COURSE OVERVIEW

CMPS 122: COMPUTER SECURITY

OVERVIEW

RESOURCES & UPCOMING ASSIGNMENTS

- Course pages
  - Schedule, lecture slides, and course info: https://users.soe.ucsc.edu/~owen/courses/cmps122/win19/
  - Assignments and submissions: https://canvas.ucsc.edu/courses/19818
  - Announcements and discussion: http://piazza.com/ucsc/winter2019/cmps122
- Always check Canvas for assignment due dates
  - Admin 1, due 23:59 Friday, Jan 11, 2019
  - Homework 1, due 23:59 Thursday, Jan 17, 2019
  - Lab 1, due 23:59 Thursday, Jan 24, 2019

WHAT IS COMPUTER SECURITY?

- Keeping computing systems functioning as intended
  - Free of abuse
- Keeping data we care about safe
  - Accessed only as appropriate
- Securing access to our resources & capabilities
  - Storage space, processing time, network access, etc.
- Enabling privacy and anonymity
  - When appropriate and possible

WHY IS COMPUTER SECURITY SO HARD?

- Programmers design for common cases, attackers exploit exceptional cases
- Security of a system is often hard to reason about
- Issues cut across computers, applications, and abstraction boundaries
- Have to consider not only what to enforce, but how to enforce it effectively
- The security arms race
  - “Secure against all the attacks we could think of”
- What is feasible for attackers to pull off continues to evolve

OVERVIEW

COURSE INFO

- People
  - Owen Arden (Instructor) owen@soe.ucsc.edu
  - Ana McTaggart (TA) womctagg@ucsc.edu
  - Nitesh Singh (TA) nsingh32@ucsc.edu
- Office hours
  - Arden: E2-349A, Wednesdays, 4:30pm-5:30pm (or by appointment)
    - Make a 15min appointment here: https://calendly.com/owenarden/office-hours-appointment
  - McTaggart: BE 118, Tuesdays, 11am-12pm
  - Singh: BE 118, Fridays, 11am-12pm
- No web browsing, playing games, etc. during lecture. Participate! Ask Questions!

OVERVIEW

WHAT IS COMPUTER SECURITY?

- Keeping computing systems functioning as intended
  - Free of abuse
- Keeping data we care about safe
  - Accessed only as appropriate
- Securing access to our resources & capabilities
  - Storage space, processing time, network access, etc.
- Enabling privacy and anonymity
  - When appropriate and possible

OVERVIEW

WHY IS COMPUTER SECURITY SO HARD?

- Offense/defense asymmetry:
  - Defenses must be total; attackers only need one “in”
  - Defenses are public; attackers develop and test in private
  - Defenses are often costly and time-consuming to change; attackers can adapt more easily
- Minimal deterrence
  - Internet facilitates anonymity
- Attacks from/to different jurisdictions complicate prosecution

OVERVIEW

WHY IS COMPUTER SECURITY SO HARD?

- Offense/defense asymmetry:
  - Defenses must be total; attackers only need one “in”
  - Defenses are public; attackers develop and test in private
  - Defenses are often costly and time-consuming to change; attackers can adapt more easily
- Minimal deterrence
  - Internet facilitates anonymity
- Attacks from/to different jurisdictions complicate prosecution
OVERVIEW

Why is computer security so hard?

▸ Security is expensive!
  ▸ Performance overhead
  ▸ Increased development time
  ▸ Specialized expertise

▸ Benefits are invisible
  ▸ Hard to measure cost-to-benefit ratio, even after deployment

OVERVIEW

Computers do what they are told, not what they are supposed to

▸ The difference between data and code is small: data can be executed and code can be modified

▸ Functionality, performance, and security are often in tension

▸ Too much complexity leads to vulnerabilities

▸ ...and so do oversimplified abstractions.

▸ Absolute security is usually impossible, infeasible, or undesirable
  ▸ Goal: effective risk management given assumptions about security threats

OVERVIEW

Discuss statement on the next slide for 2 minutes with your neighbors

▸ Decide:
  ▸ There is no chance this happened
  ▸ There may be a small possibility
  ▸ It’s likely
  ▸ It’s certain
  ▸ You don’t know
  ▸ After 2 minutes, we will have show of hands for each
  ▸ I will ask for a volunteer to make a case for each one

OVERVIEW

During this lecture, your phone was attacked by a grad student with a laptop sitting just outside the door.

▸ Decide (2 minutes):
  ▸ There is no chance this happened
  ▸ There may be a small possibility
  ▸ It’s likely
  ▸ It’s certain
  ▸ You don’t know

How could we be more certain about whether it happened? Would it be worth it?

OVERVIEW

What is this class about?

▸ How to think adversarially about computer systems

▸ How to assess threats for their significance

▸ How to build programs & systems with robust security properties

▸ How to gauge the protections provided by, and security limitations of today’s computer systems

▸ How attacks work in theory and in practice

OVERVIEW

Warning

In this class we will be discussing attacks that can do significant damage to computer systems.

▸ This is in no way an invitation to undertake these attacks or derivations thereof other than with the informed consent of all involved parties.

▸ If you want to hack your own machine, fine—but if that exploit escapes and damages another computer, you may be prosecuted and potentially imprisoned.

▸ The existence of a security hole is not an excuse to exploit it.

▸ Not just about ethics: UCSC policy, and Californian, United States, and international law.
“The University is not engaged in making ideas safe for students. It is engaged in making students safe for ideas.”
Clark Kerr (1911-2003)

OVERVIEW

DEFENSE AGAINST THE DARK ARTS

No required textbook, but additional reading is recommended
You may want to look at

OVERVIEW

LEARNING OBJECTIVES

▸ Students who pass this course should be able to explain:
  ▸ The principles of computer security
  ▸ Principles and application of basic cryptography including public-key schemes
  ▸ Authentication & trusted operating systems
  ▸ Secure network protocols including Kerberos and SSL
  ▸ Firewalls, intrusion detection, and countermeasures
  ▸ Program security & bug exploits
  ▸ Malicious code: viruses, worms, trojan horses, and more
  ▸ Legal, ethical, governmental, and societal issues in computer security

▸ Students who pass this course will have gained:
  ▸ Experience in writing simple to moderately complex security attacks
  ▸ An appreciation of the impact upon performance of design choices including selection of algorithms with which a security attack might hide its presence
  ▸ Increased familiarity with Unix/Linux; C programming, APIs, and System Calls

OVERVIEW

ASSESSMENT

<table>
<thead>
<tr>
<th>Component</th>
<th>Percentage</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Admin tasks</td>
<td>(mandatory)</td>
<td></td>
</tr>
<tr>
<td>Homework x5</td>
<td>25%</td>
<td>(1 week each)</td>
</tr>
<tr>
<td>Lab 1</td>
<td>5%</td>
<td>(2 weeks)</td>
</tr>
<tr>
<td>Lab 2</td>
<td>15%</td>
<td>(3 weeks)</td>
</tr>
<tr>
<td>Lab 3</td>
<td>20%</td>
<td>(4 weeks)</td>
</tr>
<tr>
<td>Final Exam</td>
<td>35%</td>
<td>(closed book)</td>
</tr>
</tbody>
</table>

NOTE: Administration tasks are mandatory and you must pass every component (homework, labs, and final) to pass the class. For example, doing well on the labs and final but submitting poor homework solutions will see you fail the class.

HOWEVER: Doing badly on one homework or lab, but well on the others, is fine so long as the aggregate grade for the homework / labs is a pass.

REMEMBER: If you do not pass the final, you do not pass the class.

OVERVIEW

COLLABORATION

▸ Labs are individual assignments (no groups), however...
  ▸ Working, studying, and thinking together is good!
  ▸ But the final product must be yours and yours alone
  ▸ Do NOT share code!
  ▸ DO share ideas
  ▸ Homeworks are also individual
  ▸ No discussion with anyone
  ▸ No late submissions allowed
  ▸ Plagiarism checks are performed
LAB ASSIGNMENTS

▫ All labs are submitted via Canvas
▫ Late penalties apply (15% per day), max 3
▫ Automatically graded
▫ Follow submission directions or you will lose credit!
▫ Check that your submission works “out of the box” with supplied tests (when applicable)
▫ Automatically checked for collaboration & plagiarism
▫ Your lab report is to authenticate the originality of your submission. We will typically only read it if your code is flagged by the plagiarism checker

DEFINING SECURITY:

POLICIES

▫ A system’s security policies describe
▫ What the system is supposed to do
  ▫ Store and provide access to a user’s personal files.
▫ What the system is not supposed to do
  ▫ Do not allow other users to access or modify a user’s files, unless explicitly permitted to.

DEFINING SECURITY: ATTACKS AND VULNERABILITIES

▫ An attack tries to violate security policies by exploiting vulnerabilities
▫ A vulnerability is an unintended aspect of a system’s design, implementation, or configuration
  ▫ storing client permissions on the client
  ▫ unchecked array bounds
  ▫ world-writable configuration files
  ▫ initializing pseudorandom generator with a constant seed

ATTACKS REQUIRE RESOURCES

▫ Some attacks can only be launched by adversaries with significant resources.
  ▫ SHA-1 collision: two PDFs with same hash [1]
  ▫ 6,500 CPU years + 110 GPU years
  ▫ ~$110,000 of cloud time
  ▫ MD5 collision for two Windows Update images: $0.65
▫ But violating security properties can have great value


HIGH PROFILE BREACHES

THIS WEEK: ETC ATTACK

Hackers may have spirited $1 million from the Ethereum Classic blockchain in a “51%” attack

Evaluation and Defense of Ethereum Classic blockchain

“51%” Edition of the
Ethereum Classic blockchain was attacked, resulting in $1 million worth of
Ethereum Classic (Ethereum Classic is a fork of Ethereum that removed a code change that
led to the 2016 DAO hack“51%” Edition of the
Ethereum Classic blockchain was attacked, resulting in $1 million worth of
Ethereum Classic (Ethereum Classic is a fork of Ethereum that removed a code change that
OVERVIEW

WHO IS THE ADVERSARY?

▸ Honest but clumsy (or curious) users
▸ L33t H@X0rs looking for a challenge
▸ Disgruntled employees
▸ Criminals interested in financial gain
▸ Intelligence agents seeking information
▸ Militaries intent on overthrowing a government

---------------------------------------------

OVERVIEW

THE PATH TO SECURITY

▸ What threats are considered?
▸ Who might gain from compromising the system?
▸ What vulnerabilities are exploitable?
▸ Nation states have bigger budgets than 14-year olds
▸ What assurance is there that the system is free of vulnerabilities?

---------------------------------------------

OVERVIEW

THE EVOLUTION OF THREATS

▸ 1950’s to 1990’s – Intellectual Curiosity
  ▸ The Morris Worm (1988)
  ▸ War Games – 1983

▸ 1990’s to 2000’s – Bragging Rights
  ▸ Rise of hacker organizations and underground publications
  ▸ “Script Kiddies”

▸ 2000’s onwards – Financial gain
  ▸ Spam, phishing, credit card theft, identity theft, ransomware (more recently)

▸ 2010’s onwards – Political motivation & organized cyber crime

---------------------------------------------

OVERVIEW

THE EVOLUTION OF THREATS

▸ Swordfish – 2001
▸ The Matrix – 1999

▸ Mr. Robot – 2015
OVERVIEW

PAUSE FOR THOUGHT

▸ To protect systems, you must know your enemy
▸ Absolute security is usually impossible, infeasible, or undesirable.
▸ The best approach is to fully assess what threats you are likely to encounter and create defenses that can stop those threats.

Question: which gates are better?

OVERVIEW

LAB 1 - PASSWORD CRACKER

▸ Early UNIX systems stored users' one-way encrypted passwords in /etc/passwd, a publicly readable file with entries like:

```
smithj:Ep6mckrOLChF.:561:561:Joe Smith:/home/smithj:/bin/sh
```

▸ Using the ~90 characters on keyboards of the time, a "strong" 8 character password would take between 10 and 100 years to crack

▸ As computers became faster, this "password recovery" time decreased rapidly

▸ Soon the one-way encrypted passwords were moved from /etc/passwd to /etc/shadow, a file readable only by the super user; /etc/passwd entries now look like this:

```
smithj:x:561:561:Joe Smith:/home/smithj:/bin/sh
```

OVERVIEW

LAB 1 - POP QUIZ

▸ If we assume:

▸ 96 keyboard characters

▸ Passwords between 6 and 12 (inclusive)

▸ How many possible passwords are there?

▸ Total: 619,159,333,724,000,000,000,000 (0.62 septillion)

▸ Question: Are the 6, 7, 8, 9, 10, 11 character search spaces significant?

```
6 chars = 96^6 = 7.8 x 10^11
7 chars = 96^7 = 7.5 x 10^13
8 chars = 96^8 = 7.2 x 10^15
9 chars = 96^9 = 6.9 x 10^17
10 chars = 96^10 = 6.6 x 10^19
11 chars = 96^11 = 6.4 x 10^21
12 chars = 96^12 = 6.1 x 10^23
```

OVERVIEW

LAB 1

▸ Teaching server for CMPS 122 is grunhilda.soe.ucsc.edu

▸ Log in with your CruzID Blue password:

```
ssh <cruzid>@grunhilda.soe.ucsc.edu
```

▸ If on Windows, use Putty or similar ssh client (http://www.putty.org/)

```
https://users.soe.ucsc.edu/~owen/courses/cmps122/win19/assignments/CMPS122-Lab1.pdf
```

OVERVIEW

LAB 1

▸ Scenario:

▸ You have obtained a copy of an old-style /etc/passwd file containing passwords you are reliably informed are exactly four bytes long and use only upper and lower case characters, and numeric digits.

▸ Requirements:

▸ Basic (50%): Crack a single four-character password

▸ Advanced (30%): Crack six four-character passwords

▸ Stretch (10%): Crack six four-character passwords in under 15 minutes on grunhilda.soe.ucsc.edu

▸ Extreme (10%): Crack a single four-character password without ever using more than 15% of the available CPU resources, in under 15 seconds on grunhilda.soe.ucsc.edu

▸ Run the grading harness before you submit!
Lab 1

- Run the grading harness before you submit!
- Run the grading harness before you submit!
- Run the grading harness before you submit!
- Run the grading harness before you submit!
- Run the grading harness before you submit!
- Run the grading harness before you submit!
- Run the grading harness before you submit!
- Run the grading harness before you submit!
- Run the grading harness before you submit!
- Run the grading harness before you submit!