This challenge will be analysing a malware sample.

**Learning Outcomes**

1. Foundations of analysing malware samples safely.
2. Fundamentals of .NET binaries.
3. dnSpy Tool for decompiling malware samples written in .NET.
4. Building an essential methodology for analysing malware source code.

**Key Information**

Sandbox is like a pretend computer setup that acts like a real one. It’s a safe place for experts to test malware and see how it behaves without any danger. Having a sandbox environment is essential when conducting malware analysis because it stops experts from running malware on their actual work on computers, which could be risky and harmful.

A typical environment setup of malware sandbox contains the following:

1. Network Controls
   1. Sandboxes often have network controls to limit and monitor the network traffic the malware generates. This also prevents the propagation of malware in any other assets.
2. Virtualisation
   1. Many sandboxes use technologies like VMware, VirtualBox, or Hyper-V to run the malware in a controlled, isolated environment. This allows for easy snapshots, resets, and disposal after analysis.
3. Monitoring and Logging
   1. Sandboxes record detailed logs of the malware’s activities, including system interactions, network traffic, and file modification. These logs are invaluable for analysing and understanding the malware’s behaviour.

A screenshot of a computer program

Description automatically generatedThe malware that is going to be used as the sample is “JuicyTomatoy\_defanged”.

To get to the main function, click on the red “Main” text to direct you.

A computer screen with text

Description automatically generatedNow I can see the entire code.

A screenshot of a computer program

Description automatically generatedInside the Assembly Explorer panel you can see some of the functions contained within the main file.

This would help further understand how the malware will work. Just by looking at these functions, the malware will encrypt, decrypt, and execute commands. I also noticed that there is an implant function. This could allow backdoors to be installed.

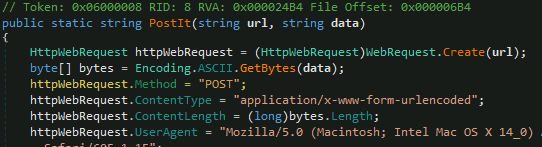
A screen shot of a computer code

Description automatically generated**Question 1) What HTTP User-Agent was used by the malware for its connection requests to the C2 server?**

The highlighted section is the User-Agent to look for in further investigations (maybe in packet captures), this can identify the browser or device.

**Answer**: "Mozilla/5.0 (Macintosh; Intel Mac OS X 14\_0) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/17.0 Safari/605.1.15

**Question 2) What is the HTTP method used to submit the command execution output?**



This talks about submitting data for command execution to the command server, this would be a post request.

**Answer**: POST

A screen shot of a computer screen

Description automatically generated**Question 3) What key is used by the malware to encrypt or decrypt the C2 data?**

The key was found in both the encrypt and decrypt functions of the malware.

**Answer**: youcanthackthissupersecurec2keys

**Question 4) What is the first HTTP URL used by the malware?**

A computer screen shot of a program code

Description automatically generated

The first URL that is mentioned is a combination or ‘str’ and ‘url’ variables. This was found in the main function.

**Answer**: http://mcgreedysecretc2.thm/reg

A black background with white text

Description automatically generated**Question 5) How many seconds is the hardcoded value used by the sleep function?**

A black screen with white and orange text

Description automatically generatedThe sleeper function showed that it took a variable as input. I then went into the main function to determined where the function was called.

A computer screen with colorful text

Description automatically generatedNext, I needed to find where the count variable was declared.

**Answer**: 15 seconds

A screen shot of a computer program

Description automatically generated**Question 6) What is the C2 command the attacker uses to execute commands via cmd.exe?**

The malware will take in text from the command server and decrypt it. It is then stored in a string variable ‘a’. Then there is a series of checks to be done, ‘sleep’…’implant’…’shell’…’quit’… Once the command has been decrypted and it is decrypted as ‘shell’ then the command will be executed.

***// Step 1: Split the decrypted string with space***

array = Decryptor(it).Split(' ')

***// "shell net localgroup administrators".Split(' ') --> ["shell", "net", "localgroup", "administrators"]***

***// Step 2: Store the first element into the "a" variable***

a = array[0] **// a = "shell"**

text = ""

***// Step 3: Combine the remaining elements (excluding the first) using space***

IF array.length > 1

THEN text = combine with space(["net", "localgroup", "administrators"]) **// text = "net localgroup administrators"**

**Answer**: Shell

**Question 7) What is the domain used by the malware to download another binary?**

A computer code on a black background

Description automatically generatedInside the main function you can see another domain is called to download a spykit executable. This is supported by the naming of the function called ‘Implant’.

**Answer**: stash.mcgreedy.thm