Network Security and Privacy

Vitaly Shmatikov

http://www.cs.utexas.edu/~shmat/courses/cs361s/
Course Personnel

◆ **Instructor:** Vitaly Shmatikov
  - Office: GDC 6.812
  - Office hours: Tuesday, 1-2pm
  - Open door policy – don’t hesitate to stop by!

◆ **TA:** Oliver Jensen
  - Office: GDC 6.818A
  - Office hours: Wednesday, 11am-12n

◆ **Watch the course website**
  - Assignments, reading materials, lecture notes
Prerequisites

◆ Required: working knowledge of C and JavaScript
  • The first project is about Web security
  • The second involves writing buffer overflow attacks in C
    – You must have detailed understanding of x86 architecture, stack layout, calling conventions, etc.

◆ Recommended: Introduction to Computer Security; Cryptography; Computer Networks; Compilers and/or Operating Systems
  • Not much overlap with this course, but will help gain deeper understanding of security mechanisms and where they fit in the big picture
Course Logistics

◆ Lectures
  • Tuesday, Thursday 11a-12:30p
◆ Three homeworks (30% of the grade)
◆ Two projects (10 + 15% of the grade)
  • A fair bit of C coding and PHP/JavaScript hacking
  • Can be done in teams of 2 students
  • Security is a contact sport!
◆ Midterm (20% of the grade)
◆ Final (25% of the grade)
◆ UTCS Code of Conduct will be strictly enforced

No make-up or substitute exams! If you are not sure you will be able to take the exams in class on the assigned dates, do not take this course!
Late Submission Policy

- Each take-home assignment is due in class at 11am on the due date
  - 5 take-home assignments (3 homeworks, 2 projects)
- You have **3 late days** to use any way you want
  - You can submit one assignment 3 days late, 3 assignments 1 day late, etc.
  - After you use up your days, you get 0 points for each late assignment
  - Partial days are rounded up to the next full day
Course Materials

◆ **Textbook:**
Kaufman, Perlman, Speciner. “Network Security”
- Lectures will *not* follow the textbook
- Lectures will focus on “big-picture” principles and ideas of network attack and defense
- Attend lectures! Lectures will cover some material that is *not* in the textbook – and you will be tested on it!

◆ **Occasional assigned readings**
- Start reading “Smashing the Stack For Fun and Profit” by Aleph One (from Phrack hacker magazine)
- Understanding it will be essential for your project
Other Helpful Books

- **Ross Anderson’s “Security Engineering”**
  - Focuses on design principles for secure systems
  - Wide range of entertaining examples: banking, nuclear command and control, burglar alarms

- **“The Shellcoder’s Handbook”**
  - Practical how-to manual for hacking attacks
  - Not a required text, but you may find it useful for the buffer overflow project

- **Kevin Mitnick’s “The Art of Intrusion”**
  - Real-world hacking stories
  - Good illustration for many concepts in this course
Main Themes of the Course

◆ Vulnerabilities of networked software
  • Worms and botnets, denial of service, attacks on Web applications, attacks on infrastructure

◆ Defensive technologies
  • Protection of information in transit: cryptography, application- and transport-layer security protocols
  • Protection of networked software: memory integrity, firewalls, antivirus tools, intrusion detection

◆ Study a few deployed protocols in detail: from design principles to implementation details
  • Kerberos, SSL/TLS, IPsec (if time permits)
What This Course is **Not** About

- **Not** a comprehensive course on computer security
- **Not** a course on ethical, legal, or economic issues
  - No file sharing, DMCA, piracy, free speech issues
  - No surveillance
- **Only** a cursory overview of cryptography
  - Take CS 346 for deeper understanding
- **Only** some issues in systems security
  - Very little about OS security, secure hardware, physical security, security of embedded devices...
Motivation

https://
YOU ACKNOWLEDGE THAT NEITHER WELLS FARGO, ITS AFFILIATES NOR ANY OF THEIR RESPECTIVE EMPLOYEES, AGENTS, THIRD PARTY CONTENT PROVIDERS OR LICENSORS WARRANT THAT THE SERVICES OR THE SITE WILL BE UNINTERRUPTED OR ERROR FREE; NOR DO THEY MAKE ANY WARRANTY AS TO THE RESULTS THAT MAY BE OBTAINED FROM USE OF THE SERVICES OR THE SITE, OR AS TO THE TIMELINESS, SEQUENCE, ACCURACY, RELIABILITY, COMPLETENESS OR CONTENT OF ANY INFORMATION, SERVICE, OR MERCHANDISE PROVIDED THROUGH THE SERVICES AND THE SITE.
“Privacy, Security and Legal”

“As a Wells Fargo customer, your privacy and security always come first.”

- Privacy policies
  - Privacy policy for individuals
  - Online privacy policy
  - Social Security Number protection policy
  - International privacy policies

- Your online security
  - How we protect you
  - Online security guarantee

- Fraud information center
  - How fraudsters operate
  - How to protect yourself

- USA PATRIOT ACT information
What Do You Think?

What do you think should be included in “privacy and security” for an e-commerce website?
Desirable Security Properties

- Authenticity
- Confidentiality
- Integrity
- Availability
- Accountability and non-repudiation
- Access control
- Privacy of collected information

...

◆ Basics of cryptography
  • Symmetric and public-key encryption, certificates, cryptographic hash functions, pseudo-random generators

◆ Authentication and key establishment
  • Case study: Kerberos

◆ Web security
  • Case study: SSL/TLS

◆ IP security (if time permits)
  • Case study: IPsec protocol suite
Syllabus (2): Attacks and Defenses

◆ Web attacks
  • Cross-site scripting and request forgery, SQL injection

◆ Network attacks
  • Worms, viruses, botnets
  • Spam, phishing, denial of service
  • Attacks on routing and DNS infrastructure

◆ Buffer overflow / memory corruption attacks

◆ Defense tools
  • Firewalls, antivirus, intrusion detection systems

◆ Wireless security
Peek at the Dark Side

The only reason we will be learning about attack techniques is to build better defenses.

Don’t even think about using this knowledge to attack anyone.
A Security Engineer’s Mindset

[Bruce Schneier]
Ken Thompson

ACM Turing Award, 1983
“Reflections on Trusting Trust”

- What code can we trust?
- Consider "login" or "su" in Unix
  - Is Ubuntu binary reliable? RedHat?
  - Does it send your password to someone?
  - Does it have backdoor for a “special” remote user?
- Can't trust the binary, so check source code or write your own, recompile
- Does this solve problem?

http://www.acm.org/classics/sep95
“Reflections on Trusting Trust”

Who wrote the compiler?

Compiler looks for source code that looks the login process, inserts backdoor into it

Ok, inspect the source code of the compiler... Looks good? Recompile the compiler!

Does this solve the problem?

http://www.acm.org/classics/sep95
“Reflections on Trusting Trust”

The compiler is written in C ...

```c
compiler(S) {
    if (match(S, "login-pattern")) {
        compile (login-backdoor)
        return
    }
    if (match(S, "compiler-pattern")) {
        compile (compiler-backdoor)
        return
    }
    .... /* compile as usual */
}
```
“Reflections on Trusting Trust”

http://www.acm.org/classics/sep95

“The moral is obvious. You can't trust code that you did not totally create yourself. (Especially code from companies that employ people like me.)”
Network Stack

Only as secure as the single weakest layer...
... or interconnection between the layers
Network Defenses

People
- Password managers, company policies...

Systems
- Firewalls, intrusion detection...

Blueprints
- TLS, IPsec, access control...

Building blocks
- RSA, DSS, SHA-1...

Cryptographic primitives

Protocols and policies

Implementations

End uses

All defense mechanisms must work correctly and securely
Correctness versus Security

- **System correctness:** system satisfies specification
  - For reasonable input, get reasonable output

- **System security:** system properties preserved in face of attack
  - For unreasonable input, output not completely disastrous

- Main difference: active interference from adversary

- Modular design may increase vulnerability ...
  - Abstraction is difficult to achieve in security: what if the adversary operates below your level of abstraction?

- ... but also increase security (small TCB)
What Drives the Attackers?

- 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
  • US: $100-150 / 1000 downloads, “global mix”: $12-15
  • 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, ICQ 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
Reading Assignment

◆ Review Kaufman, section 1.5
  • Primer on networking

◆ Start reading buffer overflow materials on the course website
  • “Smashing the Stack for Fun and Profit”
  • You will definitely need to understand it for the buffer overflow project