CSCE 5550 Spring 2025 Project “Ransomware”

Due on April 28, 2025, at 11:59pm

This project will be focused on a study of ransomware attacks and the related mitigation techniques. You will be working on a virtual machine (VM) to ensure the required isolation. You must not work on the project directly on your host machine as it may cause damage to your operating system and personal data.

For the attack part, you will develop your own ransomware tool, which will infect your guest OS in the VM and encrypt a designated directory. For the cybersecurity part, you will detect the ransomware and take action to mitigate the problem. For that purpose, you need to monitor the activities on your guest VM, detect if there is a ransomware, block it, and attempt recovery.

Group work: For this project, you may build a group of up to 5 students. Each student is expected to provide an adequate contribution. Normally, all the group members will receive the same grade based on the project evaluation. Underperforming students may have their grade reduced at the discretion of the instructor.

Group formation: due 02/16

Alternative project proposals (optional): due 02/16

Project Plan

You will follow the steps listed below. You are expected to submit a short memo (DOCX or PDF format) describing your progress every week—via Canvas. This memo will be updated as your project proceeds—submit the updated versions via Canvas. It will be used by the instructor to guide your work on the project. Also, it will serve as a basis of your project report.

Feel free to propose any methods and approaches for every step. You may use an operating system of your choice for the victim machine—note however that it must be available as a guest OS VM.

The instructor will check your project plan and may recommend some revisions for it, to avoid similarities with other groups.

1. **Research on ransomware techniques:** First, you will do a basic literature survey on ransomware to figure out how this type of malware works. There are multiple approaches for building ransomware. You may select one of them or come up with your own.

   Preparing the project plan. For each of the Steps 2-6 below, select a specific method, which you will implement for your project components. Submit your project plan as a short memo via Canvas. **Due: March 10, 2025**.
2. **Action:** In this step, you will implement your own encryption code.
   You may use a programming language of your choice. Then, you will need to choose a cryptographic library. Then, you will choose an appropriate encryption algorithm. Alternatively, you may use a cryptographic utility and write a shell code.

Note: Your encryption component must encrypt a directory recursively, that is it must encrypt all its subdirectories and files in them.

Suppose that the victim machine has a directory “critical” (create it). Then, place a copy of this project description file into it. Then, in the directory “critical”, create subdirectories “lab1”, “lab2”, “lab3”, and place the respective lab manuals into them. This the directory structure which your ransomware will attack. Specifically, your encryption component will encrypt this directory structure.

Next, you will construct the decryption component and verify that the result can be decrypted. Note: the decryption component will be transferred to the victim once the ransom is paid.

Write the encryption/decryption component and submit its description as a short memo via Canvas. **Due: March 24, 2025**.

3. **Infection:** In this step, you need to implement a method for infecting a victim. Example: A victim may open a phishing email which contains a malicious executable/script as an attachment.

   Note: You need to come up with your own proposed method of infection.

   For this step, you may use existing cybersecurity attack toolsets. However, you need to implement your method and demonstrate that it is successful.

   Design the infection method and submit its description as a short memo via Canvas. **Due: March 31, 2025**.

4. **Monitoring:** To be able to detect the ransomware, we need to be able to monitor your environment. Example: OSSEC Host IDS has a file integrity monitoring functionality. Provide your own detection component. You may use external libraries or tools to help your monitoring activities, but you will need to demonstrate that your detection component works. The monitored data must be also logged in a database or other structured file.

   Implement the monitoring component and submit its description as a short memo via Canvas. **Due: April 7, 2025**.

5. **Detection:** In this stage, you will analyze the data collected by the monitoring component. You need to define a policy: which operations on files are permitted and which are not (and by whom). Decide on how to detect the policy violation, i.e., create a rule which triggers your mitigation measure. For example, you may check the data obtained in the previous step against some set of rules, match it to a certain pattern, or even use machine learning. An important point is to demonstrate that your tool successfully detects ransomware, without creating false positives for legitimate processes.

   Implement the detection component and submit its description as a short memo via Canvas. **Due: April 14, 2025**

6. **Mitigation:** In this stage, you will define a countermeasure which is triggered by the above-mentioned rule. Example: an alert is sent to the system administrator, a process violating the policy is suspended or terminated. All aspects of the mitigation process, including prevention methods must be discussed in detail. A “defense in depth” approach must be used: consider what to do when your countermeasure fails.
   You must discuss the backup and recovery options.

   Implement the mitigation component and submit its description as a short memo via Canvas: **Due: April 21, 2025**.

7. **Report Preparation:** Compile all your memos into the project report.
   Collect the deliverables following the guidelines below. **Due: April 28, 2025**.

Deliverables

At the end of the project, you will submit the following three deliverables:

- **Project code:** It will include all the code written by the students along with a README file that shows the step-by-step instructions on how to duplicate your environment.
  It must have information on the environment, libraries needed, the dependencies, and others. Your code must be properly commented.

  Note: Maximum size of the submission is 50 MB. The submission must include only your own code. If you prepared your own VM, share it via OneDrive and provide a link in the README file. Also use README for providing links to all the libraries and third-party tools used for your project.

- **Demo video:** It will include a demonstration of your implementation and its features, showing how it works, what it does, and confirming that it works successfully.
  - Length of the video: 5 to 15 minutes.
    (The evaluation will be based on the first 15 minutes.)
  - Use a video hosting service for sharing the video – provide only its link.
- **Report:** You need to follow an outline of a regular research paper.
  It should contain:

  - Abstract: A brief overview of your project. Approximately 250 words.
  - Introduction: Explaining the problem you are trying to solve, why it is an important problem, a brief overview of your approach, a brief overview of your accomplishments, etc. This section should take approximately one page.
  - Related works: A brief description of other studies and articles that you have found. You need to present what the others have done, how, what they have

observed, etc. At least 4-5 sources need to be presented and summarized. This section should take approximately one page.

- Approach: For each step of the project, you need to explain your approach, your architecture, steps, implementation details, and so on. This is the main part of your report, and it should take around 2 pages.
- Results: You need to present your results here. This section should take approximately one page.
- References: Add the references (bibliography) in the IEEE format.

Rubric: Each of the components (action, infection, monitoring, detection, and mitigation) is worth 20 points. Up to 5 extra points may be given for the technical quality of the report. The total grade will not exceed 100 points.