

Secure Program Development—An Introduction

System Architecture and Security

René Rydhof Hansen

Aalborg University

19 SEP 2008

Goals: Course

You will **not**

- become an expert program auditor
- become an expert security developer

You will

- learn basics of IT security with focus on secure program development
- be able to perform simple security assessments of software and IT systems in general
- know how and where to find relevant and up-to-date information on vulnerabilities, exploits, and other security related topics
- know what common pitfalls to avoid when designing/developing programs
- acquire a new mindset(!)

Course (Part) Organisation

- Goal: to change your way of thinking
 - Security is not a static property
 - Fast-changing: must be able to cope with new attack forms
- Active discussion
 - In class
 - On course news group

Goals: Lecture

- You should be able to *analyse* simple scenarios using the CIA security model
- You should know the basics of secure program development projects
- You should be able to *find* information on security related issues in current software

Why bother?

- “The Internet Worm Incident”
- “Controlling the phone-lines” (Kevin Poulsen, 1990)
- “Citibank Incident” (Vladimir Levin, 1994)
- “The Omega Engineering Incident”
- “Brisbane Incident” (disgruntled employee, 2000)
- “The Microsoft Incident” stolen source code (????, 2000)
- Slammer
- “Virus at Hvidovre Hospital”
- San Francisco Network Admin

- Phishing, malware, identity-theft, ...

Why bother?

- “The Internet Worm Incident”
 - Who: Robert Morris
 - Why: curiosity(?)
 - When: November 1988
 - What: Internet-wide (unintended) DoS
 - How: Worm exploiting buffer overflows and bad passwords
- “Controlling the phone-lines” (Kevin Poulsen, 1990)
- “Citibank Incident” (Vladimir Levin, 1994)
- “The Omega Engineering Incident”
- “Brisbane Incident” (disgruntled employee, 2000)
- “The Microsoft Incident” stolen source code (????, 2000)
- Slammer
- “Virus at Hvidovre Hospital”
- San Francisco Network Admin

- Phishing, malware, identity-theft, ...

Why bother?

- “The Internet Worm Incident”
- “Controlling the phone-lines” (Kevin Poulsen, 1990)
 - Who: Kevin Poulsen
 - Why: Porche
 - When: 1990
 - What: took over phone-lines of local radio station (102nd caller gets a Porche)
 - How: long-time deep access to Pacific Bell’s switching networks
- “Citibank Incident” (Vladimir Levin, 1994)
- “The Omega Engineering Incident”
- “Brisbane Incident” (disgruntled employee, 2000)
- “The Microsoft Incident” stolen source code (????, 2000)
- Slammer
- “Virus at Hvidovre Hospital”
- San Francisco Network Admin

- Phishing, malware, identity-theft, ...

Why bother?

- “The Internet Worm Incident”
- “Controlling the phone-lines” (Kevin Poulsen, 1990)
- “Citibank Incident” (Vladimir Levin, 1994)
 - Who: Vladimir Levin
 - Why: \$10mill.
 - When: 1994
 - What: Simple money transfer
 - How: Bought passwords for highly insecure bank-network
- “The Omega Engineering Incident”
- “Brisbane Incident” (disgruntled employee, 2000)
- “The Microsoft Incident” stolen source code (????, 2000)
- Slammer
- “Virus at Hvidovre Hospital”
- San Francisco Network Admin

- Phishing, malware, identity-theft, ...

Why bother?

- “The Internet Worm Incident”
- “Controlling the phone-lines” (Kevin Poulsen, 1990)
- “Citibank Incident” (Vladimir Levin, 1994)
- “The Omega Engineering Incident”
 - Who: Tim Lloyd
 - Why: disgruntled employee
 - When: 1996
 - What: critical file server killed: losses of \$12mill., 80 people fired.
 - How: timed logic-bomb
- “Brisbane Incident” (disgruntled employee, 2000)
- “The Microsoft Incident” stolen source code (????, 2000)
- Slammer
- “Virus at Hvidovre Hospital”
- San Francisco Network Admin

- Phishing, malware, identity-theft, ...

Why bother?

- “The Internet Worm Incident”
- “Controlling the phone-lines” (Kevin Poulsen, 1990)
- “Citibank Incident” (Vladimir Levin, 1994)
- “The Omega Engineering Incident”
- “Brisbane Incident” (disgruntled employee, 2000)
 - Who: Vitek Boden
 - Why: disgruntled employee
 - When: 2000
 - What: sewage control computers reprogrammed to dump raw sewage (approx. 1.000.000 litres)
 - How: “hacking programs”
- “The Microsoft Incident” stolen source code (????, 2000)
- Slammer
- “Virus at Hvidovre Hospital”
- San Francisco Network Admin
- Phishing, malware, identity-theft, ...

Why bother?

- “The Internet Worm Incident”
- “Controlling the phone-lines” (Kevin Poulsen, 1990)
- “Citibank Incident” (Vladimir Levin, 1994)
- “The Omega Engineering Incident”
- “Brisbane Incident” (disgruntled employee, 2000)
- “The Microsoft Incident” stolen source code (????, 2000)
 - Who: ???
 - Why: ???
 - When: 2000
 - What: source code for Windows was stolen
 - How: Trojan-based password interception(?)
- Slammer
- “Virus at Hvidovre Hospital”
- San Francisco Network Admin
- Phishing, malware, identity-theft, ...

Why bother?

- “The Internet Worm Incident”
- “Controlling the phone-lines” (Kevin Poulsen, 1990)
- “Citibank Incident” (Vladimir Levin, 1994)
- “The Omega Engineering Incident”
- “Brisbane Incident” (disgruntled employee, 2000)
- “The Microsoft Incident” stolen source code (????, 2000)
- Slammer
 - Who: Chinese hackers(?)
 - Why: ???
 - When: 2003
 - What: rapid infection, led to DoS on emergency services (911), ATM machines, airline booking, several electrical and water utilities
 - How: Exploiting a buffer overflow
- “Virus at Hvidovre Hospital”
- San Francisco Network Admin
- Phishing, malware, identity-theft, ...

Why bother?

- “The Internet Worm Incident”
- “Controlling the phone-lines” (Kevin Poulsen, 1990)
- “Citibank Incident” (Vladimir Levin, 1994)
- “The Omega Engineering Incident”
- “Brisbane Incident” (disgruntled employee, 2000)
- “The Microsoft Incident” stolen source code (????, 2000)
- Slammer
- “Virus at Hvidovre Hospital”
 - Who: ???
 - Why: ???
 - When: 2006
 - What: major virus outbreak; 1500+ machines infected, 2+ weeks of disinfection and cleaning
 - How: Infected laptop/USB-stick(?)
- San Francisco Network Admin
- Phishing, malware, identity-theft, ...

Why bother?

- “The Internet Worm Incident”
- “Controlling the phone-lines” (Kevin Poulsen, 1990)
- “Citibank Incident” (Vladimir Levin, 1994)
- “The Omega Engineering Incident”
- “Brisbane Incident” (disgruntled employee, 2000)
- “The Microsoft Incident” stolen source code (????, 2000)
- Slammer
- “Virus at Hvidovre Hospital”
- San Francisco Network Admin
 - Who: Terry Childs
 - Why: disgruntled or fanatic employee
 - When: 2008
 - What: Unclear; suspected covert access and shutdown capability
 - How: trusted insider
- Phishing, malware, identity-theft, ...

Security — A Simple Model

Security — A Simple Model

Auditability, accountability, confidentiality, non-interference, delimited release, non-repudiability, authentication, assurance, trusted computing base, non-malleability, intrusion detection, intrusion prevention, tamper-proof, tamper evident, robust declassification, anonymity, privacy, fail-stop, fail-safe, access control, Biba, trust management, non-disclosure, intransitive non-interference, Bell-LaPadula, tranquility, insider, decentralised label model, secure information flow, cryptographic mix nets, onion router, eternity service, selective dependency, erasure policies, principle of least privilege, sandboxing, penetration test, trusted path, separation of duty, integrity, cryptography, hash function, key exchange protocol, Diffie-Hellman, steganography, tempest, side channel attack, man in the middle, biometrics, red/black networks, firewall, polymorphic virus, worm, denial of service, buffer overflow, privilege escalation, one-time pad, stream cipher, reference monitor, Feistel network, differential power analysis, Dolev/Yao attacker, time of check to time of use, Common Criteria, rainbow series, VPN, availability, stack inspection, . . .

Security — A Simple Model

Auditability, accountability, **confidentiality**, non-interference, delimited release, non-repudiability, authentication, assurance, trusted computing base, non-malleability, intrusion detection, intrusion prevention, tamper-proof, tamper evident, robust declassification, anonymity, privacy, fail-stop, fail-safe, access control, Biba, trust management, non-disclosure, intransitive non-interference, Bell-LaPadula, tranquility, insider, decentralised label model, secure information flow, cryptographic mix nets, onion router, eternity service, selective dependency, erasure policies, principle of least privilege, sandboxing, penetration test, trusted path, separation of duty, **integrity**, cryptography, hash function, key exchange protocol, Diffie-Hellman, steganography, tempest, side channel attack, man in the middle, biometrics, red/black networks, firewall, polymorphic virus, worm, denial of service, buffer overflow, privilege escalation, one-time pad, stream cipher, reference monitor, Feistel network, differential power analysis, Dolev/Yao attacker, time of check to time of use, Common Criteria, rainbow series, VPN, **availability**, stack inspection, . . .

Security — A Simple Model

- *Confidentiality*
- *Integrity*
- *Availability*

Security — A Simple Model

- *Confidentiality*

- Keeping secrets... secret
- Example: “no send after file read”
- Example: “credit card numbers are not stored permanently”

- *Integrity*

- Making sure data can be trusted
- ... and is only changed in an authorised way
- Example: “no write after untrusted read”
- Example: “two-signature schemes”

- *Availability*

- Making sure authorised users can use their data
- Example: ???

Incidents Revisited

- “The Internet Worm Incident”
- “Controlling the phone-lines” (Kevin Poulsen, 1990)
- “Citibank Incident” (Vladimir Levin, 1994)
- “The Omega Engineering Incident”
- “Brisbane Incident” (disgruntled employee, 2000)
- “The Microsoft Incident” stolen source code (????, 2000)
- Slammer
- “Virus at Hvidovre Hospital”
- San Francisco Network Admin

Incidents Revisited

- “The Internet Worm Incident”
 - Who: Robert Morris
 - Why: curiosity(?)
 - When: November 1988
 - What: Internet-wide (unintended) DoS
 - How: Worm exploiting buffer overflows and bad passwords
- “Controlling the phone-lines” (Kevin Poulsen, 1990)
- “Citibank Incident” (Vladimir Levin, 1994)
- “The Omega Engineering Incident”
- “Brisbane Incident” (disgruntled employee, 2000)
- “The Microsoft Incident” stolen source code (????, 2000)
- Slammer
- “Virus at Hvidovre Hospital”
- San Francisco Network Admin

Incidents Revisited

- “The Internet Worm Incident”
- “Controlling the phone-lines” (Kevin Poulsen, 1990)
 - Who: Kevin Poulsen
 - Why: Porche
 - When: 1990
 - What: took over phone-lines of local radio station (102nd caller gets a Porche)
 - How: long-time deep access to Pacific Bell’s switching networks
- “Citibank Incident” (Vladimir Levin, 1994)
- “The Omega Engineering Incident”
- “Brisbane Incident” (disgruntled employee, 2000)
- “The Microsoft Incident” stolen source code (????, 2000)
- Slammer
- “Virus at Hvidovre Hospital”
- San Francisco Network Admin

Incidents Revisited

- “The Internet Worm Incident”
- “Controlling the phone-lines” (Kevin Poulsen, 1990)
- “Citibank Incident” (Vladimir Levin, 1994)
 - Who: Vladimir Levin
 - Why: \$10mill.
 - When: 1994
 - What: Simple money transfer
 - How: Bought passwords for highly insecure bank-network
- “The Omega Engineering Incident”
- “Brisbane Incident” (disgruntled employee, 2000)
- “The Microsoft Incident” stolen source code (????, 2000)
- Slammer
- “Virus at Hvidovre Hospital”
- San Francisco Network Admin

Incidents Revisited

- “The Internet Worm Incident”
- “Controlling the phone-lines” (Kevin Poulsen, 1990)
- “Citibank Incident” (Vladimir Levin, 1994)
- “The Omega Engineering Incident”
 - Who: Tim Lloyd
 - Why: disgruntled employee
 - When: 1996
 - What: critical file server killed: losses of \$12mill., 80 people fired.
 - How: timed logic-bomb
- “Brisbane Incident” (disgruntled employee, 2000)
- “The Microsoft Incident” stolen source code (????, 2000)
- Slammer
- “Virus at Hvidovre Hospital”
- San Francisco Network Admin

Incidents Revisited

- “The Internet Worm Incident”
- “Controlling the phone-lines” (Kevin Poulsen, 1990)
- “Citibank Incident” (Vladimir Levin, 1994)
- “The Omega Engineering Incident”
- “Brisbane Incident” (disgruntled employee, 2000)
 - Who: Vitek Boden
 - Why: disgruntled employee
 - When: 2000
 - What: sewage control computers reprogrammed to dump raw sewage (approx. 1.000.000 litres)
 - How: “hacking programs”
- “The Microsoft Incident” stolen source code (????, 2000)
- Slammer
- “Virus at Hvidovre Hospital”
- San Francisco Network Admin

Incidents Revisited

- “The Internet Worm Incident”
- “Controlling the phone-lines” (Kevin Poulsen, 1990)
- “Citibank Incident” (Vladimir Levin, 1994)
- “The Omega Engineering Incident”
- “Brisbane Incident” (disgruntled employee, 2000)
- “The Microsoft Incident” stolen source code (????, 2000)
 - Who: ???
 - Why: ???
 - When: 2000
 - What: source code for Windows was stolen
 - How: Trojan-based password interception(?)
- Slammer
- “Virus at Hvidovre Hospital”
- San Francisco Network Admin

Incidents Revisited

- “The Internet Worm Incident”
- “Controlling the phone-lines” (Kevin Poulsen, 1990)
- “Citibank Incident” (Vladimir Levin, 1994)
- “The Omega Engineering Incident”
- “Brisbane Incident” (disgruntled employee, 2000)
- “The Microsoft Incident” stolen source code (????, 2000)
- Slammer
 - Who: Chinese hackers(?)
 - Why: ???
 - When: 2003
 - What: rapid infection, led to DoS on emergency services (911), ATM machines, airline booking, several electrical and water utilities
 - How: Exploiting a buffer overflow
- “Virus at Hvidovre Hospital”
- San Francisco Network Admin

Incidents Revisited

- “The Internet Worm Incident”
- “Controlling the phone-lines” (Kevin Poulsen, 1990)
- “Citibank Incident” (Vladimir Levin, 1994)
- “The Omega Engineering Incident”
- “Brisbane Incident” (disgruntled employee, 2000)
- “The Microsoft Incident” stolen source code (????, 2000)
- Slammer
- “Virus at Hvidovre Hospital”
 - Who: ???
 - Why: ???
 - When: 2006
 - What: major virus outbreak; 1500+ machines infected, 2+ weeks of disinfection and cleaning
 - How: Infected laptop/USB-stick(?)
- San Francisco Network Admin

Incidents Revisited

- “The Internet Worm Incident”
- “Controlling the phone-lines” (Kevin Poulsen, 1990)
- “Citibank Incident” (Vladimir Levin, 1994)
- “The Omega Engineering Incident”
- “Brisbane Incident” (disgruntled employee, 2000)
- “The Microsoft Incident” stolen source code (????, 2000)
- Slammer
- “Virus at Hvidovre Hospital”
- San Francisco Network Admin
 - Who: Terry Childs
 - Why: disgruntled or fanatic employee
 - When: 2008
 - What: Unclear; suspected covert access and shutdown capability
 - How: trusted insider

Biggest Threat to Security?

Biggest Threat to Security?

- Humans!
 - Error-prone
 - Social engineering
 - Solution?
- This course: focus on software
 - Almost always involved somehow
 - Foundation for all IT-systems
 - Ubiquity of IT-systems, e.g., embedded systems
- Book perspective: Risk Management
 - Traditional perspective: binary
 - Maybe security is not the most important aspect
 - Cost/benefit for entire business/organisation/application

The Main Problem

Developing a bug-free program is hard!

- Ubiquity of computer networks
 - Remote access
 - “Anonymous” access
- Size and complexity
 - Windows NT approx. 35 mill. LOC
 - Industry standard: approx. 1 error pr. 1000 LOC
- Widespread use of low-level languages
 - C/C++
 - Assembly (embedded systems)
 - High-level languages = no problems?
- Extensibility
 - Plug-ins
 - Scriptability
 - Auto-update

A sidenote: Security \subseteq Reliability

Definition (Security)

Enforcing a policy that describes rules for accessing resources.

- Pre-supposes that security violations are “enabled” by bugs
- Reliable programs are secure?

Traditional approach: Penetrate and patch

- Security implemented as post-hoc reviews
- Find a bug (penetrate), make patch

Problems:

- Patches (usually) only cover known problems
- Patches are often rush jobs
- Patches only fix the symptom
- Patches are often not applied

Better approach: sooner good, later bad

Much much cheaper to find and remove bugs earlier

- Integrate security into every phase of the project
- Set up security goals/requirements

Security Goals

According to the book:

- Prevention
- Traceability and Auditing
- Monitoring
- Privacy and Confidentiality
- Multilevel Security
- Anonymity
- Authentication
- Integrity

Goals? Mechanisms? Better suggestion?

Security Goals/Policies

Based on: Confidentiality, Integrity, Availability

Pitfalls

- Two classes: architectural and implementation
- Specific to security: trivial implementation errors can have drastic consequences
- Other problems: Social engineering, network attacks
 - Eavesdropping
 - Tampering
 - Spoofing
 - Hijacking
 - Capture/replay

“Functional” Software Project Goals vs. Security

- Functionality
- Usability
- Efficiency
- Time-to-market
- **Simplicity**

The Risk Management Perspective

Don't let security get in the way of the real goals

Software Risk Management for Security

- “Spiral” project development (not waterfall)
- Security engineer/lead
 - Don't wait til end-of-project
 - Should be integrated from start of project
 - Deriving requirements
 - Risk assessment
 - Design for security
 - Implementation
 - Security Testing
 - Should recognise legitimate business concerns (no firewall-fascists or non-interference nazis)
- Avoid: Black Box Testing
- Avoid: Red Teaming

Interesting Quote

“..., XP will probably have a negative impact on software security”

Common Criteria

- Standard for IT-security products
- No inherent project-development model
- Problem: politics

Summary

- Security incidents
- CIA Model
- Threat(s) to security
- Security goals/policies
- Risk management and software development projects

Next time: vulnerabilities, exploits, and auditing