CSE509 System Security

Logistics & Introduction

Nick Nikiforakis
nick@cs.stonybrook.edu
Who am I, and where can you find me?

• Nick Nikiforakis
  – Assistant Professor in Department of Computer Science
  – Research interests:
    • Web security and Privacy
    • Low-level security
    • Client-side security mechanisms
  – Office: 341
  – Office Hours: Monday & Wednesday, 5:30-6:30PM
For those not yet registered

• There is a physical limitation as to how many people can take this class
• I will not give (and have not given) any personal exceptions to anyone
  – It’s not because I don’t like you, it’s because I don’t have the time to review individual cases
  – If you need the class to graduate please find Kathy Germana and plead your case with her
Required background

• In addition to the official prerequisites, it will be really helpful if you have a working knowledge of C and JavaScript
  – I will try to cover some basics in the class but you will likely need more than that
Course Logistics

• Lectures
  – Monday, Wednesday: 4:00PM – 5:20PM

• Grade breakdown (subject to minor tweaks):
  – 10% Assignments
  – 20% Midterm exam
  – 35% Final exam
  – 35% Project
Code of Conduct

• The work that you present as your own, should be your own
  – Cite the resources that you used (other people’s code, documents, etc.)
  – Don’t allow your code/paper summaries to be copied
  – Don’t copy other people’s code or paper summaries

• Anything short of the above, will be grounds for immediate failing of the class and a stain in your record
Late submission policy

• Based on the structure of the class, the only thing that you need to hand-in is your project and a couple of small assignments
  – Hand them in, on time.
  – For every day that you are late, there will be a 15% penalty
Course Materials

• There is no official textbook
  – You can find textbooks that I am drawing inspiration from the course’s website

• There are no notes
  – You will have to attend and keep your own notes
  – Showing up for class will help you do well in the midterm and final

• All the material will be posted on the course’s website
High-level vision for this class

• Get you all familiar with core ideas in system security, with more focus on web and low-level security
  – We will cover the very basics of cryptography
  – If you are interested in more, perhaps consider taking Network Security the following semester

• Get you comfortable tackling a serious semester-long project
  – You have to be comfortable coding large projects
  – No hand-holding
  – No individual milestones
Low-level
• The browser is a program written in memory-unsafe languages
• Has plugins written in memory-unsafe languages (Flash, Java, etc.)

Attacks
• Buffer overflows and overreads
  • Stack
  • Heap
• Integer overflows
• Execution of attacker-provided/desired code
  • Shellcode
  • Ret2libc
  • ROP

Web
• The browser is a container and execution environment of web applications

Attacks
• Against the web application
  • Client-side
  • Server-side
• Against the user
• Against the browser
  • Against web login
  • Against the underlying OS
Ethics

• In this class we will be looking at a series of attacks against local programs, web applications, and web servers
• It is ILLEGAL to try out these attacks against third parties (unless you have informed consent by them)
  – I will not bail you out
• There are many websites and systems made to be hacked (e.g. Overthewire, hackthissite, etc.)... be smart, use them
What is security?

• Wikipedia

  – Security is the degree of resistance to, or protection from, harm. It applies to any vulnerable and valuable asset, such as a person, dwelling, community, nation, or organization.
What is computer security?

• Everyone has their own definition
  – No single one is perfect
• Computer security deals with protecting data, programs, and systems against intelligent adversaries.

• Safety vs Security
  – What’s the difference between the two?
  – Do they interact?
Why is it important?

In just a few months...
Cybercrime + Cyber espionage  
> 100 billion dollars a year  
(just for the US)
Cybercrime + Cyber espionage > 100 billion dollars a year (just for the US)
Not just about money...
Why is it hard?

• It is hard to get security right because:
  – Security is hard to test for
    • Testing correctness versus security
  – It requires a deep understanding of all technologies involved in the design and implementation of a system
    • Really hard in large real systems
  – Users are typically the weakest link
  – **Asymmetry** between attack and defense
CIA

• What do we mean by “protection”?  
• Security is about CIA

  – Confidentiality: Keeping data and resources hidden or protected from unauthorized disclosure

  – Integrity: Data and Programs are modified in specified and authorized ways. Data integrity and origin integrity.

  – Availability: Systems and networks are available for use by legitimate users
Identify the properties

• A social networking site requires users to login with their usernames and passwords before they are able to access their profiles

• In a typical operating system, the permissions of files can be setup in a way that allows only their owners (or the groups in which the owner belongs) to modify them
Identify the properties

• More and more sites use HTTPS, not only to protect their information, but to stop middlemen from injecting malware in cleartext traffic

• A bank sets up an IDS that temporarily bans a user’s IP address if the user tries to login more than 5 times unsuccessfully
Identify the properties

• Tripwire is an application that creates checksums of important system programs (e.g., ls, ps) and alerts the user when the checksum of a program does not match its original checksum.

• Certain mechanisms for programs (like stack canaries) attempt to detect attacks-in-progress and terminate the program upon detection of an attack.
Attacker mentality

• Common things attackers do
  – Put up a fake financial website, collect users’ logins and passwords, empty out their accounts
  – Insert a hidden program into unsuspecting users’ computers, use it to spread spam or for espionage
  – Subvert copy protection for music, video, games
  – Stage denial of service attacks on websites, extort money
  – Wreak havoc, achieve fame and glory in the blackhat community
Marketplace for Vulnerabilities

• Option 1: **bug bounty programs**
  – Google: up to $3133.7 in 2010, now up to $20K per bug
  – Facebook: up to $20K per bug
  – Microsoft: up to $150K per bug
  – Pwn2Own competition: $10-15K

• Option 2: **vulnerability brokers**
  – ZDI, iDefense: $2-25K

• Option 3: **gray and black markets**
  – Up to $100-250K reported (hard to verify)
  – A zero-day against iOS sold for $500K (allegedly)
It’s a Business

• Several companies specialize in finding and selling exploits
  – ReVuln, Vupen, Netragard, Exodus Intelligence
  – The average flaw sells for $35-160K
  – $100K+ annual subscription fees

• Nation-state buyers
  – “Israel, Britain, Russia, India and Brazil are some of the biggest spenders. North Korea is in the market, as are some Middle Eastern intelligence services. Countries in the Asian Pacific, including Malaysia and Singapore, are buying, too” -- NY Times (Jul 2013)
Marketplace for Stolen Data

[Dell SecureWorks, 2013]

- Single credit card number: $4-15
- Single card with magnetic track data: $12-30
- “Fullz”: $25-40
  - Full name, address, phone, email addresses (with passwords), date of birth, SSN, bank account and routing numbers, online banking credentials, credit cards with magnetic track data and PINs
- Online credentials for a bank account with $70-150K balance: under $300

Prices dropped since 2011, indicating supply glut
Marketplace for Victims

• Pay-per-install on compromised machines
  – Can be used to send spam, stage denial of service attacks, perform click fraud, host scam websites

• Botnets for rent
  – DDoS: $10/hour or $150/week
  – Spam: from $10/1,000,000 emails

• Tools and services
  – Basic Trojans ($3-10), Windows rootkits ($300), email, SMS, IM spamming tools ($30-50), botnet setup and support ($200/month, etc.)
Bad News

• Security often not a primary consideration
  – Performance and usability take precedence
• Feature-rich systems may be poorly understood
• Implementations are buggy
  – Buffer overflows are the “vulnerability of the decade”
  – Cross-site scripting and other Web attacks
• Networks are more open and accessible than ever
  – Increased exposure, easier to cover tracks
• Many attacks are not even technical in nature
  – Phishing, social engineering, etc.
Better News

• There are a lot of defense mechanisms
  – We’ll study some, but by no means all, in this course

• It’s important to understand their limitations
  – “If you think cryptography will solve your problem, then you don’t understand cryptography... and you don’t understand your problem”
  – Many security holes are based on misunderstanding

• Security awareness and user “buy-in” help

• Other important factors: usability and economics
Definitions
Definitions

• Computer Security is always related to a specific attacker/threat model

• **Threat**: A potential violation of security by an attacker

• Is this program/website secure? -> Who is the attacker? Who are you defending against?
  – Your bf/gf
  – A script kiddie
  – An opportunistic hacker
  – A hacker collective who has targeted you
  – A government
Definitions

• **Threat model**: A set of possible attacks and attackers that a system tries to protect against.
  – Physical attacks
  – Outsider attacks
  – Insider attacks (In your local network)
  – Attacks over the Internet
  – ...
Policies and mechanisms

- **Security policy**: A statement of what is and what is not allowed
  - E.g. Only computer science students are allowed to enter the lab

- **Security mechanism**: A method, tool, or procedure for enforcing a security policy
  - How do we enforce the above desired policy?

- A security policy can be implemented by a variety of security mechanisms
When faced with an attacker...

- A security mechanism can:
  - Prevent the attack
    - Attack will fail
  - Detect the attack
    - Attack may succeed but we know about it and can thus react
  - Recover from the attack
    - Stop the attack and go back to a “healthy” state
    - System functions correctly even in the presence of attacks
Achieving Secure Computing

• How do we secure a system?
  – Procedural approaches (secure practices)
    • Things that have to be done (behavior) by users to strengthen the overall security of a system (e.g. Guidelines for setting passwords, or document shredding)
  – Functions, Tools, and Mechanisms (Hardware/Software)
    • Mechanisms that are deployed to make the system more secure
  – Education and awareness
    • Understanding of threats and vulnerabilities
    • Ability to reason beyond set procedures
Prevent vs Recover

• For some organizations, it may be cheaper to recover from an attack, rather than try to prevent it
  – Money, time, complexity of system, user education VS
  – Bad publicity, loss of customers, loss of money

• Problem with this approach is the difficulty of predicting the cascade effects
  – Systems trusting each other
  – Passwords shared among systems
Necessary assumptions

• A security mechanism will work when:
  – The security policy correctly describes the security requirements of a real-world system
  – The security mechanism is implemented correctly
  – The security mechanism is installed and administered correctly
  – The security mechanism cannot be bypassed
Trust

• At the end of the day, security is ultimately based on particular assumptions and trust relationships
  – Systems
  – People
  – Policies

• Must read: “Reflections on Trusting Trust” by Ken Thompson
  – How do we know that critical executables on our machines are not malicious? E.g. “su” sending out your passwords to Ubuntu?
Action points

• Subscribe to Piazza and Blackboard
  – You can find the links to that from the course’s website
    • http://www.securitee.org/teaching/cse509
Questions?
Credits

• Slides on attacker motivation and prices of underground market by Vitaly Shmatikov