INF523: Assurance in Cyberspace Applied to Information Security

Subversion Case Studies
Potential Mitigations

Prof. Clifford Neuman

Lecture 12
13 November 2020

Network File Service (NFS) Security

- Case study of NFS subversion demonstration
  - Running example by US Navy masters student
  - Emory A. Anderson, III, for Prof Cynthia Irvine (NPS)
  - Shown to Richard Clarke, “first cybersecurity czar”
- First, consider security implications for system
  - How deeply rooted are adverse consequences
- Second, explore applicability to other systems
  - Address whether attack approach is limited to NFS
  - Briefly examine Anderson SSL subversion design
- Follow on – Later NFS case study of mitigation
  - Compare to Anderson recommended solution

Likely Tool of Professional Attacker

- Subversion is technique of choice [And 1.D]
  - Professional distinguished from amateur
- A primary objective is avoiding detection
  - Amateur often motivated by desire for notoriety
- Professional often well-funded
  - Resources to research and test in closed environment
  - Amateur tends to numerous attempts on live target
  - Flawless execution reduces risk of detection
- Coordinated attacks are mark of a professional
- Professional will invest and be patient to use
  - Subverter is likely different than attacker

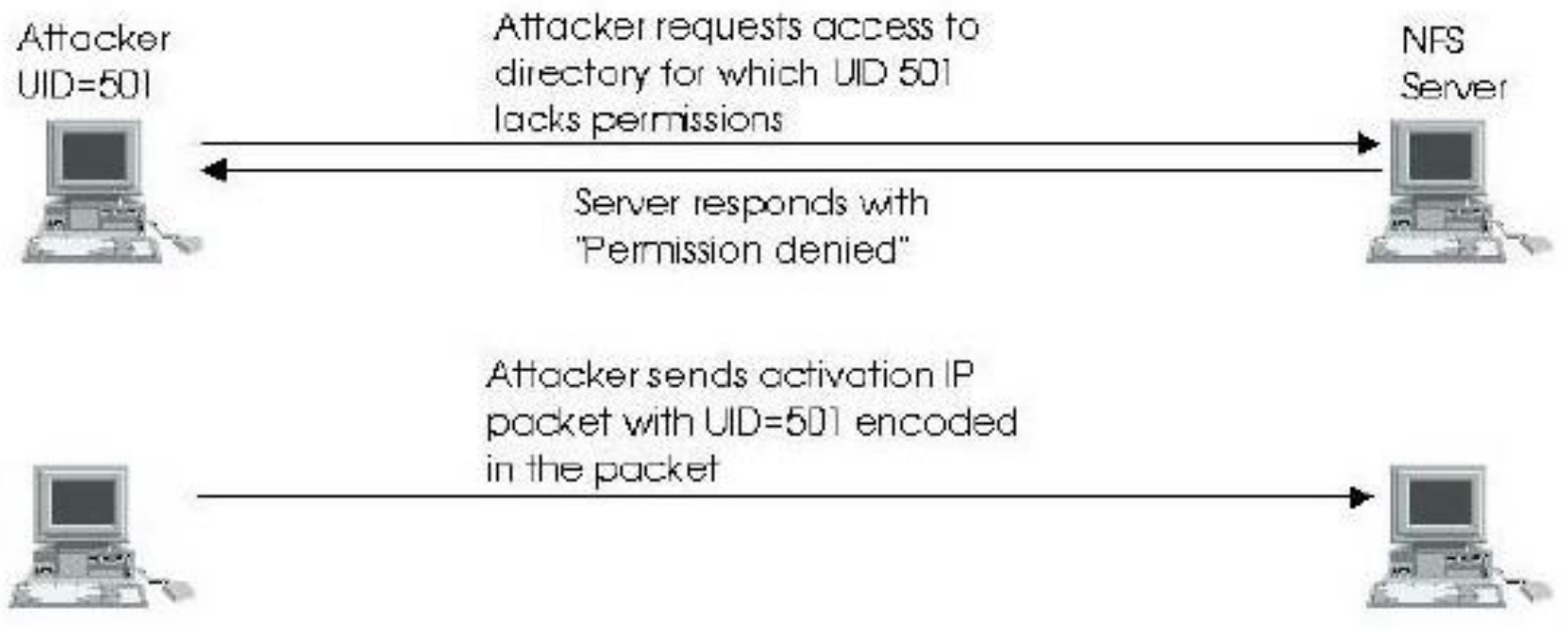
Demonstration of Subversion

- Obfuscation of artifice not given serious attention
  - Would be of utmost importance to professional attack
- Subversion can occur multiple points in lifecycle
- Selected distribution phase for demonstration
  - Driven by limited resources and access of student
  - Facilitated by NFS on open source Linux system
  - Representative of attacker mirror site opportunities
- Closed source not daunting for professional
  - May involve reverse engineering application
  - Might create a binary patch to insert in binaries
  - Entirely within anticipated professional resources

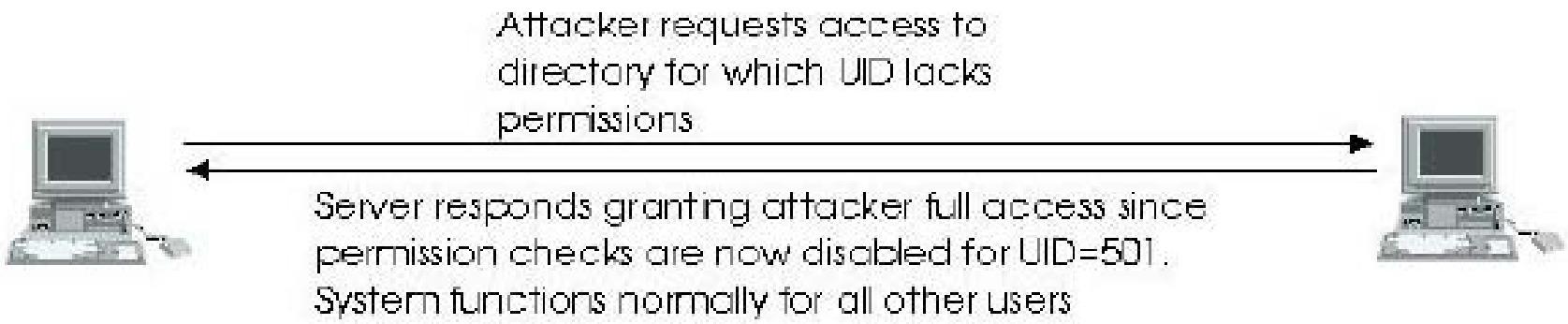
Choice of NFS as Suitable Application

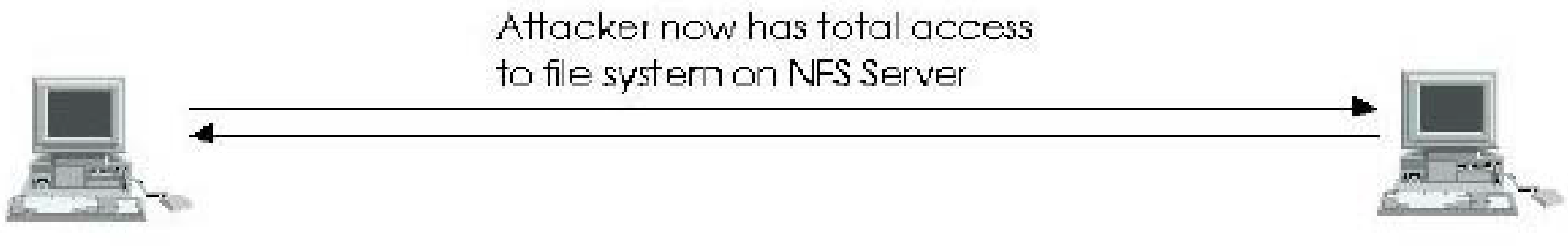
- For impact, need readily apparent significance
  - NFS is application familiar to typical IT user
  - Users understand notion of need to protect data
- Activation needs to be straightforward
  - Network interface chosen for ease of explanation
  - Internet technology is widely used
- Choose to have remote activation
  - Representative of low risk for attacker
  - Also supports local activation, e.g., via loopback
  - Trigger is a malformed Internet packet
- **Study of subversion method benefits student**

Case System and Activate the Artifice

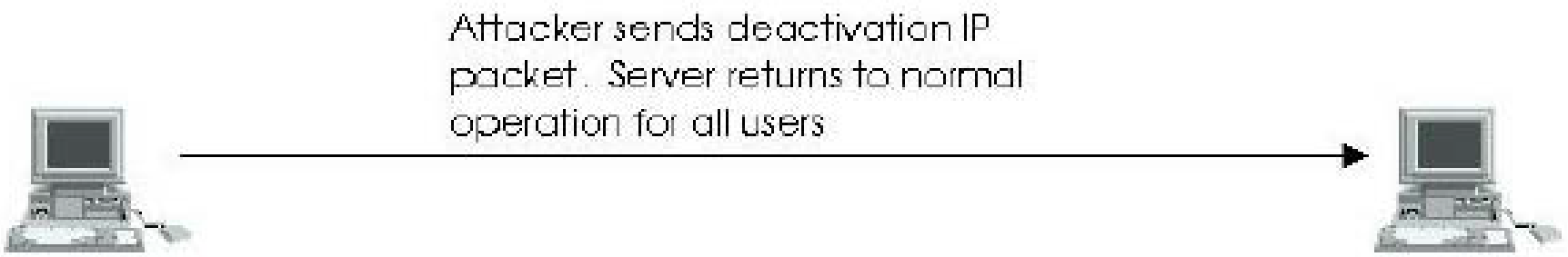


Attacker Uses Artifice for NFS Access





End Session by Deactivating Artifice



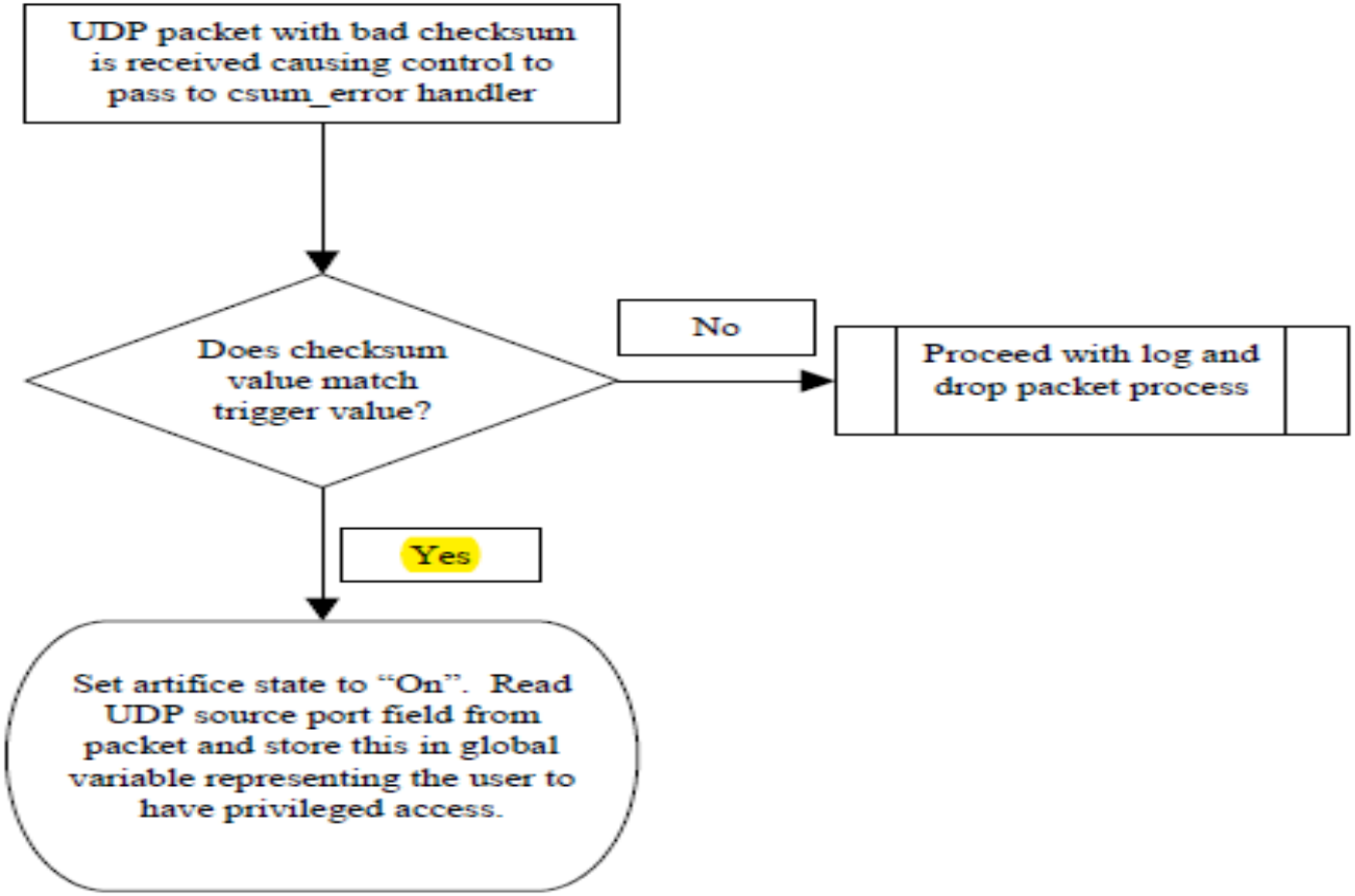
Design Properties of NFS Artifice

- Purpose of artifice to bypass file permissions
  - Bypass check for a specified user at will
  - Then re-enable the normal system operation
- Exhibits all the characteristics of subversion
  - Exception was no attempt for hide or obfuscate
- Artifice is small – eleven C statements
  - Small in relation to millions LOC in Linux kernel
  - Unlikely to be notice by those in Linux development
- Can be activated and deactivated
  - Further complicates attempts to discover existence
- Does not depend on activities of a system user

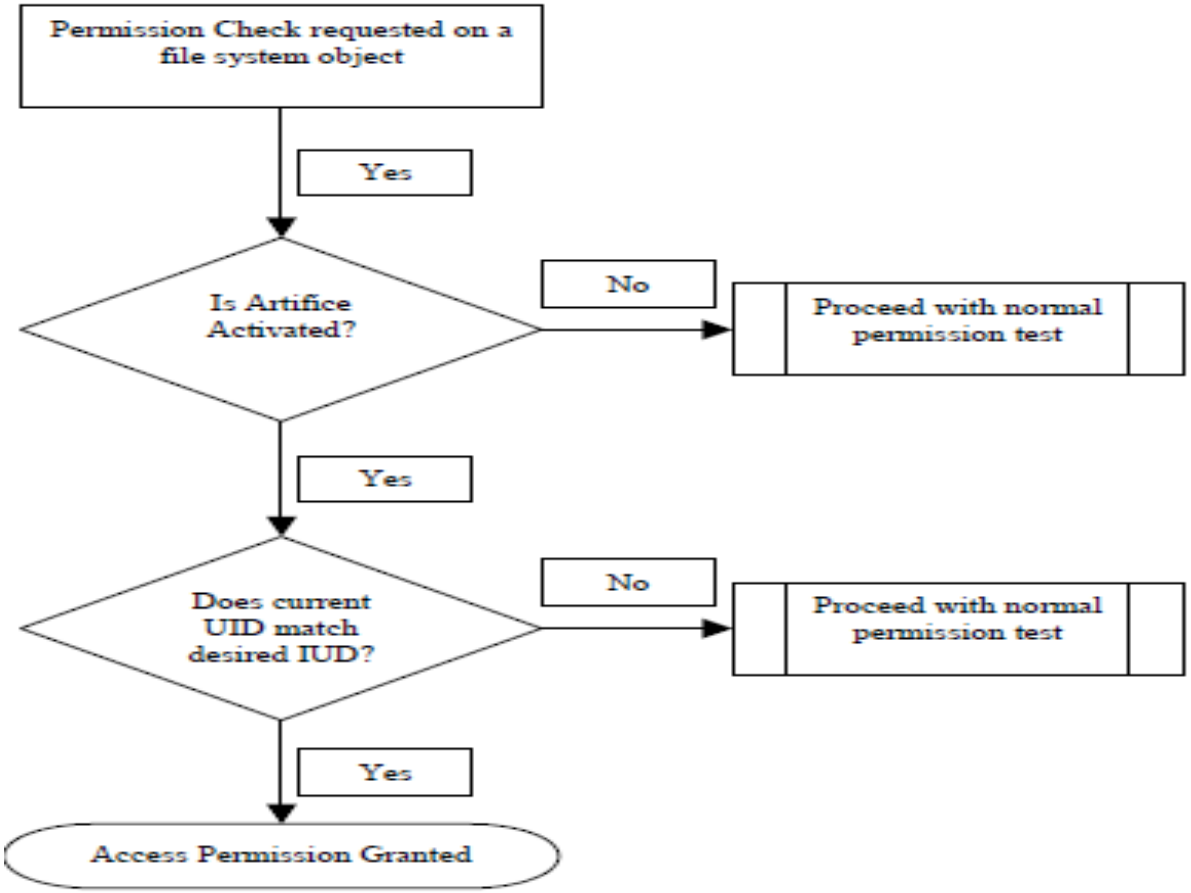
Artifice Functions

- Composed of two parts in two unrelated areas
- Subvert a portion of kernel that receives packets
  - Recognize a packet with distinguishing characteristics
  - Activation based on trigger known only to subverter
  - Extends normal check for packet checksum
- Activation recorded in global variable in kernel
- Subverts Linux file system permission checks
  - Check global kernel variable to see if activated
  - Grants attacker access to any file in the system
  - Bypass behavior limited to specified user ID
  - System functions normally for all other users

Artifice Activation



Subverted File Permission Checks



INF523: Assurance in Cyberspace Applied to Information Security

Subversion Case Studies
Potential Mitigations

Prof. Clifford Neuman

Lecture 12
13 November 2020

Extra Session: Tuesday 17 November 1:00 PM – 4:20 PM PST

- Process Control and Medical Devices (40 min)
  - Medical Devices – Jaynee Shah
  - Industrial Control Systems - Venkat Ramana Reddy Mareddy
- The Cloud and Storage/Database Infrastructure (80 min)
  - Database Servers – Di Rama
  - Cloud Security - Shagun Bhatia
  - FedRamp - Dewaine Reddish
  - Risk Management - Sarahzin Chowdhury
- Isolation and Key Management (40 min)
  - Identity Tokens and Yubikey - Arjun G. Raman
  - Isolation Technologies - Ayush Ambastha

Extra Session Thursday 19 November 1:00 PM – 4:20 PM PST

Mobile Devices

- Mobile OS - Chinmaya Pandit and Harshit Kothari
- Android - Mohammed Ababtain

Payment

- Apple Pay - Jairo Hernandez
- Apple Pay and Google Pay - MaryLiza Walker
- Apple Pay - Shanice Williams
- Apple Pay - Yang Xue
- Assurance in Payment Systems - Uddipt Sharma

Friday November 20

Operating Systems

- Linux Applications - Aditya Goindi
- Linux - Tejas Pandey
- Chrome OS - Malavika Prabhakar

Infrastructure and Vehicle Control Systems

- US Voting Infrastructure - Anthony Cassar
- Autonomous Vehicles - Chris Samayoa
- Autonomous Vehicles - Amarbir Singh
- Connected and Automated Vehicles - Abhishek Tatti
- Tesla - Dwayne Robinson
- Avionics - Pratyush Prakhar

Separate Design of SSL Subversion

- Secure Sockets Layer (SSL) widespread use
  - Secure communications between client and server
  - Client and server negotiate session keys
  - Encrypt traffic using symmetric encryption algorithm
- Options available to attacker for subversion
  - Duplicate all communications and send to attacker
  - Weaken key generation mechanism – limit entropy
  - Simply send the session keys out to the attacker
- Advantages of exfiltrating session keys
  - Attacker is passive and maintains anonymity.
  - Subverting either client or server gives total acess

NFS Subversion Technical Conclusions

- Practice for showing security inadequate at best
  - Penetration tests and add-on third party products
  - Layered defenses and security patches irrational
- Bad defense more dangerous than poor security
  - Leads to flawed belief system has adequate security
  - Can increase risk by more dependence on defense
- Have technology to provide appropriate security
  - Evaluation criteria tried and tested
  - These approaches have fallen into disfavor
- The need to address subversion is increasing
  - Threat sources multiplying and reliance increasing

NFS Subversion System Decisions

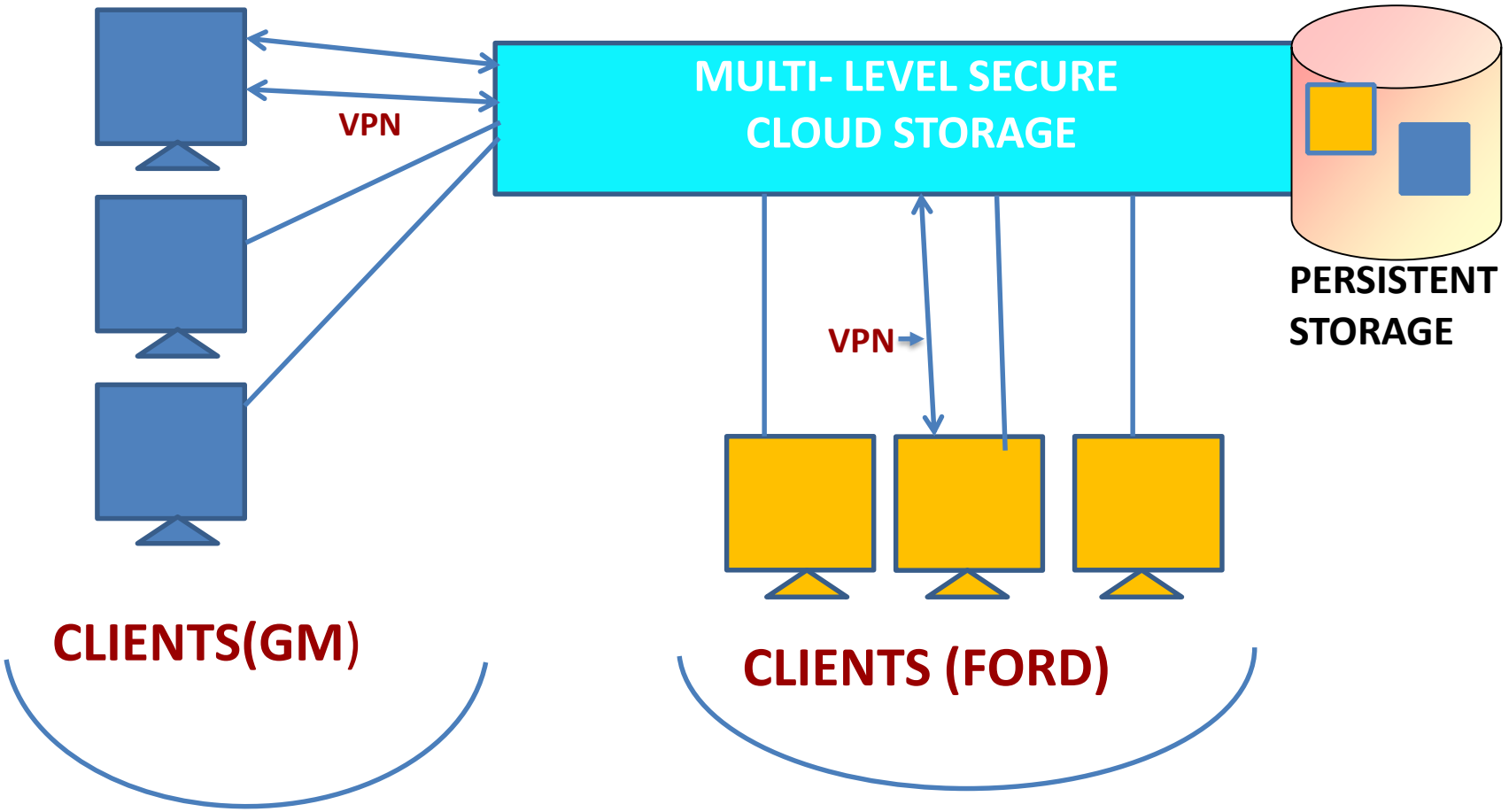
- Must address subversion for justification of trust
  - Irresponsible not to consider when deploying systems
  - Otherwise flawed belief system security is adequate
- Nurture a vast industry with add-on applications
  - Huge drain on resources for little or no assurance
- Objective of demonstration to raise awareness
  - Enable decision maker to understand the problem
    - Need to understand motive, means and opportunity
    - Consider subversion practicality and consequences
  - Make decision makers aware of proven technology
    - Verifiable protection technology applied successfully
- **Security professionals have a responsibility**

Recall Study Goals for NFS Subversion

- First, consider security implications for system
  - How deeply rooted are adverse consequences
- Second, explore applicability to other systems
  - Address whether attack approach is limited to NFS
  - Briefly examine Anderson SSL subversion design
- Next – NFS case study of mitigation
  - Compare to Anderson recommended solution

- **What else can be learned from the demo?**

Notional Cloud Storage Security



MLS File Sharing Server for Cloud

- Cloud storage service
  - Specific type of cloud computing
  - Managed resource is storage
- Needs security as good as enterprise
  - Typically replaces services of enterprise environment
  - Many of the same vulnerability as self-managed
  - Additional vulnerabilities specific to the cloud
- Current solutions are completely ineffective
  - Essential problem is construct of shared infrastructure
  - Built on low-assurance commodity technology
- Highly vulnerable to software subversion

Security Requirements of cloud

- Three primary cloud security requirements
  - Controlled sharing of information
  - Cloud isolation
  - High Assurance

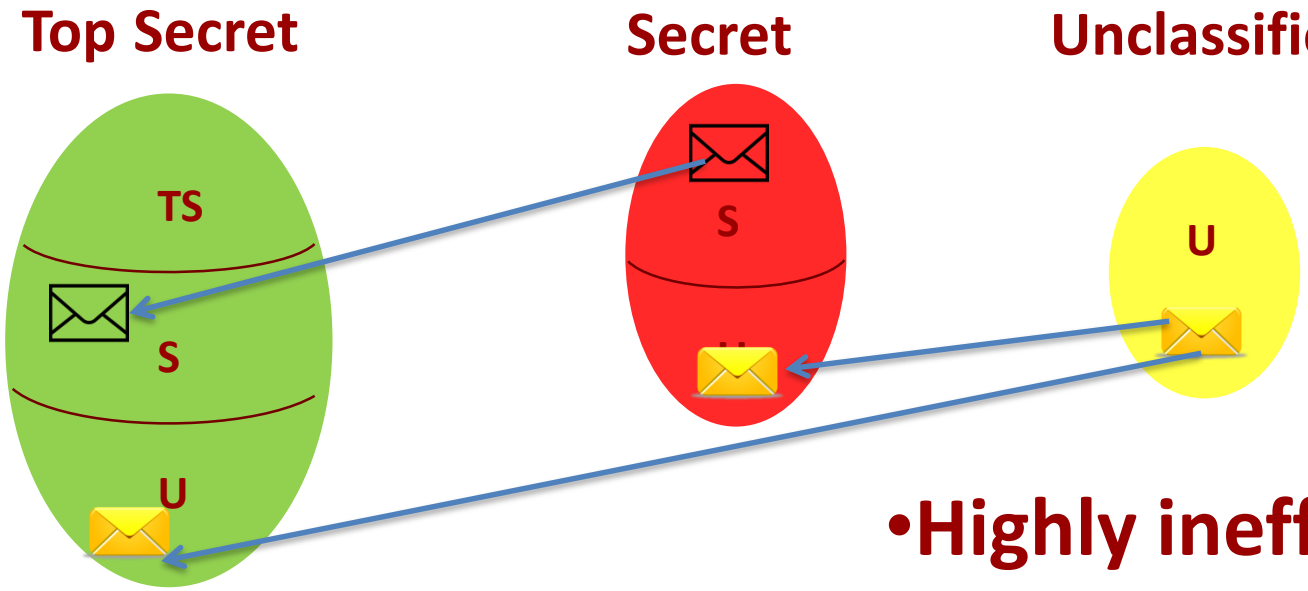
Trap Door Subversion Vulnerability

- Malicious code in platform
  - Software, e.g., operating system, drivers, tools
  - Hardware/firmware, e.g., BIOS in PROM
  - Artifice can be embedded any time during lifecycle
  - Adversary chooses time of activation
- Can be remotely activated/deactivated
  - Unique “key” or trigger known only to attacker
  - Needs no (even unwitting) victim use or cooperation
- Efficacy and Effectiveness Demonstrated
  - Exploitable by malicious applications, e.g., Trojans
  - Long-term, high potential future benefit to adversary
  - Testing not at all a practical way to detect

Alternatives for Controlled Sharing

- Three ways controlled sharing can be facilitated:
- Massive copies of data from all lower levels
  - High assurance one-way flow of information
  - Light diode interface uses physics for high assurance
- File Caching (Local Active Copy)
  - Retain at high level only actually used lower data
  - No way to securely make requests for lower data
  - Security requires manual intervention
- High assurance segmented virtual memory

Massive Copies Approach

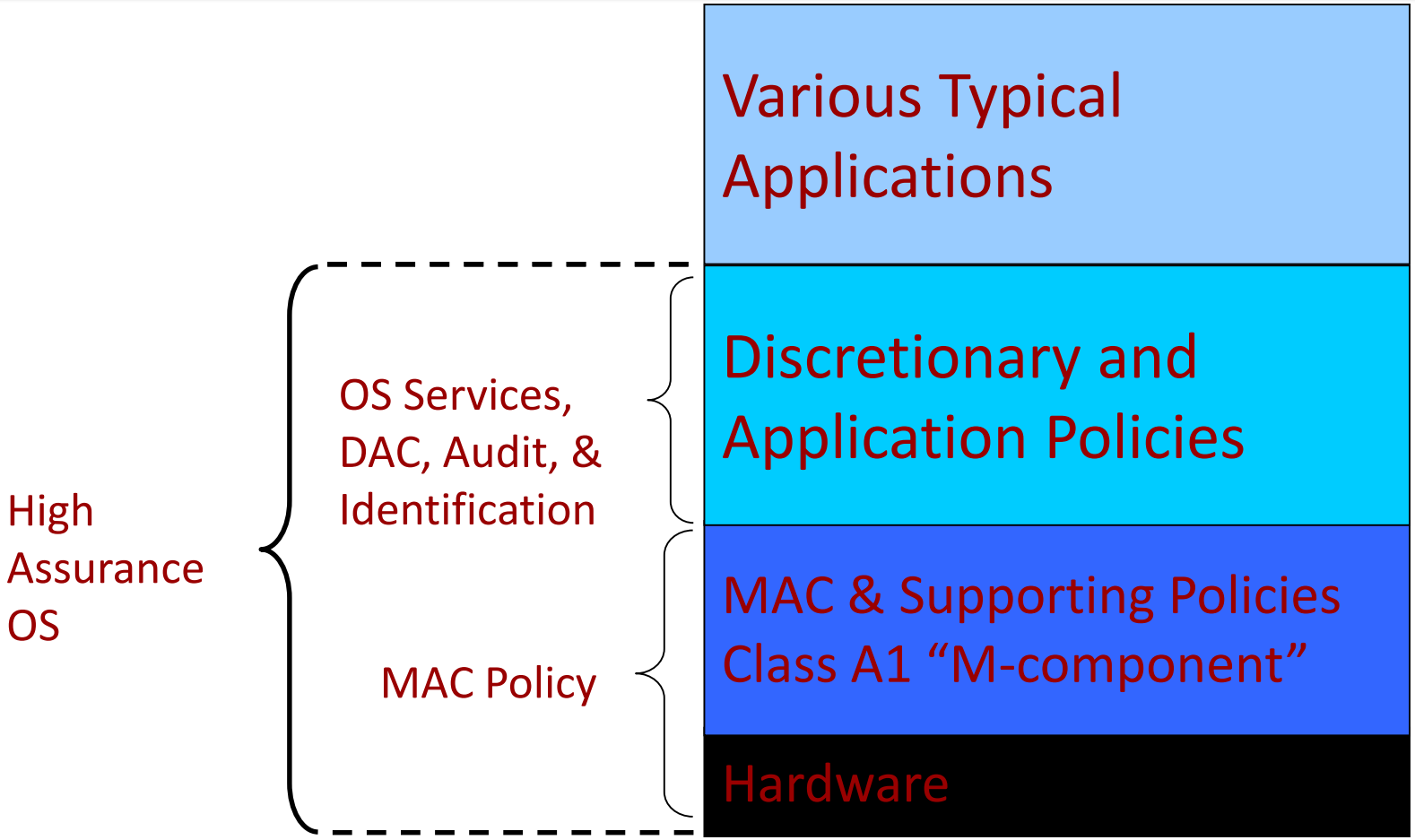


- **Highly inefficient!**
- **Does not scale!**

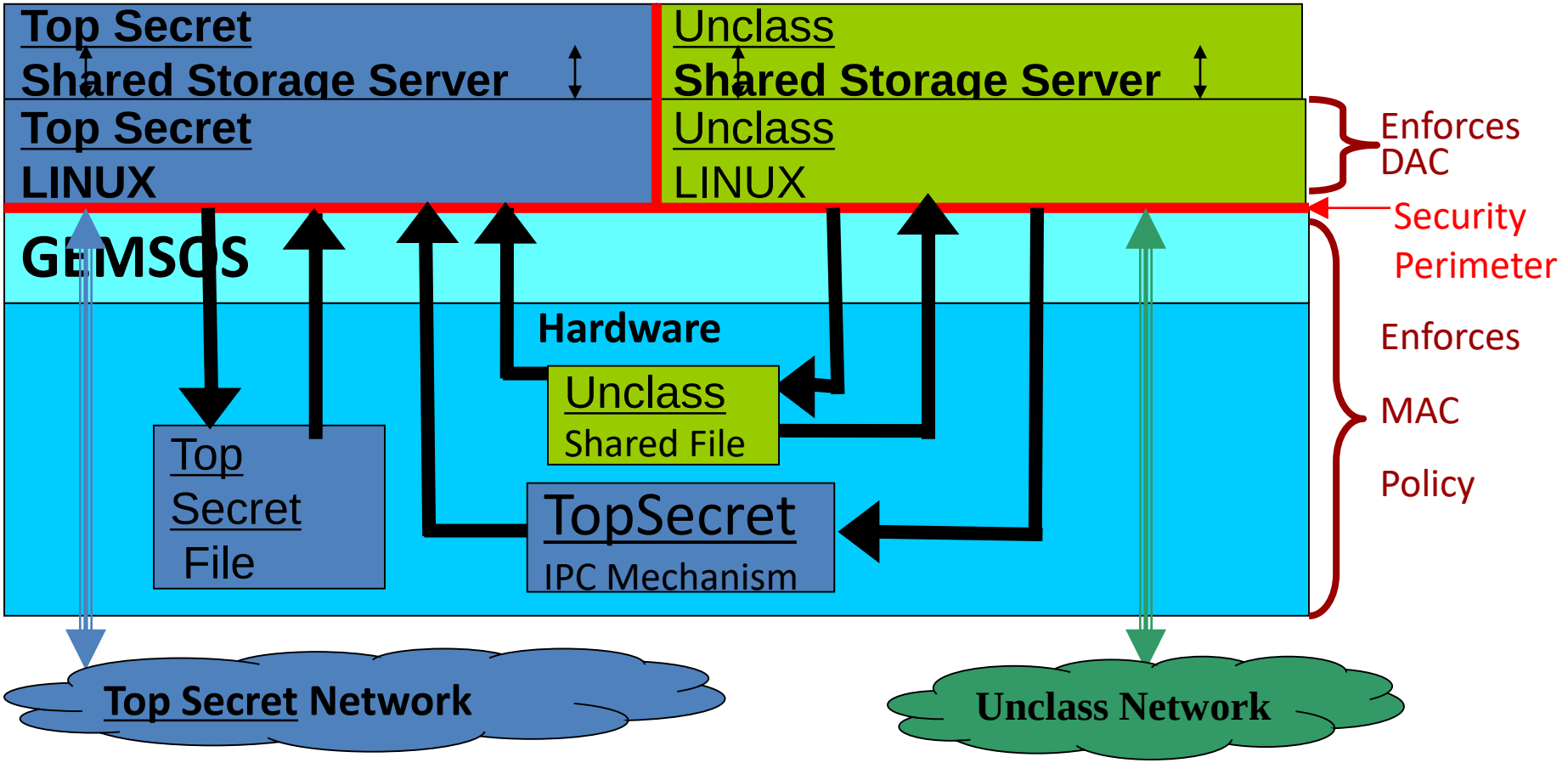
Basis to Consolidate Networks

- Foundation: Trusted Computing Base (TCB)
  - The totality of protection mechanisms
  - within a computer system
  - including hardware, firmware, and software
  - Responsible for enforcing a security policy
- The security perimeter is TCB boundary
  - Software within the TCB is trustworthy
  - Software outside the TCB is untrusted
- Mature technology -- 30 years experience
  - Derived from and used for commercial products
- Key choice is assurance: low to very high

TCB Subsets –"DAC on MAC"



Use Platform for Controlled **Sharing**



Motivation to Address Cloud Security

- Cloud storage flexible, cost-effective
- How to implement in multi-level environment?
  - Duplicate for each level? Loses advantages.
- Tempting target for attackers
  - Can increase privilege
- Want high-assurance, MLS solution
- Want cross-domain sharing (CDS)
  - Share resources are core cloud value proposition
  - Controlled sharing of up to date information
- Solution: MLS cloud network file service
  - Based on high-assurance, evaluated Class A1 TCB

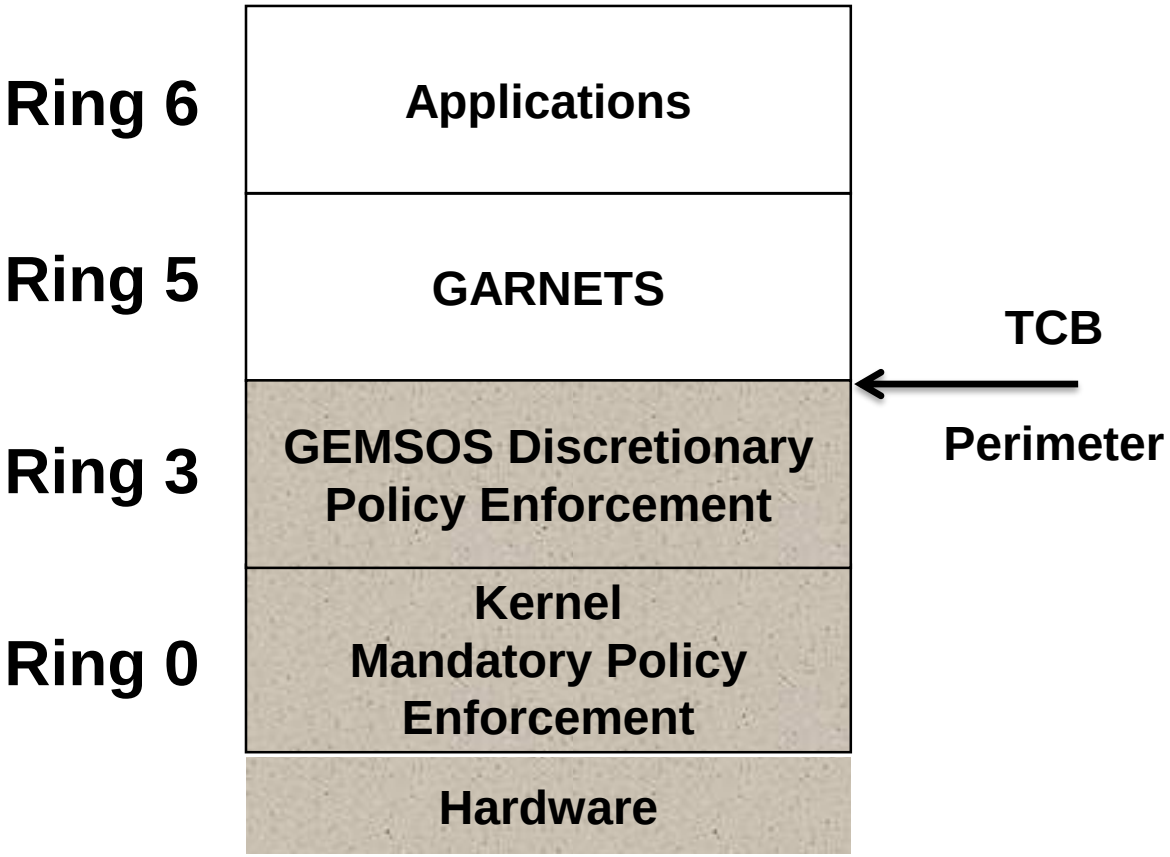
Cloud Storage Security Challenges

- Migrate storage from local security domain
  - Lose protection of “air gap” between domains
  - Dramatically expands opportunity for adversaries
- Common security approaches can’t work
  - Power of insidious software subversion attack tools
  - Exploding “cloud computing” amplifies impact & risk
- Proven reference validation mechanism (RVM)
  - Systematically codified as TCSEC “Class A1”
- Need architecture that leverages trusted RVM
  - Use verifiable (i.e., Class A1) trusted computing base
  - Want high cloud compatibility, e.g., standard NFS

Current Cloud Technology is Vulnerable

- Typically on commodity low-assurance platforms
- NetApp StorageGRID Systems
  - On Linux servers or VMware hypervisors
- OnApp Storage cloud
  - On various hypervisors, including Xen
- Amazon Elastic Compute Cloud
  - On Xen hypervisor to isolate guest virtual machines
- Xen is representative example of low-assurance
  - So-called TCB includes untrustworthy Linux, drivers
  - Largely unconstrained opportunities for subversion
- NSA said system on partition kernel too complex

Review of GARNETS Architecture

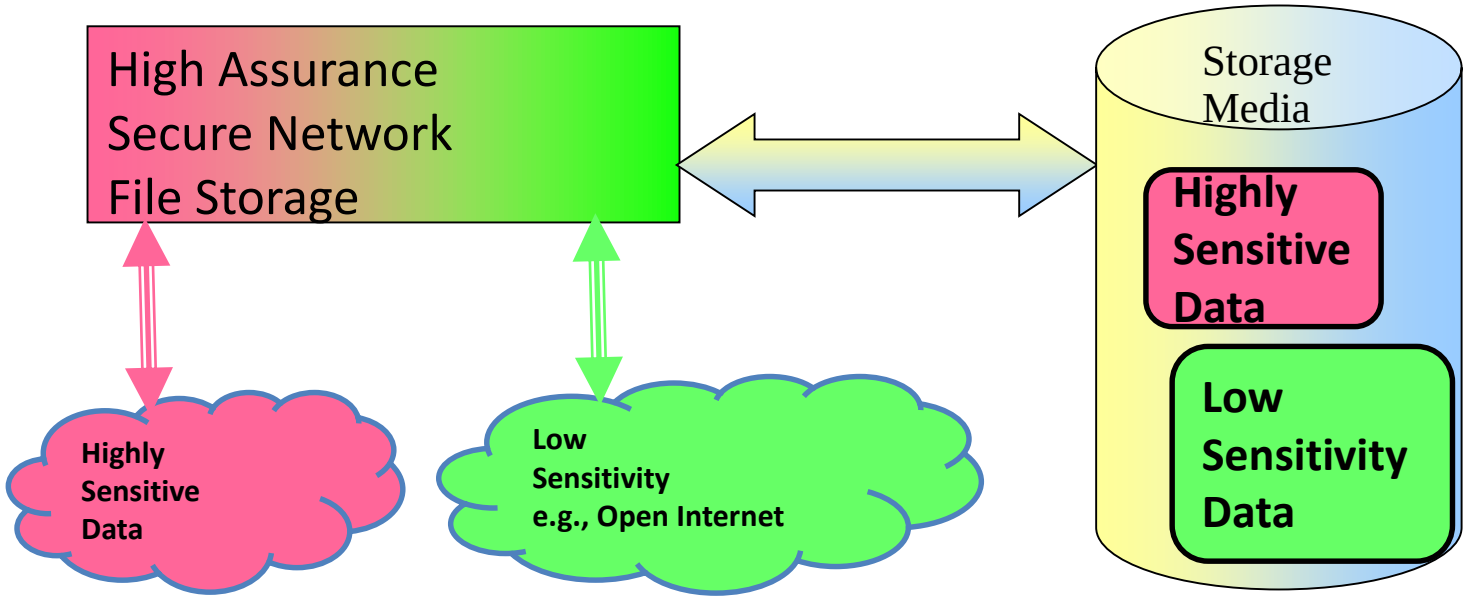


Build on MLS File System Foundation

- GARNETS MLS File is untrusted application
  - In a layer, in a separate ring, above the TCB
  - Creates file management that uses TCB objects
- MLS file system acts like standard file system
  - Spans multiple domains protected by the TCB
  - Creates one logical name space for all domains
- Run GARNETS instances at multiple domains
  - Means no massive copies needed for sharing
  - Can read both directories and files of lower domains
- For cloud storage need to add network interface
  - Multiple network interfaces for multiple domains

Target Secure Cloud Storage

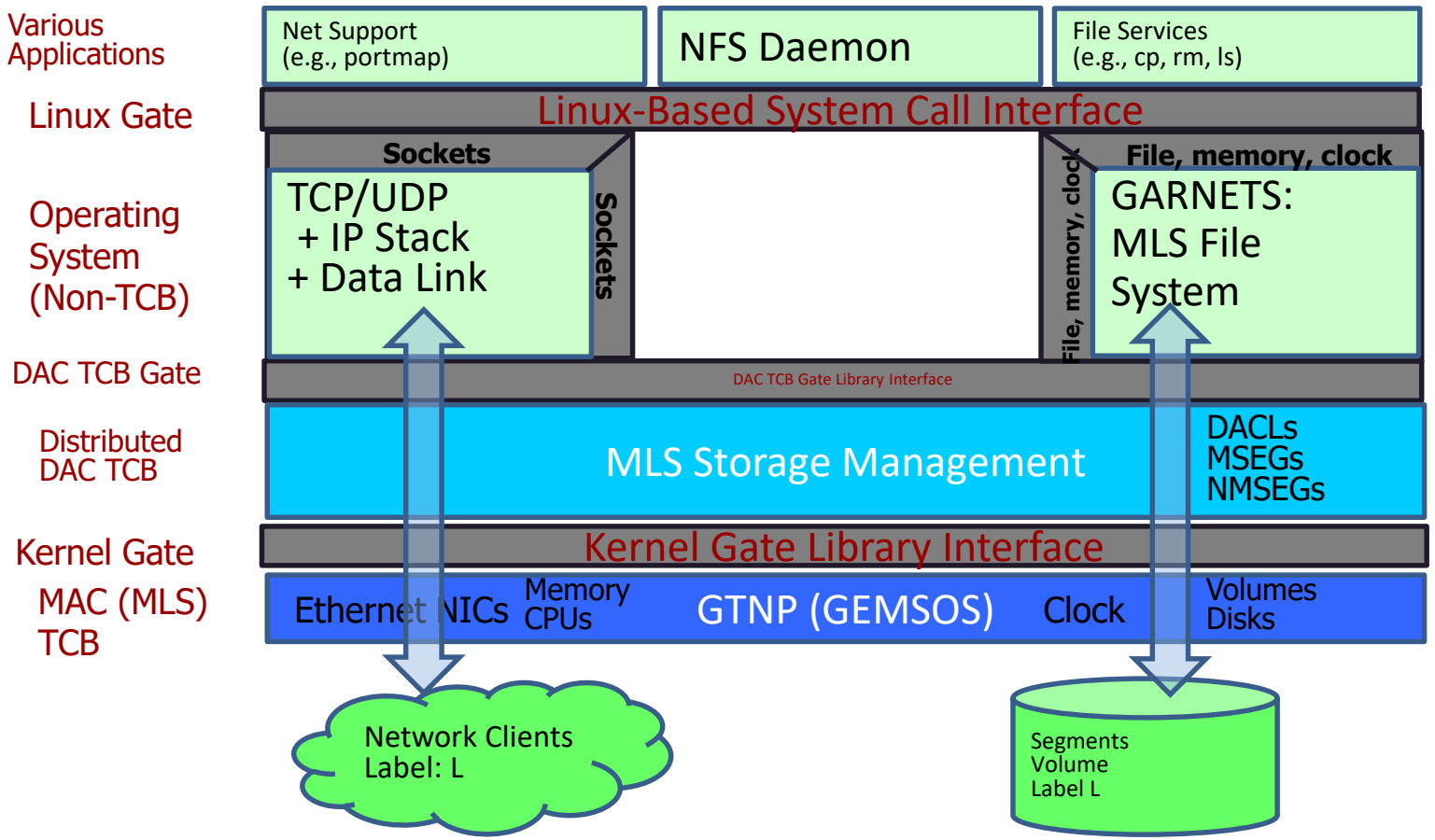
- Operates like standard network file storage
- BUT, verifiable security for
  - MAC separation of security domains



Run NFS on Top of MLS File System

- **System** challenge is a mostly compatible NFS
  - Accept the few intrinsic limitations, e.g., time last read
- Ease of porting NFS wants familiar OS interface
  - Requires creating “compatible OS” [GAS 10.7.2]
  - Demonstration chose POSIX style interface
  - Intended to support Linux source code compatibility
- In contrast, **GARNETS is not compatible**
  - Have to create a Linux system call interface
- Clients access files on server through NSF calls
  - Clients are untrusted
  - Any client using standard NSF protocol can access

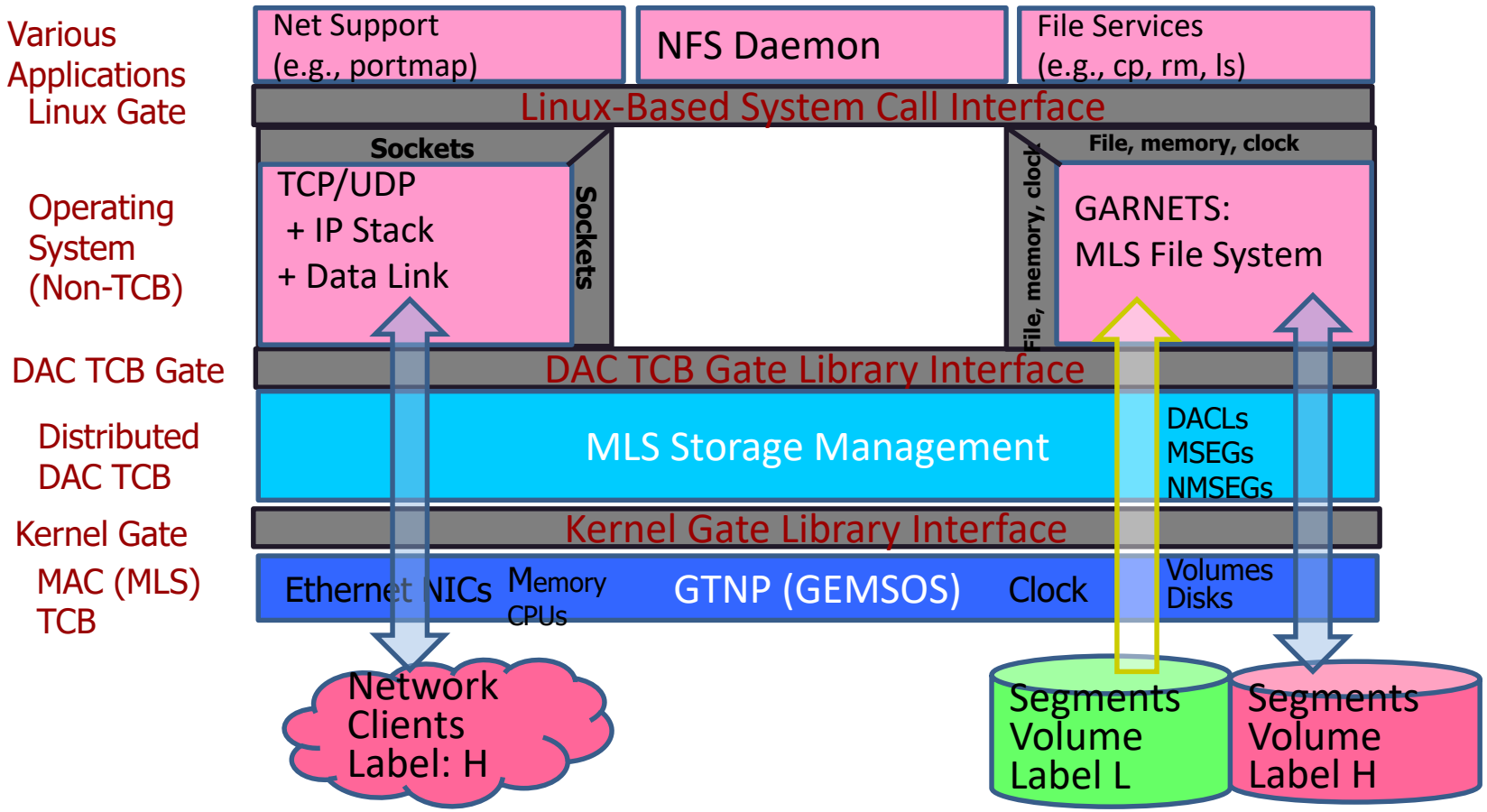
NSF Low Domain S/W Instantiation



For MLS Run Multiple NSF Instances

- Need a separate instance for each level
  - Is the only way NFS can be untrusted software
- Clouds often use virtual machine monitor (VMM)
  - Have noted low-assurance virtual machine problem
- Secure NFS demo leverages GEMSOS VMM
  - FER from NSA describes trusted MLS virtualization
  - Is NOT Type I virtualization, i.e,, cannot run binary
  - Each NFS instance runs in its own virtual machine
- MLS from TCB cannot be bypassed by VMM
  - Can be configured for typical isolation
  - In contrast to most VMMs, has controlled sharing

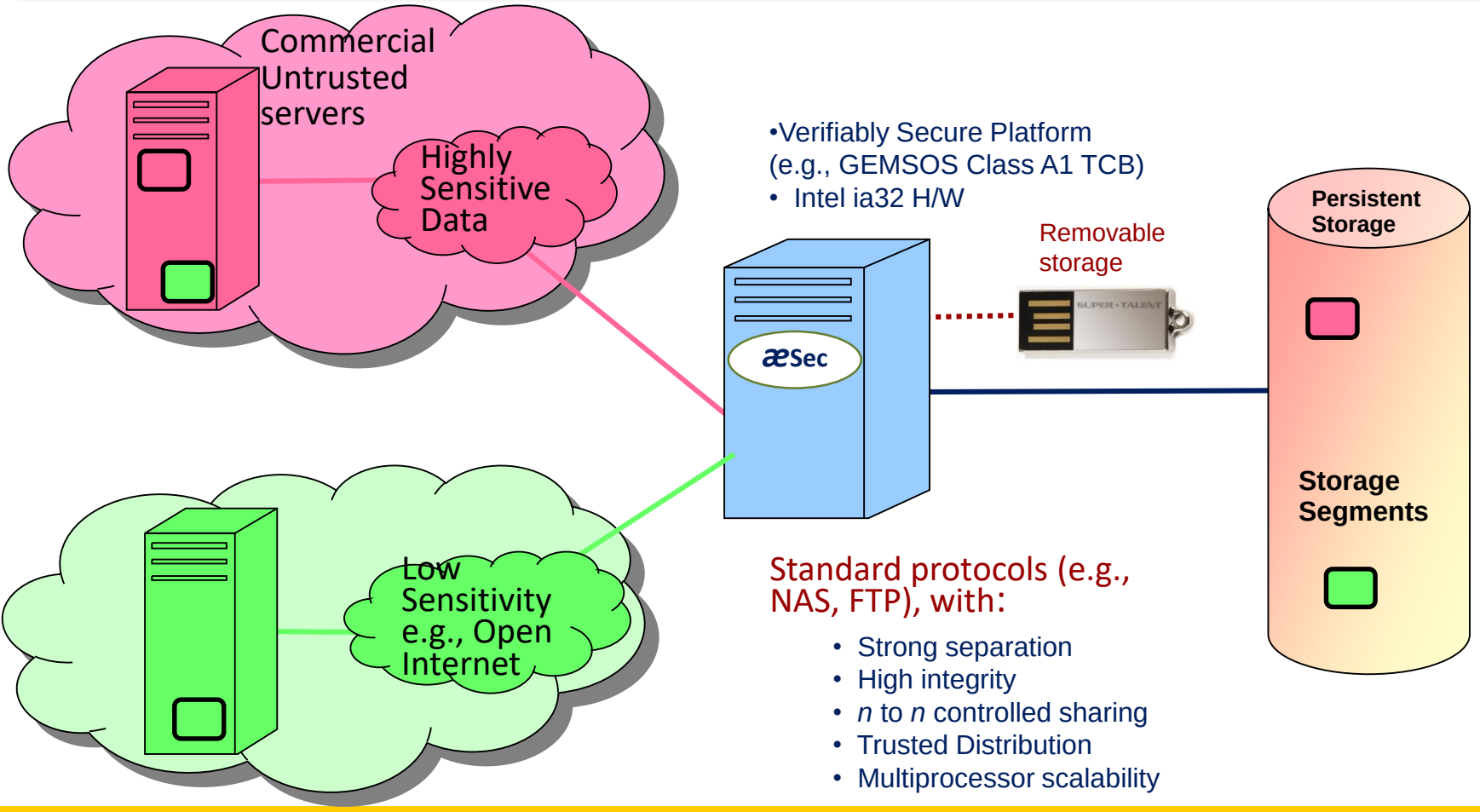
NSF High Domain S/W Instantiation



Can Support Extended Cloud Services

- Usually have metadata server for cloud services
  - Maps object names to actual locations in the cloud
- Single level servers in the network can access
  - May use FTP to transfer files to/from MLS file server
  - Applications NAS protocols can access it files
- BEWARE of sloppy system security engineering
  - Can't initiate information transfers **between** domains
  - TNI provide the network engineering framework
- Separate NIC per domain does not scale well
  - Next look at how to use a single hardware interface
  - Need equivalent of a TCSEC/TNI multilevel device

Security for Untrusted Servers



Cloud Controlled Sharing Summary

- Key is Mandatory Access Control (MAC)
  - Control isolation & sharing between security domains
- Defining properties: global and persistent
  - Control information flow (confidentiality)
    - Prevents malicious information exfiltration
  - Control contaminated modification (integrity)
  - Sound mathematical foundation
- Implement with distinct access class labels
  - Label domain of information with access class
  - User has authorized access classes, i.e., domains
- Supports MAC, viz., multilevel security (MLS)

INF523 SUPLEMENTAL MATERIAL (if time in semester)

Introduction to Crypto Seal Guards Case Study

Professor Clifford Neuman

Suplemental

Crypto Seal Guard Technology History

- Concept: label cryptographically sealed to data
- Conceived ~1980 for AF Korean Air Intelligence
- GEMSOS uses to meet TCSEC “Label Integrity”
  - Gemini Trusted Network Processor (GTNP) (1995)
  - Stored data (disk, tape) in Class A1 Evaluation
- GEMSOS uses for “Trusted Distribution”
  - Authoritative distribution media crypto sealed
  - Only sealed TCB software can be installed and run
- POC applied to packets exchanged by guards
  - Each guard is MLS – both a high and low interface

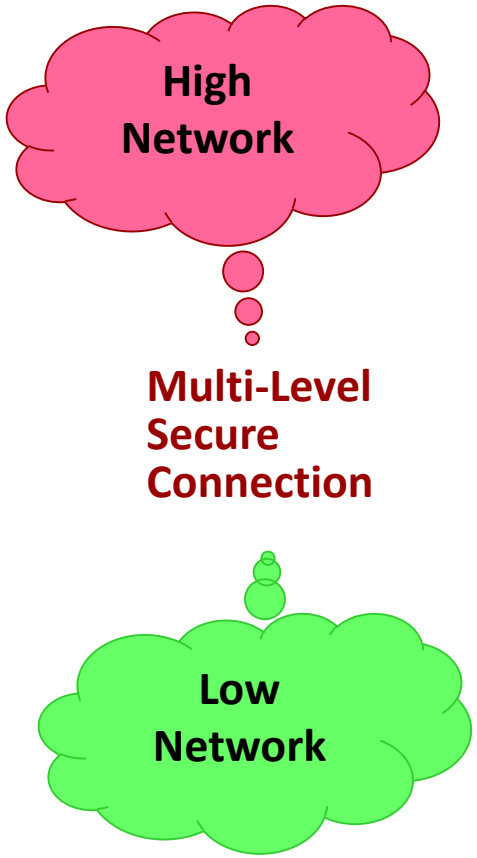
GEMSOS Support for Crypto Seals

- GEMSOS used crypto seals to meet Class A1
  - To meet Class A1 Label Integrity requirements
  - Integral to Trusted Recovery & Trusted Distribution
- GEMSOS publishes security services via APIs:
  - Data Sealing Device (and Cryptographic Services)
  - Key Management
  - Trusted Recovery & Distribution
- GemSeal uses GEMSOS APIs for crypto seals
  - Previously evaluated, stable, public interfaces
  - Minimal new trusted code
    - Generate seal
    - Validate integrity/authenticity of sealed packet & label

Overview of Seals for Shared Networks

- Proof of Concept (POC) demonstration done
  - Crypto seal release guards
  - Preproduction Class A1 MLS platform
- Access low network across system high network
  - Controlled interface protects system high data
  - Vertical data fusion with reduced footprint
- Benefits of crypto seal release guards
  - Swift implementation for MLS systems
  - Available core enabling technology for MLS
  - Rapid path to certification and accreditation (C&A)
  - Supports entire range of security domains
  - Mature deployed NSA Class A1 TCB and RAMP plan

Constraints to Access Lower Networks



- Any low connection = Multi-Level
  - Must be Multi-Level Secure (MLS)
  - Low/Medium assurance ineffective
    - Doesn't protect against subversion
    - Vulnerabilities unknown (unknowable)
- Isolation obstructs missions
  - Vertical data fusion
  - Tactical situational awareness
  - Timely access to open source data
  - Efficient utilization of resources

GemSeal POC Uses MLS Technology

- Class A1 TCB - GEMSOS™ security kernel
- Class A1 Ratings Maintenance Plan (RAMP)
- MLS aware crypto seal release guard
  - Gemini call it the GemSeal™ concept
- Technology Benefits
  - Minimize new trusted code development
  - Extensible to gamut of MLS capable systems
- High assurance resists subversion
  - Verifies absence of malicious code
  - Effective application of science
  - Key enabler for demanding accreditation, e. g., PL-5

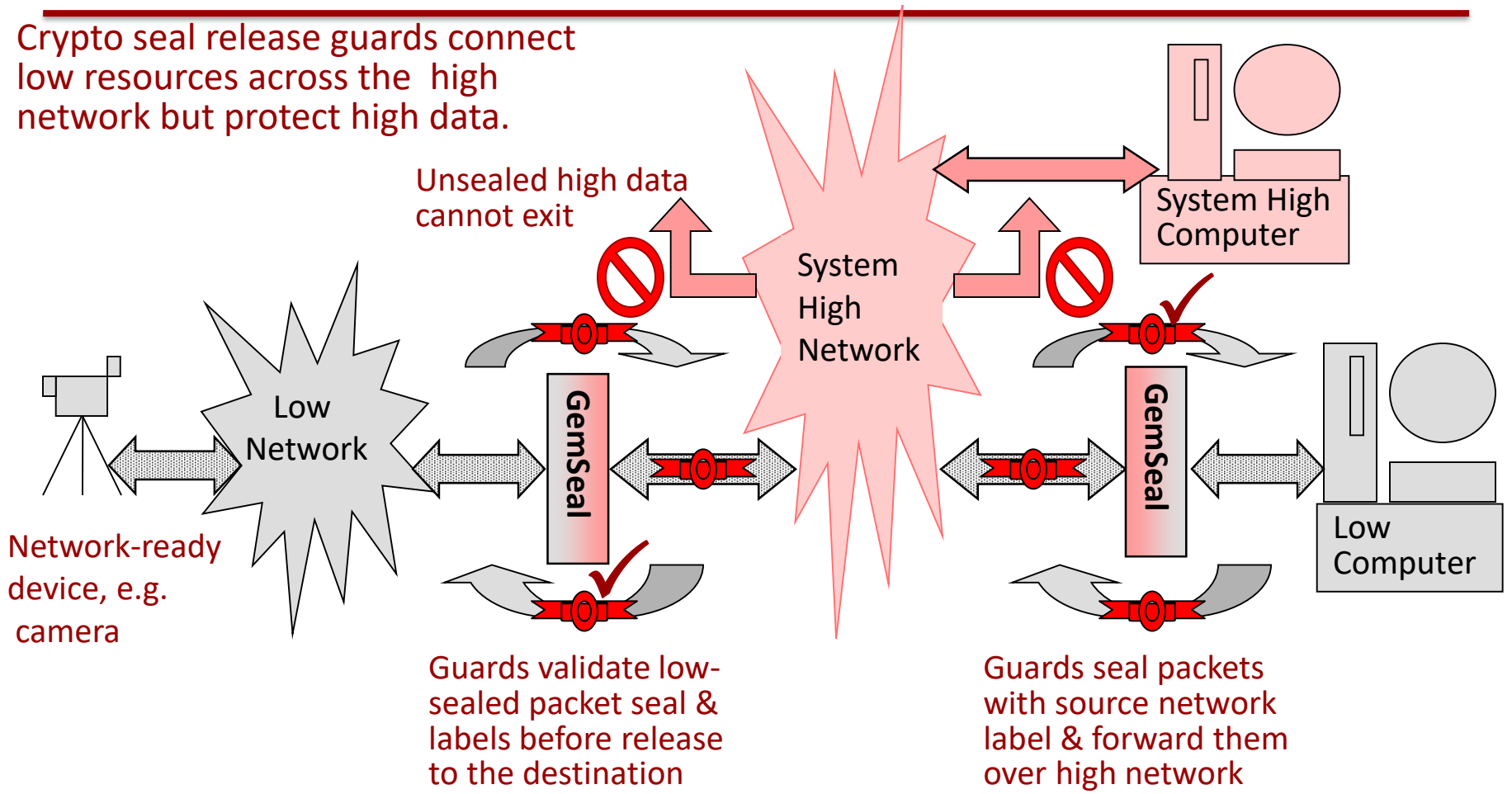
How Guard Seals a Packet

- Packet switched network design, e.g. Internet
- Concept involves multiple guards
  - POC has one or more “workstation” guards
  - POC has one or more “sensor” guards
  - Connected via a common system-high network
- Each guard has both high and low interfaces
- Sealing packets – forwarding from low to high
  - Bind source interface (low) label to each packet
  - Generate cryptographic seal of packet data + label
  - “Low-Sealed” packets include packet data + seal
  - “Low-Sealed” packets via high network interface

How Guard Releases a Packet

- Releasing packets – delivering from high to low
  - Release ONLY packets with seal-validated labels
  - Seal and label are removed before being released
- Only released to interfaces matching labels
  - Allows low data to traverse & exit high network
  - Concept supports multiple release guards
- Assures integrity of BOTH data AND label
  - Packet data is not altered
  - Source sensitivity label is authentic for this packet

AF Crypto Seal POC Demonstration



Summary of AF POC Demonstration

- Sensor (video) stream + command and control
  - Low sensor to low workstation connectivity
    - Uses existing high network infrastructure
  - Delivers access to low devices
    - For users lacking low network infrastructure
    - From controlled interface
- High network data is protected and unchanged
  - Guard validates low-sealed packets before release
  - Unsealed high packets cannot exit via guard

Summary of POC Configuration

- Two untrusted workstations with browsers
  - One ("Low") connected to "workstation guard"
  - One ("High") connected to high network
- One web server
  - Connected to low-side of the "sensor guard"
- A "high" Ethernet LAN
  - Connected to high-side of both guards
  - Also connected to second system high workstation
- The demonstration shows that
  - "Low" workstation can browse the "Low" web server
  - "High" workstation has no access to "Low" web server

Prior Evaluation Aids Accreditors

- Simplify job with reusable accreditation results
  - Certify or assess the platform once
  - Focus on system-specific additions & configurations
- GTNP provides evaluable TCB platform
  - Previously evaluated Class A1 for TNI M-Component
  - Class A1 RAMP in place and already proved useful
- Outside of the GTNP trusted computing base
  - Most of the application software will be untrusted
  - Only cryptographic seal operations need be trusted
    - Generate seals & release packets with validated seals
- Customer's certification and accreditation needs
  - The verifiably secure MLS TCB and

POC to Deployable System Summary

- Don't have to evaluate platform first
  - RAMP is already proven
  - No formal specification changes anticipated
- First: Evaluate and accredit the parts separately
  - Platform (very stable, accredit new hardware ports)
  - Crypto Seal implementation (as a trusted application)
  - Guard applications themselves evaluated separately
    - Supporting policies - audit, DAC, etc.
    - Untrusted application pieces, including network stack
    - Each protected by security kernel
- Last: refresh platform evaluation + accreditation
  - Because already successfully evaluated & accredited

Introduction to RECON Guard Security

- Review a classic and seminal paper
  - Cite: J. P. Anderson, "On the Feasibility of Connecting RECON to an External Network," Technical Report, J. P. Anderson Co., March 1981
  - Often cited for both databases and communications
- RECON is on-line interactive data base
  - Citations for both raw and finished intelligence reports
  - Also overnight batch and canned query capability
  - User may specify which file(s) to search
- Sponsor's security concerns are twofold
  - Subject to penetration from external network
  - Spillage of sensitive information from internal failure

Data Security Protection

- The data base contains two kinds of records
  - Those which can be widely distributed
  - Those whose distribution is restricted
    - Compartmented
    - Proprietary
    - Originator-controlled
- Operative aspects of the security problem
  - Commodity mainframe operating system
  - Must be prudently assumed that trapdoors exist
    - In some or much of application or operating systems
    - May be activated from externally connected users

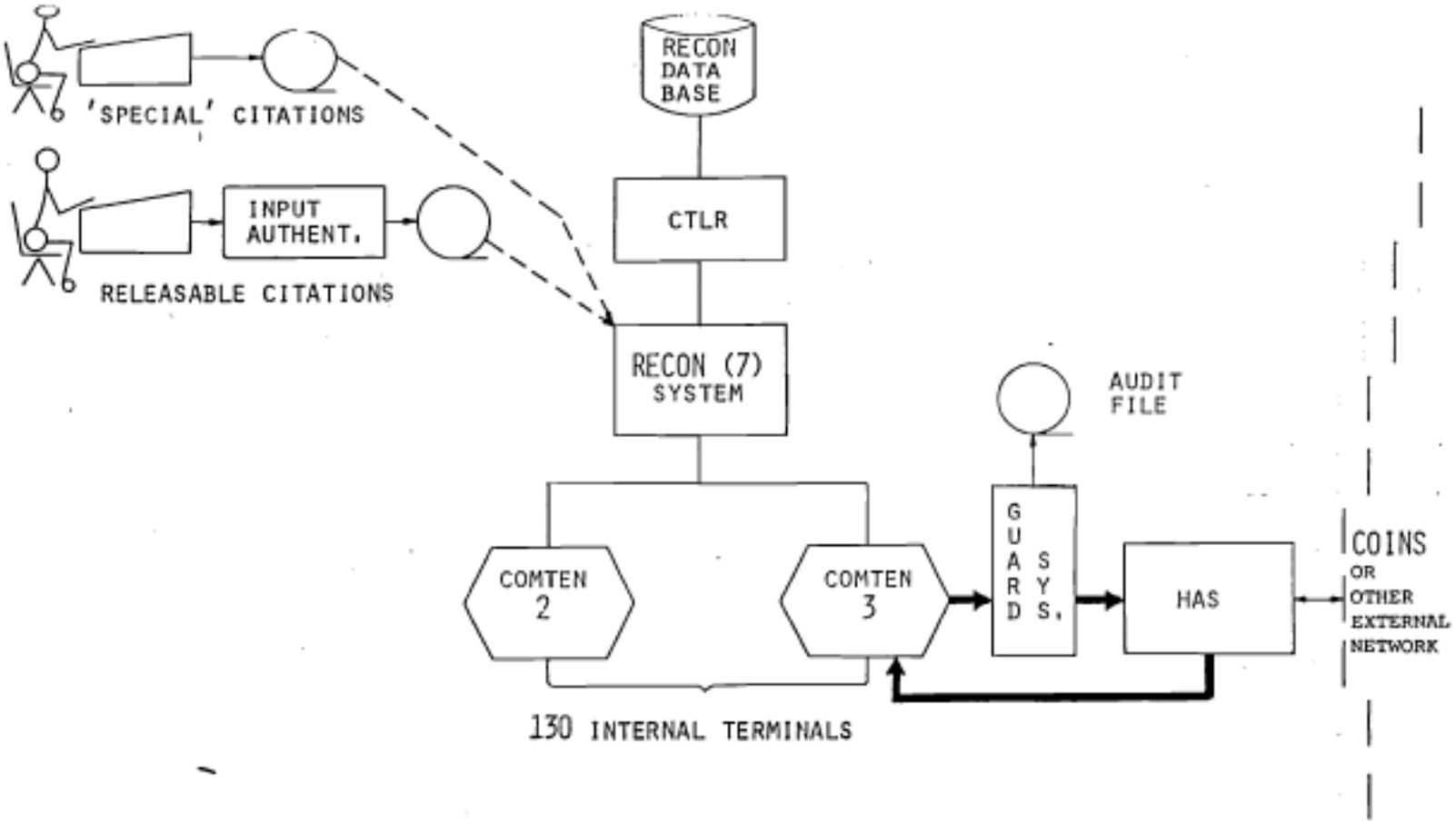
Previously Considered Approaches

- Put two kinds of records in separate systems
  - Make entries not deemed "special" accessible
  - Protected the sponsor's assets from penetration
  - Rejected because of the cost of duplicate facilities
- Multilevel secure operating system
  - In principle, would go far to defeat direct attacks
  - Could defeat placing trapdoors and Trojan Horses
  - Produce totally incompatible (with anything!) systems
  - Very expensive
- Filters added to RECON software to limit access
  - Nothing to control internal or external penetration

Guard Authenticate Releasability

- Is akin to the problem of "sanitizing" SCI
  - For release to activities without proper clearances
- Permit arbitrary queries by all users
  - Route query result of uncleared users to sanitizer
  - Sanitization officer would manually examine output
- Sanitization officer approach works in principle
  - Not practical solution because of excessive delays
  - Delays cascade to produce large response times
- Adapted as proposal to solve RECON problem
  - Adopt the idea of “sanitization” in a GUARD station
  - Automate the identification of releasable citations

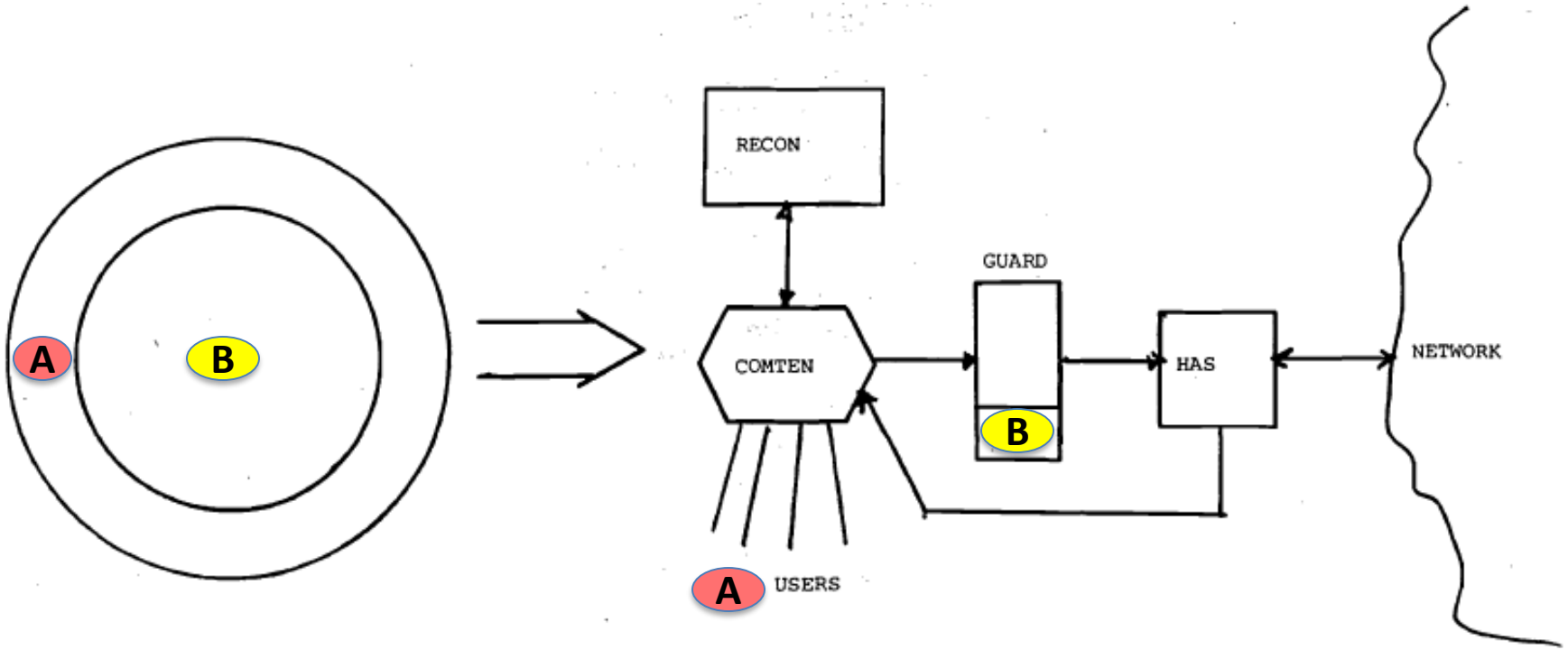
RECON Guard Technical Approach



RECON Guard Concept of Operation

- Consider all citations in one of two cases
  - Releasable even if not approved for "special" citations
  - Releasable only to approved individuals
- Each RECON entry designated by **originator**
  - Whether (or not) it is releasable to external users
- Create cryptographic checksum for releasable
  - Computed as the data enters the system
  - A function of the entire record
  - Computed by a special authentication device
  - Checksum is appended to the record and stays with it

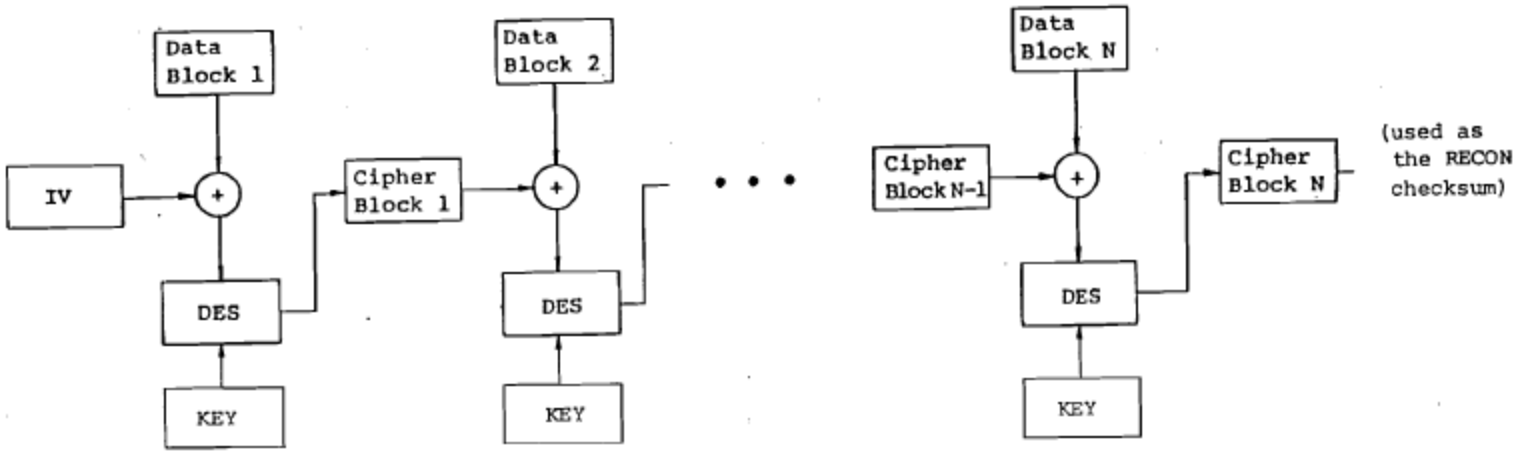
Representation of Basic Capability



Cryptographic Checksums Properties

- Principle need is assuring checksum not forged
  - Good modern crypto algorithm
  - Perform checksum functions outside RECON hosts
    - Separate entities to create and do Guard functions
- Secret key is known only to checksum devices
  - Key is never available within RECON system
  - Hardwired on board with crypto processor
  - Only method to forge is random guess (brute force)
- Key used for block-chained encipherment
  - Excellent error or tampering detection
  - Initial variable (IV) is used as half of the “secret”
  - A security “kernel” in devices control their operation

Process to Create Crypto Checksum



- ⊕ Exclusive OR
- The secret key(s) are the Initial Variable (IV) and the KEY.

BLOCK CHAINING

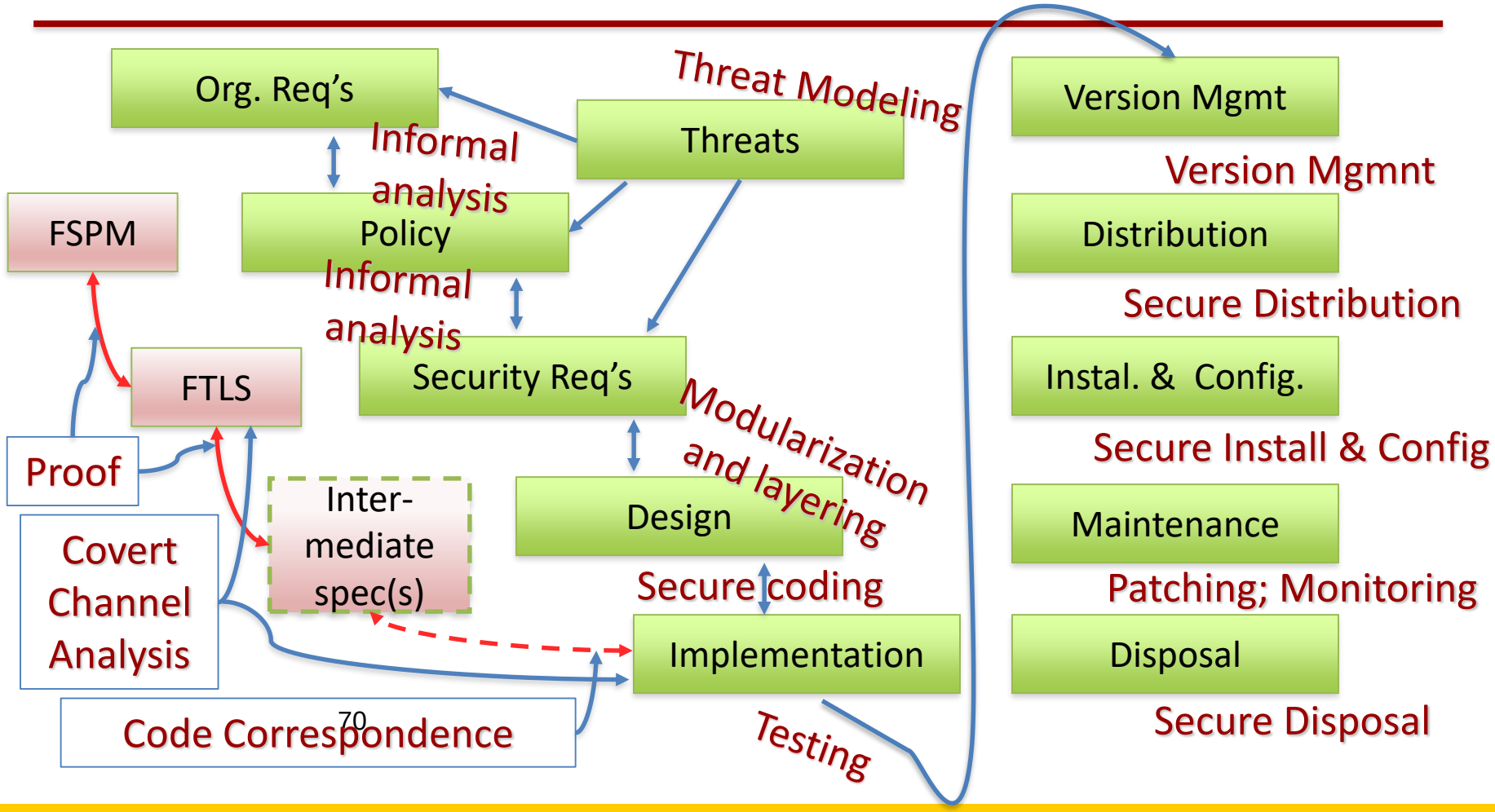
Security Properties of Guard

- No spill from RECON failure or compromise
- No manipulation of RECON will cause release
- Will “fail safe” if checksum detached from data
- Not protecting against manipulation of data
- Not preventing denial of service
- Guard system itself defends against its failure
  - Advanced design techniques, e.g., formal specs
  - Programs placed in read-only memory
  - Permits RECON to test guard message w/ loop back

Review for Final Exam

- The Final exam will be held on Wednesday December 9th from 11AM to 1PM.
- The final exam is comprehensive, in that I can ask you questions about material that was presented throughout the semester.
- However, the emphasis will be on material since the mid-term. Questions on material from before the mid-term will likely be asked in combination with material from later in the semester, i.e. in a way that will require you to discuss the relationships between multiple concepts.
- The exam is open book and open note. You will be allowed to use electronic devices, and you will edit your answers into a document in the same manner you did for the mid-term exam.

"Assurance Waterfall"



2018 Final – Q1Short Answer

a) Assume that we have evaluated assurance evidence for an operating system and found that we are satisfied with level of assurance provided. Explain how we can be sure that the software to be installed on a computer (including the operating system) is the same software and version for which those assurance arguments applied? (10 points)
b) Explain how MAC and DAC policies are enforced in GARNETS despite the code for GARNETS being untrusted. Please provide sufficient detail regarding access within the untrusted GARNETS components. (10 points)
c) Suggest reasons that a system like the Sun Network File System (NFS) running over linux was so vulnerable to the subversion (insertion of the artifice) in the NFS Subversion case study. Suggest characteristics of systems that might make them less vulnerable to this kind of attack. (10 points)
d) Explain the difference between a "side-channel" and a "covert timing channel". (10 points)

2018 Final – Q2 Longer Answer

a) Why is it that a system whose security model has been formally verified might still exhibit vulnerabilities? Consider the existence of covert channels as an example when answering this question. (15 points)
b) For which packets is the crypto-sealing of a label critical and why? Under the Crypto Seal Guard Architectures, which devices are considered trusted? (15 points)

2018 Final – Q3 Design

Assurance for next generation telecommunication systems.

You have been hired to perform an assurance evaluation for candidates for next generation telecommunications networks (5G). You are to focus your evaluation on the infrastructure side of such systems, but you are still to consider how changes to the application architecture on such devices can affect the assurance requirements for the infrastructure as well. It is critically important that in assessing assurance for such a system you consider the insider threat that might be present within infrastructure providers (services, or hardware or software vendors).

2018 Final – Q3 Continue

a) Create an attack-defense tree for the telecommunications network. Be sure to consider all the possible goals of an adversary. For defensive technologies in your AD tree you may consider changes to the end devices including calling and communications apps that might mitigate some of the attacks. Be sure to highlight potential threats that you might NOT be able to mitigate through changes to the end devices and describe the impacts that might result from a successful attack. (10 points)

b) Provide some examples of subversion to which the described system might be vulnerable. Discuss both the operational impact (what policies might be violated) as well as possible activation methods, and what the system might do (or not do) upon activation. (10 points)

c) Describe some of the requirements you would advise be placed on the hardware, software, development processes, distribution, operation and maintenance of the system to strengthen the assurance arguments that can be made about the system once it is deployed. (10 points)