- **If you choose to do a presentation project**
  - **Please email me 3 project topics you are interested in by Nov. 3 2022 (Thursday), indicating your first, second, and third choices. You can also come up with your own cybersecurity-related topic/paper – in this case, please email me the paper you would like to present by Nov. 3 2022.**
- **If you choose to do a systems project or a K-12/community outreach project**
  - **Please email me the project topic by Nov. 3 2022 (Thursday)**
- **If you choose to do a programming project**
  - **Please do NOT email me**

## Presentation Projects
## Presentation: end of November/beginning of December
## Presentation slides submission: 11:59pm Dec. 8

In the presentation project, you will present one/two papers using powerpoint slides. Each presentation should last about 20-25 minutes. The presentation will be scheduled at the end of Nov. or beginning of Dec. (You will receive an email from me about the schedule). The presentation slides should be uploaded to brighspace by Dec. 8.

1. Ransomware I

   https://cdn2.hubspot.net/hubfs/5218324/DE-PDFs-web/DR-EN-definitive-guide-to-ransomware_12-03.pdf

   https://www.lehman.edu/itr/documents/AST-0168043_cso_ransomware_survival_guide_knowbe4-upload.pdf

2. Ransomware II

   https://www.cisa.gov/sites/default/files/publications/CISA_MS-ISAC_Ransomware%20Guide_S508C_.pdf

   https://simplehelix.com/wp-content/uploads/2020/09/the-complete-guide-to-ransomware-white-paper.pdf

3. Solarwind Attack

   Study online articles on Solarwind attack and give an about 30-min presentation.

4. Identity Theft I

   https://www.codb.us/DocumentCenter/View/10521/identitytheft?bidId=
   https://benefits.uasys.edu/media/1788/id-watchdog-guide-to-identity-theft.pdf

5. Identify Theft II
   https://cippic.ca/sites/default/files/bulletins/Techniques.pdf

6. Online Phishing

   https://www.stage2data.com/wp-content/uploads/2020/02/S2D-ebook-comprehensive-guide-to-phishing-v3.pdf
   https://www2.deloitte.com/content/dam/Deloitte/sg/Documents/risk/sea-risk-cyber-101-part10.pdf

7. Social engineering

https://comum.rcaap.pt/bitstream/10400.26/38593/1/paper17.pdf

https://nsarchive.gwu.edu/sites/default/files/documents/4060577/CERT-UK-An-Introduction-to-Social-Engineering.pdf

8. Deep Reinforcement Learning for Cyber Security

https://arxiv.org/pdf/1906.05799.pdf

9. The threat of Offensive AI to Organizations

https://arxiv.org/pdf/2106.15764.pdf

10. HARPO: Learning to Subvert Online Behavioral Advertising

https://web.cs.ucdavis.edu/~zubair/files/harpo-ndss2022.pdf

11. SoK: Explainable Machine Learning for Computer Security Applications

https://arxiv.org/pdf/2208.10605.pdf

12. LEMNA: Explaining Deep Learning based Security Applications

https://gangw.cs.illinois.edu/ccs18.pdf

13. Deep Reinforcement Learning for Cybersecurity

https://edge.seas.harvard.edu/files/edge/files/deeprl_cybersecurity.pdf

14. User Preference-aware Fake News Detection

https://arxiv.org/pdf/2104.12259.pdf

15. Graph Neural Networks with Continual Learning for Fake News Detection from Social Media

https://arxiv.org/pdf/2007.03316.pdf

Guideline (ppt slides)

- Motivation: E.g. Why role-based access control?
- Background: E.g. Syntax of xml
- Technical details
- Use enough examples/pictures to illustrate technical details.
- Related work

Guideline (presentation)

- Present slowly and clearly.
- Do not directly read everything from slides.
- Do not read from the presentation notes.
- Use examples to illustrate technical details.

# Programming Projects

**No presentation**

**Submission deadline: 11:59pm, Dec. 8**

**(source code + readme)**

The programming project is done individually or by a group of 2 students. You can use the existing implementation of RSA, AES/3DES. SHA1 or MD5 (e.g. provided in java.security, openssl, etc), if necessary. You can use C, C++, Java, or Python. You are not required to implement a graphical user interface. **If you choose to do the programming project alone, you will get 10 points extra credits.**

In this project, you will implement an ***iterative*** secure banking system consisting of a bank server and multiple clients (i.e., users' computers). Each bank user can deposit money to his/he raccount. The bank server maintains a file "passwd" that stores users' ID and **hashed** passwords. Passwords are hashed using SHA1 or MD5. Assume that there are three users: Alice, Bob, and Tom. Alice's ID is alice and password is 1234. Bob's ID is bob and password is 5678. Tom's ID is tom and password is 9012.

The bank server maintains a **file "balance",** which stores the balance of the account of each user. Initially, the file "balance" contains the following information (i.e., the initial balance for all users is $10000):

alice 10000
bob 10000
tom 10000

Public key encryption is used for confidentiality. Let **Kpub** and K**prb** be the public and private key of the bank server, respectively. Assume that all users have bank's public key. The public and private keys can be generated manually and stored on the disk.

The client is invoked as:

client <*Bank server's domain name*> <*Bank server's port number*>

The bank server is invoked as:

bank <*Bank server's port number*>

The detailed steps are given below:

**S1:** The client connects to the bank server.
**S2:** The client prompts the user to enter his/her ID and password.
**S3:** The client sends E(Kpub, ID||password) to the bank server, where id and password are the user's ID and password entered, respectively.
**S4:** The bank decrypts E(Kpub, ID||password) using Kprb and gets the ID and the password. Next, the bank computes the hash of the password and compares it against the hash stored in file "passwd". If the ID and the password are correct, then the bank sends 1 to the client; otherwise 0.
**S5:** If the password is incorrect, the client prompts the user to enter the id and password again. If the password is correct, the client displays the following message:

Your account balance is <balance>. Please select one of the following actions:
1. Deposit
2. Exit

In the above, <balance> is the balance obtained from file "balance".
**S6:** If the user selects 1, then the user is prompted to enter the amount of money deposited. The client then sends them to the server. The server then updates the "balance" file, and sends 1 to the client. Next, the client displays the main menu with the updated balance:

Your account balance is <balance>. Please select one of the following actions:
1. Deposit
2. Exit

**S7.** If the user selects 2, then the client sends 2 to the server, and both the client and the server close the connection socket, and the server continues listening for connection.

**Note: To simplify your program, you do not need to handle any errors (e.g., the user enters 4, instead of a number 1, 2, or 3).**

***Submission guideline***

- Please hand in your **source code, public/private keys, files passwd and balance,** and a **Makefile (C/C++/Java)** electronically (**do not submit .o or executable code**). Please make sure that this code compiles and runs correctly on remote.cs.binghamton.edu.
- Write **a README** file (text file, do not submit a .doc file) which contains
  - The name and email address of the group members.
  - The programming language you use (C/C++/Java/Python)
  - Code for performing encryption/decryption
  - Whether your code was tested on remote.cs.binghamton.edu.
  - How to execute your program.
  - Anything special about your submission that the TA/grader should take note of.
- Place all your files under one directory with a unique name (such as proj-[userid] for project, e.g. proj-pyang).
- Tar the contents of this directory using the following command. **tar –cvf [directory_name].tar [directory_name]**
- E.g. tar -cvf proj-pyang.tar proj-pyang/
- Use brightspace.binghamton.edu to upload the tared file you created above.

***Grading Guideline***

- Readme: 5'
- Makefile (C/C++/Java): 5'
- Encryption/decryption: 20'
- Hashed password: 10'
- Other functionality: 60'
- Extra-credits (work alone): 10'

## Systems Projects

**Presentation + demo: Dec. 8 (in class)**
**Submission deadline: 11:59pm Dec. 8**

**Students will give presentations and show demos (if applicable). Students will also need to submit presentation slides/videos by 11:59pm on Dec. 8.**

### 1. Buffer Overflow Attack

This project can be done individually (10 points extra credits) or by a group of two students. In this project, you will perform and demonstrate the buffer overflow attack. You can use any example to demonstrate the attack.

### 2. Kernel Rootkit

This project is done individually. Download a windows rootkit that enables attackers to hide files or processes, and demonstrate how to do it. You will need to first install a virtual machine (e.g. virtualbox) and then execute the rootkit inside the virtual machine.

### 3. Virus (language: C)

This project is done individually (10 points extra credits) or by a group of two students.

Design and implement a virus that can infect all executable files under a specific directory. Note that the program can still execute after the program is infected with the virus. When the infected program executes, the virus will execute first, and then the program.

### 4.Capture the Flag Challenge

Solve one challenge in https://w3challs.com/, and demonstrate and explain how to solve the challenge. This project is done individually.

## K-12/Community Outreach Projects

**Presentation + demo: Dec. 8 (in class)**
**Submission deadline: 11:59pm Dec. 8**

### 1. K-12 Education video

This project is done individually.

Create 1-3 videos (around 15 minutes in total) for K-12 cybersecurity education. Sample topics include simple encryption/decryption techniques, online safety, cyberbullying, introduction to computer security, cybersecurity career. The videos should be age-appropriate and engaging.

### 2. K-12/Community Outreach Activities

This project is done individually.

Come up with ideas of TWO community/K-12 cybersecurity outreach activities (e.g., cybersecurity activities for K-12 students in school and on the community day of the engineering week). The activities should be fun and engaging. You can also search online for existing cybersecurity outreach activities and select two activities that are fun and engaging.

### 3. K-12 Online Cybersecurity Games

This project is done individually (10 points extra credits) or by a group of two students.

Develop an online cybersecurity game to teach K-12 students cybersecurity concepts. Sample topics include simple encryption/decryption techniques, threats and defense, online safety, cyberbullying, introduction to computer security. You can use any programming language for this project.