Computer Security from the Trojan Wars to the Present

Charles P. Pfleeger
Exodus Security Services
charles.pfleeger@exodus.net
Overview

- Prehistory
- Cave Dwellers
- End of Isolationism
- Penetrating the Fortress
- Gilded Age
- Storms Brewing
- Today
- Tomorrow
Message

- Those who fail to study history are doomed to repeat it
- Yesterday’s unsolved problems don’t go away
- Not everything in computer security was discovered since 1995
- There is much value in some of the fundamental/foundational papers in computer security
What is Security

• **Confidentiality**
  – information available for reading only when authorized

• **Integrity**
  – information available for modification only when authorized

• **Availability**
  – information available for use when authorized
Threat Examples

- **Confidentiality**
  - unauthorized viewing
- **Integrity**
  - unauthorized modification
- **Availability**
  - denial of authorized access
Method - Opportunity - Motive

• **Method**
  – tools, techniques, knowledge

• **Opportunity**
  – access, ability

• **Motive**
  – desire

• **Work factor**
  – difficulty, time
Prehistory
Prehistory

- **Alan Turing: Bletchley Park**
  - Robinson, Colossus, ACE, Manchester
    Automatic Digital Machine (Madm)
  - “I suppose when [computers] get to that stage, we shan’t know how [they] do it.”

- **Mark I, Harvard:**-IBM
- **ENIAC, Edvac, Binac, Univac→Rand→Sperry→Unisys**
- **Total demand for computers:** approx. 10
- **The first bug**
Dawn of History: circa 1955

- Security? Who, me? What threat?
Dawn of History: circa 1955

- **Security? Who, me? What threat?**
  - single-user systems
  - user is main threat
    - stored program
    - correctness
  - hardware reliability

- **Real “security through obscurity”**
  - handful of computer literates
  - strong physical security
Cave Dwellers Discover Fire: 1960
Cave Dwellers Discover Fire: 1960

- Multiuser systems
- Protecting whom from whom
- Hardware-enforced protection
Multiuser Systems

• **Mode of use**
  – serial use
  – serial reuse
  – shared access
    • code (programs, libraries)
    • data

• **Executive**

• **User in control**
Protecting Whom from What

• **Threats**
  – user error: code integrity
    • harm self
    • harm others
  – user error: denial of service
    • harm self
    • harm others
  – hardware/system error/failure
    • harm stored code/data
Hardware-Enforced Protection

- Memory separation
  - separation between system and process space
- Privileged mode of execution
- Timer
- Integrity checking/correction
  - parity, other error coding
Concurrent Multiprogramming

- **User-user separation**
- **Threat**
  - Incompetent (non-malicious) co-users
  - Malicious users
- **System in control**
Multiprogramming Operating Systems

- Hardware/operating system combined
- Largely single vendor
- Example families:
  - IBM OS
  - Burroughs B5000
  - GE 645
  - Honeywell
How the Grinch Stole Systems
How the Grinch Stole Systems

Willis Ware (chair), 1967 Defense Science Board Study

• Problem: Significant number of systems being acquired for military use

• Charge: Formulate recommendations for hardware and software safeguards to protect classified information in multi-user, resource-sharing computer systems
End of Isolationism

• Isolation and physical protection no longer adequate/appropriate/feasible

• Geographic spread
  – remote access
  – sharing across distance

• User-user threat model no longer adequate

• Vulnerabilities
  – accidental disclosure
  – deliberate penetration
  – active infiltration
  – passive subversion
Observations

“As of [1969]

- “It is virtually impossible to verify that a large software system is completely free of errors and anomalies
- “The state of system design of large software systems is such that frequent changes to the system can be expected...
- “System failure modes are not thoroughly understood, catalogued, or protected against
- “Large hardware complexes cannot be absolutely guaranteed error-free.”
Software

• **Language processors:**
  – Assembler languages and processors for them pose problems because seemingly safe instruction sequences can execute to disrupt service or bypass security controls

• **Supervisor program**
  – As much of the supervisor to run in user state as possible
Research Required

- Hardware and software to maintain absolute segregation
- Automatic recertification procedures for system itself
- Comprehensive automatic monitors for security controls
- Self-checking hardware controls
- Methodology for identifying failure modes
- “New architectures whose security controls minimally affect system efficiency or cost”
Penetrating the Fortress
Penetrating the Fortress

• **Primary security validation method**
  – Gain confidence
  – Assess vulnerabilities
  – Identify flaws for repair
  – Specify future system requirements
  – Clarify unresolved R&D issues

• **Success = finding flaw(s)**

• **Flaw Hypothesis methodology**
  – Generate flaw hypotheses
  – Confirm (refute) hypothesis that flaw exists
  – Generalize confirmed flaws into new hypotheses
Generic Flaws

• Inadequate identification/authentication
• Incomplete checking
  – Unclear point of check
  – Incomplete conditional case analysis
• Unauthorized control
  – Time-of-check to time-of-use
  – Read before write, read past EOF
  – Self-modifying code
  – Uncoordinated concurrency
Typical Flaw Areas

- Resource sharing mechanisms
- User interface
- Configuration management controls
- Authentication controls
- Added-on features; design modifications
- Parameter checking
- Error handling
- Side effects
- Parallelism
- Complex interfaces
- Duplication of function
- Access to residual information
- Violation of design principles
Characteristics of Methodology

- **Positive**
  - Cheap
  - Powerful
  - Systematic

- **Negative**
  - Human-centered
    - Labor-intensive
    - Variable
  - Not a formal demonstration of correctness

- **Observation**
  - Typically 3-6 calendar month effort; 3-6 persons
Technology to the Rescue (?)
James P Anderson

- Problem: How to provide information systems secure against the threat from a malicious user
- “It is clear to the panel that solutions will not occur ... from the various well-intentioned attempts to provide security as an add-on to existing systems.”
“In order to defend against a malicious user, one must design the security controls into the operating system of a machine to control the actions not only of each user, but of the many parts of the operating system itself, if it is acting on a user’s behalf.”

“The issue of computer security is one of completeness rather than degree.”

“Completeness [requires] that security be designed into systems at their inception.”
Regarding Penetration Exercises

- Tiger teams expend bounded energy to demonstrate the security inadequacy of standard or security-upgraded systems.
- Even if corrections made to fix flaws found, no assurance all flaws found and corrected.
- “It is a commentary on contemporary systems that none of the known tiger team efforts has failed [to find a flaw] to date.”
A Rigorous Security Design Model

- Controlled sharing
- Reference monitor
  - tamperproof
  - always invoked
  - small enough to be subject to analysis and tests, the completeness of which can be assured
- Building a secure system
  - define threats
  - define conceptual secure design
  - implement correctly
The Age of Dinosaurs: 1970s
The Age of Dinosaurs: 1970s

- **More complex operating systems**
  - capabilities, segmentation, indirection, scheduling, multitasking, multiprocessing, ...
  - many implications on protection
- **System becomes a computing utility**
  - reliability (protection from others and from nature) required
- **Computer becomes indispensable**
- **Genetic diversity**
B. Lampson

- Motivation for protection mechanisms: protect one user from malice or error of another user
- Reasons for protection just as strong if “user” is replaced by “program.”
- “A system can be complete from the point of view of a community of friendly and infallible users, without any protection at all.”
Dinosaurs Beget a Eunuch (or two)
Dinosaurs Beget a Eunuch (or two)

- Frustration with big, clumsy, costly, inefficient, uncontrollable mainframes
- Small, lightweight, modular, simple operating system of composable pieces
- For researchers, scientists
- Small user community
“Then We Won’t Know How It Does It”
“Then We Won’t Know How It Does It”

Ken Thompson

• Curse of the stored-program computer concept

• Q: “Why rob a bank?” A: “That’s where the money is.”

• Ken Thompson’s Trojan horse compiler

• “You can’t trust code that you did not totally create yourself.”
Unix Security

• **Password security**
  – Original model based on human user
  – Password crackers
    • brute force attacks
    • likely passwords

• “Superuser”

• Login screen spoofs

• “It is one thing to clean up a system by plugging open holes, and quite another to install security machinery that collects evidence of possible chicanery.”
Now We Know How to Do It Right
Now We Know How to Do It Right

Saltzer & Schroeder
- Economy of mechanism
- Fail-safe defaults
- Complete mediation
- Open design
- Psychological acceptability (ease of use)

- Least privilege
- Separation of privilege
- Least common mechanism
- Large work factor
- Compromise recording
1975-1985: The Gilded Age
1975-1985: The Gilded Age

- 1970s: period of intense research efforts in computer security
- Trusted systems
  - KSOS
  - PSOS
  - KVM
  - UCLA Secure Unix
- Computer Security Act
- Evaluation criteria
- U.S. National Computer Security Center
Jim Anderson

- **Shared responsibility:**
  - designers
  - manufacturers
  - government
Mid-1980s: Storms Brewing

• We Haven’t Reached Nirvana Yes
• New Kid on the Block
• The Winds of War
Crocker & Bernstein

• Communications backbone for large, complex U.S. Strategic Defense Initiative (SDI).

• “From a security perspective, assured service within the communication network is paramount ... Without assured service confidentiality and integrity are irrelevant.”

• Redundancy to counter expected errors is well understood; study’s goal is to eliminate flaws in design and implementation
## Disruption Causes

<table>
<thead>
<tr>
<th>Date</th>
<th>Failure</th>
<th>Cause</th>
<th>Severity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1970</td>
<td>Reassembly lockup</td>
<td>Algorithm flaw</td>
<td>3</td>
</tr>
<tr>
<td>1970</td>
<td>S&amp;F lockup</td>
<td>Algorithm flaw</td>
<td>(3)</td>
</tr>
<tr>
<td>1971</td>
<td>Black hole</td>
<td>Fault intolerance</td>
<td>3</td>
</tr>
<tr>
<td>1973</td>
<td>Christmas lockup</td>
<td>Resource exhaustion</td>
<td>3</td>
</tr>
<tr>
<td>1973</td>
<td>Masquerade</td>
<td>Fault intolerance</td>
<td>3</td>
</tr>
<tr>
<td>1973</td>
<td>Routing storm 1</td>
<td>Fault intolerance</td>
<td>3</td>
</tr>
<tr>
<td>1974</td>
<td>Resequence deadlock</td>
<td>Algorithm flaw</td>
<td>(3)</td>
</tr>
<tr>
<td>1974</td>
<td>Single packet turbulence</td>
<td>Use beyond intention</td>
<td>1</td>
</tr>
<tr>
<td>1974</td>
<td>Routing loops</td>
<td>Algorithm flaw</td>
<td>(2)</td>
</tr>
<tr>
<td>1976</td>
<td>Piggyback lockup</td>
<td>Algorithm flaw</td>
<td>(2)</td>
</tr>
<tr>
<td>1976</td>
<td>Phasing</td>
<td>Resource exhaustion</td>
<td>1</td>
</tr>
<tr>
<td>1980</td>
<td>Routing storm 2</td>
<td>Fault intolerance</td>
<td>3</td>
</tr>
<tr>
<td>1986</td>
<td>Crossed nets</td>
<td>Fault intolerance</td>
<td>3</td>
</tr>
<tr>
<td>1987</td>
<td>SRI IMP Crash</td>
<td>Configuration control</td>
<td>2</td>
</tr>
<tr>
<td>1987</td>
<td>NEE bug</td>
<td>Inadequate specification</td>
<td>2</td>
</tr>
<tr>
<td>1988</td>
<td>Routing storm 3</td>
<td>Fault intolerance</td>
<td>3</td>
</tr>
<tr>
<td>1988</td>
<td>IST table overflow</td>
<td>Resource exhaustion</td>
<td>2</td>
</tr>
</tbody>
</table>
Contributing Factors

- **ARPANET routing algorithm very complex**
  - distributed, adaptive nature
  - error in one node may quickly affect entire network

- **ARPANET software has evolved over time**
  - new functions, hardware, interfaces
  - maintenance changes have introduced problems
Middle Ages:
The Plague (Viral)
Viruses

• Virus vs. Trojan horse

• Origins
  – 1981: Apple II attacks
  – 1986: PC - Brain

• Types
  – boot sector
  – system
  – application

• 1985-1990

• 1995-present
Information Warfare: A Schell Game

Grant & Riche 1983

- Prediction of enemy takeover by malicious code infiltration of electronic infrastructure
The Eagle’s Own Plume

- Ease of introduction of Trojan horses into sensitive systems
- Can affect military and commercial systems
- Documented cases of both
- Size, complexity, decomposition, isolation allow attack
- Size and complexity also make it difficult to determine what attack has been planted, or if an attack is discovered, what is the effect of that attack
Calls for Action

- Expertise in software engineering, effective implementation of hardware components, and design of resource-sharing networks small relative to other technical disciplines
- This country is the world leader in computer technology, with a qualitative edge based upon research. It would be negligent and foolish to blunt this edge by ignoring the computer security problem.
The Integri-Tea Party

Welke & Mayfield 1990
• What do we mean by “integrity”?
Flavors of Integrity

- Modified only by authorized subjects
- Modified only by authorized processes
- Modified only in authorized ways
- What is stored/transmitted is what is retrieved/received
- Internally consistent
- Precise; precise enough
- Fit for purpose
Integrity Enforcement

• No one size fits all
• Example techniques
  – Access control
  – Error detection/correction code
  – Binding of objects to methods
  – Domains of execution
• Research needed
Today: The Web
Web Characteristics

- Wide availability, to the masses
- Mandatory presence
- Very low cost of entry
- Very low skill to enter
- Low genetic diversity
- Very rapid technology turnover
- High demand for “oh, wow”
Script Kiddies

- **Satan, Crack**
  - repetitive probing analysis

- **Ping of death, Smurf**
  - protocol failures

- **Unnamed**
  - buffer overruns
  - packet sniffing
Hostile Mobile Code

- Java applets, linked objects
- Code runs with privilege of victim
Cookies

- Encrypted token
- Retain state between separate web server accesses
- Format, content proprietary
- Harmless by themselves, but
- Vehicle for transmission in conjunction with other attack code
Web Site Takeovers

- **New York Times**
  - down for entire weekend
- **Department of Justice**
  - several attacks
- **CIA,**...
Easter Eggs

- **Microsoft Excel 97**
  - Open a new worksheet
    - Press `<F5>`, type X97:L97 `<Enter><Tab>`
    - Hold `<Ctrl-Shift>` and click Chart Wizard
  - Next appears ...

Flight Simulator

Use mouse to navigate: right mouse button for forward, left for reverse
Or list of developers’ names
The Future
The Future

- Those who fail to study history are doomed to repeat it
- Pace of technological advance; pace of advance in computer security
- Relationship between marketing—development—design
- Research
  - government (defense) funded
  - government (non-defense) funded
  - commercial funded
Bibliography (1)


Bibliography (2)

Bibliography (3)

- Thompson, K., “Reflections on Trusting Trust,” Commun. of the ACM,