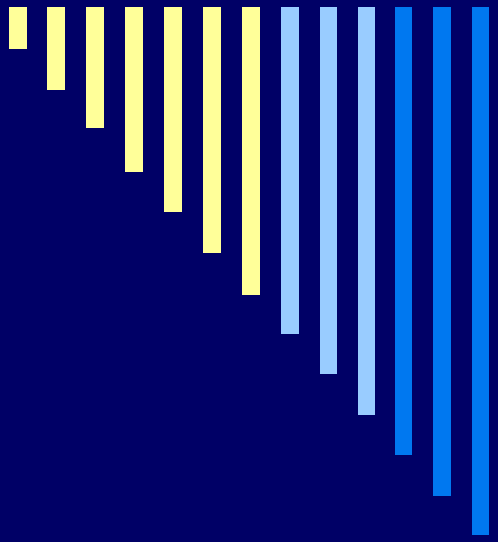


Who Are The Enemies? What Can They Do?

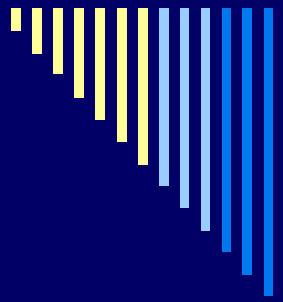
Internet Software Security Issues in the Software Development Process

Dr Charles P Pfleeger

Pfleeger Consulting Group

chuck@pfleeger.com

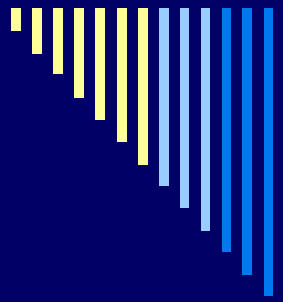
Overview



Overview

- Security Process
- Many Kinds of Threat Agents
- Countering Threats
- Software Process
- Improving Security in Software

What Developers Should Know about Security

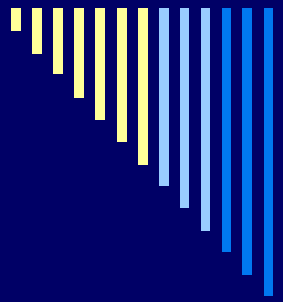


What Developers Should Know about Security

- Security is not automatic
- People and situations are hostile
- Security can be tricky
- No magic technique or checklist leads to security
- *BUT* security *can* be compatible with good development processes

Security Considerations

- Goals
- Threats and Vulnerabilities
- Method + Opportunity + Motive
- Controls



What Security Is

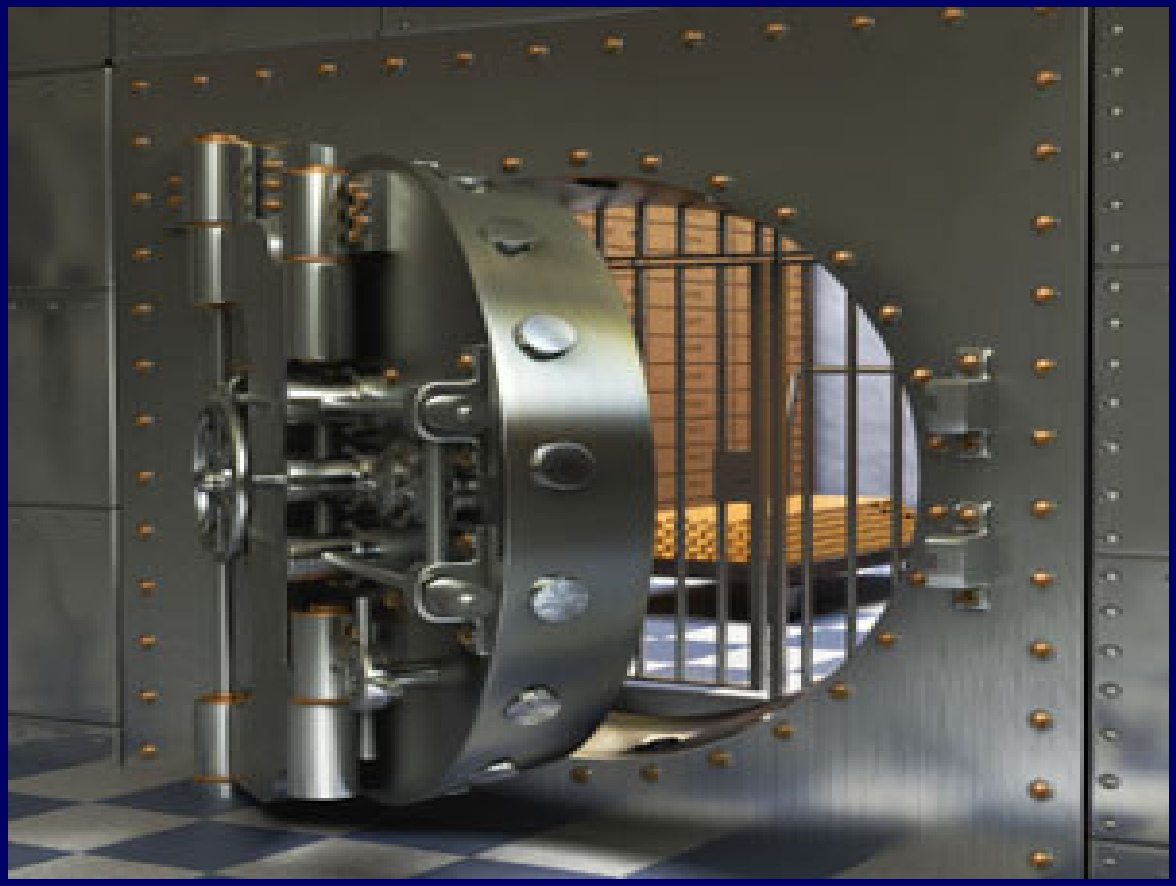
- Confidentiality
  - information available for reading only when authorized
- Integrity
  - information available for modification only when authorized
- Availability
  - information available for use when authorized

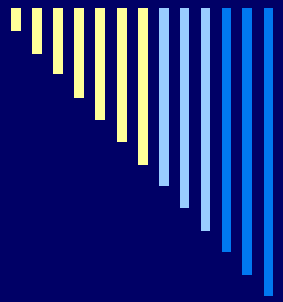
Threat Sources

Threat Sources

- Natural
  - Fire, flood, hardware failure
- Human
  - Unintentional
    - Errors (accidental file deletion)
    - Omissions (missing data)
  - Intentional
    - Outsiders
    - Insiders

Method + Opportunity + Motive

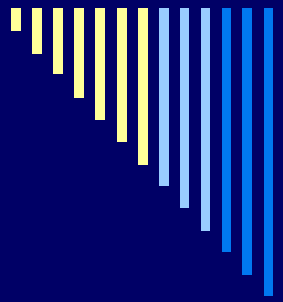




Method + Opportunity + Motive

- Method
  - tools, techniques, knowledge
- Opportunity
  - access, ability
  - work factor: difficulty, time
- Motive
  - reason (if any)

Types of Harm



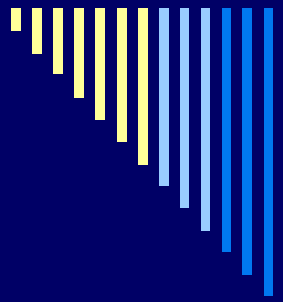
Types of Harm

- Confidentiality
  - Reveal private data
- Integrity
  - Damage or delete data
- Availability
  - Denial of access or service

Controls

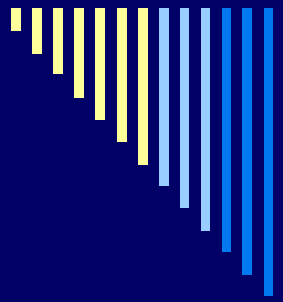


- Physical
  - Guards, fences, sprinklers
- Administrative — Logical
  - Policies, laws
- Technical
  - Devices (login tokens)
  - Software (intrusion monitors)
  - Combinations (network monitoring)



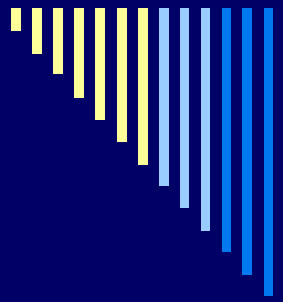
Why Security is Hard (1)

- The defender must counter all threats; an attacker needs only one vulnerability
- Developers can introduce/allow flaws without knowing it
- Testing may not demonstrate security
- Harm can come from intentional or unintentional causes



Why Security is Hard (2)

- Is 1000 the same as 1,000?
- Is invisible text acceptable?
- Does efficiency–or inefficiency–matter?
- Does order matter?



Why Security is Hard (3)

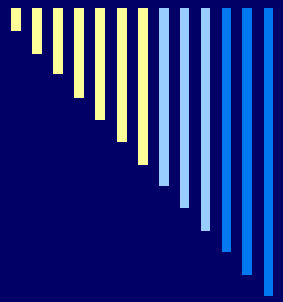
- How can you demonstrate protection?
- How can you justify spending for security?
- Security is not just a set of features

Security Conclusions

Security Conclusions

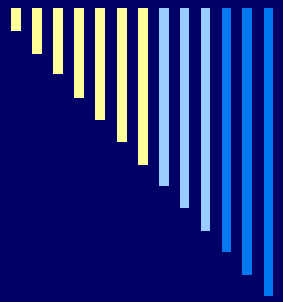
- No one person–or group–alone can be responsible for security
- Solid security is not added at the end
- Security is hard

Security Design Concepts



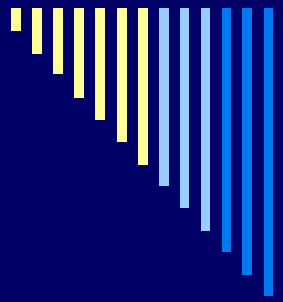
Security Design Concepts

- *Common sense*
- Saltzer and Schroeder documented and published in 1975
- Others (Viega & McGraw, Howard & LeBlanc, NIST, OWASP, more) build on Saltzer and Schroeder



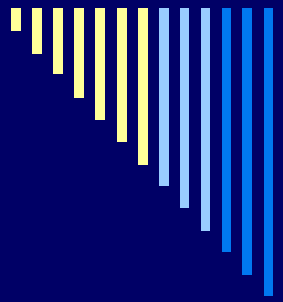
Saltzer and Schroeder

- Economy of mechanism
- Fail safe defaults
- Complete mediation
- Open design
- Separation of privilege
- Least privilege, permission-based
- Least common mechanism
- Ease of use



Viega and McGraw

- Secure the weakest link
- Keep it simple
- Promote privacy
- Remember that hiding secrets is hard
- Be reluctant to trust
- Use community resources



Howard and LeBlanc

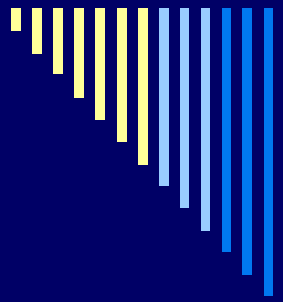
- Minimize attack surface
- Assume external systems are insecure
- Security features are not the same as secure features
- Don’t mix code and data
- Plan for failure
- Learn from mistakes
- Backward compatibility is tricky
- Fix security issues correctly

Security Control Philosophy

Security Control Philosophy

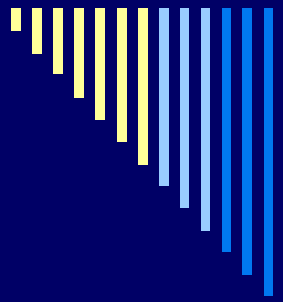
- Secure perimeter
- Control of users and environment (devices, software)
- Defense in depth
- “Perfect” (secure) software

Web Characteristics



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”



Software Development vs. the Web

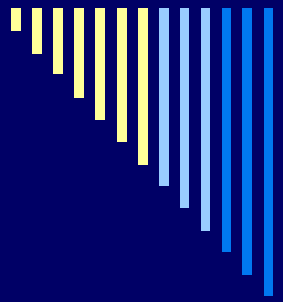
- Orderly, managed process
- Well-defined system
- Well-defined requirements
- Integrated testing
- Continuous improvement
- Controllable => securable

- Unmanaged
- No perimeter; constant change
- Programs used-reused for any situation
- Testing varies
- Little feedback
- Uncontrolled => unsecured

Some Sources of Computer Incidents

- Errors
- Insiders
  - Nonmalicious and malicious
- Hackers
- Organizations
- Terrorists

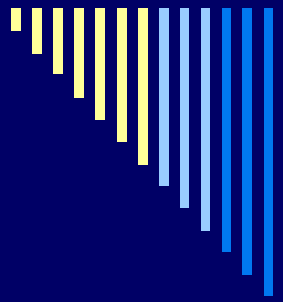
Errors



Errors

- Humans are human
- Errors are unintentional
  - Pressure, distraction, confusion
- Random

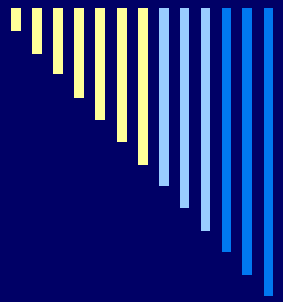
Insiders



Insiders

- Insiders account for much harm
  - Many insiders, many actions
  - Much power
- Most insider harm is unintentional
- Intentional harm from anger, pay-off, values conflict

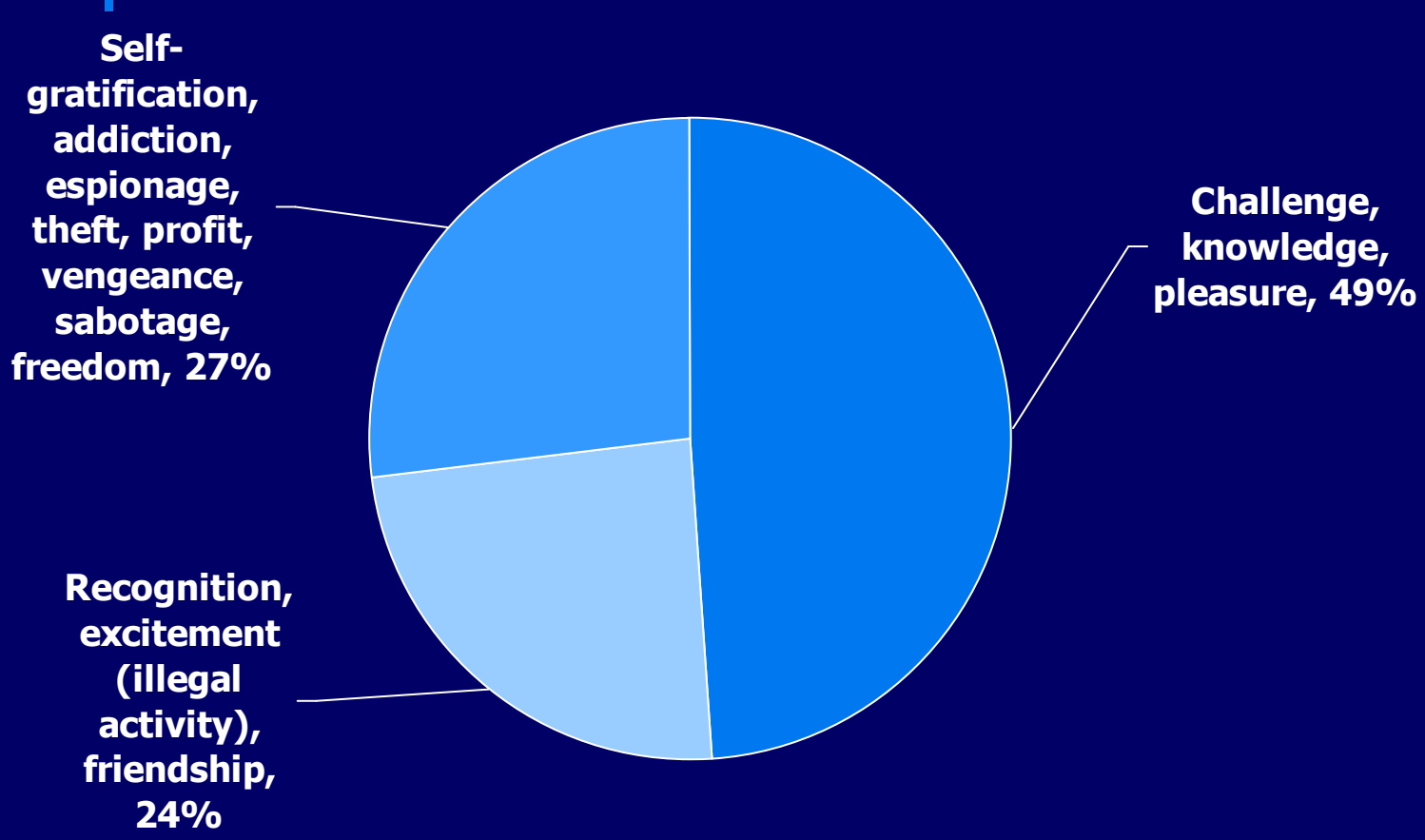
Hackers



Hackers

- Individuals, loose federations
- Motivation
  - Personal
- Objective
  - Experience, fame
- Impact
  - Modest
  - Sometimes accidental side effects

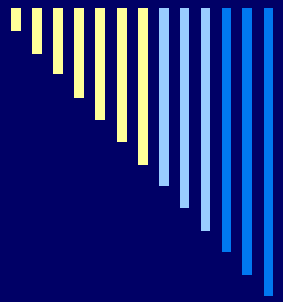
Hackers' Motivations



Denning 1999, citing Chantler

Disturbing Hacker Trends

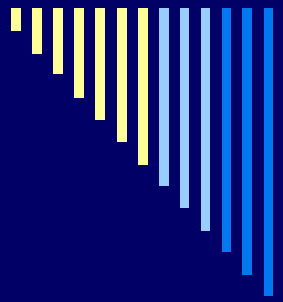
- ~0-day exploits
- Bagle, Netsky, Zafi worm writers join forces
- Enormous botnets for rent
- Conficker very sophisticated; evolving



Organized Crime

Organized bureaucracies (computer work is only one activity)

- Motivation
  - Financial
- Objective
  - Fraud, extortion
- Impact
  - e.g., identity theft, credit



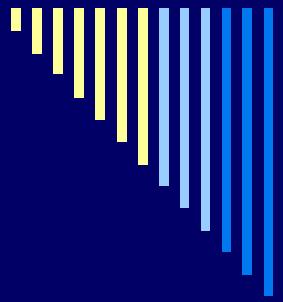
Intelligence Services

- Motivation
  - Data: political, economic, military
    - needs of military during engagement
    - about potentially hostile nations
    - transnational threats (drugs, WMD)
- Objective
  - Mostly to learn
  - Sometimes to influence world
  - Occasionally to act

Terrorists

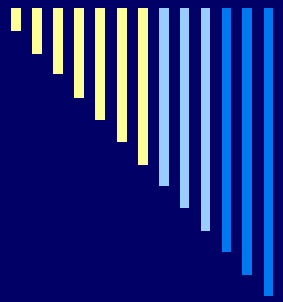
Terrorists

- Motivation
  - Ideological, political
- Objective
  - Intelligence gathering
  - Psychological
    - fear, panic, persuasion, misinformation
  - “Mass annoyance” to “mass disruption”



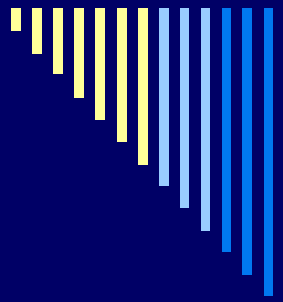
Cyber Warfare Activities

- Disruption
- Sabotage
- Denial of service
- Misinformation
- Deception



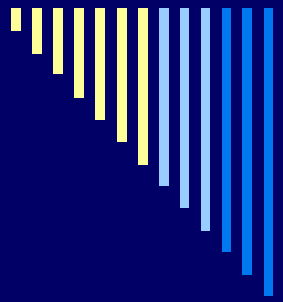
Uses of Computers in Terrorism

- Disseminate information
  - web pages, desktop publishing
- Communicate with each other
- Gather, hold, analyze information
  - data mining, web browsing
- Target computers (uncommon)
  - unauthorized access (read, write, delete)
  - denial of service



Why Cyber-Advertising (and Spam) Is Attractive

- Remote operation, anonymous
- Massive reach, low cost
- Agile, easy to change, tune
- Message tuned to audience (coupled with data mining)
- Novel campaign garners media attention

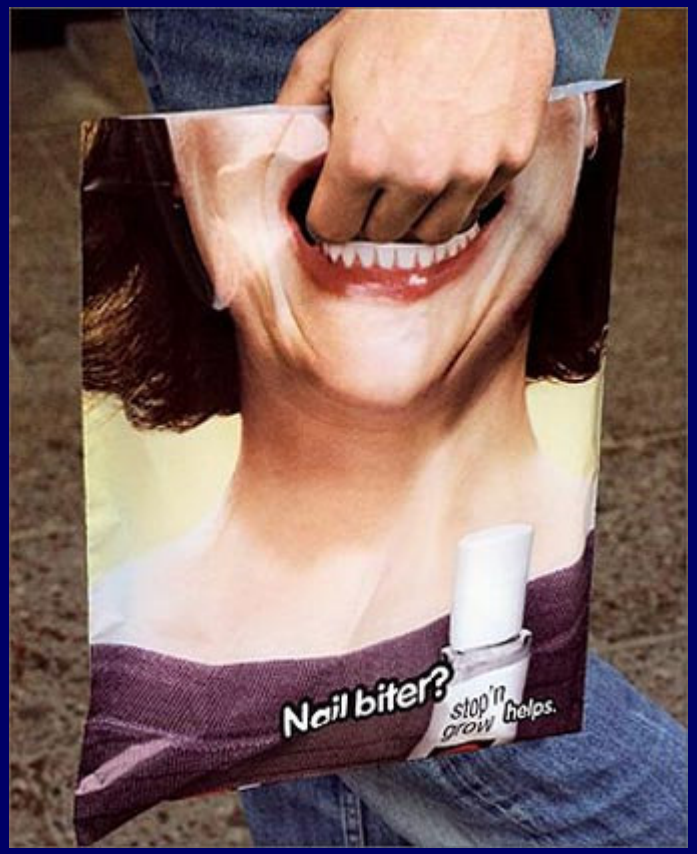


Why Cyber-Terrorism Is Also Attractive

- Remote operation, anonymous
- Massive reach, low cost
- Agile, easy to change, tune
- Message tuned to audience (coupled with data mining)
- Novel campaign garners media attention

The Medium Is the Message

- The Internet may be worth more as a communications medium than a target
- Individual nodes—or whole subdomains—can still be targets



Lessons (To Be) Learned

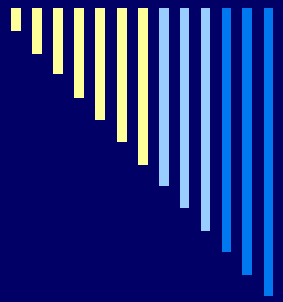
- Evidence of vulnerability is visible to others
- Attackers work together
- Talk (articles, blogs) may alert them to weaknesses

Recent Events

- 2008 attack on firewall of China
  - Apparently political
- 2006 top level domain server DOS attack
  - 2002 disabled 8/13 root name servers
    - Apparently a simple ICMP packet flood
- 2007 attack on Estonia web
  - Motive unknown; testing?
- 2003 power blackout in NE Canada, US
  - Apparently tree limbs, cascading errors
- 2000 attack on wastewater treatment control center
  - Apparently a disgruntled former employee

Interdependent Risk

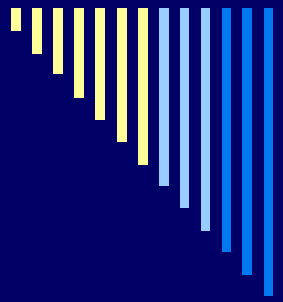
- One company, two divisions, both at risk of catastrophic (company) failure
  - If Division 1 invests in risk-reduction, it remains at risk unless Division 2 does also
  - In fact, Division 1's investment is wasteful if Division 2 does not invest
- What is incentive for an interdependent multi-corporation industry (power, finance, telecommunications) to invest in security?



Playing Security Analyst: A Smart Terrorist Would …

- Plan, research, experiment
- Plant attacks today for use tomorrow
- Diversify the approach
- Implement multiple concurrent attacks
- Use different approaches to “come from all directions”

Software Process



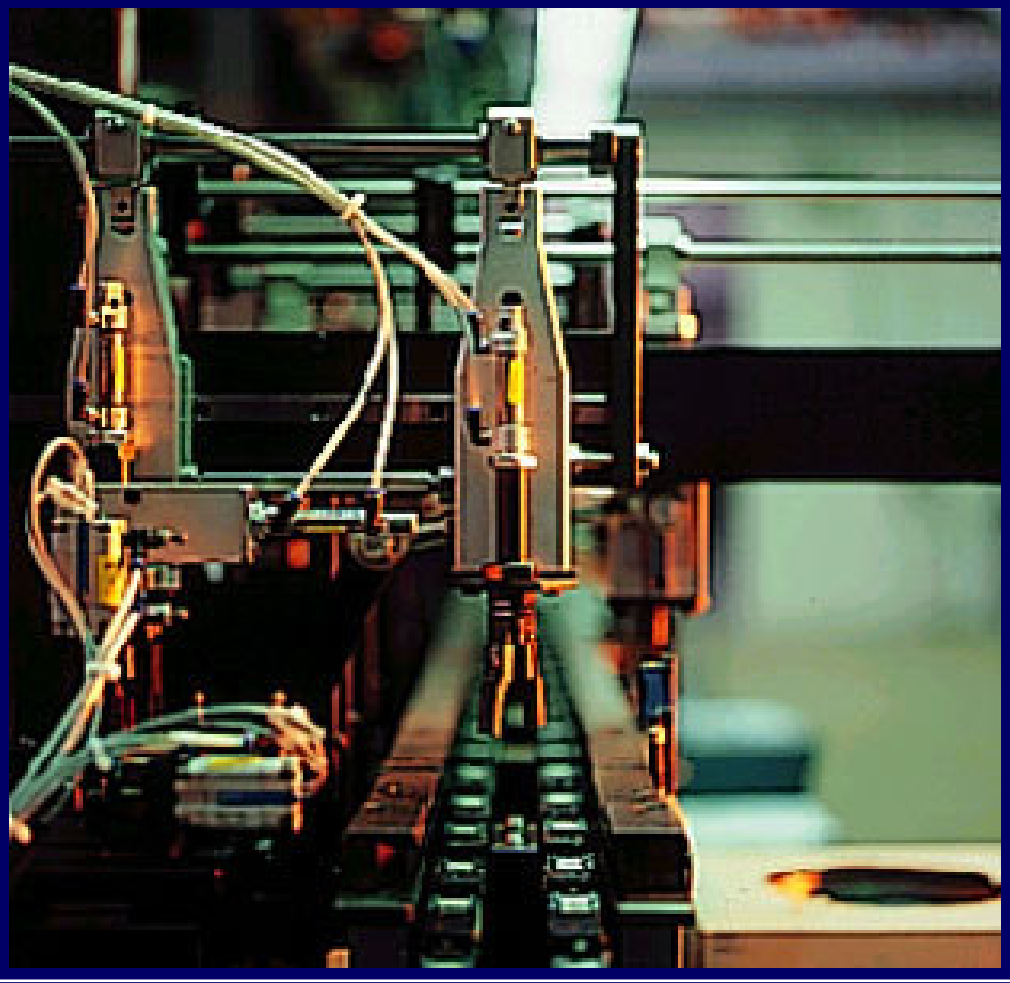
Software Process Models

- MPS.BR
- CMMI
- ISO 9000
- Others

Common Aspects

- Establish an organizational policy
- Plan and define a process
- Monitor and control the process
- Evaluate adherence
- Collect improvement information
- Correct root causes of problems
- Ensure continuous improvement

Basic Process Areas



Basic Process Areas

- Requirements development
- Project planning
- Project monitoring and control
- Measurement and analysis
- Process and product quality assurance
- Configuration management

Measurement

Measurement

- What is important to measure
- How to measure it
- How precise the measurement must be
- How to interpret the measurement
- How to use measurement to improve

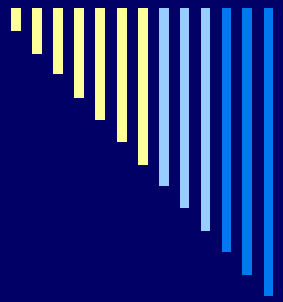
System Development Controls

- System development lifecycle
- Requirements determination
- Protection specifications development
- Design review
- System test review
- Certification and accreditation
- Service level agreement

Conclusions: What to Do?

What to Do: Process

- Requirements development
- Project planning
- Project monitoring and control
- Measurement and analysis
- Process and product quality assurance
- Configuration management
- Include security requirements
- Security overseer
- Threat–control analysis
- Security control coverage
- Security testing at all stages
- Security review prior to stage release



What to Do: Design

- Include security from the start
- Make security trade-off decisions visible and based on balance of competing factors
- Ensure a security expert is integral part of design team
- Ensure that project team members are all sensitive to security

What to Do: Measurement

- What is important to measure
- How to measure it
- How precise the measurement must be
- How to interpret the measurement