CS 443/643 *Security & Privacy in Computing*

Root the (Ballot) Box

You will work in groups of 2 or 3 students on the semester project, which will start on October 7 at 10am EDT. The goal will be to take a voting machine and utilize the attacks and vulnerabilities taught in the course to create one "good" working voting machine and one "bad" *vulnerable* voting machine with several *back doors* that would allow a knowledgeable attacker to cheat in the election. Then you will be given two voting machines from other groups to attack, attempting to find all of the vulnerabilities (intentional or not) in their system. You will either be given one good machine and one bad machine or two bad machines. It is up to you to figure out what type of machines you have. Finally, your group will present on both your own bad machine and your experience attacking the other group's bad machine.

Part 1 *The "good" voting machine*

The base of your project will be the good voting machine. You may find the base voting machine code online (with proper citations/references) or build your own. The voting system must have the following features:

- Be able to compile and run inside the class VM
    - If this requires installing external dependencies, include those in a folder with your project so other teams can easily run your code
    - If you need a database, use SQLite for ease of installation
- Include multiple elections for various offices
    - President, Congressional districts/constituencies, etc
    - Do not have to be US government positions
- A registration check to ensure a valid voter
- A voting process where a registered voter makes their choices
    - A legitimate voting system allows only one ballot per user
- A "Cast Ballot" action where the user's votes are submitted, displaying the vote selections for each office
- A process for an administrator to perform the following functionality:
    - Create a new election
    - Open an election for voting
    - Close an election to prevent further votes (may be auto-closed at a given time)
- The "good" voting machine **CANNOT** include any tricks to manipulate others into thinking that it is a "bad" machine.

The "machine" is not a physical device; we expect your group to create a software package that provides the above functionality at minimum. Consider, for example, a "machine" that is a simple CGI-based web application, or a command-line C program. We encourage you to be creative, and add additional features!

***Part 2** The "bad" voting machine*

After your group finalizes the voting machine code for the "good" machine, the second phase will be to (covertly) add in vulnerabilities as back doors to the code for the "bad" machine. Your group, in its final write-up, should clearly show (*i.e.*, with screenshot evidence) how the vulnerabilities can be used to compromise the integrity of the voting machine. Remember that the goal of this project is to *change the outcome of the election*, not just "break" the application. This doesn't just mean changing vote totals -- there are many ways to compromise an election, so be creative!

You will not be graded on the quality of the voting machine (outside of it meeting this basic functionality), and you may use whatever sources you want (cite them) to get the legitimate voting machine code. What you will be graded on is the quality of your back doors, the cleverness of your inserted vulnerabilities, and your use of the techniques taught in this class.

We expect at least *five* vulnerabilities explicitly coded into the voting machine. Three of these must come directly from the ones studied in class and the SEED Labs; the rest can either be a more difficult variant of a previously-studied vulnerability, or a brand-new one entirely. We encourage you to be creative in this as well. Your "bad" voting machine **CAN** be made to appear to be "good."

What to turn in
Building the voting machines is half of the project. This part is due on November 8 at 10 pm EST via Gradescope.

- Two gzipped files with...
  - All of your source code and build environment (such as makefiles) and installation instructions for **EACH** machine
  - Make sure to label your machines as 'machine A' 'and machine B' to prevent from indicating which machine is good or bad
  - A write-up indicating which machine is good/bad, how your voting machine works and what vulnerabilities you inserted in the bad machine and how you would exploit them
  - Documentation with screenshots of your good voting machine running correctly and then documentation with screenshots of your bad voting machine with a cheater changing the outcome
  - A user manual for an unsuspecting user to use the voting machine correctly
- A third gzipped distribution of just your source code and user manual. This will be used by another group in part 2 of the project. Make sure there are no comments explaining your vulnerabilities or any indications of which machine is bad or good.

Part 3 *The red team*

For this part of the project, you will be given two gzipped distributions of source code and a user manual for another group's voting machines. The machines you are given may be both bad or one bad and one good. You will not be given two good machines. Your team's goal is to perform a security analysis of the voting machine. You will first try to identify whether a machine is good or bad. Then, you will try to find all of the five vulnerabilities that the other team put into their bad voting machine, plus any other unintentional vulnerabilities. There will be extra credit if you find vulnerabilities (with exploits) that they did not intend to create.

What to turn in

On November 29, by 10 pm EST, you will submit a report to Gradescope identifying the machines as good/bad, outlining all of the techniques you used to try to find vulnerabilities, describing the vulnerabilities that you found and if you were able to exploit them, and documentation (including screenshots) of exploiting them.

Part 4 *The class presentation*

Student presentations will be on November 30 and December 2 in class. Both sections will meet together in the same Zoom meeting, and each group will present for 12 minutes. First you will describe your voting machines and the backdoors and vulnerabilities that you inserted. Then, you will describe the voting machines you were assigned to analyze and you will explain what your red team analysis uncovered. Demos of both systems would be useful. Provide a more in-depth exploration of the most interesting (in your opinion) three vulnerabilities, and mention the others you find.

After each presentation the group whose voting machine was red teamed in the previous presentation will go next. Participation grades in the course will take into account the attendance at the student presentations on both days, especially for students who present on November 30.

What to turn in

All teams must turn in their project and their final presentation slides to Grapdescope by November 29 at 10pm EST, and all teams must be ready to present on demand in class on either day.