January 11, 2016

Introduction to Computer Security 2

COMPGA02

Emiliano De Cristofaro and Gianluca Stringhini

Thanks to Aurelien Francillon,Vitaly Shmatikov, and Giovanni Vigna for letting us reuse some of their slides

Last edited, 2016-01-05

Important Info 1/3

- **Who are we**
  - Lecturers: Emiliano De Cristofaro and Gianluca Stringhini
  - Demonstrators (labs): Jeremiah Onaolapo, Lucky Onwuzurike
    - {e.decristofaro, g.stringhini, jeremiah.onaolapo.13, lucky.onwuzurike.13}@ucl.ac.uk

- **Format**
  - 10 x 2h lectures (Mondays 11am-1pm)
    - Weeks will be numbered 1-10 – i.e., not counting reading week after W5
  - 5 x 2h lab (Tuesdays 9-11, MPEB 1.21)
    - Divided in two groups, alphabetically, based on your last name
    - Odd Weeks: A-O; Even Weeks: P-Z
  - Can I switch group? Yes, but:
    - Only before the end of Week 2 and you can't switch back
    - Up to you to find someone to take your place
    - Let both demonstrators know via email

Important Info 2/3

- **Assessment**
  - 70% closed-book exam (2.5 hours)
  - 30% project (in-lab, week 9 and/or 10)
    - Let us know well ahead of time if you can’t be there (and why)

- **Office Hours**
  - Emiliano: Mondays 1-2pm, MPEB 6.04 (after class)
  - Gianluca: Wednesdays 1-3pm, MPEB 7.02
  - Jeremiah/Lucky: TBD
  - Do not call or come outside office hours unless you email first
  - Use the discussion forum on Moodle to ask questions

Important Info 3/3

- **Moodle**
  - All slides and announcements posted there, suggested readings, etc.
    - Make sure you are subscribed to the announcements (“news forum”)
    - It is your responsibility not to miss important announcements and material
  - There is also a discussion forum
    - Ask any questions to lecturers, demonstrators, or colleagues
    - Much better than email so everyone can see the answers
    - Make sure you are subscribed to receive email
  - **Enrolment key:** imahacker!
    - But you should be automatically enrolled via Portico

Course Aim

From syllabus:

> *“Providing an advanced understanding of network and computer security vulnerabilities and countermeasures in real-world systems.”*

In other words:

- Learn to think critically about security
- Formalize adversarial threat model
- Design appropriate responses to security challenges
- Apply what you learned in GA01 to concrete situations

Think as an attacker

- One can't secure a system without being aware of ways to break it...
  - *“You can’t make something secure if you don’t know how to break it.”* (Marc Weber Tobias)
- Schneier’s “Law”:
  - *“Any person can invent a security system so clever that he or she can't imagine a way of breaking it.”*
  - https://www.schneier.com/blog/archives/2008/03/the_security_mi_1.html
- **Caveat emptor!**
  - The only reason we will be learning about attack techniques is to build better defenses
  - That is, don’t use this knowledge to perform attacks!!!

Ethics & Law

- Malicious hacking/cracking is illegal
- Discussing vulnerabilities/how they are exploited is useful
  - E.g., for education, awareness, …
- Full disclosure policy
  - The information about vulnerability has been already distributed to parties that may provide a solution to the problem (e.g., vendors)
  - See: Responsible vulnerability disclosure process (IETF Internet Draft)
  - Preventing similar mistakes from being repeated

Academic Conduct

- High standard expected in academic conduct:
  - Regulations on how to avoid plagiarism
  - Reference and credit sources appropriately

- High standard expected in professional conduct:
  - Computer Misuse and Data Protection
  - Procedures for research with human subjects
  - Responsible research and disclosure procedures
  - Compliance and risk based assessments

Tentative List of GA02 Topics

- Denial of service
- MiTM/Spoofing
- Network Security
- Wireless Security
- SQL, CSRF, XSS, Clickjacking
- Windows/Android Security

- Race Conditions
- Malware
- Memory corruption
- Buffer overflow
- Intrusion/Anomaly Detection
- Firewalls

Course Material

- Slides only provide an outline
  - We guarantee you won’t be very successful by only reading the slides ☺
- Papers and books
  - We do **expect** you to read from additional resources
- Of course, use the **Internet**!
  - Also follow blogs/media to keep up, e.g.:
    - The Registry (Security), Wired Threat Level, New Scientist (Tech), Slashdot, Darkreading, Schneier on Security, Light Blue Touchpaper, MIT Tech Review (Computing/Web), etc.

Textbooks

Pfleeger and Pfleeger. “Analyzing Computer Security: A Threat/Vulnerability/Countermeasure Approach”

Dieter Gollmann. “Computer Security”

Ross Anderson. “Security Engineering”

Kaufman, Perlman, Speciner. “Network Security”

Koziol, et al. “The Shellcoder’s Handbook”

Mitnick. “The Art of Intrusion”

Stallings. Cryptography and Network Security

Correctness vs. Security

- Program or system **correctness**: program satisfies specification
  - For reasonable input, get reasonable output
- Program or system **security**: program properties preserved in face of attack
  - For unreasonable input, output not completely disastrous
- Main difference: **adversary**
  - Active interference from a malicious agent
  - It is very difficult to come up with a model that captures all possible adversarial actions

Threat Model

- Threat modeling is among the hardest tasks of a security researcher
- Adversary **resources** and **capabilities**:
  - Every power that the adversary has
  - E.g. parts of the system observed, parts of the system that can be influenced, parties they can corrupt
- Strategic Adversary:
  - The adversary will chose to commit resources **optimally** to violate the security properties

Protection

- What is it that you want to protect?
  - Defining **assets**
- What are the goals of the protection efforts?
  - Security **properties**
- What do you want to protect against?
  - Attack: any maliciously intended act against a system or a population of systems; any action that violates a given security policy

Threats vs Vulnerabilities

- **Threats**
  - Define who might attack against what assets, using what resources, with what goal in mind, when/where/why, and with what probability
- **Vulnerabilities**
  - Specific weakness in security that could be exploited by adversaries with a wide range of motivations and interest in a lot of different assets
- Source:
  - http://jps.anl.gov/Volume4_iss2/Paper3-RGJohnston.pdf

Threat vs Vulnerability

- Example 1:
  - Threat: Thieves could break into our facility and steal our equipment
  - Vulnerability: The lock we are using on the building doors is easy to pick
- Example 2:
  - Threat: Adversaries might install malware so they can steal social security numbers for identity theft
  - Vulnerability: My computer does not have up-to-date virus signatures and/or has an insecure browser

Harm vs Attack

- **Harm**
  - Negative consequence of an actualized threat
  - E.g., a stolen computer, modified or lost file, revealed private letter, or denial of access
  - Usually, harm occurs when a threat is realized against a vulnerability
- **Attack**
  - An attempt by an adversary to cause harm to valuable assets, usually by trying to exploit one or more vulnerabilities

More definitions

- **Threat Assessment**
  - Attempting to predict the threat
- **Vulnerability Assessment**
  - Attempting to discover security vulnerability
- **Risk**
  - The combination of the probability of an event and its consequence
- **Risk Management**
  - Attempting to minimize (security) hazards by deciding intelligently how to deploy, modify, or re-assign security resources. Involves TA, VA.

Countermeasures

- Countermeasure (or control):
  - A means to counter threats
  - To protect against harm, we can neutralize the threat, close the vulnerability, or both.
- Typical countermeasure involve:
  - Prevention: blocking the attack or closing the vulnerability
  - Dissuasion: making the attack harder but not impossible
  - Deflection: making another target more attractive
  - Mitigation: making its impact less severe
  - Detection: either as it happens or some time after the fact
  - Recovering from attack, making sure it doesn’t happen again

Some Numbers

- Adware industry is worth $2 billion/year, malware industry is $105 billion/year
- 50%-80% of computers connected to Internet are infected with spyware
- 81% of emails is spam (Symantec report 2011)
- 90% of web applications are vulnerable (Cenzic 2009)
- 5.5 billion malware attacks in 2011 (Symantec 2011)
- 2012: 42% increase in target attacks
- In UK, £1B lost on cybersecurity attacks every year
  - 1 in 5 individuals affected
- Good news:
  - Cyber Security market in 2011 was worth $63.7 billion, expected to grow to about $120.1 billion by 2017

Some reasons

- System and network administrators are not prepared
  - Insufficient resources
  - Lack of training
- Attackers leverage the availability of broadband connections
  - Many connected home computers are vulnerable
  - Collections of compromised home computers are “good“ weapons for attacks
    - High speed networking, powerful CPUs, always on

Bugs and failure

- Hardware and software are developed by humans and therefore are not perfect
- A human error may introduce a **bug** (or fault)
- When a fault gets triggered, it might generate a **failure**…
  - If the fault is “security-related”, it is usually called a vulnerability
  - When the vulnerability is triggered (exploited) can lead to the **compromise**

Some history

- 1960s - mainframe computers like the MIT’s Artificial Intelligence Lab became staging ground for hackers
- 1970s - hackers start tampering with phones (the largest network back then)
  - 1972, John Draper finds that the whistle that comes with the Cap’n Crunch cereal produces a sound at the 2600 Hz (the same used by AT&T to authorize long-distance calls)
  - Start of phone **phreaking**

Some history (cnt’d)

- 1973 - Bob Metcalfe wrote RFC 602: “The Stockings Were Hung by the Chimney with Care”
  - ARPA computer network is susceptible to security violations
  - *“many people still use passwords which are easy to guess: their first names, their initials, their host name spelled backwards, a string of characters which are easy to type in sequence”*
- 1980/81 - Two hacker groups form
  - Legion of Doom (US)
  - Chaos Computer Club (DE)
- 1982 - The term “cyberspace” is coined in the novel *Bourning Chrome*

Some history (cnt’d)

- 1983 - The movie Wargames introduces hackers to the public
- 1986 - German hackers penetrate Lawrence Berkeley Laboratory systems and try to obtain secrets to be sold to the KGB
  - Cliff Stoll found an intruder while investigating a $0.75 accounting discrepancy for CPU time
  - He decided to monitor the intruder to find out who he/she was and how he was able to gain privileged access

Some history (cnt'd)

- 1980: John Shock and Jon Hepps of Xerox PARC developed the first 5 worms
  - Designed to preform helpful tasks
  - Some worms were quite simple, simply traveling throughout the network posting announcements
  - Others were quite clever and complex, idle during the day, taking advantage at night of the largely idle computers
- 1988: the Internet worm, developed by Robert T. Morris, brings down the Internet
  - A mistake in replication led to unexpected proliferation
  - Internet had to be "turned off", damages were estimated in the order of several hundred thousand dollars
  - The CERT (Computer Emergency Response Team) is formed
- 1994: Kevin Mitnick attacks the Supercomputer Center in San Diego using a TCP spoofing attack
  - Arrested in 1995 and sentenced to 46 months in prison

Some history (cnt’d)

- 1990 - Operation Sundevil: secret service arrests hackers in 14 U.S. Cities for credit-card theft and telephone and wire fraud
- 1992 - Release of the movie *Sneakers*
- 1993 – The first DefCon conference is held in Las Vegas. It is so popular that it will become an annual event
- 1995 – A Russian cracker siphon 10M $ from Citibank and transfer the money to banks around the world
- 1995 – The movie *Hackers* is released
- 1999 – The Melissa worm causes large problems to the email systems

Some history (cnt’d)

- 2000 – ILOVEYOU, a VBScript worm infects millions of computers within a few hours of its release
- 2001 – CodeRed: overflow in MS-IIS server, 300,000 machines infected in 14 hours
- 2002 - Bill Gates announces the 'Trustworthy Computing’ initiative, a new direction in Microsoft's software development strategy aimed at increasing security
- 2003 – The SQL Slammer worm infected 75,000 machines (90% of the possible targets) in 10 minutes
  - Starts the fear for the flash worms
- 2004 – Sasser: overflow in Windows LSASS, around 500,000 machines infected
- 2005-today – Worms are replaced by botnets
- 2010 – Stuxnet attacks centrifuge systems in nuclear facilities in Iran
  - Completely new (and unexpected) level of sophistication
  - Most likely state sponsored

Changing Nature of the Threat

- Attackers are more prepared and organized
- Attacks are easy, low-risk and difficult to trace
- Increasingly sophisticated but also easy to use
- Source code is not required to find vulnerabilities
- The complexity of Internet-related applications and protocols are increasing – and so is our dependency on them

Terminology

- The term "hacker" was introduced at MIT in the 60s to describe "computer wizards"
  - *[...] someone who lives and breathes computers, who knows all about computers, who can get a computer to do anything. Equally important, though, is the hacker's attitude. Computer programming must be a hobby, something done for fun, not out of a sense of duty or for the money.* (Brian Harvey, University of Berkeley)
- The term was later associated to "malicious hackers" or "crackers", that is, people that perform intrusions and misuse computer systems

Terminology (cnt’d)

- Black Hat: a cracker, someone bent on breaking into the system you are protecting
- White hat: usually associated to friendly security specialists
- Script Kiddie: lowest form of cracker; script kiddies do mischief with scripts and programs written by others, often without understanding the exploit they are using

Insecure Software

- Technical factors
  - Complexity of task, composition, changes
- Economic factors
  - Open-source vs closed-source
  - Security is not a feature
  - Deadlines
  - Insufficient funding/resources
- Human factors
  - Mental models
  - Social factors
  - Poor risk analysis

Attack Methods

- Eavesdropping
  - Get copies of information without authorization
- Masquerading
  - Send messages with other‘s identity
- Message tampering
  - Change content of message
- Replaying
  - Store a message and send it again later, e.g. resend a payment message
- Exploiting
  - Use bugs in software to get access to a host
- Combinations
  - E.g., Man in the middle attack

Social Engineering

- “The art and science of getting someone to comply to your wishes”
  - The weakest link, the user, is often the target
- Performed in many different forms
  - Social engineering by phone
  - Dumpster diving
  - Reverse social engineering
  - Malware disguised as fake anti-virus
- Secret Service favorite

Security Overview

- Security issues at various stages of application life-cycle
  - Mistakes, vulnerabilities, and exploits
  - Avoidance, detection, and defense
- Architecture
  - Security considerations when designing the application
- Implementation
  - Security considerations when writing the application
- Operation
  - Security considerations when the application is in production

Security Overview

Architecture and design

- Validation of requirements (building the right model)
- Verification of design (building the model right)

Common problems

- Authentication and privileges
  - Session replay
  - Principle of least privilege
- Communication protocol design
  - Sniffing, man-in-the-middle
  - Session killing, hijacking
- Parallelism and resource access
  - Race conditions
- Denial of service

Security Overview

Implementation

- Verification of implementation
- Classic vulnerabilities (often programming-language-specific)

Common problems

- Buffer overflows
  - Static: stack-based buffer overflows
  - Dynamic: heap-based buffer overflows
- Input validation
  - URL encoding
  - document root escape
  - SQL injection
- Back doors

Security Overview

Operation

- decisions made after software is deployed
- often not under developer’s control

Common problems

- denial of service (DOS)
  - network DOS
  - distributed DOS, zombies
- administration problems
  - weak passwords
  - password cracking
  - unsafe defaults

Security Architecture

- What is a security architecture?
  - A body of high-level design principles and decisions that allow a programmer to say "Yes" with confidence and "No" with certainty.
  - A framework for secure design, which embodies the four classic stages of information security: protect, deter, detect, and react.
- Security is a measure of the architecture’s ability to resist unauthorized usage
  - At the same time, services need to be provided to legitimate users

What if architecture is flawed?

- Some history: The Swedish warship Vasa
  - In Stockholm, Vasa Museum
  - A reminder for all engineers
  - The ship was built well, but its architecture was *flawed*
    - On its first trip, it fired its guns to salute the port and…
- So what does Vasa have to do with security?
  - Your code might be engineered well, but if your architecture is bad from a security point of view, your system may be broken by attacker

Security and design

- Systems are often designed without security in mind
  - Developer is often more worried about solving the problem than protecting the system
  - Security is ignored because either the policy is generally not available, or it is easier to ignore security issues
- Organizations and individuals want their technology to survive attacks, failures and accidents
  - Critical systems need to be survivable

Design principles

- Design is a complex, creative process
- No standard technique to make design secure
  - But general rules derived from experience
- 8 principles according to Saltzer and Schroeder (1975) *“The protection of information of computer systems”*
  - Economy of Mechanism
  - Fail-safe defaults
  - Complete mediation
  - Open design
  - Separation of privilege
  - Least privilege
  - Least common mechanism
  - Psychological acceptability

Practice Defense in Depth

- Have several layers of security
  - Preventing is not enough, you also need detection and mitigation mechanisms
  - Two controls are better than one
- No single point of failure

"The only system which is truly secure is one which is switched off and unplugged, locked in a titanium-lined safe, buried in a concrete bunker, and surrounded by nerve gas and very highly paid armed guards. Even then, I wouldn't stake my life on it" (Gene Spafford)

Important Info 1/3

- **Who are we**
  - Lecturers: Emiliano De Cristofaro and Gianluca Stringhini
  - Demonstrators (labs): Jeremiah Onaolapo, Lucky Onwuzurike
    - {e.decristofaro, g.stringhini, jeremiah.onaolapo.13, lucky.onwuzurike.13}@ucl.ac.uk

- **Format**
  - 10 x 2h lectures (Mondays 11am-1pm)
    - Weeks will be numbered 1-10 – i.e., not counting reading week after W5
  - 5 x 2h lab (Tuesdays 9-11, MPEB 1.21)
    - Divided in two groups, alphabetically, based on your last name
    - Odd Weeks: A-O; Even Weeks: P-Z
  - Can I switch group? Yes, but:
    - Only before the end of Week 2 and you can't switch back
    - Up to you to find someone to take your place
    - Let both demonstrators know via email

Important Info 2/3

- **Assessment**
  - 70% closed-book exam (2.5 hours)
  - 30% project (in-lab, week 9 and/or 10)
    - Let us know well ahead of time if you can’t be there (and why)

- **Office Hours**
  - Emiliano: Mondays 1-2pm, MPEB 6.04 (after class)
  - Gianluca: Wednesdays 1-3pm, MPEB 7.02
  - Jeremiah/Lucky: TBD
  - Do not call or come outside office hours unless you email first
  - Use the discussion forum on Moodle to ask questions

Important Info 3/3

- **Moodle**
  - All slides and announcements posted there, suggested readings, etc.
    - Make sure you are subscribed to the announcements (“news forum”)
    - It is your responsibility not to miss important announcements and material
  - There is also a discussion forum
    - Ask any questions to lecturers, demonstrators, or colleagues
    - Much better than email so everyone can see the answers
    - Make sure you are subscribed to receive email
  - **Enrolment key:** imahacker!
    - But you should be automatically enrolled via Portico