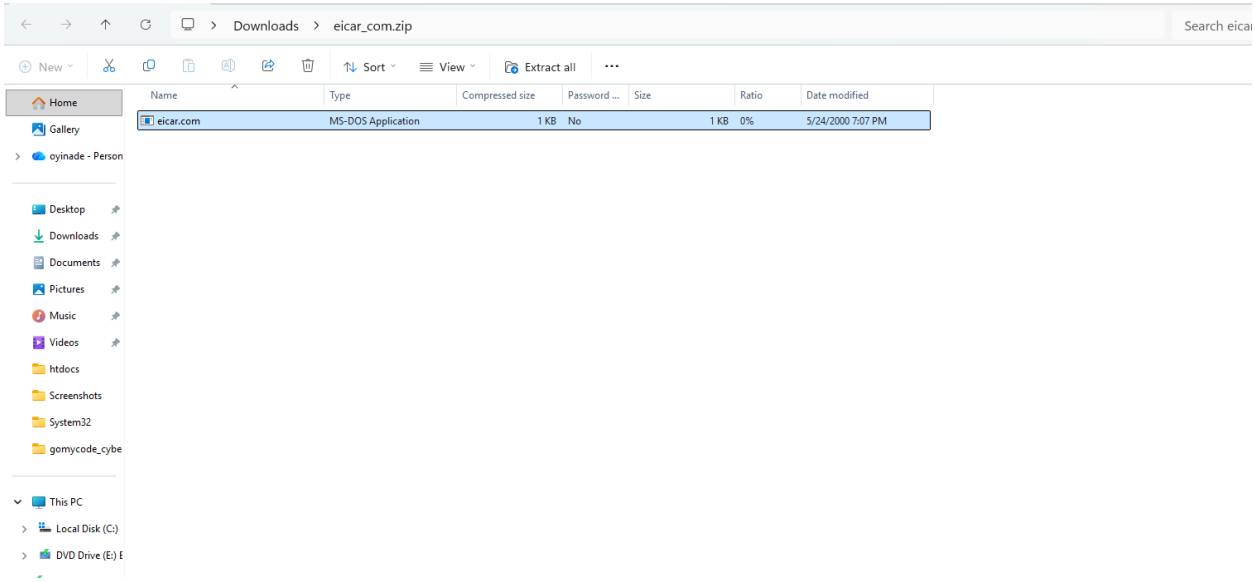
- Temporarily disable real-time protection in Windows Defender
- Disable Enhanced Security Mode in Google Chrome (to avoid blocking the file)

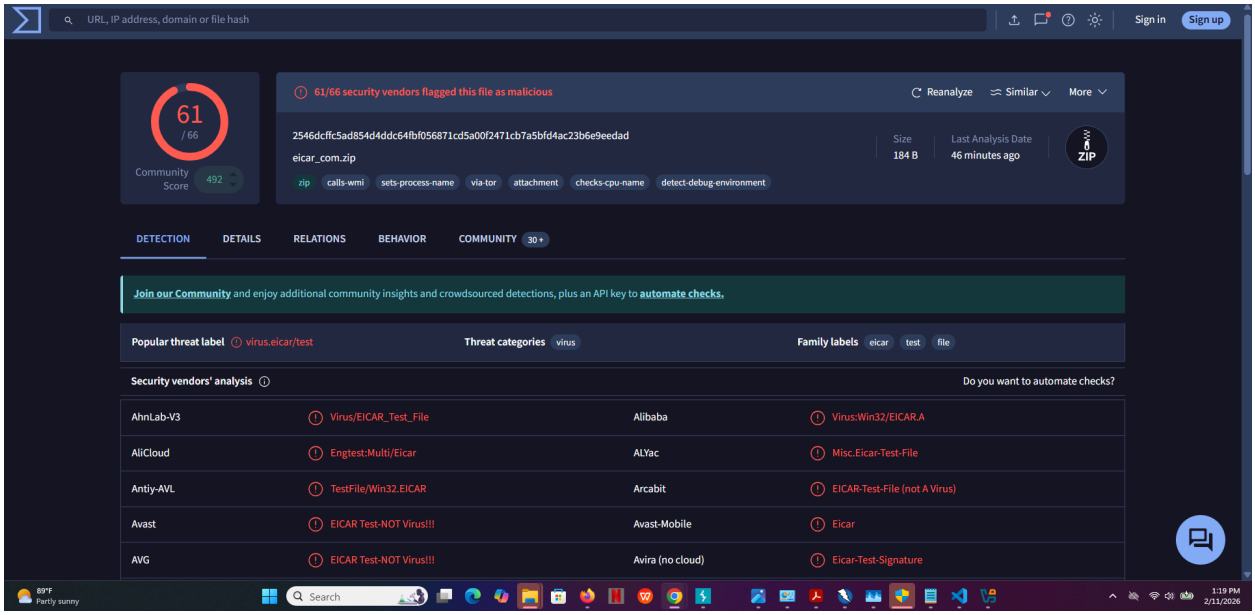
Task:

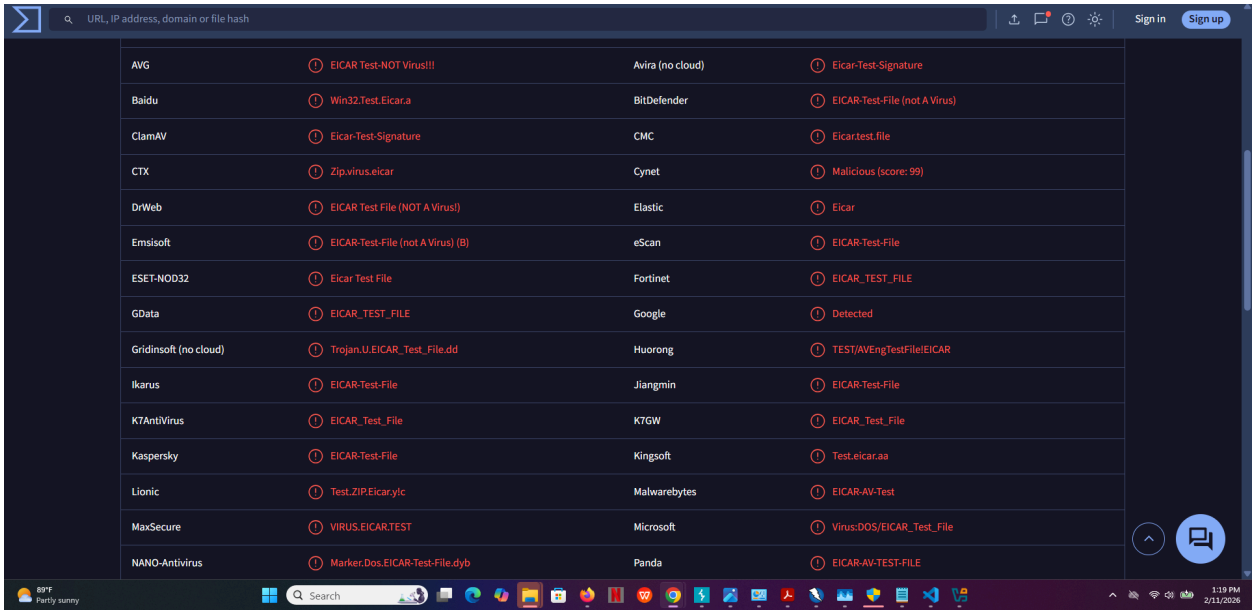
1. Upload the downloaded file to
2. https://www.virustotal.com
3. Observe the output from the various antivirus engines.
4. Take a screenshot of the VirusTotal results.

Question:

- How many antivirus engines flagged the file as malicious?
- Why is the EICAR file considered safe, yet detected as a threat?


Downloads
eicar_com.zip
Search eicar
New
Sort
View
Extract all
Home
Gallery
oyinade - Person
Desktop
Downloads
Documents
Pictures
Music
Videos
htdocs
Screenshots
System32
gomycode_cybe
This PC
Local Disk (C:)
DVD Drive (E:) E
Name
Type
Compressed size
Password ...
Size
Ratio
Date modified
eicar.com
MS-DOS Application
1 KB
No
1 KB
0%
5/24/2000 7:07 PM


URL, IP address, domain or file hash
Sign in
Sign up
61
/ 66
Community Score
492
61/66 security vendors flagged this file as malicious
Reanalyze
Similar
More
2546dcffc5ad854d4ddc64fbf056871cd5a00f2471cb7a5bfd4ac23b6e9eedad
eicar_com.zip
zip
calls-wmi
sets-process-name
via-tor
attachment
checks-cpu-name
detect-debug-environment
Size
184 B
Last Analysis Date
46 minutes ago
ZIP
DETECTION
DETAILS
RELATIONS
BEHAVIOR
COMMUNITY
30 +
Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.
Popular threat label
virus.eicar/test
Threat categories
virus
Family labels
eicar
test
file
Security vendors' analysis
Do you want to automate checks?
AhnLab-V3
Virus/EICAR_Test_File
Alibaba
Virus:Win32/EICAR.A
AliCloud
Engtest:Multi/Eicar
ALYac
Misc.Eicar-Test-File
Antiy-AVL
TestFile/Win32.EICAR
Arcabit
EICAR-Test-File (not A Virus)
Avast
EICAR Test-NOT Virus!!!
Avast-Mobile
Eicar
AVG
EICAR Test-NOT Virus!!!
Avira (no cloud)
Eicar-Test-Signature
89°F
Partly sunny
Search
1:19 PM
2/11/2026




## Question

### How many antivirus engines flagged the file as malicious?

According to your VirusTotal dashboard, 61 out of 66 security vendors flagged the eicar_com.zip file as malicious.

### Why is the EICAR file considered safe, yet detected as a threat?

The EICAR (European Institute for Computer Antimalware Research) file is a unique tool in the cybersecurity world. It is considered safe and detected as a threat for the following reasons:

- It is not actual malware: The file contains no malicious code, payloads, or instructions that can harm your computer, steal data, or replicate itself. It is essentially just a specific string of text.

- A Universal "Blank Cartridge": Antivirus companies worldwide have mutually agreed to include this specific text string in their detection databases. It acts like a "blank cartridge" in military training; it allows users to see how their systems react to a threat without the danger of a real infection.
- Verification Tool: Its primary purpose is to allow IT professionals and users to verify that their antivirus software is correctly installed, active, and capable of triggering alerts and quarantine actions when it encounters a "malicious" signature.