Introduction

Computer Security Lecture 1

Mike Just[1]

School of Informatics
University of Edinburgh

11th January 2010

[1] Based on original lecture notes by David Aspinall

Outline

- Overview
- Lectures and Tutorials
- Assessment
- Advisory
- Timeline
- Reading

Computer security is a concoction of science, technology, engineering, and human factors. A secure system is only as strong as the weakest link: each factor must be secured, using multiple layers to provide "defence in depth".

Outline

What is Computer Security?

- **Security** is about protecting assets.
- **Computer Security** concerns assets of computer systems: the information and services they provide.
- Just as real-world physical security systems vary in their security provision (e.g., a building may be secure against certain kinds of attack, but not all), so computer security systems provide different kinds and amounts of security.
- Computer security is quite vast in scope, touching on many areas besides computer science. In this course we will study some *fundamentals*, some current *internet technologies*, and a little bit about *engineering and management* aspects.
- This short lecture describes the organization of the course, an outline of the topics from the syllabus, and gives a timeline of notable computer security events.

Security Fundamentals

By “fundamentals” we mean the basic concepts of secure systems, as well as the low-level design of security protocols.

- Security properties worth guarding: *confidentiality*, *integrity*, *authentication*, *availability*, *accountability*.
- A useful science: *cryptography*.
- Communicating carefully: *security protocols*.
- Increasing confidence: *formal techniques*.

Security Technologies

Which include . . .

- Privacy for the masses: *PGP*, *S/MIME*
- The consumer favourite: *SSL / TLS*
- Securing networks: *firewalls*, *DNSSec*, *IPSEC*
- Securing connections: `ssh`, *VPNs*.
- Programming securely: *Java security model*

We'll examine these kind of technologies in outline to understand how they are implemented and what they provide.

Engineering and Management

System security is an engineering and ultimately a management issue; technology is only part of a security “solution”.

We’ll consider engineering aspects of system security and higher-level issues, such as:

- Secure kernels and trusted computing bases
- Malicious code and network defences
- Security policy models, multi-level systems, security standards
- Real-world issues (usability and human factors; economics; legalities)

Outline

Lecture plan

- I will give **16 lectures covering main topics** from the syllabus.
- Main topics (over 16 lectures in a mixed order)
  - Introduction
  - Threats
  - Crypto
  - Security Protocols
  - Security models
  - Internet and network
  - Software
  - Web and email security
  - Usability and security
- Possible additional topics
  - More crypto (e-cash, zero-knowledge, steganography)
  - Formal techniques (info flow, static analysis)
  - Evidence-based certification.
  - Defences, IDS. Content protection (DRM). Legalities.

What lectures will not cover

- War stories, hacking tales, etc.

  You can read about these in many places, for example, Anderson's textbook, books like The Cuckoo's Egg, websites like `kevinmitnick.com`.

- Crypto application HOWTOs, personal firewall recommendations, . . .

  Though you should be able to apply what you learn in this course to such applications.

- Low-level details of security APIs

  Though the practical exercise will expose you to the use of APIs.

- Mathematical details of modern cryptography

  We'll cover crypto at a high level; you are encouraged to study the mathematics independently

Lectures will concentrate on underlying computer science behind computer security and engineering aspects of secure programming.

Outline

Assessment for the course

- You will be assessed on the **exam, weighted 100%**.
- Questions in the **exam** will test material covered in my lectures, the required reading, and material from the practical assignments.
- Concepts from other parts of the syllabus or the other lectures may be used as a basis for exam questions, but without assuming detailed knowledge.
- Past papers (some with solutions) are available from the ITO web page.

Practical assignments and Tutorials

- There also be **2 class tutorials**.
  Times and dates TBA on the course web page.
- There will be **2 assignments**, to be completed in advance of the tutorials, and will be issued 2 weeks prior. Solutions to the assignments will be discussed at the tutorials.
- You are **not required to submit answers** to the assignments. We will discuss solutions at the tutorials.
- Since the assignment material may contribute to the exam content, it is **highly recommended that you complete the assignments in advance of the tutorials** (even though the assignements themselves will not be assessed).

Outline

Standard security course advisory

- *Nothing in this course is intended as incitement to crack!*
- Breaking into systems to “demonstrate” security problems at best causes a headache to overworked sysadmins, at worst compromises systems for many users and could lead to **prosecution**.
- If you spot a security hole in a running system, **don’t exploit it**, instead consider contacting the relevant administrators confidentially.
- But be aware that keeping abreast with latest security patches and methods is difficult; practical security is a matter of weighing up risks and costs, so your advice may be not be quickly acted on.
- This is especially true in a relatively low security environment such as a university, where open access has traditionally been put above security, and resources for sysadmin are very tight.

Responsible security experiments

- If you want to experiment with security holes, play with your own machine, or better, your **own private network of machines**.
- One (mostly) harmless way of doing this is using a form of *virtualisation*: e.g., VMWare, UML, Xen.
- If you discover a new security hole in a standard application, or operating system routine that may be running at many sites, then consider contacting the vendor of the software (or vendor of the operating system which contains the software) in the first case. You might also raise the issue in a security forum for discussion, perhaps without providing complete details of the hole.
- The software vendor or other security experts will be able to confirm or deny, and work can begin on fixing the problem (if you haven't already suggested a fix).

Outline

Is there a computer security crisis?

- Almost every month: high-profile case of computer security failure reported in media. This gives the impression that security problems are prevalent.
- Partly salacious reporting: viruses, “underground” criminals, cyber-terrorism.
- . . . But the high frequency of security faults and incidents reported, e.g., on **BugTraq** and **CERT**, testify to many security problems in widely deployed systems.
- Yet there is a good body of knowledge in Computer Security and a history of carefully designed secure systems. It seems that most present-day security problems are due to *poor design of commodity systems and/or insufficient investment in ensuring security*. Under-investment may be misguided or deliberate policy.

Security timeline, part 1

Security attacks begin in 1950s and security mechanisms were designed for operating systems since the beginning. Early attackers were near the machines. Now the Internet allows millions of anonymous attackers to target any connected system. “White-hats” and “black-hats” are in an arms race...

- 1960 Memory protection hardware: partitioning, virtual memory.
- 1962 File access controls in multiple-access systems.
- 1967 One-way functions to protect passwords.
- 1968 Multics security kernel (BLP model)
- 1969–89 ARPANET ⇝ Internet; TCP/IP in 1977.

Infamously, ARPANET was built to withstand nuclear attack but was nearly crippled in 1988 by the Morris Internet Worm. ARPANET assumed centralised administration which no longer applies in the Internet: a dramatic example of a change in environment invalidating security.

Security timeline, part 2

1975	Unix-Unix copy protocol (UUCP) and mail trapdoors
1976	Public-key cryptography and digital signatures
1978	RSA public-key cryptosystem.
1978	First vulnerability study of passwords (intelligent search).
1978	E-cash protocols invented by David Chaum.
1983	Distributed domain naming system (DNS), vulnerable to spoofing.
1984	Viruses receive attention of researchers.
1985	Advanced password schemes.
1986	Wily hacker attack (Clifford Stoll's "Stalking...")
1988	Internet Worm: 6,000 computers (10% of Internet).
1988	Distributed authentication realised in Kerberos.
1989	Pretty Good Privacy (PGP) and Privacy Enhanced Mail (PEM).

Security timeline, part 3

1990 Anonymous remailers (protocols prevent tracing).

1993 Packet spoofing; firewalls; network sniffing.

1994 Netscape designs SSL v1.0 (revised 1995).

1996 SYN flooding. Java exploits. Web-site hacking.

1997 DNSSec security extension for DNS proposed.

1998 Script kiddies' scanner tools. IPSec proposals.

1999 First DDoS attacks. DVD encryption broken

2000 VBscript worm ILOVEYOU (0.5 – 8 million infections). Cult of the Dead Cow's Back Orifice 2000 Trojan.

Security timeline, part 4

- 2001 Code Red, Nimbda worm infects Microsoft IIS.
- 2002 Palladium; chipped XBox blocked from online play.
- 2003 W32/Blaster worm. Debian and FSF are cracked.
- 2004 First mobile phone virus Cabir
- 2005 Flaws in SHA-1. Sony's "rootkit" with broken DRM.
- 2006 RFID cracks.
  Microsoft Vista released; vulnerabilities discovered.
- 2007 Data breaches: TJX Inc (94m), UK HMRC (24m).
  iPhone released & cracked.
- 2008 Kaminsky discovers major DNS flaws. CIA reports power utility cyber-extortion. Oyster Cards cloned and UK e-passports faked.

Security timeline, part 5

2009 Conficker virus
iPhone worm
DoS attacks on social networks (Twitter, Facebook)
Numerous data breaches
Hacktivism
TJX Hacker indicted
BT & Phorm
“Privacy” at Facebook, Google, . . .
Cloud computing
. . .

Course feedback

- Security is a fast moving subject, with new research, security breaches and security technologies all appearing daily.
- I try to keep this course as up-to-date and interesting as possible, but with a necessarily academic focus on foundations and well-established tools.
- If there is some part of the course that you think could be improved, some topic or news item that you think deserves mention, please let me know.
- You are welcome to send comments or suggestions directly to me by email at `mjust@inf.ed.ac.uk`. If you wish to send feedback anonymously, find out how to use an anonymous remailer, or use the low-tech solution of leaving a note in my mail box.

Outline

Reading: textbooks

- Dieter Gollmann, *Computer Security*, 2nd Ed, John Wiley & Sons, 2006.
  A textbook providing a good (but brief) overview.
- Ross Anderson, *Security Engineering: A Comprehensive Guide to Building Dependable Distributed Systems*, John Wiley & Sons, 2nd Edition, 2008.
  More detail of whole-system and non-software aspects of security. Lots of interesting examples.
- Matt Bishop, *Computer Security: art and science*, Addison Wesley, 2003.
  Provides a good coverage of computer security topics, with pointers to research areas.
- Bruce Schneier, *Applied Cryptography*, J. Wiley & Sons, 2nd Ed, 1996.
  Classic practical crypto text; many algorithms and source code, but now somewhat dated (little mathematics).

Reading: other textbooks

- Nigel Smart, *Cryptography: An Introduction*, McGraw-Hill, 2003.
  Crypto and other computer security issues, more rigorous than Schneier. Electronic 3rd Ed at `http://www.cs.bris.ac.uk/~nigel/Crypto_Book/`.
- Michael Huth, *Secure communicating systems: design, analysis, and implementation.*, CUP, 2001.
  Includes cryptography and other techniques including protocols and information flow analysis.
- John Viega and Gary McGraw, *Building Secure Software: How to Avoid Security Problems the Right Way*, Addison-Wesley, 2001.
  The first book to be written on secure coding. Biased towards Unix programming; look for 2nd Edition of *Writing Secure Code* by Howard and LeBlanc (MS Press, 2003) for a Windows bias.

Reading: specialist books

- Simson Garfinkel and Gene Spafford, *Practical UNIX and Internet Security*, O'Reilly, 1996.
  Sound advice on security and good coverage of UNIX specifics. Internet side a little dated by now.
- William R. Cheswick and Steven M. Bellovin, *Firewall and Internet Security*, 2nd Edition, 2004.
  Classic book on internet security; original 1st ed online at `http://www.wilyhacker.com`.
- Alfred J. Menezes and Paul C. Van Oorschot and Scott A. Vanstone, *Handbook of Applied Cryptography*. CRC Press, 1996.
  A bible of crypto. Available online at `http://cacr.math.uwaterloo.ca/hac`.

Reading: background

- Bruce Schneier, *Secrets & Lies — Digital Security in a Networked World*, John Wiley & Sons, 2000.
  An entertaining and compulsive account of computer security needs and techniques, addressed at non-experts.
- Dorothy E. Denning, *Information Warfare and Security*, Addison-Wesley, 1999.
  Chilling accounts of impact of information in modern warfare.
- Simson Garfinkel, *Database Nation*, O'Reilly, 2001.
  The personal impact of ubiquitous electronic data storage (mainly from a US perspective, but concerns are relevant globally).

Reading: web resources

The web is particularly rich in resources for this subject. Interesting sites include home pages of security researchers, (often biased) technology news discussion forums, security services and advisory services, etc. A representative sample:

- `http://www.cl.cam.ac.uk/users/rja14/`
  — Ross Anderson's home page.
- `http://www.slashdot.org`
  — Slashdot, news for nerds (mostly CS students).
- `http://www.theregister.co.uk`
  — The Register (UK industry news)
- `http://www.cert.org/`
  — CERT advisories on security, incident statistics
- `http://www.securityfocus.com`
  — Hosts of popular mailing lists.
- `http://www.inf.ed.ac.uk/teaching/courses/cs`
  — Our homepage, with many more links.