Final Project

CSCI 4971 – Secure Software Principles

Rensselaer Polytechnic Institute

Spring 2010

In the autumn of 2004, [Daniel J.] Bernstein taught a course about computer software security, titled “UNIX Security Holes.” The 16 members of the class discovered 91 new UNIX security holes[, and] publicly announced 44 of them with sample exploit code.

Project Options

Bughunter

Use your skills to find, exploit, and patch vulnerabilities that put open source projects at risk.

Codeslinger

Write a tool to aid in code auditing, such as by locating problematic source code, fuzzing, or automating exploitation.

`Undefineable`

Work on another project related to the class. Talk to the TAs about your idea.

Requirements

- Find at least 2 vulnerabilities in open source projects.
    - They don't necessarily have to be from the same project.
    - Actively maintained projects are preferred.
    - Work from the latest (preferably trunk/tip/HEAD) version of the codebase.
- Develop a proof-of-concept exploit.
    - It doesn't have to get you root, but demonstrate the severity of the vulnerability.
- Submit a patch to fix it.
    - A well-documented write-up will suffice if the vulnerability is complex.
- Work individually.

A Note on Web Vulnerabilities

Let's face it, most web applications are, shall we say, "low-hanging fruit." But, if you insist:

- No XSS, CSRF, or SQL injection.
  - But if you find them, be nice: Tell the maintainers.
- Go for tough bugs; come up with clever exploits, not just proofs-of-concepts.

Caveat hackor

DO NOT attempt to exploit production servers. Always deploy your own instance of the software. We may be able to help if you are having trouble with this.

Submissions

- E-mail the TAs (`ssp-ta@cs.rpi.edu`) as soon as you've identified a vulnerability.
  - Briefly describe it and how you plan to try to exploit it.
  - We won't penalize you if it's independently found by someone else.
- Submit your patch to the maintainers however they prefer (mailing list, Bugzilla, etc.) and e-mail a copy to the TAs.
- Prepare a 5 minute presentation on each vulnerability.
  - How did you find it?
  - What is the specific flaw in the code?
  - How did you exploit it?
  - What could an attacker gain from exploiting it?

Project Options

Bughunter

Use your skills to find, exploit, and patch vulnerabilities that put open source projects at risk.

Codeslinger

Write a tool to aid in code auditing, such as by locating problematic source code, fuzzing, or automating exploitation.

`Undefineable`

Work on another project related to the class. Talk to the TAs about your idea.

Requirements

- Write or extend an open source tool to aid in security audits.
    - Use whatever language you feel comfortable with. We can read anything.
- Demonstrate how your program or feature can be used.
- Work individually or in a team of two.

Submissions

- E-mail the TAs (`ssp-ta@cs.rpi.edu`) before you start. Wait for us to approve it.
  - Describe the project and set yourself a reasonable goal for the end of the semester.
- Submit your finished code to the TAs, with documentation on how to use it.
  - We'd like you to use version control so we can see your progress.
  - If you're extending an existing project, consider submitting your code to them as well.
- Prepare a 10 minute presentation on the tool or feature.
  - Give some background and how you contributed.
  - What problem does your feature address?
  - How does it compare to other (e.g., commercial) solutions?
  - Don't forget to give a demonstration.

Project Options

Bughunter

Use your skills to find, exploit, and patch vulnerabilities that put open source projects at risk.

Codeslinger

Write a tool to aid in code auditing, such as by locating problematic source code, fuzzing, or automating exploitation.

`Undefineable`

Work on another project related to the class. Talk to the TAs about your idea.

Schedule

- Project proposal due April 1
  - What open source projects are you investigating?
  - What project or feature are you writing?
- In-class presentations May 6 & 10

Grading

Quoth the syllabus,

25 percent of your grade will be based on an end-of-semester project. Students *must* have a passing grade on the project to pass the class.

Don't worry much over your grade. If it is evident that you learned something during the project, you'll be fine. We will judge projects individually.