Computer Security from the Trojan Wars to the Present

Charles P. Pfleeger

Exodus Security Services

charles.pfleeger@exodus.net

EXODUS®

Overview

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

EXODUS®

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

EXODUS®

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

EXODUS®

Threat Examples

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

EXODUS®

Method - Opportunity - Motive

- **Method**
  - tools, techniques, knowledge
- **Opportunity**
  - access, ability
- **Motive**
  - desire
- **Work factor**
  - difficulty, time

EXODUS®

Prehistory

EXODUS®

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**

EXODUS®

Dawn of History: circa 1955

- Security? Who, me? What threat?

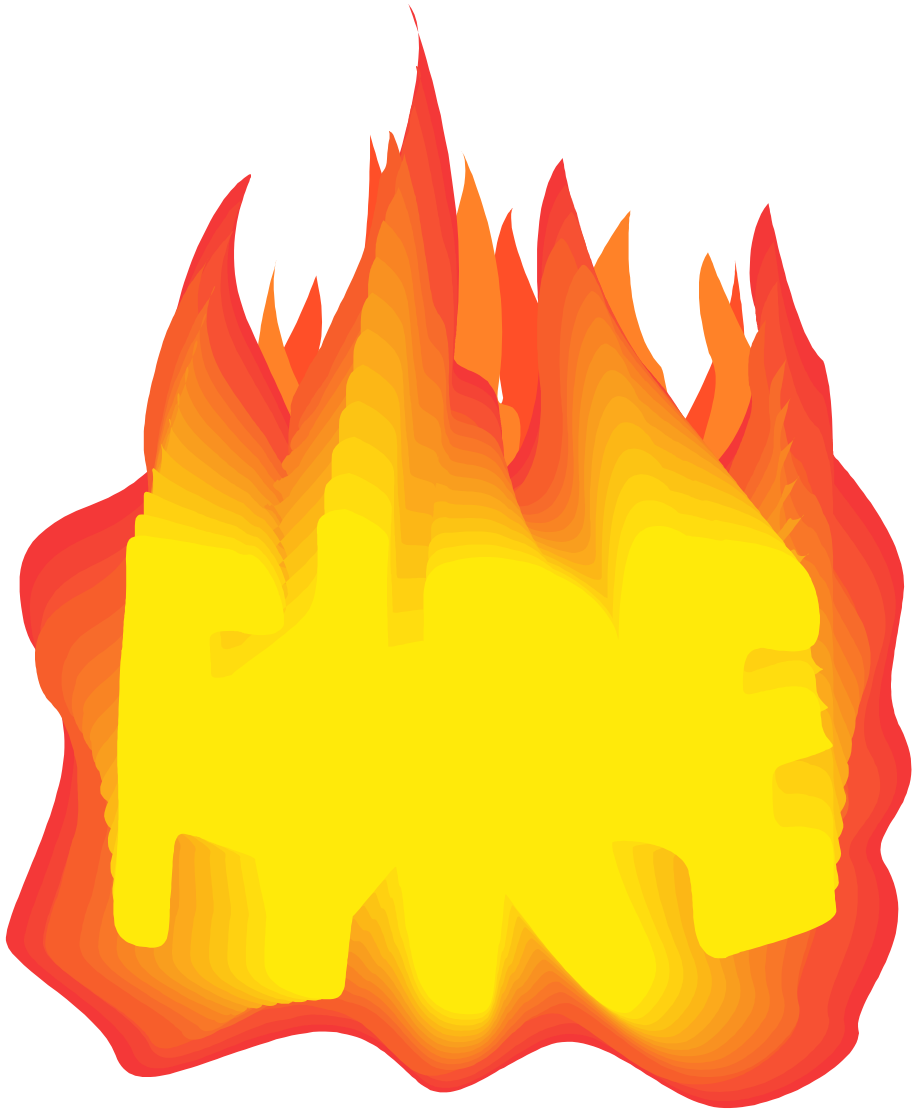
EXODUS®

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

EXODUS®

Cave Dwellers Discover Fire: 1960



EXODUS®

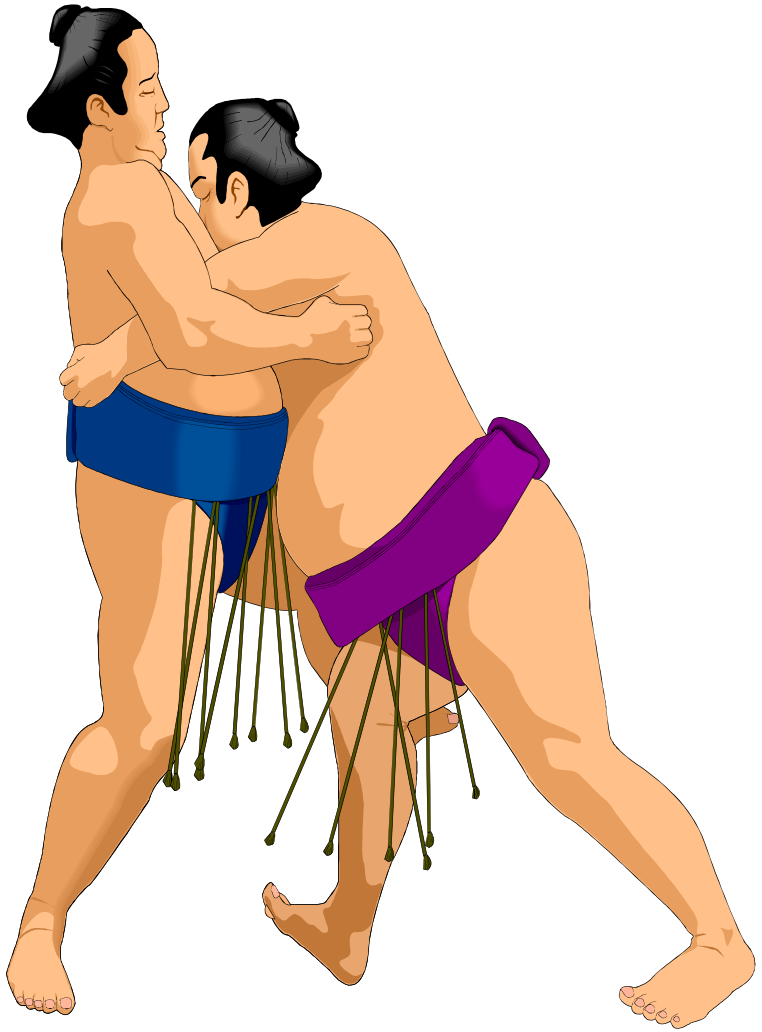
Cave Dwellers Discover Fire: 1960

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

EXODUS®

Multiuser Systems

- **Mode of use**
  - serial use
  - serial reuse
  - shared access
    - code (programs, libraries)
    - data
- **Executive**
- **User in control**



EXODUS®

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

EXODUS®

Hardware-Enforced Protection

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

EXODUS®

Concurrent Multiprogramming

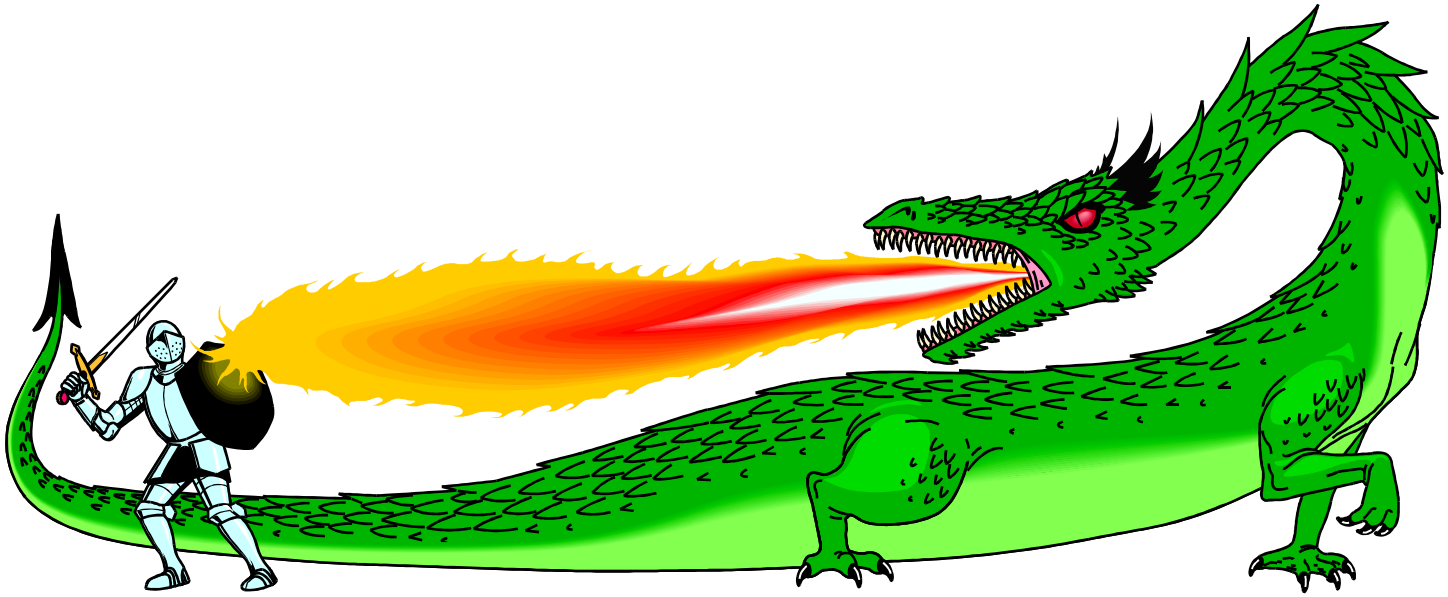
- **User-user separation**
- **Threat**
  - Incompetent (non-malicious) co-users
  - Malicious users
- **System in control**

Multiprogramming Operating Systems

- **Hardware/operating system combined**
- **Largely single vendor**
- **Example families:**
  - IBM OS
  - Burroughs B5000
  - GE 645
  - Honeywell

EXODUS®

How the Grinch Stole Systems



EXODUS®

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**

EXODUS®

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

EXODUS®

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."

EXODUS®

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

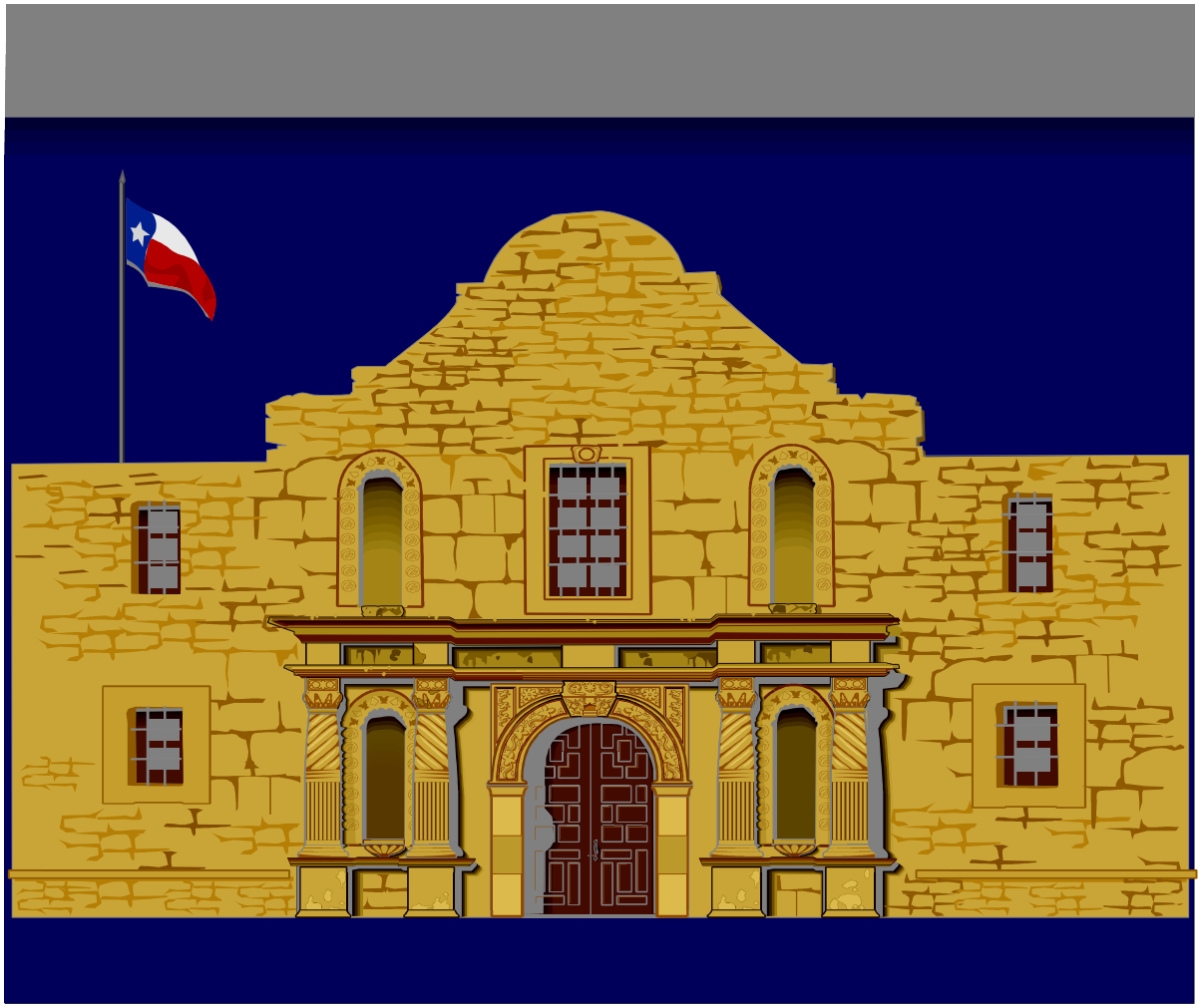
EXODUS®

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”

EXODUS®

Penetrating the Fortress



EXODUS®

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

EXODUS®

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

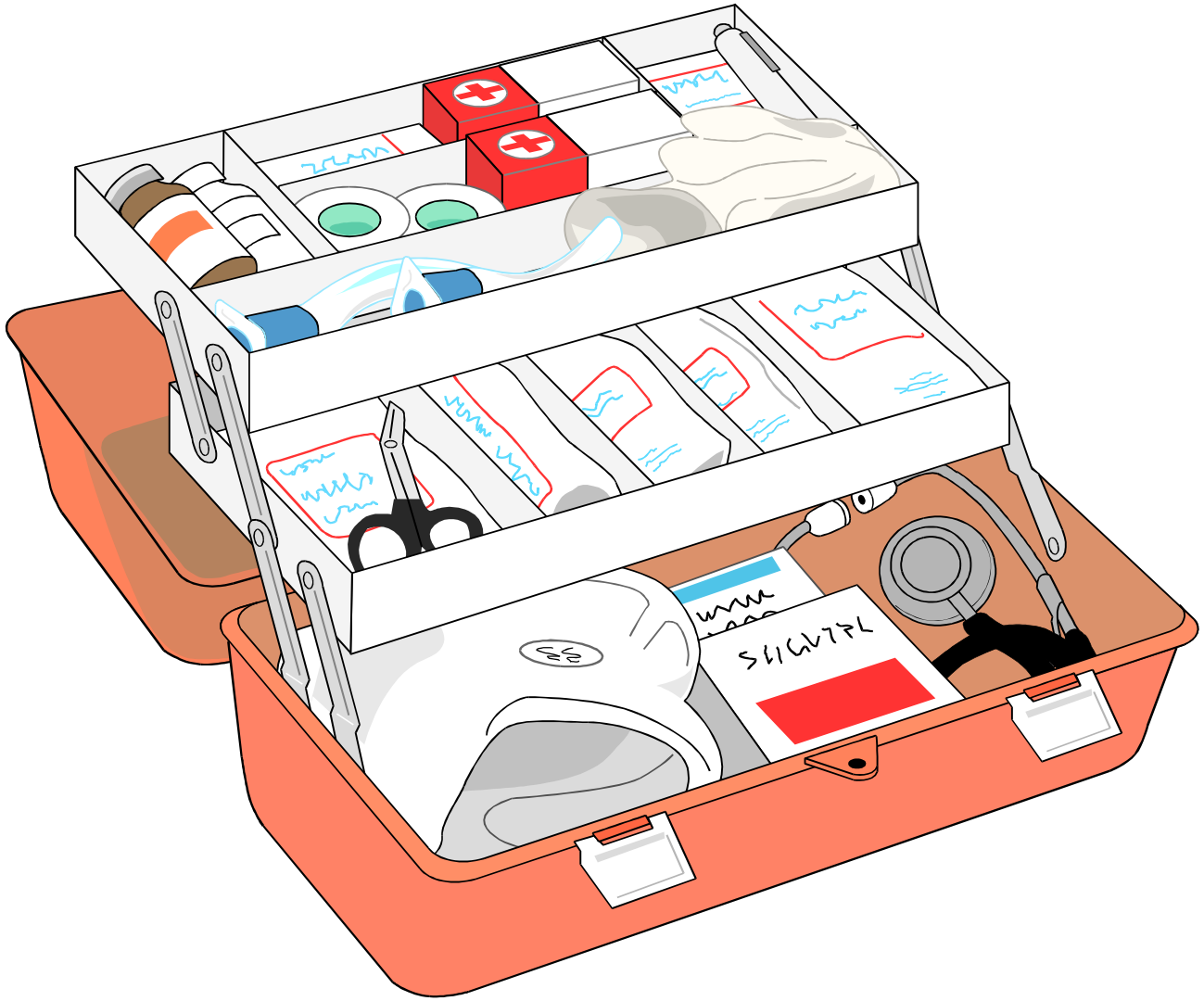
EXODUS®

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

EXODUS®

Technology to the Rescue (?)



EXODUS®

Technology to the Rescue (?)

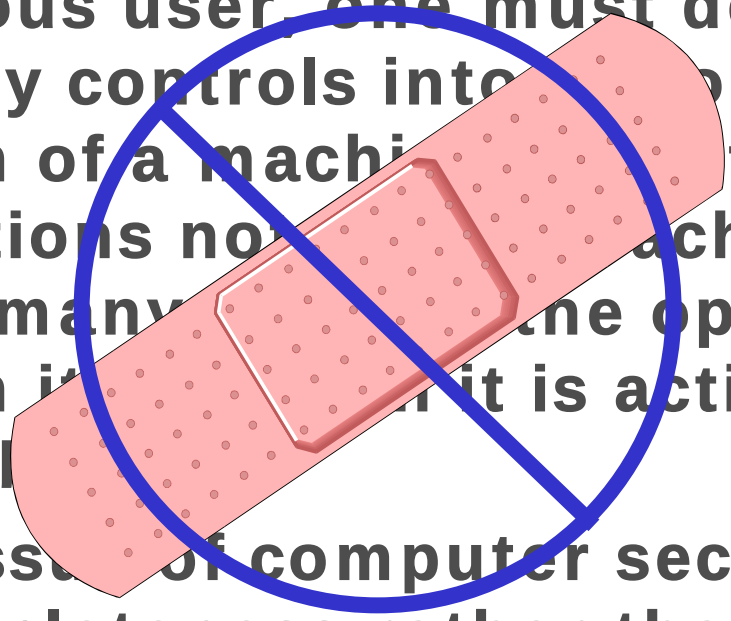
James P Anderson

- **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.”**

EXODUS®

Add-Ons Rejected

- "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 of each user, but of the many [illegible] the operating system it is on when it is acting on a user's [illegible]
- "The issue of computer security is one of completeness rather than degree."
- "Completeness [requires] that security be designed into systems at their inception."



EXODUS®

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.”

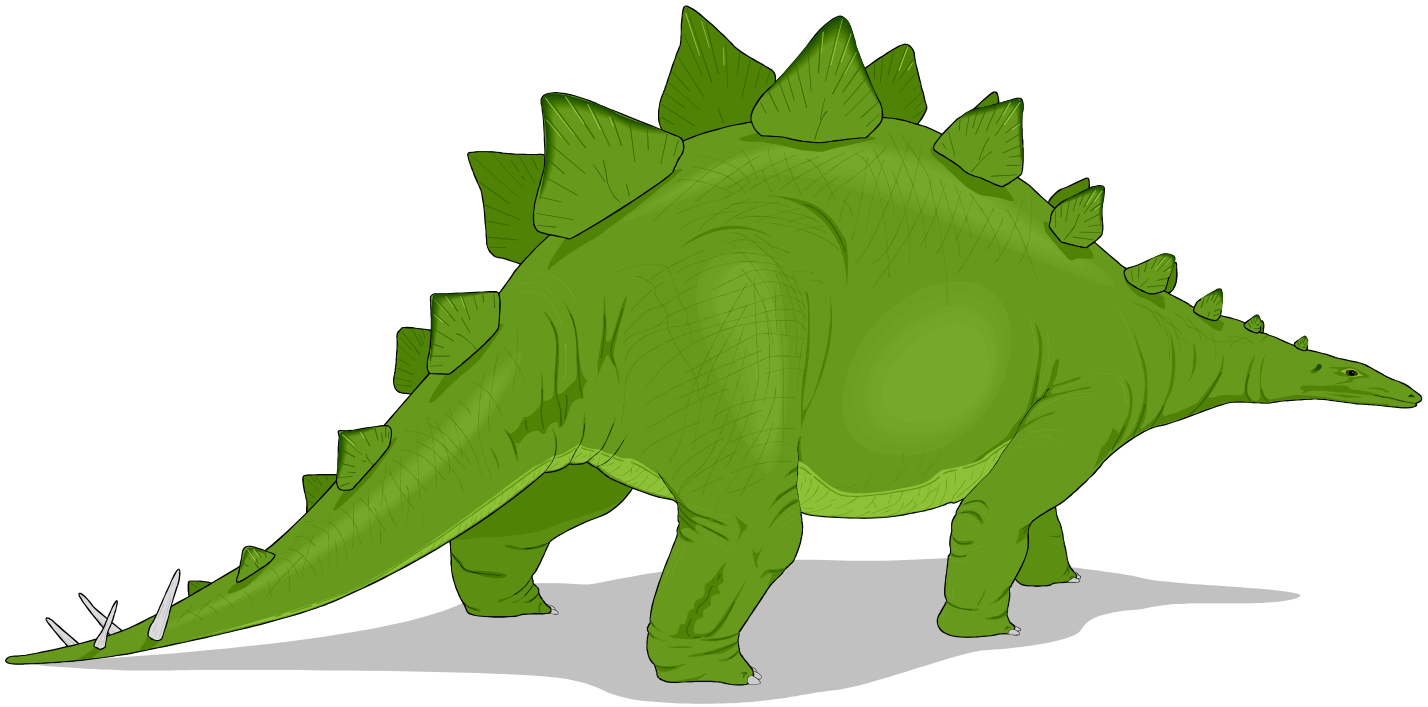
EXODUS®

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

EXODUS®

The Age of Dinosaurs: 1970s



EXODUS®

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**

EXODUS®

Protection

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."**

EXODUS®

Dinosaurs Beget a Eunuch (or two)

EXODUS®

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

EXODUS®

“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.”**

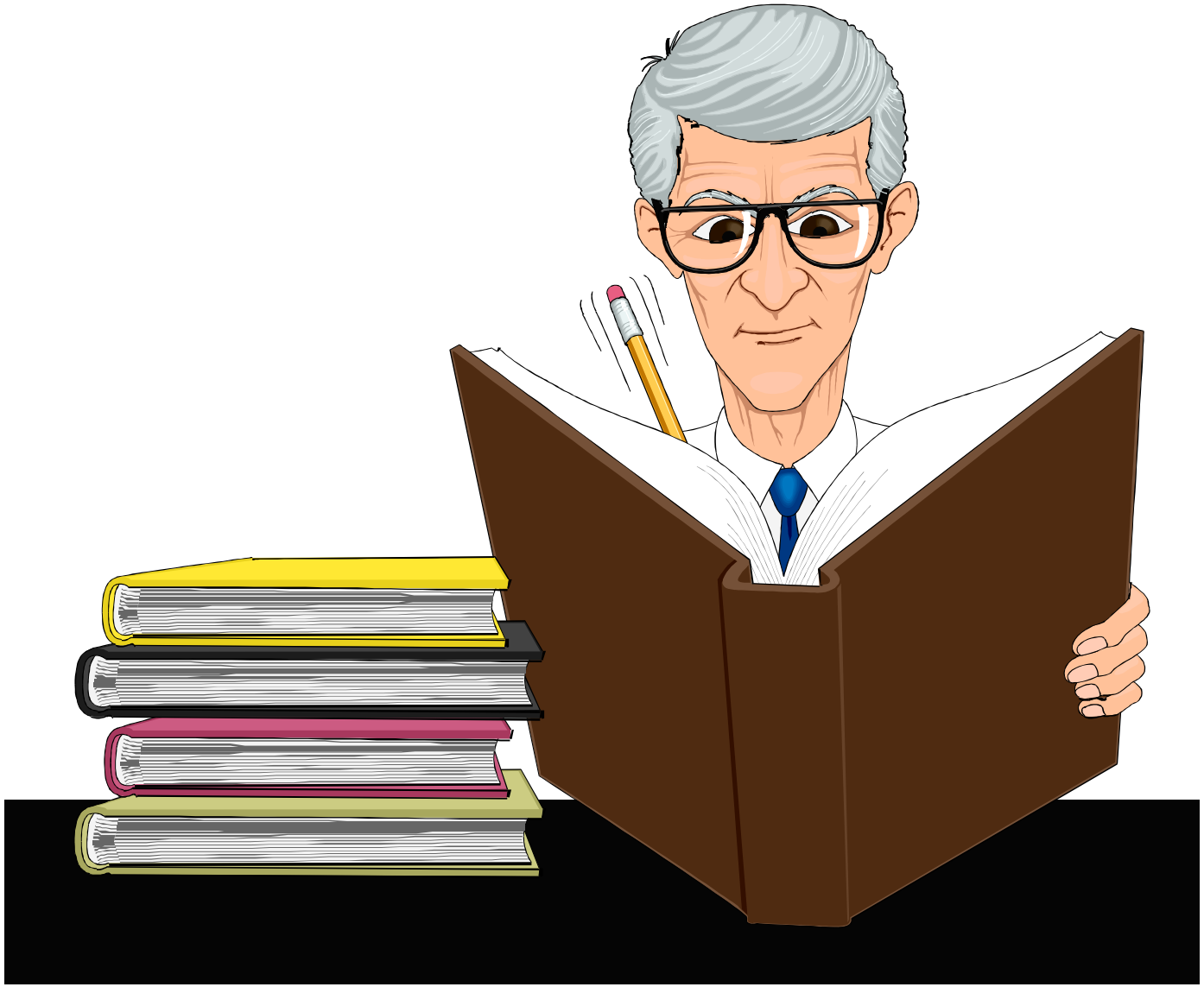
EXODUS®

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.”**

EXODUS®

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

EXODUS®

1975-1985: The Gilded Age

EXODUS®

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**

Composition in Three-Part Harmony

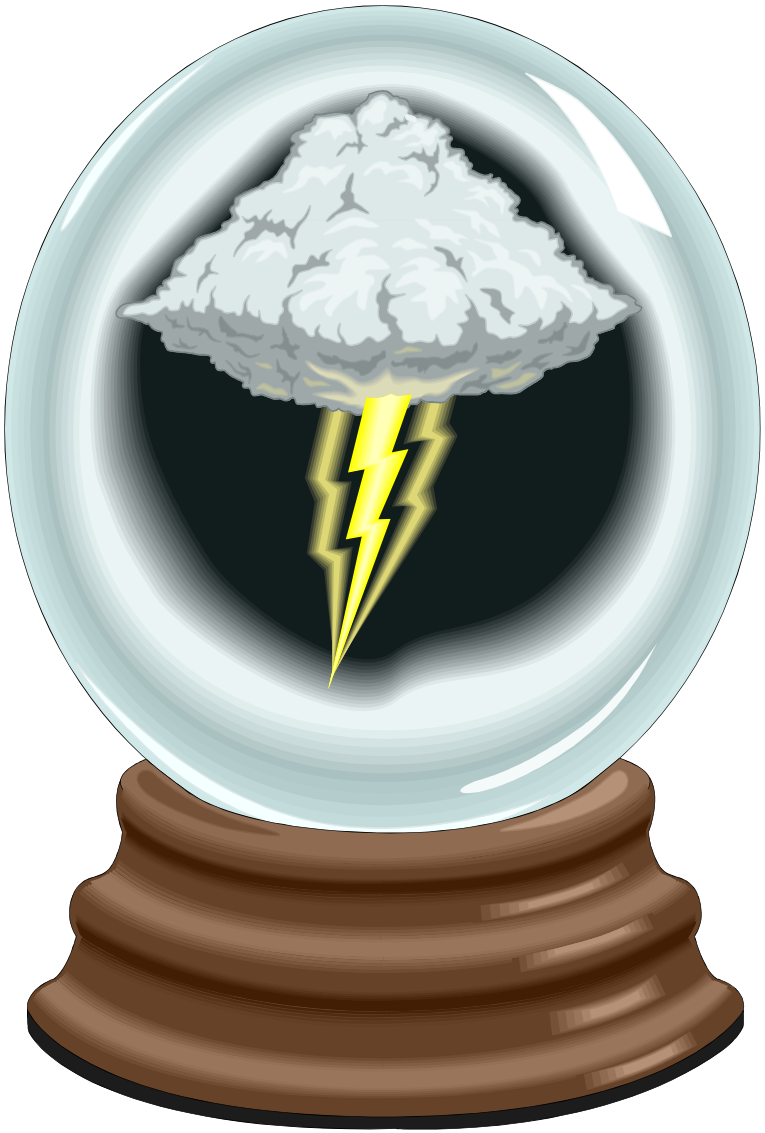
Jim Anderson

- **Shared responsibility:**
  - designers
  - manufacturers
  - government

EXODUS®

Mid-1980s: Storms Brewing

- We Haven't Reached Nirvana Yes
- New Kid on the Block
- The Winds of War



EXODUS®

To Err is Human: [D]ARPA-|INTER-NET Disasters

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

EXODUS®

Disruption Causes

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

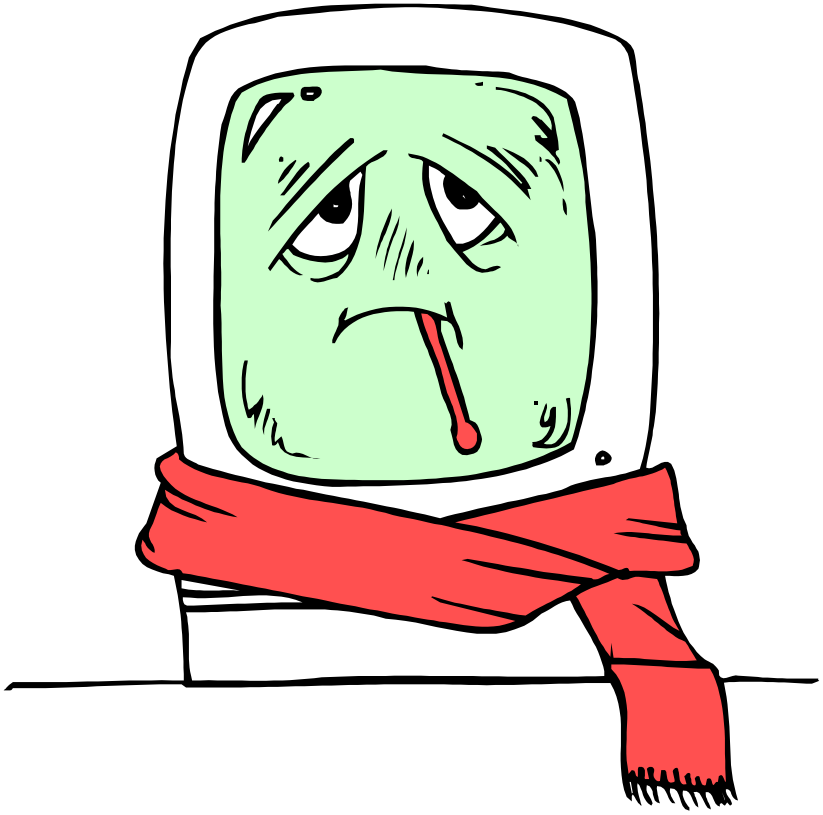
EXODUS®

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

EXODUS®

Middle Ages: The Plague (Viral)



EXODUS®

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

EXODUS®

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

EXODUS®

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”?

EXODUS®

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

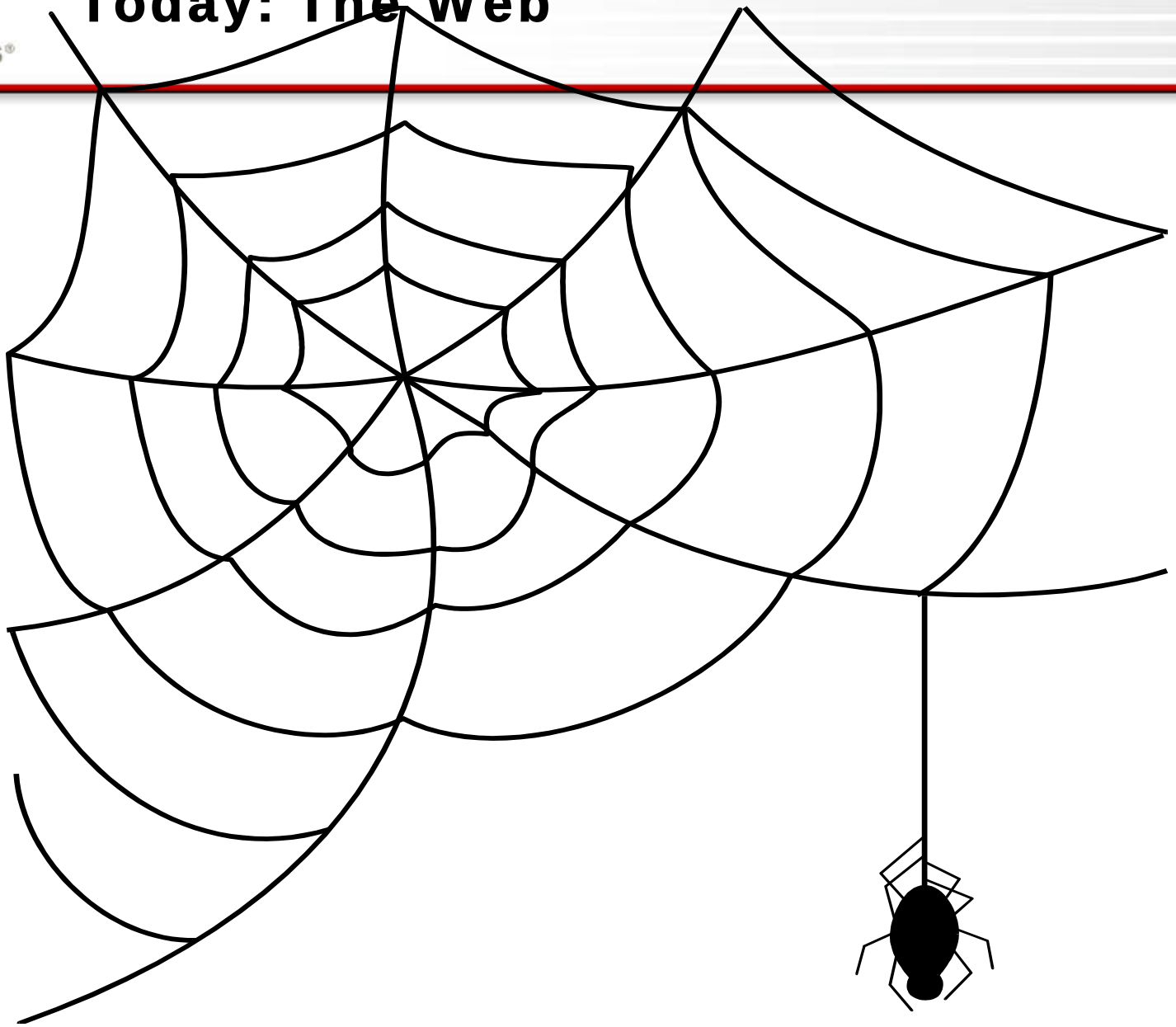
EXODUS®

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**

EXODUS®

Today: The Web



EXODUS®

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”

EXODUS®

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

EXODUS®

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 …

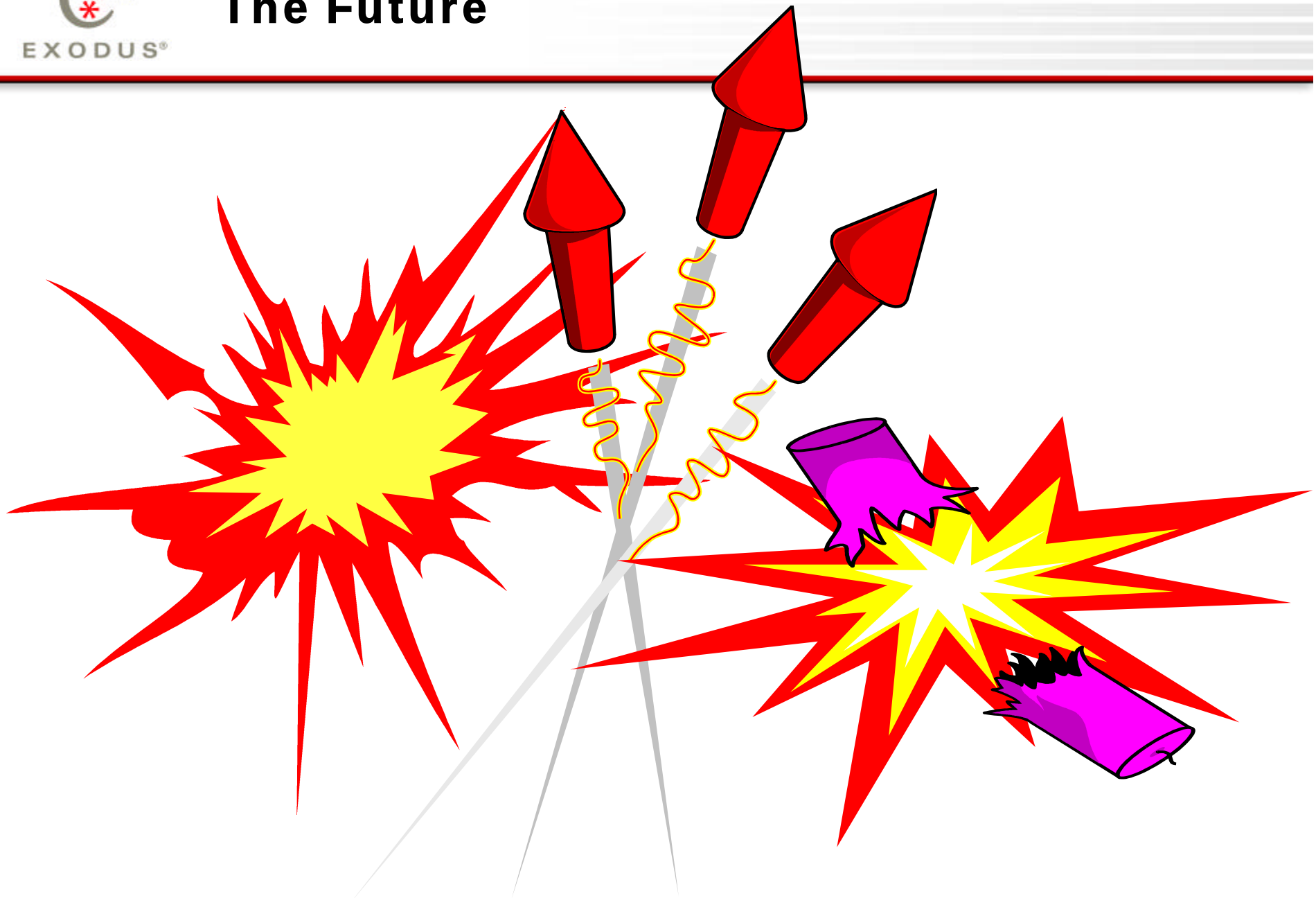
Flight Simulator

Use mouse to navigate: right mouse button for forward, left for reverse

Or list of developers' names

EXODUS®

The Future



EXODUS®

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

EXODUS®

Bibliography (1)

- Anderson, J., "Computer Security Technology Planning Study," *U.S. Air Force Electronic Systems Div. Tech. Rpt. 73-51,* Oct 1972.
- Anderson, J., "Accelerating Computer Security Innovations," *Proc. 1982 IEEE Symp. on Sec. and Privacy.*
- Feiertag, R. and Neumann, P, "The Foundations of Provably Secure Operating Systems (PSOS)," *Proc. 1979 Nat'l Comp. Conf.*
- Gold, B., *et al.*, "A Security Retrofit of VM/370," *Proc. 1979 Nat'l Comp. Conf.*
- Grampp, F. and Morris, R., "Unix Operating System Security," *Bell Systems Tech. Journal,* Aug 1984.

EXODUS®

Bibliography (2)

- Grant, P. and Riche, R., "The Eagle's Own Plume," *U.S. Naval Institute Proceedings*, July 1983
- Lampson, B., "Protection," *Proc. 5th Princeton Conf. on Information Systems, 1971,* **reprinted in** *ACM Operating Systems Review,* Jan 1974.
- Linde, T., "Operating System Penetration," *Proc. 1975 Nat'l Comp. Conf.*
- McCauley, E. and Drongowski, P., "KSOS—The Design of a Secure Operating System," *Proc. 1979 Nat'l Comp. Conf.*
- Morris, R. and Thompson, K., "Password Security: A Case History," *Commun. of the ACM, Nov 1979.*

EXODUS®

Bibliography (3)

- Popek, G, et al., "UCLA Secure Unix," *Proc. 1979 Nat'l Comp. Conf .*
- Saltzer, J. and Schroeder, M, "The Protection of Information in Computer Systems," *Proc. of the IEEE,* Sept 1975.
- Thompson, K., "Reflections on Trusting Trust," *Commun. of the ACM,*
- Ware, W., "Security Controls for Computer Systems," *Rand Corp. Tech. Report R-609-1,* 1970 **(reissued 1979).**
- Welke, S., "A Taxonomy of Integrity Models, Implementations, Mechanisms," *Proc. 1990 Nat'l Comp Security Conf.*