CSE 543: Computer Security
Fall'21
Project 3 - Research Project (300 points)

1 Introduction

In this month long project, students will conduct research on network or systems security in teams of 1-2 people under the instructor's supervision. All teams and project topics must be approved by the instructor. The outcome of this project will be similar to a conference/workshop-style paper of 6-8 pages maximum in which you build an argument for studying the open research problems that you advocate investigating — identify why this problem is important and why you believe it is solvable in the near future. The grade will be based on the novelty, depth, correctness, rigor of execution, clarity of presentation and effort.

2 Project Theme

"Thinking like an adversary" is critical to understand— (1) the current security and privacy posture of a system; (2) how the existing security policies can be broken; and (3) how to enforce better and robust security policies. In this course, each project, therefore, should be aligned with the following theme: *"security analysis of an existing system"*. The overarching goal of each project would be to design and implement a security evaluation framework for an application/protocol/system. Students may achieve this goal in one of the following ways:

1. Reproducing an existing attack (part 1) and identify new ones (part 2): As a first step, you may reproduce a known attack on an existing system. Your task is to understand a known vulnerability and corresponding attack, and write an original exploit. You should not contact the paper authors for the exploits or are not allowed to use any pre-packaged attack tools. You may, however, use any open-source and general purpose tools (e.g., gdb, wireshark, and packet sniffers) and scripts. Note that some vulnerabilities may have been fixed in newer software or libraries (e.g., TCP or TLS), so you may need to target an old version. For part 2, you will identify new attacks based on your experience with part 1. It is okay if you cannot identify new attacks but you need to demonstrate tangible efforts (e.g., an attempt to replay a packet) to identify new ones.

According to the course ethics policy, you must not test your attack against systems owned by other people. If you need to set up a victim/target, set up your own isolated device/machine/VM as the victim. You can perform the attack only against your own device/machine.

You need to present a demo of the attacks in which you will demonstrate— (1) what is the vulnerability (2) how it can be exploited and how the attack works and (3) what are the implications of the attack. You will also present how you implemented the exploits, set up the attacks, and while doing so what technical challenges you run into.

2. Design and implement a systematic security analysis framework: You may also choose to design and implement a systematic security analysis framework for an application/protocol/system based on formal verification, program analysis (e.g., static and dynamic program analysis), and software testing (e.g., fuzzing).

You need to present a demo of your security analysis tool in which you will demonstrate— (1) the type of vulnerabilities (e.g., low-level memory corruptions or functional bugs) your tool is designed to evaluate (2) the underlying scientific techniques enabling the security analysis of the target application, and (3) the findings, i.e., the known and unknown vulnerabilities your analysis technique can determine. You will also present how you implemented the tool and while doing so what technicalMilestone 1: challenges you run into.

3 Project Topics

The tentative list of topics is, but not limited to, the following:

1. Network Security
2. System Security
3. Web Security
4. IoT Security (see the description below)
5. Embedded Device Security
6. Hardware Security
7. Fuzzing for Software/Hardware Security
8. Language Based Security
9. Host Intrusion Detection
10. Cloud Computing Security
11. Android System Security
12. Smartphone security
13. Blockchain Security:

4 Description of Some of the Project Topics

IoT Security: There are more IoT devices on earth than people. Most of the IoT devices have to be cheaper, smaller in size, power efficient and with minimal latency. More often than not this

leads to a trade off when it comes to security. IoT devices are one of the most vulnerable devices. For the project one can look into security of the apps that control them, firmware, access control, network protocol vulnerabilities, botnets, IoT malware, IoT crypto-miners, IoT ransomware etc. You can find some of the papers in https://github.com/nsslabcuus/Recent-IoT-Security-Paper

Fuzzing for Software/Hardware Security: Fuzzing is one of the most popular method to test security. One can do protocol fuzzing, application fuzzing, kernel fuzzing, IoT fuzzing, Smart contract fuzzing, browser fuzzing and so on. One can improve exiting fuzzers by incorporating new heuristics or methodology. A new emerging field is anti-fuzzing where one tries to make binaries fuzzing resistant (Look up AntiFuzz and Fuzzification from USENIX'19). Fuzzing has very broad applications. You can choose fuzzing for most of the fields and incorporate lots of various techniques like ML, symbolic execution, value set analysis, Taint analysis , etc. One example of fuzzing + deep learning + network/protocol security: https://dl.acm.org/doi/10.1145/3203217.3203241.

Blockchain Security: Blockchain has been a hot topic fro past few years. It all started with emergence of Bitcoin. Blockchains like Ethereum are not bring programmability to the Blockchain. One can work on the smart contracts (the program part) or protocols or decentralized application running on top of the Blockchain. One can apply forensic analysis, fuzz testing, formal analysis, etc. Following two papers are a good starting point to understand some of the landscape for Ethereum: 1) https://arxiv.org/pdf/1908.04507.pdf 2) https://eprint.iacr.org/2016/1007.pdf There have been multiple papers in USENIX Security, IEEE S&P in past 2-3 years that deal with smart contracts and blockchain in general. I recommend reading abstract of most of them if no all to get a sense of current state of the field.

5 Paper Selection for the Project

For identifying known attacks or security analysis frameworks for existing systems/networks/applications, students are encouraged to check the last few years proceedings of IEEE Security and Privacy (Oakland), ACM CCS, NDSS and USENIX Security. Students interesting in CTF and ethical hacking can also find known attacks in Blackhat or DEFCON too. A list of papers (https://syed-rafiul-hussain.github.io/index.php/teaching/cse543-f21/papers.html) from Oakland, CCS, USENIX Security and NDSS has already been compiled for you. Dsiclaimer: the list could be outdated. You may not find the most recent (late 2020,2021) papers there.

IEEE S&P (aka., Oakland):
- 2021: https://www.ieee-security.org/TC/SP2021/program.html
- 2020: https://www.ieee-security.org/TC/SP2020/program.html
- 2019: https://www.ieee-security.org/TC/SP2019/program.html
- 2018: https://www.ieee-security.org/TC/SP2018/program.html
- 2017: https://www.ieee-security.org/TC/SP2017/program.html
- 2016: https://www.ieee-security.org/TC/SP2016/program.html

CCS:
- 2021: https://www.sigsac.org/ccs/CCS2021/

- 2020: https://www.sigsac.org/ccs/CCS2020/
- 2019: https://www.sigsac.org/ccs/CCS2019/
- 2018: https://www.sigsac.org/ccs/CCS2018/index.html
- 2017: https://www.sigsac.org/ccs/CCS2017/
- 2016: https://www.sigsac.org/ccs/CCS2016/index.html

NDSS:
- 2021: https://www.ndss-symposium.org/ndss-program/ndss-2021/
- 2020: https://www.ndss-symposium.org/ndss-program/2020-program/
- 2019: https://www.ndss-symposium.org/ndss-program/ndss-symposium-2019-program/
- 2018: https://www.ndss-symposium.org/ndss2018/programme/
- 2017: https://www.ndss-symposium.org/ndss2017/ndss-2017-programme/
- 2016: https://www.ndss-symposium.org/ndss2016/ndss-2016-programme/

USENIX Security:
- 2021: https://www.usenix.org/conference/usenixsecurity21/technical-sessions
- 2020: https://www.usenix.org/conference/usenixsecurity20/technical-sessions
- 2019: https://www.usenix.org/conference/usenixsecurity19/technical-sessions
- 2018: https://www.usenix.org/conference/usenixsecurity18/technical-sessions
- 2017: https://www.usenix.org/conference/usenixsecurity17/technical-sessions
- 2016: https://www.usenix.org/conference/usenixsecurity16/technical-sessions

BlackHat:
- US'21: US:https://www.blackhat.com/us-21/briefings/schedule/index.html
- Europe'21: https://www.blackhat.com/eu-21/briefings/schedule/index.html
- Asia'21: https://www.blackhat.com/asia-21/briefings/schedule/index.html

Defcon:
- US'21: US:https://defcon.org/html/defcon-29/dc-29-schedule.html
- US'20: https://defcon.org/html/defcon-safemode/dc-safemode-schedule.html

6 Project Milestones

¡p¿ The following important dates and checkpoints are subject to changes as the semester progresses. All the changes will be notified through CANVAS. For the updated dates, please check the course schedule (https://syed-rafiul-hussain.github.io/index.php/teaching/cse543-f21/schedule.html) page.

- **October 25:** Project group information due

- **November 1 – Milestone 1 (50 points)**: Project proposal due (meet the instructor at office hours or by appointment to discuss the proposal)

• **November 2 and 4 (20 points):** Project meeting (mandatory) with the instructor to discuss the project.

• **November 18 – Milestone 2 (50 points)**: We expect **50%** of the proposed work should be completed by this milestone. The expectations for milestones 2 and 3 have been communicated with the grading report for milestone 1.

• **December 2 – Milestone 3 (100 points)**: We expect the **complete implementation** of your project and **the evaluation results**.

• **December 10 (80 points):** Final paper due and project presentation.

7 Milestone 1: Project proposal

The proposal should not exceed 2 pages (including references). Please use this ACM conference template on Overleaf for project-related submissions.

What (Problem Statement): A name for the project. Articulate the problem statement and a clear description of your goal (e.g., proposing a new attack, proposing a new security analysis technique for a particular system, proposing a new defense, improving an existing attack/defense, etc.). For part 1 of this project, which particular attack(s) do you like to reproduce?

Why (Motivation for this project): Why should we be interested in the result of the project? A detailed view of motivation and most related work.

How (Design- or implementation-level details): What resources will you use to set up the attack for part 1? What do you plan for part 2? Is there an existing code-base of any related work? How will you evaluate your results, i.e., what will be your evaluation criteria (e.g., effectiveness, efficiency, performance overhead)?

Work load and expected deliverable for other milestones: If your team has two members, who will do what? You also need to provide your expected deliverable for milestones 2 and 3 in this proposal.

8 Milestone 2: Implementation (Partial Progress)

You need to demonstrate the progress of your project's implementations. You also need to write a two-page (maximum) report of what you have done (e.g., designed, implemented and evaluated) for milestone 2. You should use the same ACM template as you did for milestone 1. There is no specific outline for the report but your report should be as detailed as possible.

9 Milestone 3: Design Overview, Implementation and Evaluation (100 points):

Your task is to implement and evaluate your proposal. You need to demonstrate some tangible results for this milestone. You need to submit a report discussing the following:

- **Design Overview:** Write a high-level overview of your proposed system/solution. You may use a diagram to summarize how your proposal works. Without giving much details of each of the component in your system, you only mention the functionalities of each major component (i.e., the building block) of your system and the flow/interactions among those components. Use appropriate diagram/figure to provide such high-level details.
- **Implementation Details and Evaluation Setup:** What software/hardware did you use/write/modify for your implementations? What does each module do in connection with the components of your system/solution?

  How did you set up the experiments for evaluation? What are the evaluation metrics? Make sure the metrics answer the research questions in Milestone 1. What dataset (if any) have you used? Mention if you have any baseline results to compare with.
- **Evaluation results:** Present the results with tables and graphs. If you have different setup for different experiments, describe experimental setup (e.g., hardware, software, and datasets used).

  For informal security analysis of your proposed defense/solution, provide justification (with text) on how your defense/solution can prevent the attacks that you mentioned in the threat model.
- **README and inline documentation:** You need to write a detailed instruction in the README.md file by following which one will be able to set up your project, execute your code and validate some of the results you submit. Failure to provide sufficient information for reproducing your results may cause some penalties. Also, your code should have inline documentation. Check here to know about the inline documentation.
- **Submission instructions:** Submit the design overview, implementation details and evaluation results in a PDF to the CANVAS portal. You need to share your code with me on GitHub. You will be added as a collaborator in `https://github.com/psu-cse543-f21` through which you can check in your code.

You may reuse the reports for milestone 1 and milestone 2. Submitting the same milestone 2 report for this submission will not work.

10 Final Report

You need to submit a minimum 8-page (without references) conference- or workshop-style paper. A typical conference/workshop paper has the following sections: Abstract, Introduction, Background, A High-level Design Overview, Design Details, Implementation, Evaluation, Discussion and Limitations, Related Work, and Conclusion and Future Work. To know what to write in each section, you may take a look at any paper (e.g., https://syed-rafiul-hussain.github.io/wp-content/uploads/2020/08/ATFuzzer-journal.pdf) published in IEEE S&P, CCS, NDSS, and USENIX Security conferences.

Find below the guideline for how to write the important sections of your paper.

Abstract: An Abstract should be relatively short: one paragraph, around 300 words. A wandering and off-topic abstract can confuse the reader. Consider the following template for writing an abstract (approximately one sentence per point):

- Area
- Problem
- Solution Idea
- Methodology
- Results
- Take-away

Introduction: The abstract and introduction are two important sections of your paper. An introduction is a longer version of this same outline, often dedicating about a paragraph per point. This is where you motivate the research problem you are solving, clearly spell out the problem statement, justify why the existing solutions are not enough, pitch your proposals, provide high-level intuitions of your ideas, i.e., why one should believe that your idea will work and present the high-level results. If you fail to get the reviewer's interest in the abstract and introduction, it is hard to recover. Furthermore, the abstract and introduction are the most-read parts of the published paper.

Background: Provide background knowledge that is necessary to read and understand your paper. Do not provide the details that have never been used or referred to in your paper. E.g., you are writing a paper on X.509 certificate validation. To explain that, you do not need to provide details about how public and private keys are generated or what is the length of the key.

Design Overview: In this section, first discuss the security assumptions, capabilities, and goals of the adversary and the trusted computing base. Then summarize what are the design challenges to your proposed solutions. Provide high-level intuitions for how to address those challenges. Finally, write a high-level overview of your proposed system/solution. You may use a diagram to summarize

how your proposal. Without giving many details of each of the components in your system, you only mention the functionalities of each major component (i.e., the building block) of your system and the flow/interactions among those components. Use appropriate diagram/figure to provide such high-level details.

Design Details: Discuss in detail all the components of your proposed system. Discuss how you have handled the most critical corner cases of your design. These corner cases are important because this will allow someone expert in the subject area to completely realize your system design. Also, it will help others integrating your solution or replicating your system for future research. Always provide the intuitions for your solutions. Saying why you have done something is more important than how have done it. Note that you should still keep the discussion in this section at the conceptual level, i.e., not at the level of implementation details (e.g., length of the key used in our implementation was 128 or the encryption algorithm we chose are AES, which are too detail). The conceptual-level discussion abstracts away a lot of implementation-level details and presents the ideas in a way that can be instantiated (i.e., implemented) in different ways in your implementations.

Implementation Details: What software/hardware did you use/write/modify for your implementations? What does each module do in connection with the components of your system/solution, and how those new modules or add-ons serve the purpose?

Evaluation: The evaluation section can be broadly broken down into two sub-sections: Evaluation setup and results.

Evaluation Setup: How did you set up the experiments for evaluation? What are the evaluation metrics? Make sure the metrics answer the research questions. What dataset (if any) have you used? Mention if you have any baseline results to compare with.

Evaluation results: Present the results with tables and graphs. If you have a different setup for different experiments, describe the experimental setup (e.g., hardware, software, and datasets used). For informal security analysis of your proposed defense/solution, provide justification (with text) on how your defense/solution can prevent the attacks that you mentioned in the threat model.

Related work: One of the most critical and often overlooked portions of a research project is a sufficient investigation of related work. For this milestone, you will write a related work section. A good related work section is not simply a laundry list of papers and corresponding summaries. Rather, a good related work tells a story of how technology and research have advanced to address problems and topics related to that considered by the paper. It is also critical to contrast your paper with the prior work, not just state what the prior work does. While your research is not yet complete, you should have enough of a grasp on the idea to contrast it with the prior work. You may also revise your related work for the final written paper. Finally, during your related work investigation, take note of how the related work section of those papers is written. Many papers have poorly written related work sections. Identify what makes a good and bad related work section.

Conclusion and Future Work: Don't summarize your work here. That's what the abstract was

for. Instead, provide some philosophical ruminations on your work and future possibilities, i.e., conclusions that you have arrived at as a result of your work.