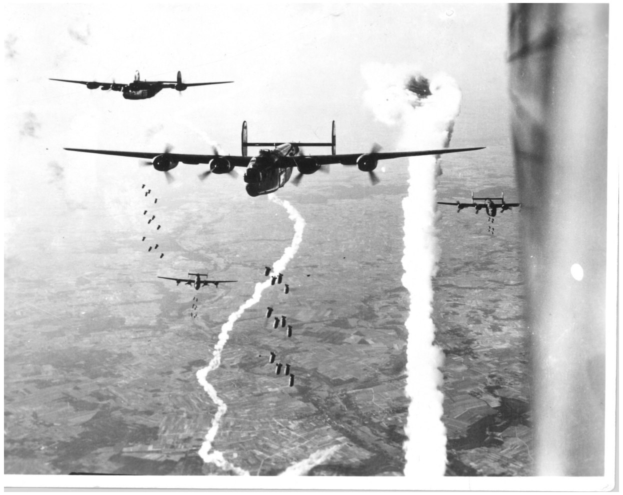
Cyber Security Briefing

How to Fight and Win the New War

and

Why What We Are Doing Now Won't Work



Dr. Steve G. Belovich
CEO, IQware
(www.iqware.us, www.iqmtm.com)

September 12, 2011

IQ! ware TM

"I must study politics and war, that our sons may have liberty to study mathematics and philosophy" John Adams, 12 May 1780

Acknowledgments

Thanks to The Heritage Foundation

for hosting this presentation

Thanks to Ted Schelenski

(CFO The Heritage Foundation) for making it happen

Thanks to Brett Schelenski

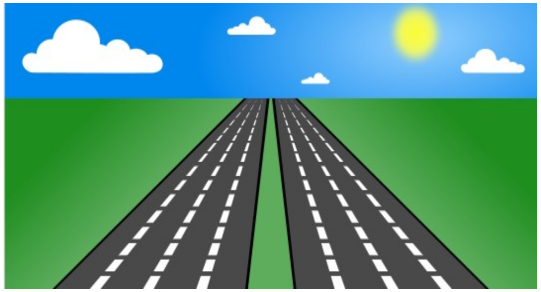
(CEO TriPod Enterprises) for suggesting it

Thanks to Lt. Col. Tom S. Belovich

(USAF 8-th Air Force, 446th BG B-24 Bombardier)

For studying WWII so I could study engineering

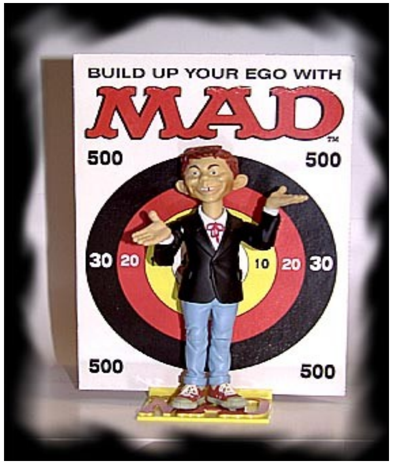
The Roadmap



- Part 1 - Background
  - Cyberwar “SitRep” (6)
  - Security Technology & Hacking Techniques (10)
  - Security History & Architecture (3)
  - Software Structure (2)
- Part 2 – What We Know Now
  - Security Research Results (3)
  - IT Classification Systems (2)
  - Observations & Achievable Objectives (4)
  - The “Dirty Little Secret” (2)
- Part 3 – Policy
  - General Guidelines (2)
  - Specific items (1)

Background - SitRep 1

Vulnerable Systems

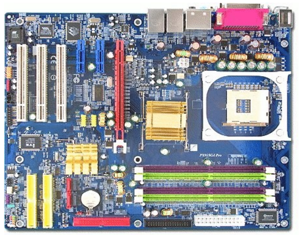


What, Me Worry?

- Cyber war can involve any IT system:
  - Government control and information management systems (e.g., CIA, NSA, DoD)
  - Control systems (e.g., nuclear facilities, power generation plants, power grid management)
  - Factory automation systems (e.g., manufacturing plants, chemical plants, weapons plants)
  - Critical infrastructure systems (e.g., water treatment, wastewater, etc.)
  - Business systems (e.g., health care, pharmacy, pharmaceutical, POS, financial, banking, stock trading)

Background - SitRep 2

Cyber Warfare Objectives



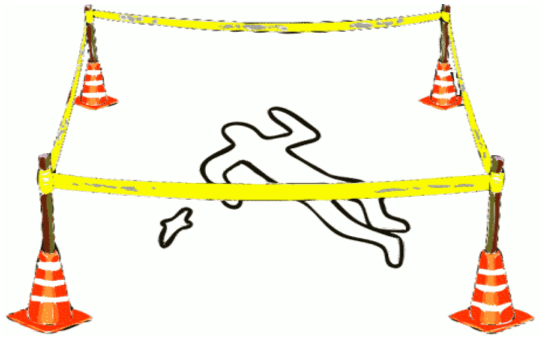
- Offensive Purpose
  - Denial of availability (can't get to the system)
  - Steal or corrupt data (ID theft, technical secrets, etc.)
  - Disruption of operational capability (critical functions can't be performed)
  - Cause irreparable damage to IT systems
  - Cause unwanted & undetected operations (e.g., Stuxnet)

- Defensive purpose
  - Maintain availability (easiest to do)
  - Protect data (medium difficulty)
  - Maintain operational capability (hard to do)
  - Prevent damage (hard to do)
  - Prevent unwanted, undetected operations (hardest to do)

Background - SitRep 3

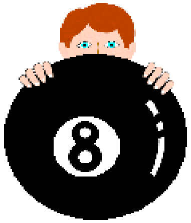
Cyber Warfare Battlefield - 1

We Put Out Fires



We Do Forensic Analysis

We Need to Think!



- We're getting beat!
  - Successful “hacks” happen daily
  - State-sponsored hacking is increasing dramatically
  - Many key systems have already been compromised
  - Current defense techniques are not working
  - Requirements for connectivity increase risks and dangers
  - Underlying technologies are flawed - not ready for “prime time”
  - Mission-critical functions have been deployed on weak and vulnerable platforms
  - Technological “fads” and “coolness” are still trumping operational effectiveness (e.g., “cloud computing”)
  - Current “thinking” does not work
  - It's an “IQ War” where “Smart” always beats “Dumb”

WARNING: It Will Get Much Worse!

Background - SitRep 4

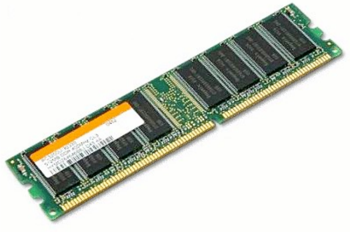
Cyber Warfare Battlefield - 2

- Personnel Issues
  - Risks are not adequately understood nor appreciated
  - IT personnel are not sufficiently knowledgeable
  - Existing policies are procedures are not followed
  - Existing policies and procedures are insufficient
- Technical Issues
  - Emphasis on connectivity and ease-of-use trump security
  - Net-centric approach allows global access – increased risk
  - IT systems grow in “layers” - weak spot is the “lower layers”
  - Existing standards have built-in flaws
  - IT systems are very poorly architected & engineered
- Economic Issues
  - Consumer market economics is preventing real innovation and driving purchasing decisions – large installed base prohibits rapid change
  - Lower layers of IT systems costly to alter / change
- Management Issues
  - Existing “certs” are nearly useless
  - Mission-critical functions are deployed on non-secure platforms
  - No “credit” given for future planning – no budget either!

Background - SitRep 5

Cyber Wareface Battlefield - 3

- Inadequate control over key layers of the IT structure
  - Hardware, RAM, secondary storage, network gear
  - Compilers & Run-time systems
  - Software development “tools”
  - Applications
- Outsourcing – are designs being altered by design tools and/or the manufacturing process?
- Off-campus chip manufacturing risk - who is building your chips and what are their real capabilities?
- Off-campus coding risk - who is really doing the work and what's really in there?
- Off-campus design and manufacturing tool risk (what is it really generating?)



Who is this guy?

WARNING: In the IT world a **“Tool” = “Building Material”**

Background - SitRep 6

Cyber Security Fallacies

- I can make my network 100% bullet-proof
  - TCP/IP was designed for ease of connectivity not security
  - the TCP/IP protocol permits challenge and response without authentication – a built-in flaw
- I can have browser access and be 100% secure
  - Browsers deployed on non-secure platforms can be hacked – it happens every day.
  - Google had their “single sign-on” code base stolen that way in 2010
- Firewalls / intrusion detection will 100% protect me
  - They're good to have but the architecture of the common protocols and platforms prevents 100% protection
- Anti-virus software will 100% protect my desktops
  - They can only detect what they know about and cannot “see” anything else
  - They also make your system run very slowly
- I have been “NIST Certified” So I am OK
  - “Certs” come from limited testing of functional subsets, not from fundamental engineering design. Little basis for assuming operational or performance confidence.

"It's not what you don't know that gets you into trouble. It's what you know for sure that just ain't so." Samuel Clemens (Mark Twain)

Security Technology – 1

Firewalls & Anti-Virus Software

What They Can Do

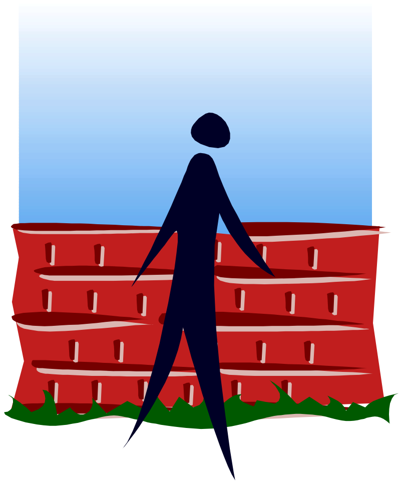
- Firewalls help protect desktops against network transmitted viruses.
- Firewalls can reduce network traffic by keeping unwanted / unknown packets out
- Anti-virus software scans and finds instances of known viruses.

What They Cannot Do

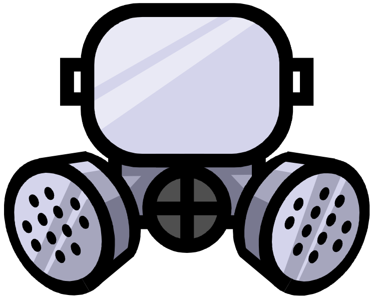
- **Prevent damage from new viral strains.**
- **Clean up damage from cyber attacks.**
- **Prevent critical data loss from cyber-thieves.**
- **Prevent damage from operator error(s)**
- **Ensure proper operation of application software**
- **Make the IT system tamper-proof.**
- **Provide 100% security.**

Security Technology -2

Firewalls - Overview



- Described by Cheswick and Bellovin in their book.
- Two Main Types
  - Application proxies - more secure, very restrictive on performance, used mostly for out-bound traffic.
  - Packet Filtering gateways - less secure, allows higher network throughput, used mostly for in-bound traffic.
- Most have built-in security flaws due to design errors
- Many breaches due to misconfiguration and/or mis-administration (e.g., default passwords, weak ACLs)



Firewall Hacking – 1

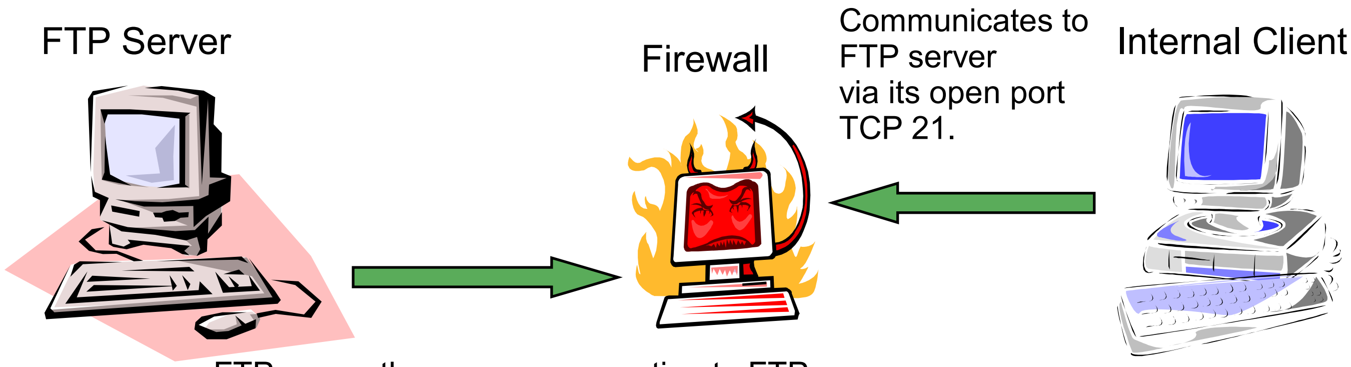
- Identification
  - Port scanning - done randomly to avoid IDS response.
  - Banner grabbing - helps determine type and version
  - Route tracing - See the responses and deduce location of firewall.
- Scanning Through Firewalls
  - Raw Packet Transmission - “Hping” tool (see www.kyuzz.org/antirez/ hping.html) sends TCP packets to ports and analyzes responses.
  - Firewalking - a popular tool at www.packetfactory.net/projects/firewalk that builds packets with IP TTL that expires one hop past firewall. If firewall allows it, it will pass, expire and generate a “ICMP TTL expired in transit” error message. Otherwise, packet will be dropped and generate a null response or ICMP type 13 admin prohibited filter packet.

TTL: Time-To-Live, field within IP packet. Each router that handles packet decrements this field.

Firewall Hacking – 2

- Source Port Scanning - works on firewalls that do not keep state information.

Normal Operation
Packet filtering firewall must keep open all connections from source port TCP 20 to high-numbered ports on its internal network to allow FTP data channel to pass through firewall (TCP 53 zone transfers are also usable for this attack)



FTP Server

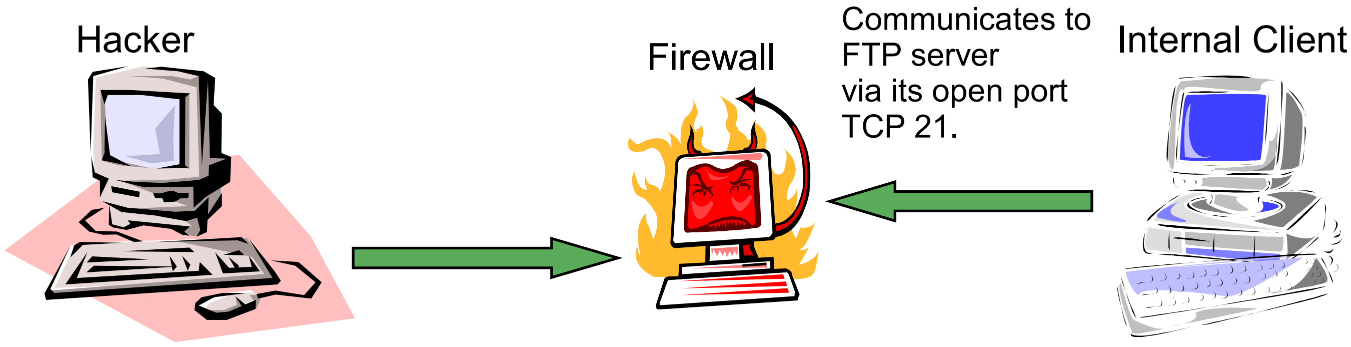
Firewall

Communicates to FTP server via its open port TCP 21.

Internal Client

FTP server then opens connection to FTP client from TCP port 20 to a high-numbered port on client for all data communications (e.g., dir, -ls, etc.)

Attack Scenario
Packet filtering firewall does not maintain state and cannot track one TCP connection with another. So, ***all connections from source port 20 to high numbered ports on its internal network are allowed*** and effectively pass through the firewall unhindered.



Hacker

Firewall

Communicates to FTP server via its open port TCP 21.

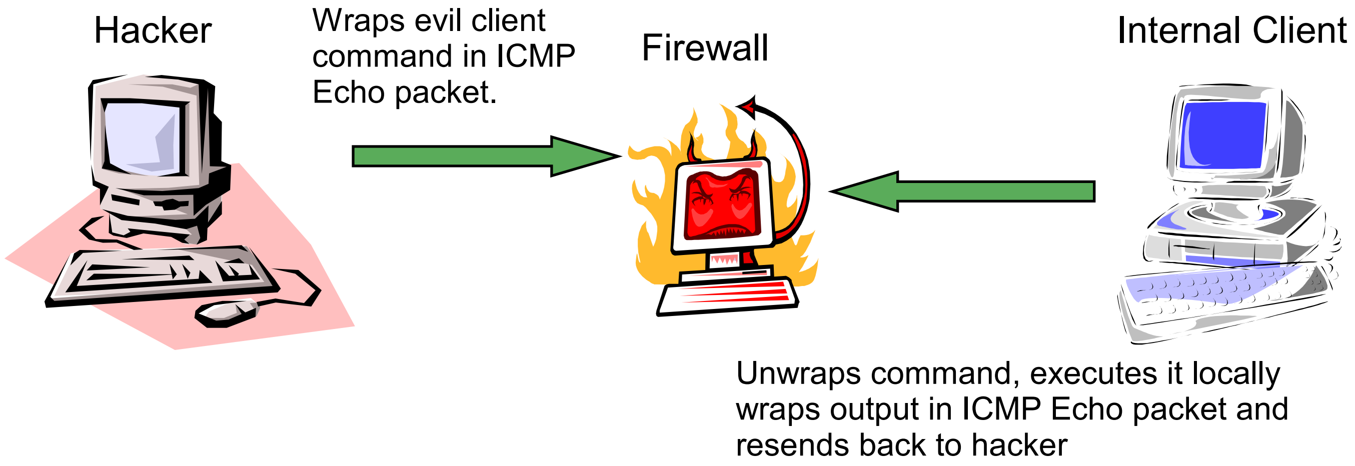
Internal Client

Attacker opens connection to FTP client from TCP port 20 to a high-numbered port on client - this allows nearly total access to the client.

Firewall Hacking - 3

- Packet Filtering
  - Weak ACLs - broad ACLs allow unintended traffic creating vulnerabilities.
  - Checkpoint Firewall Weaknesses - versions 3.0 & 4.0 open ports by default. UDP 520 (RIP), UDP 53 (DNS lookups), & TCP 53 (zone xfers) are allowed from any host to any host. Creates potential weakness once hacker has already compromised a system beyond the firewall (or used a trojan).
  - ICMP and UDP tunneling - wrapping real data in a packet header. Firewalls & routers that let ICMP Echo, ICMP Echo Reply & UDP packets through are vulnerable to this attack. Relies on an already compromised client box beyond the firewall.

Tunneling Example
Depends upon an already compromised client box - very easy to do via a trojan.



Hacker

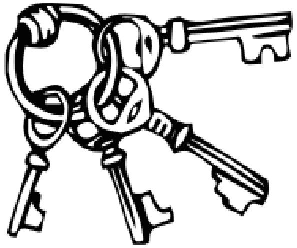
Wraps evil client command in ICMP Echo packet.

Firewall

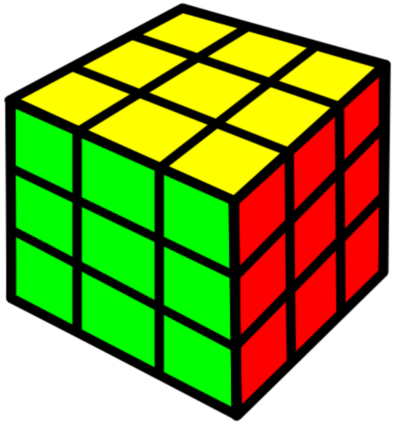
Internal Client

Unwraps command, executes it locally wraps output in ICMP Echo packet and resends back to hacker

Encryption Techniques



- Too many to list!
- CKM (Constructive Key Management) is best technology available.
- Protects data in motion and at rest.
- Relies on the proper execution of the encryption / decryption algorithms by the underlying software, firmware and hardware.
- Cannot protect data after decryption, which is needed to make data "human readable".
- Software that encrypts and decrypts is the point of vulnerability.
- Hacking targets the point of decryption so data is stolen after it has been decrypted.



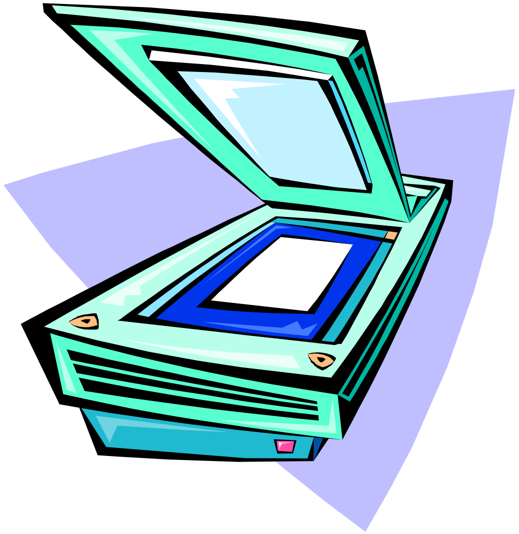
Hacker Methods -1

Footprinting

- Footprinting - discovering an organization’s network information:
  - Internet - domain names, network blocks, TCP services, system architecture, IDS, ACLs, banners, routing tables, SNMP, etc.
  - Intranet - protocols in use (e.g., IP, IPX, NetBUI, etc.), network blocks, IP addresses of reachable systems, etc.
  - Remote access - VPNs & related protocols, phone numbers
  - Extranet - Access control mechanism, connection origination/destination.

Hacker Methods - 2

Scanning



- Scanning - discovering which systems are alive and what they are running.
  - Ping Sweeps - sending ICMP ECHO(type 8) packets to target systems in a range of IP addresses to get ICMP ECHO_REPLY.
  - Port Scans - connecting to TCP (or UDP) ports on target system to identify services that are running.
  - Active Stack Fingerprinting - sending packets to target system and examining IP stack to detect an O/S specific implementation (e.g.,FIN packet probe, TCP initial window size, ACK value, ICMP message quoting, etc.).
  - Passive Stack Fingerprinting - passively monitoring network traffic for same purpose as above.

Automated tools

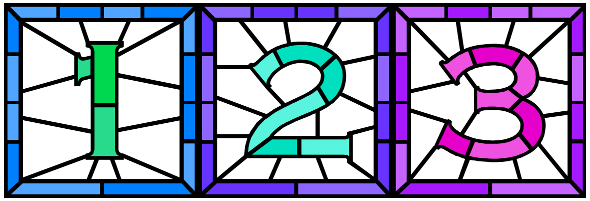
1) Superscan - www.foundstone.com/rdlabs/
termsofuse.php?filename=superscan.exe
2) NetScanTools Pro 2000 - www.nwpsw.com

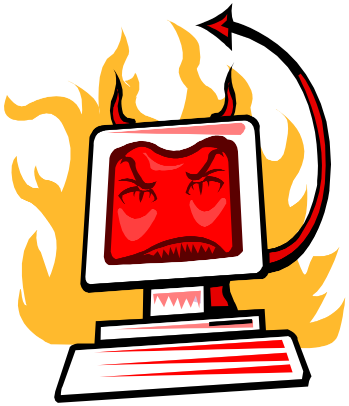
ICMP: Internet Control Messaging Protocol
TCP: Transmission Control Protocol
UDP: User Datagram Protocol

Hacker Methods – 3

Enumeration

- Enumeration - identifying valid information about the following areas:
  - Network Resources and Shares
  - Users and Groups
  - Applications and Banners
- Enumeration techniques are O/S specific, including:
  - Password guessing
  - Eavesdropping on network password exchange
  - Denial-of-Service (DOS)
  - Buffer Overflows

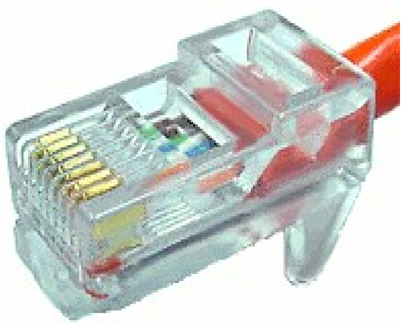




Buffer Overflow Exploiting
1) Phrack 49/14 & 55/15, www.phrack.org
2) www.cultdeadcow.com (general hacking stuff)

Hacking Methods – 4

Network Hacking

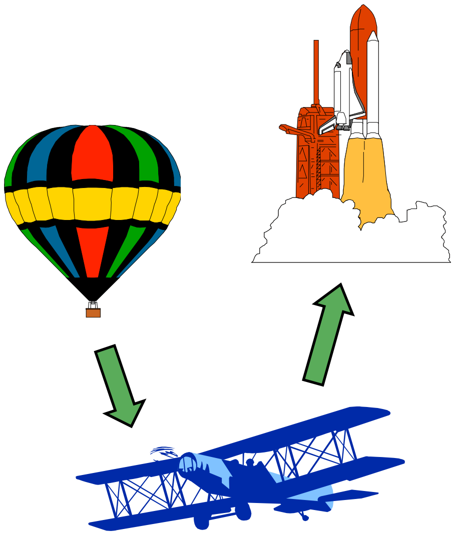


- Goal:
  - Own the network by listening to sensitive traffic & redirecting traffic to unauthorized systems.
- Methods:
  - Discovery - tracerouting, port scanning, O/S identification, Cisco (et al) Packet Leakage, banner grabbing, SNMP protocol insecurities, various router/gateway-specific weaknesses, etc.
  - Back Doors - default accounts, device specific weaknesses
  - Shared vs. Switched - detecting the network media/protocol, ARP Redirecting, RIP spoofing.
  - Wireless Network Hacking - War Driving, WEP attacks (exploiting implementation of RC4 stream cipher and the 50% chance of repeat initialization vector every 4,823 packets), WAP/WTLS attacks, etc.

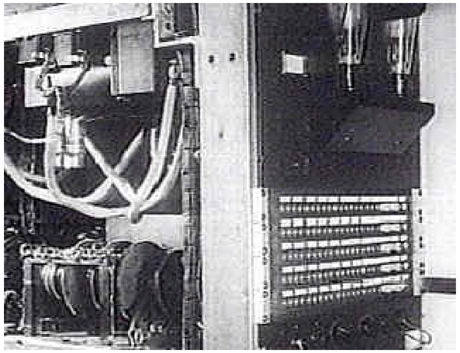
ARP: Address Resolution Protocol - maps a 32-bit IP to a 48-bit physical hardware address.
RIP: Routing Information Protocol
SNMP: Simple Network Management Protocol.
WAP: Wireless Application Protocol (cellular phone).
WTLS: Wireless Transport Layer Security, protects data transferred between cellular phones and WAP gateway.

IT Security History - 1

(How the Heck Did We Get Into This Mess?



- Why should we care?
  - It explains why we are in this condition.
  - It significantly affects our options today.
  - You can't get a solution if you don't understand the problem and ignorance is very expensive!



- The Early Years 1946 - 1960
  - Getting hardware working was the only issue.
  - O/S concepts not there yet.
  - Mainframes dominated.
  - Programming was a manual, time-consuming and painful process.
  - Need for security non-existent.

IT Security History - 2

(How the Heck Did We Get Into This Mess?)

- 1960 - 1965
  - Hardware more reliable.
  - Operating systems evolving.
  - Private Networks invented.
  - Some basic security concepts formed.
  - Small installed based allows lots of experimentation.
  - Multi-task O/Ss require task isolation, protection and scheduling (key invention #1).
- 1965 - 1970
  - Operating systems stable.
  - Growing installed base - fewer new architectures.
  - Public networks used for connectivity
  - Remote access requires user isolation & protection.
  - Multi-user O/S supports user isolation & protection (key invention #2).

IT Security History - 3

(How the Heck Did We Get Into This Mess?)

- 1970 - 1980
  - Midrange machines are introduced.
  - Large installed base (200,000-400,000 machines).
  - O/S has more features and computer languages evolving.
  - Significant remote access via public networks.
  - Security matters and a lot of research is done.
  - A secure system design is introduced in October 1972 (key invention #3 - more on this later).

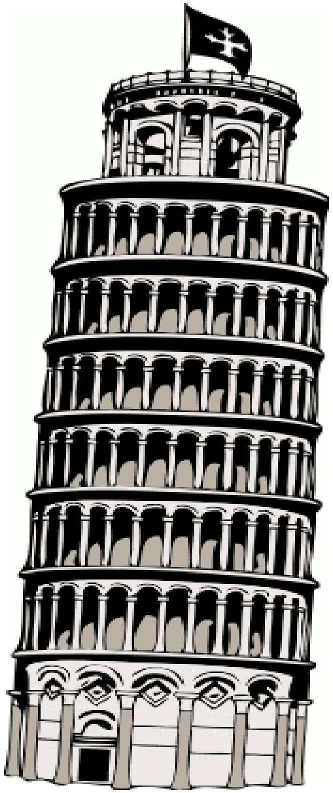
Consumer Market Economics Drove This

- The “PC / Desktop Revolution” 1980 - 2000
  - O/S starts out simple and dumb (key inventions 1, 2 & 3 were ignored – oops!).
  - Networking PCs requires security to control who can do what (privilege) and how much of a resource may be used (quota).
  - Security is deliberately eliminated (oops!).
  - O/S grows to encompass what used to be applications.
  - Cost is the driver - security & performance are secondary.
  - Multimedia and entertainment matter.
  - Huge installed base limits change.
  - Internet becomes available.

Software Structure - 1

IT Systems are Like a Multi-Story Building!

- **Business Applications**
- **Databases**
- **Network communication software, Internet**
- **Languages, Compilers & Tools**
- **Utilities & Libraries**
- **Operating system**
- **ISA (Hardware Layer)**



- **Libraries**
- **Books**
- **Chapters**
- **Paragraphs**
- **Sentences**
- **Words**
- **Alphabet**

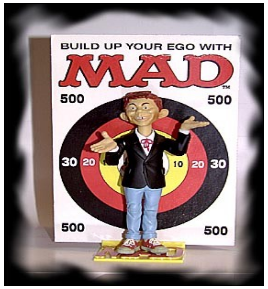
We are focusing here

We need to focus here

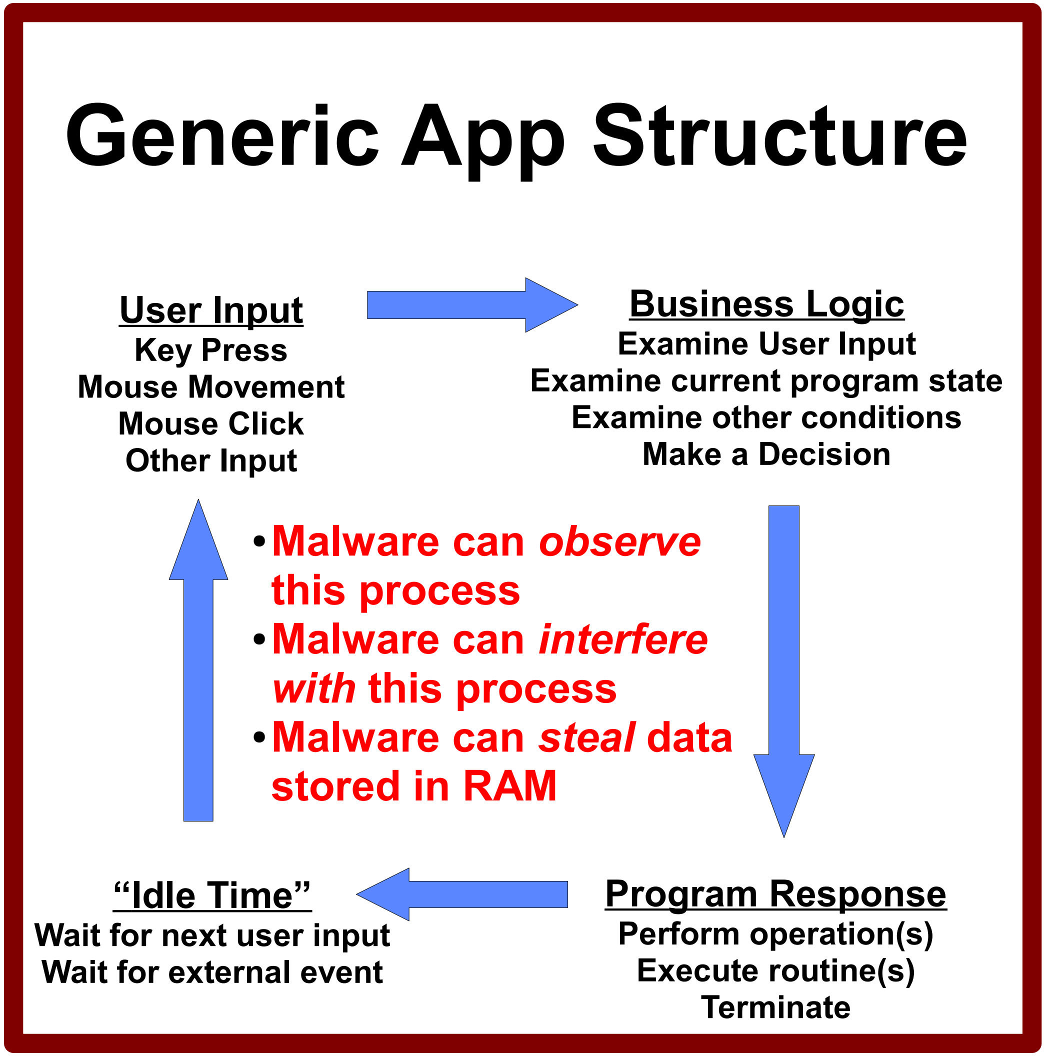
1) Layers made by different & competing vendors
2) Minimal universal standards between layers
3) Inter-layer interfaces changing often and without warning
4) Interface features come and go from release-to-release
5) There's no “Master Plan”

Software Structure – 2

Why Browsers Are Dangerous



What, Me Worry?



Generic App Structure

User Input
Key Press
Mouse Movement
Mouse Click
Other Input

Business Logic
Examine User Input
Examine current program state
Examine other conditions
Make a Decision

- **Malware can *observe* this process**
- **Malware can *interfere with* this process**
- **Malware can *steal* data stored in RAM**

Program Response
Perform operation(s)
Execute routine(s)
Terminate

"Idle Time"
Wait for next user input
Wait for external event

- Applications receive user inputs and make one or more decisions.
- Applications perform one or more operations in response to the decision.
- Applications then wait for the next user input and event.
- All this processing occurs on the desktop, laptop, mobile device.
- Encrypted data is still decrypted on the device and stored as 'plaintext' in RAM – where it is vulnerable to "spyware".
- "Spyware" and "malware" can observe and interfere with this process because it has access to RAM on the same device.
- Malware can observe and 'steal' critical data.

Security Research Results

(40 Years of Learning)

- What Secure Systems Must Do:
  - Control access to data objects.
  - Track and record access to data objects for audit purposes
  - Permit only authorized entities to read, write, create or delete information.
  - Must architect the system so that it obeys a secure system model.
  - Must have proper architecture, coding and deployment of the O/S, the application and all layers in between.
  - Discovering security flaws then fixing them one-by-one does NOT work.

What We Know Now – Research Results 2

Secure System Requirements

- **Policy**
  - **Security Policy** - System must enforce a well-defined security policy.
  - **Marking** - System must associate all objects with access control labels (sensitivity & access modes).
- **Accountability**
  - **Identification** - System must identify individuals and their various authorizations in a secure manner.
  - **Audit Trail** - System must keep & protect audit trail so actions may be traced to responsible party.
- **Assurance**
  - **Evaluation** - System must have hardware/software mechanisms that can be independently evaluated to assure that policy & accountability are enforced.
  - **Continuous Protection** - System must continuously protect trusted mechanisms that enforce policy & accountability from tampering.

Secure System Classification – 1

DoD 5200.28 (December 1986)

- **D - Minimal Protection**
- **C - Discretionary Protection**
  - **C1 - Discretionary security protection** - separates users & data, uses credible controls to enforce access limitations on an individual basis.
  - **C2 - Controlled Access Protection** - users individually accountable for their actions, security audit trail, resource isolation.
- **B - Mandatory Protection**
  - **B1 - Labeled Security Protection** - security policy model, keeps integrity of sensitivity labels, sensitivity labels must be held in all major system data structures, demonstration of reference monitor implementation.
  - **B2 - Structured Protection** - formal security policy model, discretionary & mandatory access control enforcement extended to all subjects & objects, separation of critical & non-critical system elements, stringent configuration management controls, covert channels are addressed, relatively resistant to penetration.
  - **B3 - Security Domains** - Reference monitor mediates all accesses of subjects to objects, be 100% tamperproof, TCB (trusted Computing Base) only contains security-relevant code & data structures, system engineered for minimal complexity, security-relevant events are signaled, system recovery is required, highly resistant to penetration.
- **A - Verified Protection**
  - **A1 - Verified Design** - Functionally same as B3, full mathematical verification of design.

Secure System Classification – 2

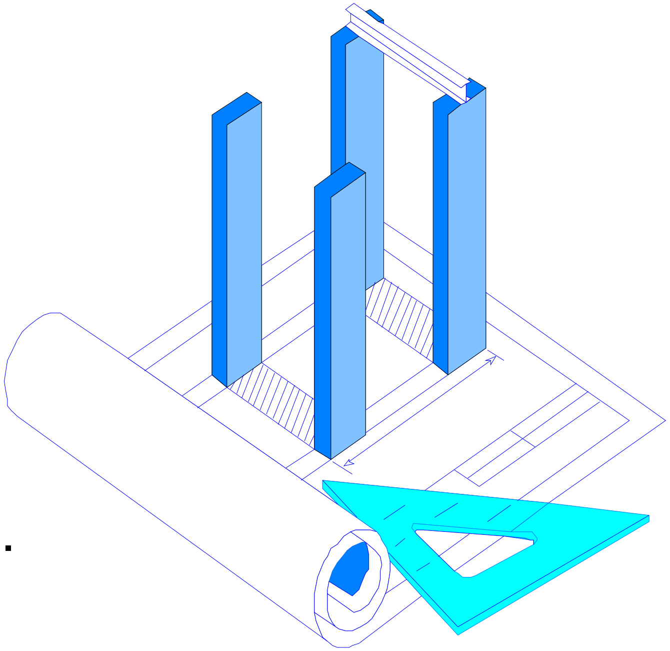
NIST / ISO 15408

(NIST - National Institute of Standards & Technology)
(NIAP - National Information Assurance Partnership)
ISO 15408

- **EAL-1 - Functionally Tested** - independent testing of selected features.
- **EAL-2 - Structurally Tested** - independent testing of selected features using limited developer design data.
- **EAL-3 - Methodically Tested & Checked** - independent testing using limited developer design data, selective developer result confirmation, evidence of develop search for obvious vulnerabilities.
- **EAL-4 - Methodically Designed, Tested & Reviewed** - independent testing using low-level vendor design data, search for vulnerabilities, development controls, automated configuration management.
- **EAL-5 - Semiformally Designed & Tested** - independent testing of all of the implementation (TOE), formal model, semiformal conformance to design specs, vulnerability assessment for attackers with moderate potential.
- **EAL-6 - Semiformally Verified Design & Tested** - independent testing of 100% of TOE, modular & layered approach to design, structured presentation, vulnerability assessment for attackers with high potential, systematic search for covert channels.
- **EAL-7 - Formally Verified Design & Tested** - same as above, but all models, specs & presentations are formal, TOE is tightly focused on security functionality, amenable to formal analysis, design complexity must be minimized.

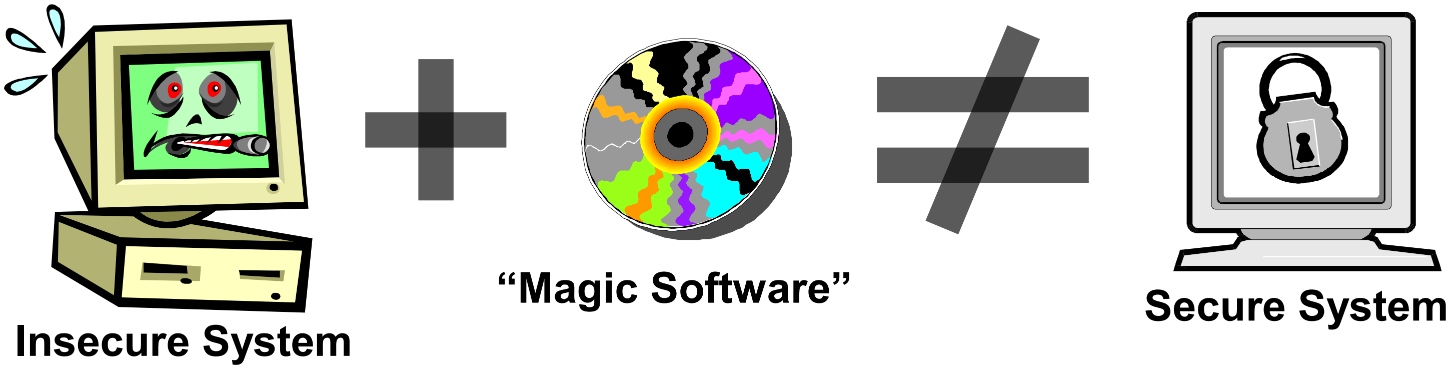
Cyber Briefing Observations - 1

- IT systems are built over time.
- Software is built in layers.
- Little control over the underlying layers.
- Focusing on the top layers can't work.
- Network security alone can't work.
- Browsers are very vulnerable.
- Desktops, laptops, palmtops are all vulnerable.
- Poor past decisions limit choices today.
- Consumer market economics drives decisions – not good.
- Critical functions were assigned to non-secure platforms – very bad.



Cyber Briefing Observations - 2

- Security cannot be just “added-on”.
- A mythical desktop security “add-on” can’t be compatible with existing IT systems
- A “magic-CD” that will 100% protect your desktop without any other changes is not possible now.
- Going over each line of code - a famous Microsoft quote - won’t fix the security problem.
- Must redesign the O/S, and application and network protocols & software architecture.
- Security is more of an economic issue than a technical one.


Insecure System

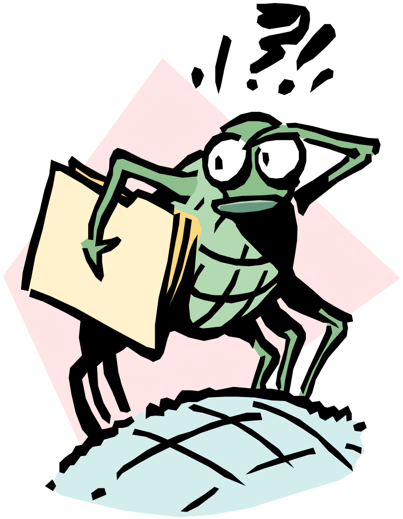
“Magic Software”

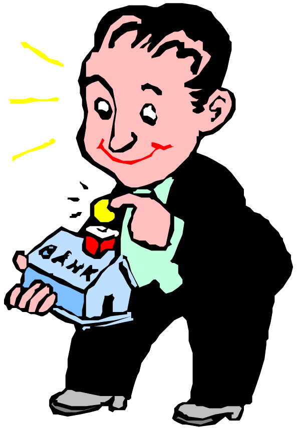
Secure System

What We Know Now – Observations & Achievable Objectives 3

Common Security Questions

- How can I protect my PC?
- How can I prevent viruses from damaging my IT systems?
- How can I clean up the damage from viruses?
- How can I prevent malicious attacks from stealing data?
- How can I prevent malicious attacks from planting a time bomb?
- How can I take advantage of new technology?
- How can I do all this and still save money?





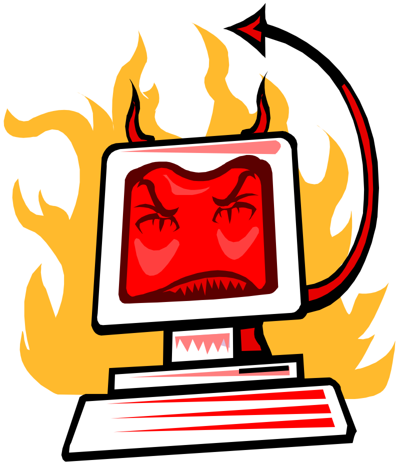
What We Know Now – Observations & Achievable Objectives 4

Very Useful, Achievable Objectives

- Protect data assets
- Provide continuity of operations
- Make IT systems tamper-proof

Unachievable Objectives

(of limited value)

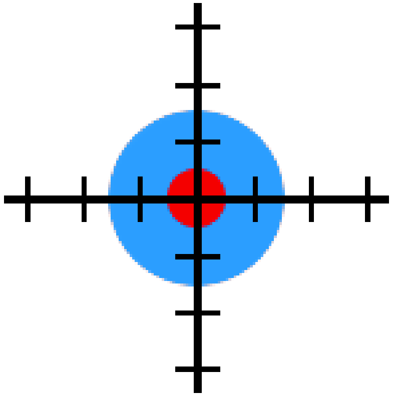


- Prevent cyber attacks
- 100% “Cyber-shield” each desktop
- Prevent all human errors
- Prevent “insider attacks”

What We Know Now – Observations & Achievable Objectives 5

The “Mess”

(How the Heck Did We Get Into This Mess?)



- The Three Mistakes
  - Small systems do not scale up.
  - Security was never built-in to the desktop O/S architecture where it is required.
  - Critical tasks were deployed on small systems which were never intended to be secure, which made those tasks and operational capability vulnerable to attack.
- So where are we?
  - Network protocol has built-in security flaws or “holes”.
  - Critical functions are deployed on vulnerable systems.
  - Browser access makes it worse because they are hackable.
  - Huge installed base prevents significant re-design because consumer market economics is the driver.
  - Huge economic dis-incentives to fix this because B-B and B-C market dynamics are at cross purposes – and the consumer market is the driver (due in part to mandating COTS).
  - Secure systems require the correct architecture at the core of the O/S – otherwise they won't be secure.

100,000
100,000
100,000
100,000
100,000
THE UNITED STATES OF AMERICA
ONE HUNDRED THOUSAND DOLLARS IN GOLD

What We Know Now – Dirty Little Secret 1

The “Dirty Little Secret”

(We Know How to Solve The Security Problem)

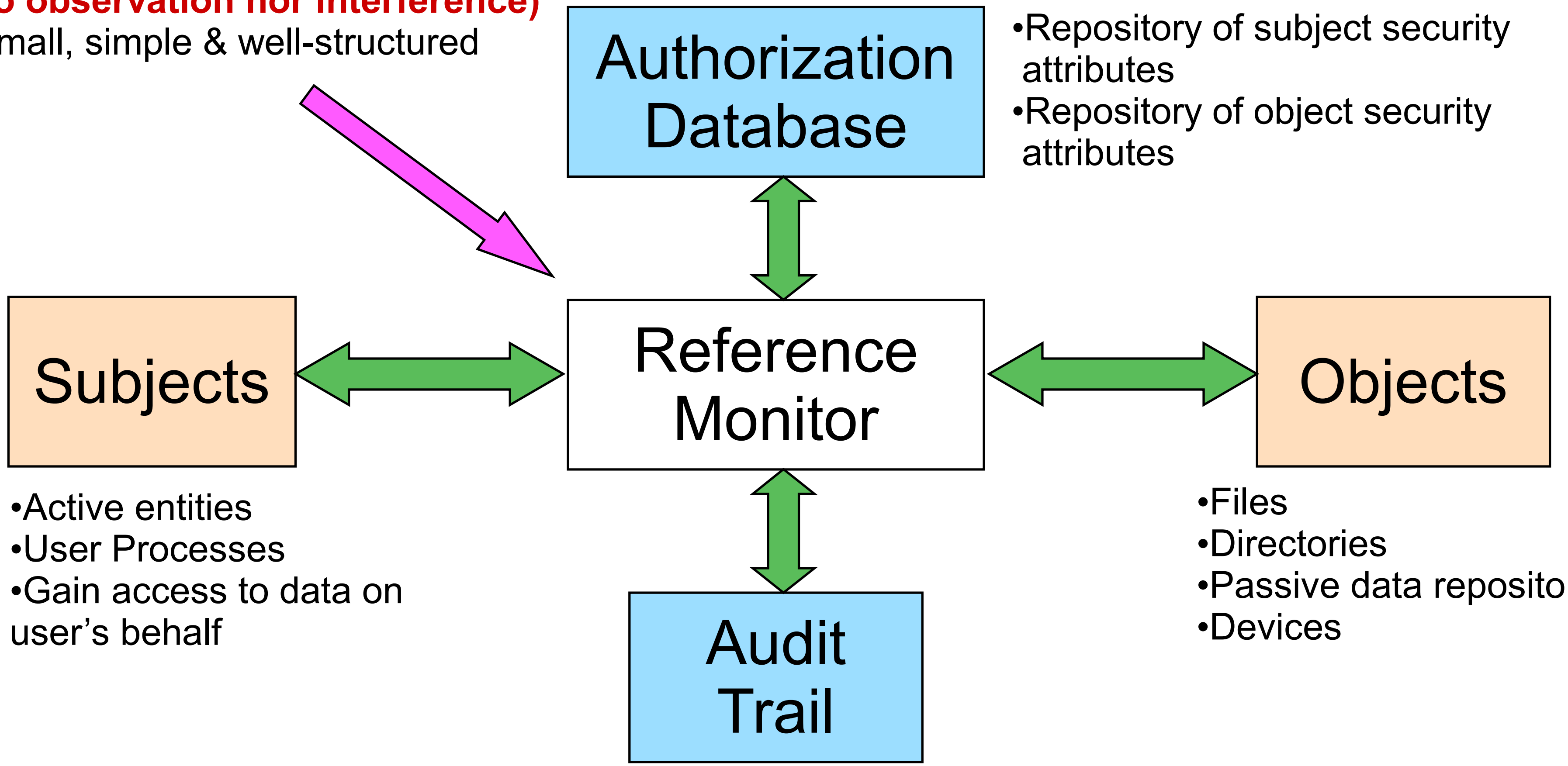
- October 1972 – USAF produces “Computer Security Technology Planning Study” (ESD-TR-73-51 Vol.II, produced per contract with James P. Anderson & Co.) – they invent the “Reference Monitor”, a secure system architecture.
- April 1974 – Barry Schrager @IBM headed up the RACF (data security stuff) project. They implement what they can but the economics of the installed base prevents wide adoption of a fundamentally new O/S architecture.
- Mid 1970s - DEC was transitioning from the 16-bit PDP-11 to the new 32-bit VAX architecture so a new O/S was warranted. They included most of the “Reference Monitor” in the design of their VMS O/S.
- Mid 1980s – Business apps start migrating to PCs because they are perceived to be cheap - no real plan for scale-up nor security.
- Mid 1990s – Deployment of critical business apps and government apps to desktops continues, networking is ubiquitous, security issues becoming important.
- 1998 - Compaq buys DEC
- 2001 - HP buys Compaq.
- 2011 – The only transaction-based, real-time O/S that has not been successfully hacked (when configured properly) is OVMS (c.f., DEFCON 9 in 2001, Kevin Mitnick's testimony).

What We Know Now – Dirty Little Secret 2

The Reference Monitor

(A Secure System Architecture USAF, October 1972)

- •Enforces security policy
- •Mediates all attempts by subjects to access objects
- •Tamperproof database & audit trail **(no observation nor interference)**
- •Small, simple & well-structured



Authorization Database

- •Repository of subject security attributes
- •Repository of object security attributes

Subjects

Reference Monitor

Objects

- •Active entities
- •User Processes
- •Gain access to data on user's behalf

- •Files
- •Directories
- •Passive data repositories
- •Devices

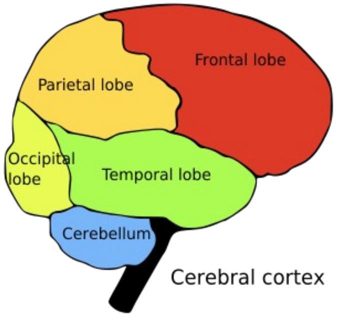
Audit Trail

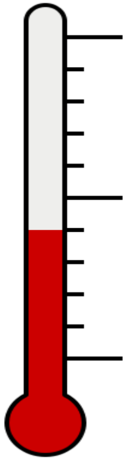
- •Record of all security-related events

Policy – General Guidelines 1

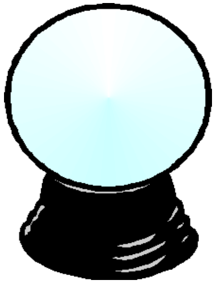
Policy – General Guidelines

- This is an “IQ War” - the “biggest brain” wins and brains beat bucks ($$$)
- It's OK to mandate a result; e.g., “Keep the data safe” or “Maintain operations”.
- It's NOT OK to mandate a detailed process; e.g., use triple DES encryption, use “Windows-7 SP4”, etc.
- Must allow the market to innovate and develop new solutions.
- Mandating “COTS” (Commercial Off-The-Shelf) helped get us into this mess – stop doing that.
- “One-size-fits-all” is also bad thinking.
- Need federal involvement in standards for accuracy, completeness and market stability.

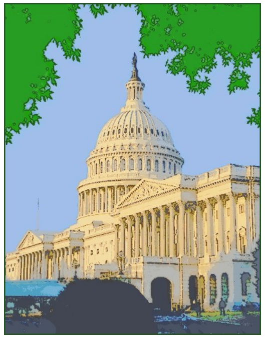


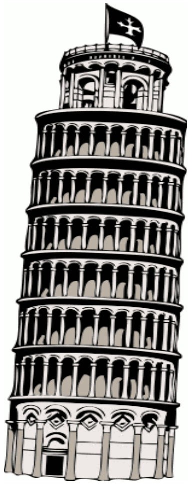


Policy – General Guidelines



- Free market can handle short-term items because the metrics are short term (quarterly reports and stock performance).
- Long-term items need federal guidance to provide stability to the market:
  - Feds can afford to spend the dollars, do the research and set the long-term standards that are useful and accurate (e.g., USAF, NASA, NSF).
  - Companies cannot justify an investment if the regulation will change dramatically down the road – market uncertainty is bad.
  - Can't have federal specifications written by consumer-market people.
  - Cannot let consumer market economics drive the security effort.
- Get physical control of the “IT Stack”, including hardware, manufacturing, design tools, O/S and apps.
- Maintain federal control of operational standards – do not let that fall to other organizations / countries with a different agenda.
- Don't place barriers in the way of private industry innovation by mandating meaningless “certs”.





Policy – Specific Items

- Do assume your network is polluted with “evil” packets
- Do maintain “standard security procedures” for existing systems. They are not perfect but it's better than nothing.
- Do segment network as much as possible and “air-gap” mission-critical systems.
- Don't allow wireless access to secure systems.
- Don't put important stuff on a hand-held or desktop device
- Don't assign mission critical operations to a non-secure platform. If you have already done so, then migrate away from that ASAP.