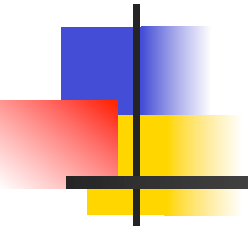


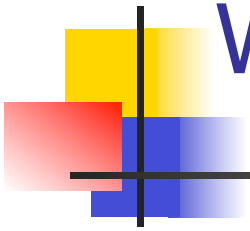
Ten Unsolved Problems in Information Assurance

Carl E. Landwehr

Senior Fellow

Mitretek Systems

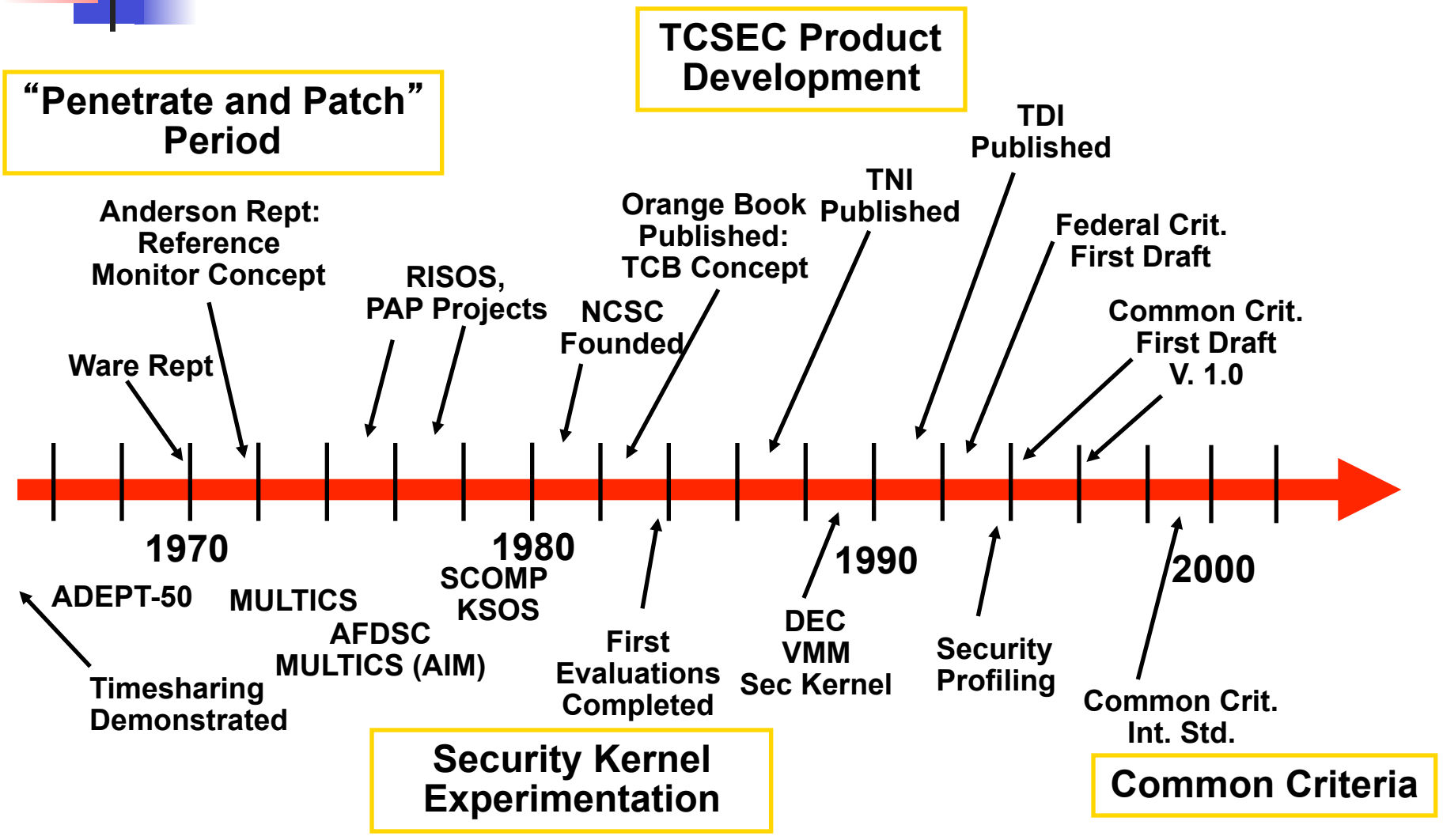
14 Sept. 2001



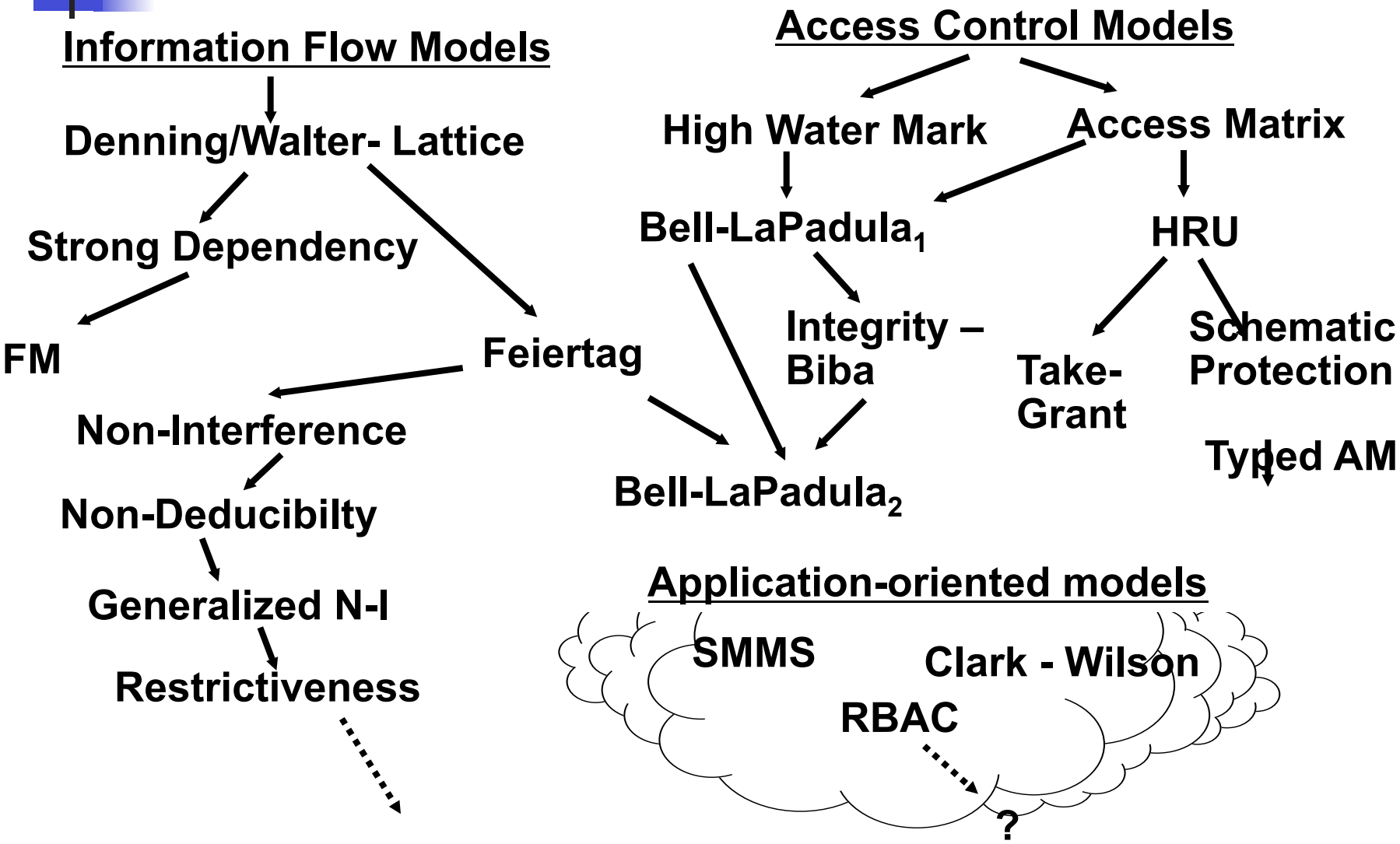
Where did computers come from?

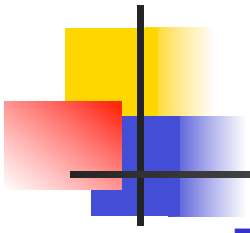
- Much early development motivated by security applications:
  - Breaking codes
  - Developing new weapons
  - Computing ballistic trajectories
- But securing the computers was not an issue: big, physically isolated, unshared

A little history of computer security



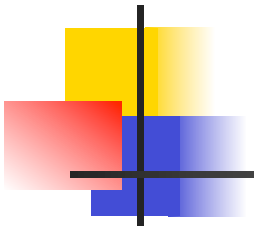
Security model genealogy





What have we learned?

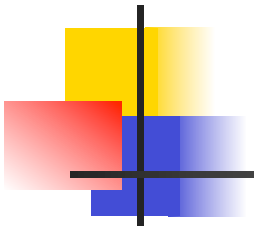
- Many good concepts relating security and operating system architecture
  - Access control matrix
  - Reference monitor model
  - Virtual machine models
  - …
- How hard some things are
  - Security policy and specification
  - Security management
  - Effective user interfaces
  - Covert channel elimination
  - Technology transfer!



What do we still not know?

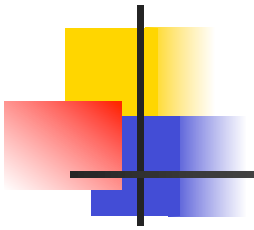
- Following is just a sampling of problems – there are lots more out there
- There are areas of overlap among these problems

How to avoid building security flaws into programs

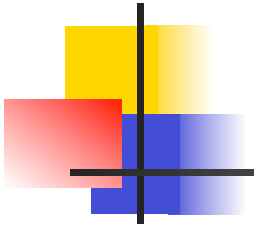


- What's the problem?
  - We keep building programs and despite best efforts, security flaws turn up in them later
  - Policy specification is part of the problem: no policy→ "no flaw"
- What do we know?
  - General software and system engineering techniques
  - Some collections of documented security flaws, organized in various ways, dating back to mid 1970's
  - Some tactics for dealing with specific kinds of flaws
    - Buffer overflows
    - Type violations
    - Code scanning for suspicious constructions
- Where are we going?
  - More, better type enforcement: PCC, IRM
  - More, better code scanning

How to know when a system has been penetrated



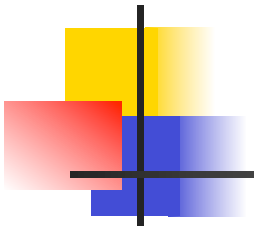
- What's the problem?
  - Intruders can masquerade as legitimate users
  - Malicious code can be inserted early in the life cycle
- What do we know?
  - How to detect attacks that have already been identified
  - How to generate lots of false positives
- Where are we going?
  - Trying to bring more information sources to bear to reduce false positives
  - Some efforts to distinguish normal and abnormal behaviors → need to have a model of what's normal



How to design systems that can tolerate intrusions

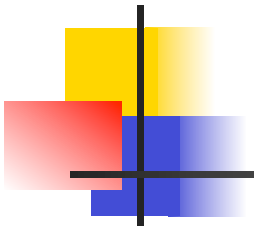
- What's the problem?
  - If we can't avoid flaws or identify intrusions, can we build systems that work anyway?
  - Must deal with malicious faults, which cannot be assumed to be uncorrelated
- What do we know?
  - Fault tolerance techniques: redundancy, masking, diversity, no single points of failure
  - Some results from group membership protocols that deal with intruders
- Where are we going?
  - Efforts underway to explore a variety of techniques and approaches under DARPA OASIS program

How to design systems with manageable security



- What's the problem?
  - Built in security controls are misconfigured, leaving vulnerabilities exposed
  - Most systems are not so much designed as patched together
- What do we know?
  - Humans don't deal well with complex interfaces, and security is rarely "Job 1"
  - Some principles for design of user interfaces
- Where are we going?
  - More automated procedures for installing patches

How to provide reasonable protection of intellectual property

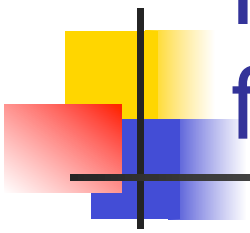


- What's the problem?
  - Owners seem to want "Mission Impossible" tapes: read once bits
  - Mechanisms pretend to be strong but are usually weak: DVD, e-Book, etc.
  - How to protect "cleartext" or provide "fair use"
- What do we know?
  - Encryption techniques, label-based techniques
  - Watermarking, steganography
- Where are we going?
  - Legal enforcement

How to support privacy enforcement technically

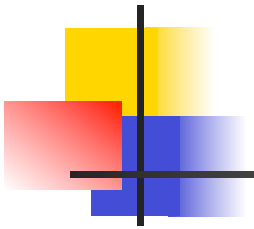
- What's the problem?
  - Privacy generally imposes constraints on information flows and linkages
  - Suitable mechanisms for flexible enforcement
  - Policy also lacking
- What do we know?
  - Conventional auditing techniques
  - Watermarking
- Where are we going?
  - Healthcare information as case in point
  - Policies may specify authorized/unauthorized linkages of private information
  - Technology needed to reveal source of linkage, or provide "one-way" links

How to get trustworthy computations from untrusted platforms

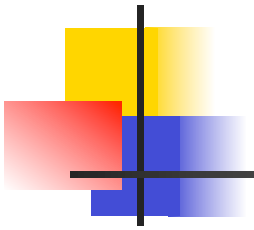


- What's the problem?
  - Most platforms are untrustworthy, yet we need to rely on their computations
  - May want to hide the computation from the platform doing the computing
- What do we know?
  - Some people are willing to donate computing time SETI@home, Distributed.net, Cosm, (Legion?)…
  - Some computations can be checked quickly
- Where are we going?
  - Mobile code techniques are beginning to be explored
  - Some restricted classes of functions identified

How to prevent/withstand denial of service attacks



- What's the problem?
  - Attackers can exhaust system resources without effective penalty
- What do we know?
  - If you're big enough it may not matter: Google
    - But it might: Yahoo!
- Where are we going?
  - Some QoS work may help
  - Policy might help (Charge for bits? Liability?)

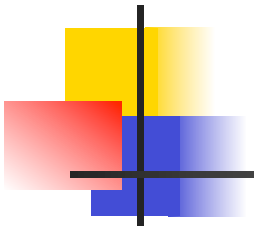


How to quantify security tradeoffs

- What's the problem?
  - Security imposes costs of various sorts
    - Extra mechanism, delay, etc.
    - Precise costs, and quantifiable benefits hard to find
  - Hard to develop rational system designs
- What do we know?
  - Tolerating N Byzantine Failures requires 3N+1 processors and many messages
- Where are we going?
  - Recent results (Castro, Liskov) beginning to show practical Byzantine Fault Tolerance
  - Work on tradeoffs of FRS schemes by Ganger and others at CMU

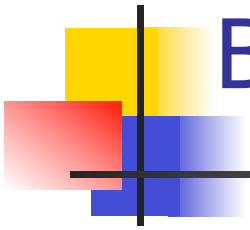
How to reveal / minimize assumptions in security system designs

- What's the problem?
  - Assumptions are often vulnerabilities
- What do we know?
  - Attackers will try to violate some system assumption
    - Kocher: differential power analysis
      - Power consumption can leak key information
  - Assumptions can be hidden in plain sight
    - Needham/Schroeder protocol:
      - assumed no old key compromised
- Where are we going?
  - Incremental progress
  - Analytic methods (e.g., protocol analysis)



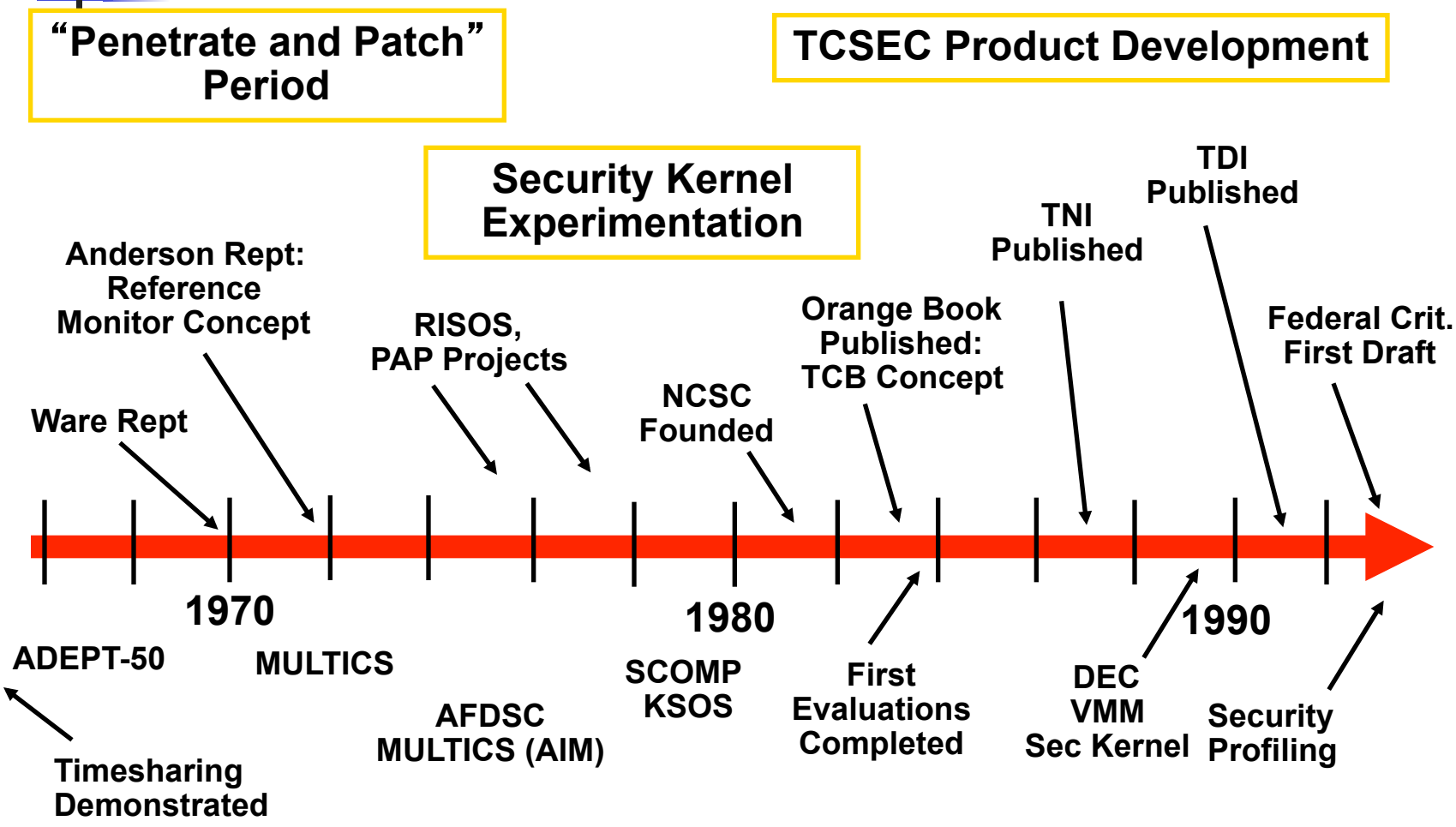
How to build programs/systems and know what they do

- The ultimate problem:
  - Be able to design and build a system and forecast its behavior under specified conditions, including categories of attacks
- Demands not only correctness, composition, but modeling and simulation



Backup

COMPUSEC History



Toward MLS Computing Service

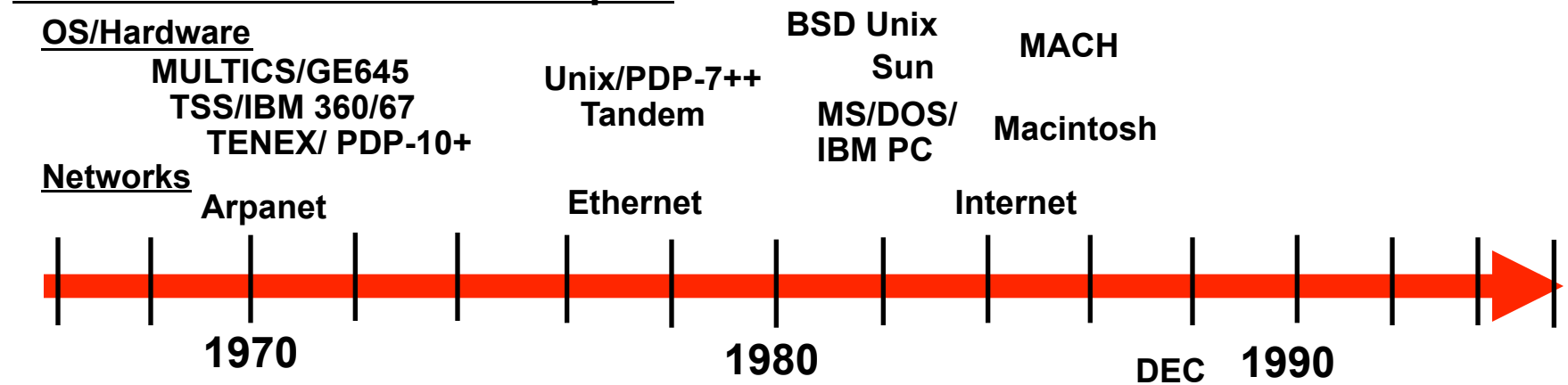
Dominant Architectures

Large Centralized Timesharing

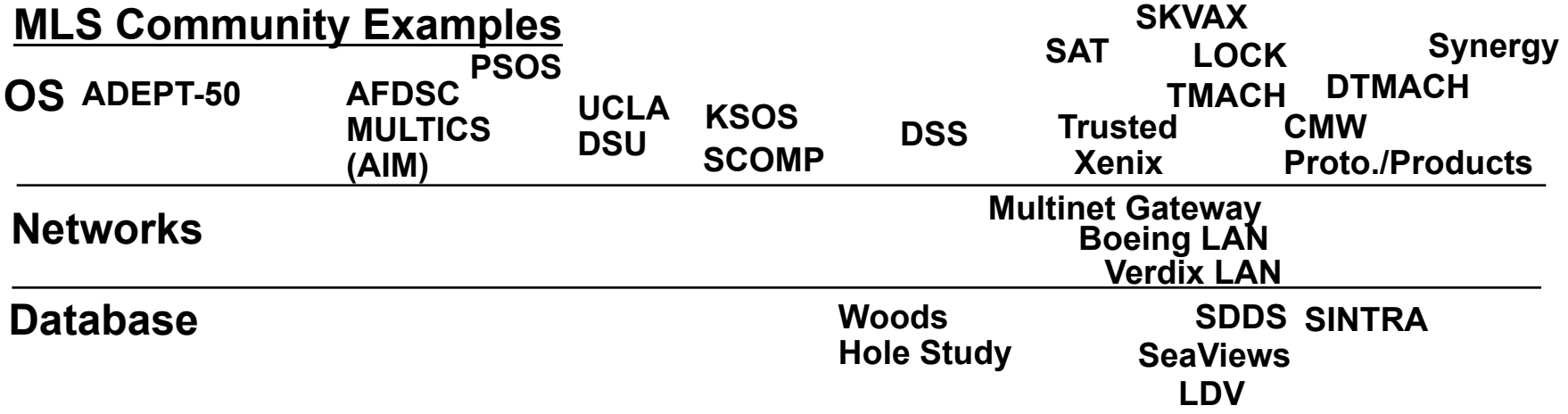
Medium Centralized Timesharing plus Networks

Workstation - based Client - Server, LAN / WAN

Research/Commercial Examples



MLS Community Examples



An Incomplete History

Programming Methodology

Dijkstra T.H.E. 68

Parnas Info. Hiding 72

Dijkstra Disc. of Prog -76

Struct. Pgming - DD&H - 72

Hoare CSP 78 - 85

Gries Sci. of Prog 81

Sufrin Z 84

Raise 85

Knuth Literate Prog. 86

Balzac 91

Program Verification

Floyd 67

Hoare 69

SRI: SPECIAL- HDM 76/ EHDM 83 / PVS 90?

IPV-PARC 73

UT / CLINC: GVE 74 / ROSE 88

SDC/Burroughs/Unisys: Ina-Jo / FDM

ISI, GE, RPI: XIVUS / AFFIRM 76

Larch 80

IP Sharp ORA-Canada: mEVES-mVerdi 83 EVES -Verdi 87

ORA-US: Romulus (Ulyssess)84? Penelope86/CLIO

HOL 85

Automated Theorem Proving

Bledsoe

Boyer-Moore 71

London

SDVS 77

LCF 77

1970

1980

1990

Clark Wilson

HWM-ADEPT-50

Ware Rept

Walter et al

Bell-LaPadula

Feiertag B-L / KSOS

Goguen.-Meseguer Non-Interference

McCullough Restrictiveness

McCullough Hook-up

Gray Probabilistic N-I

Sutherland

McLean System Z

Security Modeling & Theory

Anderson Rept - Ref Monitor

Denning Lattice

Towards Secure DBMS

Research

Codd

Hinke-Schaefer

Woods Hole Study
- integrity lock
- replication
- view-based sec.

Integ.Lock Proto.

SeaViews

LDV

SD-DBMS

SWORD

SINTRA

MUSET

Commercial

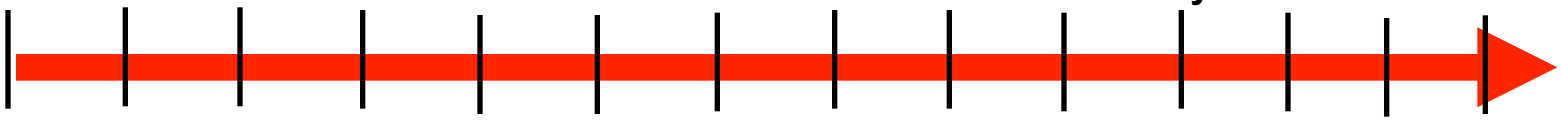
TRUDATA/SQLSentry

T-Rubix

T-Oracle 7

Informix OLS

Sybase S-SQL



1970

1980

1990

Ware Rept

Anderson Rept: Ref. Monitor

NCSC

TCSEC: TCB Concept

IFIP WG11.3

TNI

TDI

First TDI Eval. Done

Dominant Architectures

Large Centralized Timesharing

Medium Centralized Timesharing plus Networks

Workstation - based Client - Server, LAN / WAN

Network Technology

Arpanet

Ethernet

Internet

World Wide Web

Information Assurance Definition

Information Assurance:

- Information operations (IO) that protect and defend information and information systems (IS) by ensuring their availability, integrity, authentication, confidentiality, and nonrepudiation. This includes providing for restoration of information systems by incorporating protection, detection, and reaction capabilities. — *National Information Systems Security (INFOSEC) Glossary, NSTISSI No. 4009, January 1999*