Rochester Institute of Technology of Dubai

Department of Electrical Engineering and Computing

Computing Security

CSEC 202 Reverse Engineering Fundamentals

Spring 2024

Sections: 600, 601, and 602

Homework Assignment #3:

Basic Dynamic Analysis

"Show the Cat who is in control!"

Release Date: March 21, 2024

Due Date: **April 15, 2024** – 11:59:59 p.m. (GST= GMT+4)

Instructor: Emad AbuKhousa (eakcad@rit.edu)

Homework Assignment #3:

Basic Dynamic Analysis

Please pay close attention to the following essential guidelines and expectations:

- **Submission Deadline:** There will be **no extensions** under any circumstances. Ensure you manage your time effectively to meet this deadline.
- **Instructor Support Window: The** instructor will not provide support 2 days before the assignment's deadline. It's crucial to start early to maximize the opportunity for receiving guidance and support. Procrastination may limit your ability to seek help.
- **Support Availability**: Support for this assignment will be available only during office hours. Make sure to bring up any questions or concerns during these designated times to receive assistance.
- **Communication Channels:** WhatsApp is not considered an official channel for office hours or assignment support. Please use the designated communication methods provided by your instructor for all correspondence related to this assignment.

Academic Integrity:

- Zero Tolerance for Plagiarism: Plagiarism and paraphrasing of any kind will be strictly penalized. All submissions will be thoroughly checked for originality. **We have robust measures in place to detect malpractice, so do not underestimate the system's ability to identify cheating.**

- Consequences of Academic Misconduct: Being caught in acts of academic dishonesty, such as plagiarism, will result in a score of zero for this assignment. Further disciplinary actions may follow according to the academic conduct policies. It is imperative that you adhere to the highest standards of academic integrity in your work.

These guidelines are put in place to ensure a fair and conducive learning environment for all students. Adhering to these rules is not only about avoiding penalties but also about cultivating professionalism, responsibility, and integrity in your academic pursuits.

Remember:

Start working on your assignment early to navigate through potential challenges with ample time for adjustments.

Objectives:

This homework is designed to enhance your hands-on experience with basic dynamic analysis, a fundamental step towards mastering reverse engineering. Dynamic analysis plays a crucial role in your ability to comprehend the behavior of binary code, offering insights that significantly aid in a quicker and more precise understanding of assembly code during later stages of analysis. Compared to basic static analysis, dynamic analysis provides a deeper dive into the code's functionality, revealing operational aspects that static methods alone might miss.

Objective: To apply basic dynamic analysis techniques on a given binary, aiming to understand its behavior and functionalities. This practical experience will prepare you for more advanced reverse engineering tasks, improving your skills in analyzing assembly code.

Important Note: The binary provided for this assignment is intended strictly for educational purposes. Under no circumstances should it be run on a production machine. It is imperative to conduct your analysis within a controlled environment, such as a sandbox or the virtual setup introduced earlier in this course. You are expected to adhere to safe practices and bear full responsibility for any actions taken without the appropriate precautions.

Rubric:

Homework Assignment #3 Rubric	
Criteria	**Pts**
Two Detailed Reports (one for individual tasks and one for group tasks): **The group report should be submitted by only one member of the team.** a. Craft a well-organized, step-by-step report detailing your entire analysis process. b. Include clear explanations for each step, accompanied by relevant screenshots. c. Discuss the result and purpose of each tool or command you employed, demonstrating your understanding. d. Format your report professionally in a standard Word document using consistent styles and headings. e. Include a cover page listing the assignment details, team members, their contributions, and this rubric for reference.	10%
Part 1: Getting Ready [10%] • FLARE VM Setup [4%]: Comprehensive setup of FLARE VM, including the installation of the additional necessary tools. • Python HTTP Server Configuration [3%]: Successful configuration and demonstration of the Python HTTP server running. • Network Configuration Verification [3%]: Accurate display and documentation of network configuration details.	10%

Part 2: Initial Behavioral Analysis of "brbbot.exe" [20%] • Tool Utilization [5%]: Effective use of Process Hacker, Process Monitor, Regshot, and Wireshark for initial analysis. • System and Process Monitoring [5%]: Detailed monitoring and documentation of system changes and process behaviors. • Analysis and Reporting [10%]: In-depth analysis of system changes, network traffic, and process behaviors, including comprehensive reporting**.**	20%
Part 3: Intercepting "brbbot.exe" Network Traffic [20%] • ApateDNS and Wireshark Setup [5%]: Correct configuration of ApateDNS and Wireshark to capture and analyze "brbbot.exe" network traffic. • Network Traffic Analysis [10%]: Detailed examination and interpretation of DNS queries and HTTP traffic related to "brbbot.exe". • Analysis Reporting [5%]: Insightful analysis reporting, highlighting any significant findings related to network traffic interception.	20%
"Us vs. the CAT" round 3 **Show the Cat who is in control!**	
Part 4: Dynamic Analysis of "GreenCat" Malware • Malware Behavior Without C2 Server [10%]: Identifies and analyzes "GreenCat's" behavior and attempted communications in the absence of a C2 server. • System and Network Changes [10%]: Observes and documents significant system and network changes initiated by "GreenCat".	20%
Part 5: Who is in control - The Puppeteer Takes the Stage C2 Server Setup and Operation [10%]: • Demonstrates the successful setup and operation of the Python C2 server, capturing "GreenCat's" communication attempts. • Command Execution and Malware Response [10%]: Analyzes and documents "GreenCat's" responses to C2 commands, focusing on the malware's actions and any system changes.	20%
Bonus: Enhance your analysis by incorporating a broader set of tools, including: 1. Utilizing API Logger and API Monitor for detailed tracking of API calls. 2. Employing Netcat for advanced network monitoring. 3. Applying INetSim to simulate internet services in a controlled lab environment. 4. Utilizing Cuckoo Sandbox for automated malware analysis. • Note: ChatGPT and other LLM models are not applicable for use in this context	10%

Good Luck with the GreenCat

Part 1 [10%] Getting Ready:

Objective:

Prepare your reverse engineering environment by installing the FLARE VM, additional necessary tools, Python 3, and configuring a Python HTTP server. Finally, demonstrate your setup by displaying network configuration details.

Instructions:

Install FLARE VM:

- Access the FLARE VM GitHub repository: https://github.com/mandiant/flare-vm
- Follow the installation instructions provided in the README file. This involves downloading a Windows 10 or later VM, installing the FLARE VM package, and configuring the environment as per the documentation.
- Ensure your VM has network access and can reach the internet for software updates and tool installations.

Install Additional Tools:

ProcDOT:

- Download ProcDOT from its official source: https://www.procdot.com/
- Follow the installation or deployment instructions provided by ProcDOT.

ApateDNS:

Similarly, download ApateDNS from its official or a trusted source: https://fireeye.market/apps/211380

Install Python 3:

- FLARE VM might already include Python. Verify the installed version by opening a command prompt and typing ***python --version***.
- If Python 3 is not installed, or you need a different version, download the latest Python 3 installer from the official Python website.
- Run the installer, ensuring you select the option to add Python to your PATH variable.

Configure and Start a Python HTTP Server:

- Open a command prompt with administrative privileges.

- Navigate to the directory you wish to serve (this can be any directory containing files you want to be accessible over the network).
- Run the Python HTTP server on port 80 (requires admin privileges):

python -m http.server 80

- Note: Using port 80 requires administrative permissions due to it being a well-known port. If there are issues, ensure you are running the command prompt as an administrator.

Testing the webserver:

Create the HTML File:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>FLARE VM Server Test</title>
</head>
<body>
   <h1>Welcome to FLARE VM HTTP Server</h1>
   <p>This is a test page to verify that your Python HTTP server is running correctly on your FLARE VM.</p>
   <p>If you can see this message, it means your server setup is successful!</p>
<p> members:
<p> student name 1
<p> student name 2
<p> student name 3


</body>
</html>
```

Open a text editor of your choice (e.g., Notepad, Visual Studio Code, Sublime Text).

Copy the above HTML code and paste it into your text editor.

Save the file as index.html in the directory you are using as your server's root. For example, if you have chosen **C:\Users\vboxuser\Desktop\webserver** as your directory, save the file there.

Start the Python HTTP Server:

If your server isn't running already, open a command prompt with administrative privileges, navigate to your server directory, and start the server using:

python -m http.server 80

Ensure that Python is added to your system's PATH, and you're using Python 3.

Access the Page via a Web Browser:

Open a web browser of your choice.

In the address bar, enter **http://localhost** or **http://127.0.0.1** and press Enter.

If your server is set up correctly, you should see the Welcome message from the index.html file displayed in your browser.

Display Network Configuration:

In the command prompt, type ipconfig /all and press Enter.

This command will display all network interfaces and their configurations, including the IP address, subnet mask, and default gateway. Students should identify the active network adapter (usually one with an IPv4 address, subnet mask, and default gateway listed and not labeled as "Media disconnected").

Submission Guidelines for Homework 3: Part 1

You are required to submit evidence of successfully completing the setup phase of your reverse engineering environment. Specifically, you need to provide screenshots showing the following:

- **ProcDOT and ApateDNS Installation:**
  A screenshot showing the successful installation of ProcDOT.
  A screenshot showing the successful installation of ApateDNS.
- **Python Installation:**
  A screenshot showing the installed version of Python on your system. This can be obtained by opening a command prompt and typing python --version.
- **Python HTTP Server:**
  A screenshot of the command prompt running the Python HTTP server on port 80. Ensure the command used and the server running messages are visible.
- **Webpage Display:**

Modify the provided index.html file to include the names of all group members at the bottom. Save and refresh your web browser to display the changes.
A screenshot of the webpage accessed through your web browser showing the custom message with the group members' names. This verifies your server is correctly serving pages.

- **Network Configuration:**
  A screenshot of the output from running ipconfig /all in a command prompt window. Highlight or clearly indicate your IP address, subnet mask, and gateway in the image.

Important Note

Before proceeding further, it's crucial to take a snapshot of your machine at this current stage. This snapshot will serve as a clean baseline for the analysis moving forward. It is essential not to conduct a malware analysis session on top of a previous one, as this can lead to contaminated results and potential misinterpretations of the malware's behavior. Always revert to this snapshot before starting a new analysis segment to ensure the integrity and accuracy of your findings.

Part2: Initial Behavioral Analysis of brbbot.exe

Objectives:

- Understand the essential aspects of behavioral analysis for malicious Windows programs.
- Learn to effectively use key behavioral analysis tools available in the FLARE VM environment.

Preparation:

Ensure your FLARE VM is fully set up as per the instructions provided in the course materials and part 1 of this assignment. This includes having all the necessary tools installed and ready for use. If you haven't installed FLARE VM yet, refer to the FLARE VM GitHub repository for installation instructions.

Required Tools:

- Process Hacker
- Process Monitor (Procmon)
- Regshot
- Wireshark
- ProcDOT

Exercise Steps:

Launch Process Hacker and Process Monitor:

- Start by launching Process Hacker on your FLARE VM. Review the current processes to familiarize yourself with the system's normal state.

- Open Process Monitor next. Pause the capture (Ctrl+E) and clear the previous log file (Ctrl+X) to start fresh.

Take the First Snapshot with Regshot:

- Open Regshot and take the initial system snapshot. Select 1st shot > Shot to capture the current state of the registry and filesystem.

Prepare for Network Capture:

- If using Wireshark within the FLARE VM or an alternative setup, prepare it for capturing. Start Wireshark and select the appropriate network interface for monitoring (select all interfaces).

Activate brbbot.exe and Monitor its Behavior:

- With Process Monitor active, start monitoring (Ctrl+E) and execute **brbbot.exe** on your VM.
- Observe the brbbot.exe process within Process Hacker to understand its initial actions.

Terminate brbbot.exe and Complete Captures:

- Use Process Hacker to terminate the brbbot.exe process.
- Stop the capture in Process Monitor (Ctrl+E) and save the log file.
- Stop the network capture in Wireshark.

Analyze Changes with Regshot:

- Take the second snapshot with Regshot (2nd shot > Shot) and compare it to the initial one by selecting Compare. Review the report for any significant changes.

Analyze Process Monitor's Log with ProcDOT:

- Open ProcDOT and load the Process Monitor log file. Set brbbot.exe as the initial point of interest and generate the activity diagram.

Review Network Traffic in Wireshark:

- Examine the captured network traffic in Wireshark, focusing on any suspicious DNS requests or other network activities related to brbbot.exe.

Submission Requirements:

Students are required to submit a comprehensive report that includes:

1. A brief description of each step undertaken during the exercise.

2. Screenshots documenting the execution of each step, including the findings in Process Hacker, Regshot before and after comparisons, the ProcDOT analysis, and relevant Wireshark traffic.

3. A summary of the behavioral analysis findings, highlighting any indicators of malicious activity observed during the exercise.

Part 3: Intercepting brbbot.exe's Network Traffic

Objectives:

- Understand the process of examining and redirecting a malware specimen's network traffic within a malware analysis lab setup.
- Learn how to modify the lab environment to uncover additional malware behaviors.

Preparation:

- Ensure brbbot.exe is Not Running: Confirm that brbbot.exe is not currently active on the FLARE VM. Terminate the process if it's running.
- Configure DNS Using ApateDNS: Adjust the FLARE VM's network settings to direct all DNS queries to 127.0.0.1, using ApateDNS to handle these queries. This setup simulates a controlled DNS environment for the analysis.

Exercise Steps:

Launch ApateDNS:

- Open ApateDNS on the FLARE VM. Configure it to respond to any DNS queries with 127.0.0.1, capturing the malware's DNS requests.

Start Capturing Network Traffic with Wireshark:

- Initiate Wireshark on the same FLARE VM. Begin a new capture session, ensuring the correct network interface is selected for monitoring the malware's traffic.

Infect the System with brbbot.exe:

- Execute brbbot.exe to simulate infection and initiate the malware's behavior while capturing the network activity.

Launch the Web Server:

Start the web server on the FLARE VM by using the command

```
python -m http.server 80
```

This step is crucial for simulating a web environment that brbbot.exe might interact with, specifically targeting the URL http://brb.3dtuts.by.

Visit http://brb.3dtuts.by:

Using a web browser on the FLARE VM, navigate to http://brb.3dtuts.by to ensure the web server and ApateDNS are correctly configured and operational. This step confirms that brbbot.exe's network traffic can be redirected and captured for analysis.

Analyze the Captured Traffic:

After allowing sufficient time for brbbot.exe to exhibit its network behavior, terminate the process.

Stop the Wireshark capture and review the collected packets. Focus on DNS queries handled by ApateDNS and any HTTP traffic, particularly communications with http://brb.3dtuts.by.

Save the Encoded Payload:

If brbbot.exe engages in HTTP communication, use Wireshark to locate and follow the TCP stream of this session. Look for and save any encoded payloads for further analysis.

Detailed Instructions:

- Steps 1 & 2: Verify ApateDNS is redirecting DNS queries by using nslookup in the FLARE VM. Start capturing network traffic with Wireshark, filtering for DNS and HTTP protocols to streamline the analysis.

- Step 3: Monitor brbbot.exe's execution closely using tools like Process Explorer for any spawned processes or significant system changes.

- Step 4 & 5: Ensure the web server is active and accessible. This setup is pivotal for capturing brbbot.exe's interaction with simulated web resources.

- Step 6 & 7: Focus on the malware's network behavior, documenting any DNS requests for brbbot.exe-specific domains and the subsequent HTTP traffic. Save payloads encountered for decoding and analysis.

Detailed Instructions for Analysis:

- Suspicious Activities Captured by Process Monitor: Run Procmon before executing brbbot.exe to capture all system events. Look for abnormal activities such as unusual file system access, registry modifications, or network traffic.

- Files Created by the Executable: Use Procmon to filter and identify files created by brbbot.exe (if any). Note the names, types, and locations of these files.

- Remote Resources Interaction: Utilize ApateDNS alongside Wireshark to observe and analyze brbbot.exe's DNS queries and network communication protocols. Focus on the content of the data being sent and the nature of the interactions with simulated remote resources.

- Creation of Other Processes: Employ Process Explorer to monitor any processes spawned by brbbot.exe. This can indicate the malware's spreading mechanism or execution of additional malicious payloads.

- Permanent Changes to the OS: Use Regshot to take before and after snapshots of the system's registry and file system upon executing brbbot.exe. Compare these snapshots to identify any lasting changes made by the malware.

Submission Requirements:

- Students are expected to compile a comprehensive report including:
- Screenshots demonstrating the setup and configuration of ApateDNS and the web server.
- Wireshark evidence of DNS and HTTP traffic, particularly interactions with http://brb.3dtuts.by.
- Analysis of the captured network traffic, highlighting the malware's communication patterns and any encoded data transmitted.
- A discussion on the challenges encountered during the setup and analysis process and how they were resolved.

"Us vs. the CAT" round 3

Show the Cat who is in control!

Part 4: Dynamic Analysis of "GreenCat" Malware

Objectives:

- Determine the dynamic server addresses "GreenCat" attempts to communicate with in the absence of a C2 server.
- Apply behavioral analysis tools to understand "GreenCat's" initial actions.

Exercise Steps:

Preparation and Tool Setup:

- Utilize the snapshot to revert back to the baseline state.
- Ensure FLARE VM is fully configured with necessary tools: Process Hacker, Process Monitor (Procmon), Regshot, and Wireshark.
- Run Process Hacker and Procmon to monitor system processes and activities. Pause and clear previous captures in Procmon.
- Configure DNS Using ApateDNS: Adjust the FLARE VM's network settings to direct all DNS queries to 127.0.0.1, using ApateDNS to handle these queries. This setup simulates a controlled DNS environment for the analysis.

Initial System Snapshot:

- Use Regshot to take a baseline snapshot of the system's state.

Network Capture Setup:

Initiate Wireshark to capture all network traffic, particularly focusing on DNS and HTTP protocols.

Executing "GreenCat":

- Launch "GreenCat" and monitor its activities via Process Hacker and Procmon, noting any immediate behaviors.

Analyzing System and Network Changes:

- After sufficient observation, terminate "GreenCat".

- Analyze changes using Regshot, Procmon logs with ProcDOT, and network traffic captured by Wireshark.

Deliverables of this part of the report:

- A detailed report section documenting the setup, execution, and findings from each tool, emphasizing "GreenCat's" attempted communications and any significant system changes. In your report, specifically address the following key points to clarify the nature of "GreenCat's" communication strategies:
- **C2 Server's URL:** Identify and document the URL "GreenCat" is programmed to contact. This involves analyzing network traffic to discern any DNS queries or HTTP requests made by the malware in an attempt to locate its command and control server.
- **Communication Port:** Determine the specific port number "GreenCat" uses to communicate with the C2 server. This information can be gleaned from examining the details of network packets captured during the malware's active communication phase. Understanding the port used is crucial for comprehensively mapping out the malware's communication pathway and could provide insights into the protocols employed by "GreenCat".

Part 5: Who is in control - The Puppeteer Takes the Stage

"*With the stage set and your tools at the ready, you breathe life into the Python-based C2 server, stepping into the role of **the puppeteer**. Each command sent through the server's interface is a string pulled, a test of "GreenCat's" obedience. From basic inquiries to complex operations, you observe, analyze, and record, turning "GreenCat's" responses into data, into understanding.*

As you command and "GreenCat" responds, the lines between controller and controlled blur. The digital cat, once a predator lurking in the shadows of the internet, now dances at the end of your strings. This isn't just analysis; it's a ballet of bits and bytes, a performance where you lead, and "GreenCat" follows."

Objectives:

- Set up and operate a Python-based C2 server to communicate with "GreenCat."
- Analyze "GreenCat's" behavior in response to specific C2 commands.
- Utilize behavioral analysis tools to monitor and record "GreenCat's" actions, including ApateDNS, to monitor, capture, and interpret "GreenCat's" activities and communication patterns.

Preparation:

- Ensure FLARE VM is fully configured with all necessary tools.
- Download the provided Python C2 server script, SSL certificate, and key from the course GitHub portal.

Required Tools:

- Process Hacker
- Process Monitor (Procmon)
- Wireshark
- Python C2 server script
- ApateDNS

Exercise Steps:

C2 Server Setup:

- Download the Python C2 server script, certificate, and key from the course GitHub repo.
- Run the C2 server using the command provided in the instructions. Ensure it's listening for incoming connections from "GreenCat."

Monitoring Setup:

- Launch Process Monitor and begin capturing system events. Clear any previous logs.
- Open Wireshark and start capturing network traffic. Focus on the interface that "GreenCat" will use for communication.

Run "GreenCat":

Execute "GreenCat" on the FLARE VM. Monitor for an initial connection attempt to the C2 server.

C2 Server Interaction:

Once "GreenCat" establishes communication with the C2 server, begin sending the listed C&C commands through the server's command interface.

Observe and document "GreenCat's" responses and system changes in response to each command.

C&C Commands to Execute:

- shell, exit, list, list /p, list /s, list /d
- getf f ..., putf f ..., quit, 1, --sleep--
- start, start /p hello, start /s hello, whoami, v
- pidrun 30 funcs, geturl https://www.google.com google, hello

Detailed Instructions for Analysis:

- Process and Behavior Monitoring: Use Process Hacker to track processes initiated by "GreenCat" following each C2 command. Utilize Process Monitor to observe file and registry changes.
- Network Traffic Analysis: With Wireshark, examine "GreenCat's" network communications with the C2 server, especially the data exchanged following each command.
- Registry and File System Changes: Employ Regshot for before-and-after snapshots to identify permanent changes made by "GreenCat."

Submission Requirements:

1. A summary of "GreenCat's" activity in response to each C&C command, supported by screenshots or logs from Process Hacker and Process Monitor.
2. Analysis of network traffic captured during the interaction with "GreenCat," highlighting any significant data exchanges.
3. Observations of registry and file system changes resulting from "GreenCat's" responses to C&C commands.

Download Instructions:

Step 1: Cloning the Repository

- Open a terminal window or command prompt on your system.
- Navigate to the directory where you wish to clone the repository using the cd command.
- Enter the following command to clone the repository:
  git clone https://github.com/RITDubaiCSEC202/HW3.git
- Wait for the cloning process to complete. This command creates a local copy of the repository in your specified directory.

Step 2: Accessing Malware Sample

1. Once the repository is cloned, navigate to the folder containing the malware samples within the cloned directory. This can be done through the file explorer or terminal:
   cd HW3
2. You will find the malware files compressed and password-protected to prevent accidental execution or detection by antivirus software.

Step 3: Extracting Malware Samples

Use your preferred archive manager capable of handling ZIP (or the relevant archive format) files. When prompted for a password, use **infected** as the password to extract the files. Ensure you're doing this in a secure, isolated environment, such as a virtual machine dedicated to malware analysis, to prevent any risk to your system or network.

Safety Precautions:

Do Not Execute the Malware on a Production System: Always work within a controlled, isolated environment that's disconnected from any production or personal networks. Virtual machines configured for malware analysis are ideal for this purpose.

Password Protection: The use of the password "infected" for malware files is a standard practice to prevent accidental execution and to bypass certain automated scanning tools or antivirus software during transfer or storage.

Maintain Operational Security: Ensure that all interactions with the malware, including analysis and data extraction, are conducted within the confines of your secure environment. Be mindful of the malware's capabilities to communicate over networks or perform system modifications.