Terminology

Vulnerability – A weakness in a system, program, procedure, or configuration that could allow an adversary to violate the intended policies of a system.

Threat – Tools or knowledge (capabilities) that can be used to exploit a vulnerability to violate the intended policies of a system.

Attack – An attempt to exploit a vulnerability to violate the intended policies of a system.

Compromise or intrusion – The successful actions that violate the intended policies of a system.

Terminology

Trusted – Parts of a system that we depend upon for the proper enforcement of policies, whether or not the code is free of vulnerabilities (almost all systems have vulnerabilities).

Trustworthy – Our belief that a system is free of vulnerabilities that could result in the violation of the relevant security policies.

Accreditation – A statement by a third party that a system or software has been found to be trustworthy with respect to a particular set of policies and for a particular operational environment.
Incidents and Breaches

Penetration – A successful attack (intrusion) that exploits a vulnerability in the code base of a system or its configuration. The result will often be to install a subversion.

Denial of Service – An attack that prevents authorized access to a resource, by destroying a target or overwhelming it with undesired requests.

Subversion - An intentional change to the code base or configuration of a system that alters the proper enforcement of policy. This includes the installation of backdoors and other control channels in violation of the policy relevant to the system.

Subversion vectors – the methods by which subversions are introduced into a system. Often the vectors take the form of malicious code.

More Terminology

Secure – A system is secure if it correctly enforces a correctly stated policy for a system. A system can only be secure with respect to a particular set of policies and under a set of stated assumptions. There is no system that is absolutely secure.

Trusted Computing Base – That part of a system which if compromised affects the security of the entire system. One often unstated assumption made with respect to a secure system is that the TCB is correctly implemented and has not been compromised.

Attack Surface – The accumulation of all parts of a system that are exposed to an adversary against which the adversary can try to find and exploit a vulnerability that will render the system insecure (i.e. violate the security policies of the system).

Categorizing Malicious Code

How propagated
• Trojan Horses
  – Embedded in useful program that others will want to run.
  – Covert secondary effect.
• Viruses (an specialization of a Trojan horse)
  – When program started will try to propagate itself.
• Worms
  – Exploits bugs to infect running programs.
  – Infection is immediate.

Classes of Malicious Code

The perceived effect
• Viruses
  – Propagation and payload
• Worms
  – Propagation and payload
• Spyware
  – Reports back to others
• Zombies or bots or botnets
  – Controllable from elsewhere
Activities of Malicious Code

- Modification of data
  - Propagation and payload
- Spying
  - Propagation and payload
- Advertising
  - Reports back to others or uses locally
- Propagation
  - Controllable from elsewhere
- Self Preservation
  - Covering their tracks
- Subversion

Defenses to Malicious Code

- Detection
  - Virus scanning
  - Intrusion Detection
- Least Privilege
  - Don't run as root
  - Separate users ID's
- Sandboxing
  - Limit what the program can do
- Backup
  - Keep something stable to recover

Trojan Horses

- A desirable documented effect
  - Is why people run a program
- A malicious payload
  - An “undocumented” activity that might be counter to the interests of the user.
- Examples: Some viruses, much spyware.
- Issues: how to get user to run program.
Trojan Horses

- Software that doesn't come from a reputable source may embed trojans.
- Program with same name as one commonly used inserted in search path.
- Depending on settings, visiting a web site or reading email may cause program to execute.

Viruses

- Resides within another program
  - Propagates itself to infect new programs (or new instances)
- May be an instance of Trojan Horse
  - Email requiring manual execution
  - Infected program becomes trojan

Viruses

- Early viruses used boot sector
  - Instruction for booting system
  - Modified to start virus then system.
  - Virus writes itself to boot sector of all media.
  - Propagates by shared disks.

Viruses

- Some viruses infect program
  - Same concept, on start program jumps to code for the virus.
  - Virus may propagate to other programs then jump back to host.
  - Virus may deliver payload.
Viruses can be Spread by Email

- Self propagating programs
  - Use mailbox and address book for likely targets.
  - Mail program to targeted addresses.
  - Forge sender to trick recipient to open program.
  - Exploit bugs to cause auto execution on remote site.
  - Trick users into opening attachments.

Viruses Phases

- Insertion Phase
  - How the virus propagates
- Execution phase
  - Virus performs other malicious action
  - Virus returns to host program

Analogy to Real Viruses

- Self propagating
- Requires a host program to replicate.
- Similar strategies
  - If deadly to start won’t spread very far – it kills the host.
  - If infects and propagates before causing damage, can go unnoticed until it is too late to react.

How Viruses Hide

- Encrypted in random key to hide signature.
- Polymorphic viruses changes the code on each infection.
- Some viruses cloak themselves by trapping system calls.
Macro Viruses

- Code is interpreted by common application such as word, excel, postscript interpreter, etc.
- May be virulent across architectures.

Worms

- Propagate across systems by exploiting vulnerabilities in programs already running.
  - Buffer overruns on network ports
  - Does not require user to “run” the worm, instead it seeks out vulnerable machines.
  - Often propagates server to server.
  - Can have very fast spread times.

Delayed Effect

- Malicious code may go undetected if effect is delayed until some external event.
  - A particular time
  - Some occurrence
  - An unlikely event used to trigger the logic.

Zombies/Bots

- Machines controlled remotely
  - Infected by virus, worm, or trojan
  - Can be contacted by master
  - May make calls out so control is possible even through firewall.
  - Often uses IRC for control.
Spyware

- Infected machine collect data
  - Keystroke monitoring
  - Screen scraping
  - History of URL’s visited
  - Scans disk for credit cards and password.
  - Allows remote access to data.
  - Sends data to third party.

Some Spyware Local

- Might not ship data, but just uses it
  - To pop up targeted ads
  - Spyware writer gets revenue for referring victim to merchant.
  - Might rewrite URL’s to steal commissions.
  - Superfish

Theory of Malicious Code

- Can not detect a virus by determining whether a program performs a particular activity.
  - Reduction from the Halting Problem
- But can apply heuristics

Defenses to Malicious Code

- Detection
  - Signature based
  - Activity based
- Prevention
  - Prevent most instances of memory used as both data and code
Defenses to Malicious Code

- Sandbox
  - Limits access of running program
  - So doesn’t have full access or even users access.
- Detection of modification
  - Signed executables
  - Tripwire or similar
- Statistical detection

Root Kits - Subversion

- Hide traces of infection or control
  - Intercept systems calls
  - Return false information that hides the malicious code.
  - Returns fall information to hide effect of malicious code.
  - Some root kits have countermeasures to attempts to detect the root kits.
  - Blue pill makes itself hyper-root

Best Detection is from the Outside

- Platform that is not infected
  - Look at network packets using external device.
  - Mount disks on safe machine and run detection on the safe machine.
  - Trusted computing can help, but still requires outside perspective

Economics of Malicious Code

- Controlled machines for sale
- “Protection” for sale
- Attack software for sale
- Stolen data for sale
- Intermediaries used to convert online balances to cash.
  - These are the pawns and the ones that are most easily caught
Economics of Malicious Code

Source: Yuri Namestnikov
securelist.com

Ongoing Attacks

- Norse network displays a real time attack map based on data from honeypots throughout the network.
- http://map.norscorp.com

Economics of Adware and Spam

- Might not ship data, but just uses it
  - To pop up targeted ads
  - Spyware writer gets revenue for referring victim to merchant.
  - Might rewrite URL’s to steal commissions.

Current Event

- Linux.Rex.1
  - Presented by Sachim Lohith
Linux.Rex.1 – what is it?

- Security researchers from the firm Dr. Web have discovered a new Linux Trojan dubbed Linux.Rex.1 that is capable of self-spreading through infected websites composing a peer-to-peer botnet.
- The threat was designed to infect web servers that use certain content management systems (CMS). [Eg. Drupal]
- The Linux.Rex.1 Trojan was written in the Go programming language and can perform a wide range of malicious activities, including sending out spam messages, launch DDoS attacks and of course spread itself over networks.

Linux.Rex.1 – how does it work?

- The malware has the ability to hack websites by exploiting a well-known SQL injection flaw.
- Linux.Rex.1 is a Trojan that can create such P2P botnets by implementing a protocol responsible for sharing data with other infected computers. Once the Trojan is launched, a computer that has been infected starts operating as one of this network’s nodes.
- The malware program receives directives over the HTTPS protocol and sends them to other botnet nodes, if necessary.
- When commanded by cybercriminals, it starts or stops a DDoS attack on a specified IP address.
- In addition, it sends out spam email messages to website owners, threatening them with DDoS attacks on their website; requesting payment of a Bitcoin ransom to avoid the attack.

Linux.Rex.1 – fix for the issue

- Website administrators are advised to update all the services to patch any known security vulnerabilities on their websites. This is expected to protect them from Rex Linux Trojan at the moment.
- Sources:
  - https://vms.drweb.com/virus/?i=8436299&lng=en

Recent News: Security of Autonomous Cars

Presented By Aparna Himmatramka
Recent News: Security of Autonomous Cars

What happened?
- Recently, Tesla Motors Faced SEC Probe Over Possible Security Law Breach
- It is accused of hiding vital information from its investors when a man was killed by one of its test cars in May
- NHTSA is investigating the crash to determine whether it reveals systemic issues tied to development of driverless cars or a possible malware that crashed the software used.
- The damages sustained by the Model S in the crash limited Tesla’s ability to recover data from it remotely

What are we dealing with?
- It’s no surprise that semi- and fully-autonomous transportation and the potential for driverless cars have become hot topics.
- According to Gartner, driverless vehicles will represent approx. 25% of the passenger vehicle population in use in mature markets by 2030.
- Ensuring driver safety from cyber threats has become a major development focus and challenge in the automotive and security industries.

Challenges
Cyber Attacks that are likely to be targeted at highly connected and autonomous cars:
- System stability and predictability
- Ransomware
- Spyware
- Privilege Escalation

Combat Strategies
Can be tackled by targeting four domains
1. Intra-vehicle communications:
   Ex. IPS, monitoring
2. External communications:
   Ex. Firewalls, IDS
3. Connectivity Infrastructure:
   Ex. Securing cellular networks
4. High-assurance access control systems and identity management
CSci530: Security Systems
Lecture 9 – October 21, 2016
Countermeasures

Dr. Clifford Neuman
University of Southern California
Information Sciences Institute

Intrusion Prevention

• A Marketing buzzword – for better than detection
• In Reality – General Good practices fall in this category
  – We will discuss network architectures
  – We will discuss Firewalls
  – This lecture is about this kind of intrusion prevention
• Intrusion detection (next week)
  – Term used for networks
  – But applies to host as well
    • Tripwire
    • Virus checkers
• Intrusion response (part now, part next week)
  – Evolving area
    • Anti-virus tools have a response component
    • Can be tied to policy tools

Architecture: A first step

• Understand your applications
  Information Flow:
  – What is to be protected
  – Against which threats
  – Who needs to access which apps
  – From where must they access it
• Do all this before you invest in the latest products that salespeople will say will solve your problems.

What is to be protected

• Is it the service or the data?
  – Data is protected by making it less available
  – Services are protected by making them more available (redundancy)
  – The hardest cases are when one needs both.
Classes of Data

- Decide on multiple data classes
  - Public data
  - Customer data
  - Corporate data
  - Highly sensitive data
    (not total ordering)
- These will appear in different parts of the network

Classes of Users

- Decide on classes of users
  - Based on the access needed to the different classes of data.
- You will architect your system and network to enforce policies at the boundaries of these classes.
  - You will place data to make the mapping as clean as possible.
- You will manage the flow of data

Example

- Where will you place your company’s public web server, so that you can be sure an attacker doesn’t hack your site and modify your front page?
- Where will you place your customer’s account records so that they can view them through the web?
  - How will you get updates to these servers?

Other Practices

- Run Minimal Systems
  - Don’t run services you don’t need
- Patch Management
  - Keep your systems up to date on the current patches
  - But don’t blindly install all patches right away either (possible subversion vector)
- Account management
  - Strong passwords, delete accounts when employees leave, etc.
- Don’t rely on passwords alone
Countermeasures - Continued

Dr. Clifford Neuman
University of Southern California
Information Sciences Institute

How to think of Firewalled Network

Crunchy on the outside.
Soft and chewy on the inside.
– Bellovin and Merrit

Firewalls

• Packet filters
  – Stateful packet filters
    • Common configuration
  • Application level gateways or Proxies
    – Common for corporate intranets
  • Host based software firewalls
    – Manage connection policy
  • Virtual Private Networks
    – Tunnels between networks
    – Relationship to IPsec

Packet Filter

• Most common form of firewall and what one normally thinks of
• Rules define what packets allowed through
  – Static rules allow packets on particular ports and to and from outside pairs of addresses.
  – Dynamic rules track destinations based on connections originating from inside.
  – Some just block inbound TCP SYN packets
Network Address Translation

- Many home firewalls today are NAT boxes
  - Single address visible on the outside
  - Private address space (net 10, 192.168) on the inside.
- Hides network structure, hosts on inside are not addressable.
  - Box maps external connections established from inside back to the private address space.
- Servers require persistent mapping and manual configuration.
  - Many protocols, including attacks, are designed to work through NAT boxes.

Application FW or Proxies

- No direct flow of packets
  - Instead, connect to proxy with application protocol.
  - Proxy makes similar request to the server on the outside.
- Advantage
  - Can’t hide attacks by disguising as different protocol.
  - But can still encapsulate attack.
- Disadvantage
  - Can’t do end to end encryption or security since packets must be interpreted by the proxy and recreated.

Host Based Firewalls

- Each host has its own firewall.
  - Closer to the data to be protected
  - Avoids the chewy on the inside problem in that you still have a boundary between each machine and even the local network.
- Problems
  - Harder to manage
  - Can be subverted by malicious applications.

Embedded and Distributed FW

- Embedded Firewalls
  - Implemented on hardware cards (firmware)
  - Better protected against subversion
  - Requires protected management component.
- Distributed Firewalls
  - Policy managed from central location
  - Flows managed by individual host, embedded, or appliance based firewalls.
  - Coordinated view of system policies.
Virtual Private Networks

- Extend perimeter of firewalled networks
  - Two networks connected
  - Encrypted channel between them
  - Packets in one zone tunneled to other and treated as originating within same perimeter.
- Extended network can be a single machine
  - VPN client tunnels packets
  - Gets address from VPN range
  - Packets encrypted in transit over open network

IPSec

- IP Security (IPsec) and the security features in IPv6 essentially move VPN support into the operating system and lower layers of the protocol stack.
- Security is host to host, or host to network, or network to network as with VPN’s
  - Actually, VPN’s are rarely used host to host, but if the network had a single host, then it is equivalent.

IPSec Goals

- Authentication of hosts
  - Verify the source of IP packets
  - Prevention of replays
- Verify integrity of packets
  - Through use of hashes and cryptography
- Ensure confidentiality of packets
  - Protect the payload

The IPSec Security Model

Secure

Insecure
IPSec Architecture

- IPSec provides security in three situations:
  - Host-to-host, host-to-gateway and gateway-to-gateway
- IPSec operates in two modes:
  - Transport mode (for end-to-end)
  - Tunnel mode (for VPN)
Various Packets

Original

Transport mode

Tunnel mode

Authentication Header (AH)

- Provides source authentication
  - Protects against source spoofing
- Provides data integrity
- Protects against replay attacks
  - Use monotonically increasing sequence numbers
  - Helps protect against DOS attacks
- NO protection for confidentiality!

AH Details

- Use 32-bit monotonically increasing sequence number to avoid replay attacks
- Use cryptographically strong hash algorithms to protect data integrity (96-bit)
  - Use symmetric key cryptography
  - HMAC-SHA-96, HMAC-MD5-96

AH Packet Details
Encapsulating Security Payload (ESP)

- Provides all that AH offers, and
- in addition provides data confidentiality
  - Uses symmetric key encryption

ESP Details

- Same as AH:
  - Use 32-bit sequence number to counter replaying attacks
  - Use integrity check algorithms
- Only in ESP:
  - Data confidentiality:
    - Uses symmetric key encryption algorithms to encrypt packets

ESP Packet Details

<table>
<thead>
<tr>
<th>IP header</th>
<th>Next header</th>
<th>Payload length</th>
<th>Reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Security Parameters Index (SPI)</td>
<td>Sequence Number</td>
<td>Initialization vector</td>
</tr>
<tr>
<td>TCP header</td>
<td>Data</td>
<td>Pad</td>
<td>Pad length</td>
</tr>
</tbody>
</table>

Authenticated

Encrypted TCP packet

Internet Key Exchange (IKE)

- Exchange and negotiate security policies
- Establish security sessions
  - Identified as Security Associations
- Key exchange
- Key management
- Can be used outside IPsec as well
IPsec/IKE Acronyms

• Security Association (SA)
  – Collection of attribute associated with a connection
  – Is asymmetric!
    • One SA for inbound traffic, another SA for outbound traffic
    • Similar to ciphersuites in SSL
• Security Association Database (SADB)
  – A database of SAs

How They Fit Together

IPsec/IKE Acronyms

• Security Parameter Index (SPI)
  – A unique index for each entry in the SADB
  – Identifies the SA associated with a packet
• Security Policy Database (SPD)
  – Store policies used to establish SAs

SPD and SADB Example
How It Works

• IKE operates in two phases
  – Phase 1: negotiate and establish an auxiliary end-to-end secure channel
    • Used by subsequent phase 2 negotiations
    • Only established once between two end points!
  – Phase 2: negotiate and establish custom secure channels
    • Occurs multiple times
  – Both phases use Diffie-Hellman key exchange to establish a shared key

IKE Phase 1

• Goal: to establish a secure channel between two end points
  – This channel provides basic security features:
    ▪ Source authentication
    ▪ Data integrity and data confidentiality
    ▪ Protection against replay attacks

IKE Phase 1

• Rationale: each application has different security requirements
• But they all need to negotiate policies and exchange keys!
• So, provide the basic security features and allow application to establish custom sessions

Examples

• Packets sent to BofA.com must be encrypted using AES with HMAC-SHA1 integrity check
• All packets sent to address cnn.com must be integrity checked with HMAC-SHA1 and no confidentiality is required.
Phase 1 Exchange

- Can operate in two modes:
  - Main mode
    - Six messages in three round trips
    - More options
  - Quick mode
    - Four messages in two round trips
    - Less options

Phase 1 (Main Mode)

Initiator

Responder

- Establish vocabulary for further communication

Establish secret key using Diffie-Hellman key exchange
Use nonces to prevent replay attacks

Signed hash of IDi (without Cert_req, just send the hash)
IPSec (Phase 1)

- Four different ways to authenticate (either mode)
  - Digital signature
  - Two forms of authentication with public key encryption
  - Pre-shared key
- NOTE: IKE does use public-key based cryptography for encryption

IPSec (Phase 2)

- Goal: to establish custom secure channels between two end points
  - End points are identified by <IP, port>:
    - e.g. <128.9.70.63, 8000>
  - Or by destination network:
    - e.g. All packets going to 128.124.100.0/24
  - Use the secure channel established in Phase 1 for communication

IPSec (Phase 2)

- Only one mode: Quick Mode
- Multiple quick mode exchanges can be multiplexed
- Generate SAs for two end points
- Can use secure channel established in phase 1

IPsec Policy

- Phase 1 policies are defined in terms of protection suites
- Each protection suite
  - Must contain the following:
    - Encryption algorithm
    - Hash algorithm
    - Authentication method
    - Diffie-Hellman Group
  - May optionally contain the following:
    - Lifetime
    - ...
IPSec Policy

• Phase 2 policies are defined in terms of proposals
• Each proposal:
  – May contain one or more of the following
    • AH sub-proposals
    • ESP sub-proposals
    • IPComp sub-proposals
    • Along with necessary attributes such as
      – Key length, life time, etc

Android Stagefright

• Presented by

  • Aditya Mandyamdevashikamani

Android Stagefright

• In July 2015, security company Zimperium announced that it had discovered a "unicorn" of a vulnerability inside the Android operating system.

  • What is stagefright?
    Stagefright is the group of software bugs that affect versions 2.2 ("Froyo") and newer of the Android operating system, allowing an attacker to perform arbitrary operations on the victim's device through remote code execution and privilege escalation.

• How the attack was accomplished?
  Stagefright attack can be achieved when a hacker can encode a piece of malware into an MP3 or Mp4 file and then disseminate it. Any Android user who clicks on the downloaded file will prompt the OS to automatically preview the song, infecting the device. And since virtually every build of Android OS currently available shares this same auto-preview feature, the exploit works nearly universally.

  • With the phone compromised, a worm using this vulnerability could read your contacts and send malicious MMS messages to your contacts, spreading like wildfire.

  • To check if your android phone is vulnerable install the Stagefright Detector App from Google Play which was developed by Zimperium
Android Stagefright

- How to defend against the Stagefright attack?
  Google promptly patched the flaws in the library, which parses video and other media files.
- Some tips to protect phones to protect against attack like stagefright are:
  a) Update your device: Keep your device updated to the latest version at all times.
  b) Disable Auto-fetching of MMS: You will need to disable this for both Hangout and regular messaging app
- Limitation of Android Operating system: While the fix for Stagefright has been out for months, Android users have to rely on carriers and device manufacturers to push the updates onto the devices.

IPSec Policy Example

- In English:
  - All traffic to 128.104.120.0/24 must be:
    - Use pre-hashed key authentication
    - DH group is MODP with 1024-bit modulus
    - Hash algorithm is HMAC-SHA (128 bit key)
    - Encryption using 3DES
- In IPSec:
  - [Auth=Pre-Hash; DH=MODP(1024-bit); HASH=HMAC-SHA; ENC=3DES]

End of IPSec

- New Topic
Announcements

• Numeric Mid-term Grades Posted
  – Letter grades later today.
  – Average was around 62.
• All proposals have been responded to
  – If you did not get feedback, see me or resend

MITRE Challenge IOT
The IoT’s massive interconnections of devices, or “things,” will lead to new efficiencies and capabilities. Imagine an intelligent hospital system that links patient monitoring devices with drug infusion pumps to prevent overdose and reduce false alarms. Or a smart city that schedules its maintenance work to minimize street blockages.

But amid this promise are enormous challenges. With so many potential points of entry that could be exploited, how do we protect our cars, homes, and factories? Organizations, from utility providers and hospitals, to corporations and the military, are wondering how to manage the IoT to ensure security and privacy within their different operating systems and environments.

Here are the details to get yourself onboard:
• The challenge will be hosted at https://register.mitre.org/challenge_iot_signup/
• The challenge begins Mid November and live for 8 weeks
Motivation:
• Chance to win $50000 cash prize
• Recognition and promotion for coming up with a game-changing solution

Attack Paths
• Many attacks today are staged from compromised machines.
  – Consider what this means for network perimeters, firewalls, and VPN’s.
• A host connected to your network via a VPN is an unsecured perimeter
  – So, you must manage the endpoint even if it is your employees home machine.
Defense in Depth

- One should apply multiple firewalls at different parts of a system.
  - These should be of different types.
- Consider also end to end approaches
  - Data architecture
  - Encryption
  - Authentication
  - Intrusion detection and response

Case Study: Trusted Network Interpretation

- Focusing on the network structure supporting MLS policies with high assurance:
  - Informatics Students will want to Read chapter 1

MLS Components

- MLS (separating users of different clearance levels as they access different classification levels)
  - Clearance level
  - Classification level
  - Security level (generic term for either clearance level or classification level)

How to Support BLP

- Separate Machines – System High
- Trusted High Assurance Systems
  - Capable of implementing MAC (e.g. BLP)
- What about networks?
Organization of the TNI

- Two parts, three appendices, a list of acronyms, a glossary, and a list of references.
- Part I presents TCSEC statements and detailed interpretations, which together constitute the requirements against which networks will be evaluated; and rationale for the network interpretation of the TCSEC. The TCSEC statement applies as modified by the Interpretation.
- Part II is a description of other Security Services not covered in the TCSEC interpretation which may be applicable to networks.
- Appendix A describes the evaluation of network components.
- Appendix B describes the rationale for network partitioning into individual components.

Objectives:

- Swiftly gain understanding of
  - NTCB Partitions
  - NTCB Subsets
- And how to apply these concepts to real networks

Network Interpretation of TCB

- Like a stand-alone system, the network as a whole possesses a single TCB, referred to as the NTCB, consisting of the totality of security-relevant portions of the network.
- Unlike a stand-alone system, design and evaluation of the network rests on understanding how security mechanisms are distributed and allocated to various components, in such a way that the security policy is supported reliably in spite of (1) the vulnerability of the communication paths and (2) the concurrent, asynchronous operation of the network components.

TCB Partitions and Subsets

- NTCB Partitions described in [TNI] A,B (appendices)
- NTCB Subsets described in “TCB Subsets for Incremental Evaluation” by Shockley, Schell
**NTCB Partitions**

- An NTCB that is distributed over a number of network components is referred to as partitioned, and that part of the NTCB residing in a given component is referred to as an NTCB partition. [TNI]pg13

**TCB Subset**

- Is an interface to an abstract mechanism that enforces some access control policy upon subjects which attempt to access objects under its control
  - Mediates every access
  - Tamperproof
  - Well enough constructed to be evaluated for correctness
- May have an internal interface to a smaller included mechanism which is also a TCB subset enforcing a less restrictive policy
- An NTCB Partition is a TCB subset which does not include a smaller TCB subset, thus is in direct control of a particular well-defined subset of objects of one processing component of a particular network system.

**Gaps  TNI and Current Practice**

- First, recognize that there are many "gaps" between current network security practice and the concepts espoused by the TNI (especially the pre-existence of a "NSAD", etc.), and prepare to do some "thought experiments that might transcend/leap over those gaps and apply some of the TNI concepts to analyze current practice in network security.
- Consider a small, almost trivial / degenerate case of policy to be enforced, examine the logical arrangement of protection measures required to enforce it.
- Then tweak this policy a little in several different directions (considered one at a time) and see how the logical arrangement of protection measures required to enforce the policy must evolve and change, and how these protections compare and contrast with common practice in network security today.

**Background**

- Some Security Policy Enforcement Tools & Components
  - Blacker
  - IPSEC
  - IPSO/CIPSO security options
  - HAIPE
  - Suite B
  - Encryption & Authentication at various protocol layers
  - Firewalls
  - Proxy servers
Evolution of Network Mandatory Policy Enforcement tools

- Link encryptors ➔ network encryptors
  - Internet Private Line Interface (IPLI) ~1983
    - (experimental hardware similar to IPSEC/HAIPE devices)
  - Blacker ~1984-1990
    - Provided for interconnection of MLS systems
    - Fielded production units but not available today
  - IP Security Options (IPSO) ~1988 (term re-used)
  - Trunk Encryptors ~1993
    - Passed "telecom signaling" through the encryptor
  - ATM Encryptors ~1994-1998
    - First "key agile" encryptors – different key for each ATM cell stream
  - IPSEC ~1998
  - HAPE ~2001
    - Provided for interconnection of Single Level Systems
  - MPLS – Multi-Protocol Label Switching
  - VLANs

General notion of a Multi-Level Network

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000001</td>
<td>(Reserved 4)</td>
</tr>
<tr>
<td>00111101</td>
<td>Top Secret</td>
</tr>
<tr>
<td>01011101</td>
<td>Secret</td>
</tr>
<tr>
<td>10011101</td>
<td>Confidential</td>
</tr>
<tr>
<td>01100101</td>
<td>(Reserved 3)</td>
</tr>
<tr>
<td>11001101</td>
<td>(Reserved 2)</td>
</tr>
<tr>
<td>10101101</td>
<td>Unclassified</td>
</tr>
<tr>
<td>11100011</td>
<td>(Reserved 1)</td>
</tr>
</tbody>
</table>

IPSO (RFC 1108)
Protection Authority Field

Table 2 - Protection Authority Bit Assignments

<table>
<thead>
<tr>
<th>BIT NUMBER</th>
<th>AUTHORITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>GENSER</td>
</tr>
<tr>
<td>1</td>
<td>SIOP-ESI</td>
</tr>
<tr>
<td>2</td>
<td>SCI</td>
</tr>
<tr>
<td>3</td>
<td>NSA</td>
</tr>
<tr>
<td>4</td>
<td>DOE</td>
</tr>
<tr>
<td>5, 6</td>
<td>Unassigned</td>
</tr>
<tr>
<td>7</td>
<td>Field Termination Indicator</td>
</tr>
</tbody>
</table>

HAIPE/HAIPIS

- High Assurance IP Encryption
- High Assurance IP Interface Specification
- (basically a high assurance version of IPSEC)
  - Reconfigurable algorithms
  - Specified open algorithms (suite b)
  - Some restrictions, some enhancements over IPSEC

IP Security Options for labeling datagrams

IP Security Options for labeling datagrams (single level networks)
Summary

- IPSO provides a way to add sensitivity labels to datagrams
- IPSEC (HAIPE) provides a way to protect the integrity and confidentiality of these datagrams between
  - Single level networks
  - Multi-level networks (requires MLS computing nodes to complete the protection within the interconnected networks)

Summary (2)

- Blacker, IPSEC/HAIPE, IPSO are attempts to insert MAC enforcement at the boundaries between network enclaves
- Examples of DAC enforcement include:
  - Kerberos and other network authentication systems such as Remote Authentication Dial-In User Service (RADIUS)
  - Network File System remote user permissions
  - Unix file permissions (user, group, world, etc.)
- Firewalls, bastion hosts, proxy servers, etc. are attempts to insert Application level policy enforcement at the boundaries between network enclaves

Notional Interconnection of Enclaves

Blacker vs IPSEC/HAIPE

- Which requirements for Multi-level Network encryption were satisfied by “Blacker” that are not satisfied by current IPSEC/HAIPE devices?
  - Blacker handles “ranges” of security labels according to a multi-level policy
  - Current IPSEC/HAIPE devices handle only “single level” channels (do not actually process sensitivity labels)
**Protecting the Inside**

- Firewalls are better at protecting inward threats.
  - But they can prevent connections to restricted outside locations.
  - Application proxies can do filtering for allowed outside destinations.
  - Still need to protect against malicious code.
- Standalone (i.e. not host based) firewalls provide stronger self protection.

---

**Intrusion Types**

- External attacks
  - Password cracks, port scans, packet spoofing, DOS attacks
- Internal attacks
  - Masqueraders, Misuse of privileges

---

**Attack Stages**

- Intelligence gathering
  - Attacker observes the system to determine vulnerabilities (e.g., port scans)
- Planning
  - Decide what resource to attack and how
- Attack execution
  - Carry out the plan
- Hiding
  - Cover traces of attack
- Preparation for future attacks
  - Install backdoors for future entry points
Intrusion Detection

- Intrusion detection is the problem of identifying unauthorized use, misuse, and abuse of computer systems by both system insiders and external penetrators.

- Why Is IDS Necessary?

IDS types

- Detection Method
  - Knowledge-based (signature-based) vs behavior-based (anomaly-based)

- Behavior on detection
  - passive vs. reactive

- Deployment
  - network-based, host-based and application-based

Components of ID systems

- Collectors
  - Gather raw data

- Director
  - Reduces incoming traffic and finds relationships

- Notifier
  - Accepts data from director and takes appropriate action

Advanced IDS models

- Distributed Detection
  - Combining host and network monitoring (DIDS)
  - Autonomous agents (Crosbie and Spafford)
Intrusion Response

• Intrusion Prevention  
  – (marketing buzzword)
• Intrusion Response  
  – How to react when an intrusion is detected

Possible Responses

– Notify administrator
– System or network lockdown
– Place attacker in controlled environment
– Slow the system for offending processes
– Kill the process

Phase of Response (Bishop)

– Preparation
– Identification
– Containment
– Eradication
– Recovery
– Follow up

PREPARATION

• Generate baseline for system
  – Checksums of binaries
    • For use by systems like tripwire
• Develop procedures to follow
• Maintain backups
IDENTIFICATION

• This is the role of the ID system
  – Detect attack
  – Characterize attack
  – Try to assess motives of attack
  – Determine what has been affected

CONTAINMENT

• Passive monitoring
  – To learn intent of attacker
  – Learn new attack modes so one can defend against them later
• Constraining access
  – Locking down system
  – Closing connections
  – Blocking at firewall, or closer to source
• Combination
  – Constrain activities, but don’t let attacker know one is doing so (Honeypots, Jail).

ERADICATION

• Prevent attack or effects of attack from recurring.
  – Locking down system (also in containment phase)
  – Blocking connections at firewall
  – Isolate potential targets

RECOVERY

• Restore system to safe state
  – Check all software for backdoors
  – Recover data from backup
  – Reinstall but don’t get re-infected before patches applied.
FOLLOWUP

• Take action against attacker.
  – Find origin of attack
• Notify other affected parties
  – Some of this occurs in earlier phases as well
• Assess what went wrong and correct procedures.
• Find buggy software that was exploited and fix

Limitations of Monolithic ID

• Single point of failure
• Limited access to data sources
• Only one perspective on transactions
• Some attacks are inherently distributed
  – Smurf
  – DDoS
• Conclusion: “Complete solutions” aren’t

Sharing Information

• Benefits
  – Increased robustness
  – More information for all components
  – Broader perspective on attacks
  – Capture distributed attacks
• Risks
  – Eavesdroppers, compromised components
  – In part – resolved cryptographically

Sharing Intrusion Information

• Defining appropriate level of expression
  – Efficiency
  – Expressivity
  – Specificity
CIDF

- Common Intrusion Detection Framework
  - Collaborative work of DARPA-funded projects in late 1990s
  - Task: Define language, protocols to exchange information about attacks and responses

CISL

- Common Intrusion Specification Language
  - Conveys information about attacks using ordinary English words
  - E.g., User joe obtains root access on demon.example.com at 2003 Jun 12 14:15 PDT

CISL

- Problem: Parsing English is hard
  - S-expressions (Rivest)
    - Lisp-like grouping using parentheses
    - Simplest examples: (name value) pairs
      (Username ‘joe’)
      (Hostname ‘demon.example.com’)
      (Date ‘2003 Jun 12 14:15 PDT’)
      (Action obtainRootAccess)

CISL

- Problems with simple pairs
  - Confusion about roles played by entities
    - Is joe an attacker, an observer, or a victim?
    - Is demon.example.com the source or the target of the attack?
  - Inability to express compound events
    - Can’t distinguish attackers in multiple stages
  - Group objects into GIDOs
CISL: Roles

- Clarifies roles identified by descriptors
  (Attacker
   (Username ‘joe’)
   (Hostname ‘carton.example.com’)
   (UserID 501)
  )
  (Target
   (Hostname ‘demon.example.com’)
  )

CISL: Verbs

- Permit generic description of actions
  (Compromise
   (Attacker …)
   (Observer
    (Date ‘2003 Jun 12 14:15 PDT’)
    (ProgramName ‘GrIDSDetector’)
   )
   (Target …)
  )

Lessons from CISL

- Lessons from testing, standardization efforts
  – Heavyweight
  – Not ambiguous, but too many ways to say the same thing
  – Mismatch between what CISL can say and what detectors/analytics can reliably know

Worm and DDOS Detection

- Difficulty is distinguishing attacks from the background.
  – Zero Day Worms
  – DDoS
- Discussion of techniques
  – Honeynets, network telescopes
  – Look for correlation of activity
Reacting to Attacks

- How to Respond to Ongoing Attack
  - Disable attacks in one's own space
  - Possibly observe activities
  - Beware of rules that protect the privacy of the attacker (yes, really)
  - Document, and establish chain of custody.
- Do not retaliate
  - May be wrong about source of attack.
  - May cause more harm than attack itself.
  - Creates new way to mount attack
  - Exploits the human element

Virus Checking

- Signature based
  - Looks for known indicators in files
  - Real-time checking causes files to be scanned as they are brought over to computer (web pages, email messages) or before execution.
  - On server and client
- Activity based
  - Related to firewalls, if look for communication
  - Alert before writing to boot sector, etc.
- Defenses beyond just checking
  - Don’t run as root or admin

The Human is the Weak Point

- Low bandwidth used between computer and human.
  - User can read, but unable to process crypto in head.
  - Needs system as its proxy
  - This creates vulnerability.
- Users don’t understand system
  - Often trust what is displayed
  - Basis for phishing
The Human is the Weak Point(2)

- Humans make mistakes
  - Configure system incorrectly
- Humans can be compromised
  - Bribes
  - Social Engineering
- Programmers often don’t consider the limitations of users when designing systems.

Some Attacks

- Social Engineering
  - Phishing – in many forms
- Mis-configuration
- Carelessness
- Malicious insiders
- Bugs in software

Addressing the Limitations

- Personal Proxies
  - Smartcards or devices
- User interface improvements
  - Software can highlight things that it thinks are odd.
- Delegate management
  - Users can rely on better trained entities to manage their systems.
- Try not to get in the way of the users legitimate activities
  - Or they will disable security mechanisms.

Social Engineering

- Arun Viswanathan provided me with some slides on social engineering that we wrote based on the book “The Art of Deception” by Kevin Mitnik.
  - In the next 6 slides, I present material provided by Arun.
- Social Engineering attacks rely on human tendency to trust, fooling users that might otherwise follow good practices to do things that they would not otherwise do.
Total Security / not quite

• Consider the statement that the only secure computer is one that is turned off and/or disconnected from the network.
• The social engineering attack against such systems is to convince someone to turn it on and plug it back into the network.

Six Tendencies

• Robert B. Cialdini summarized six tendencies of human nature in the February 2001 issue of Scientific American.
• These tendencies are used in social engineering to obtain assistance from unsuspecting employees.

Six Tendencies

• People tend to comply with requests from those in authority.
  – Claims by attacker that they are from the IT department or the audit department.
• People tend to comply with requests from those who they like.
  – Attackers learns interests of employee and strikes up a discussion.

Six Tendencies

• People tend to follow requests if they get something of value.
  – Subject asked to install software to get a free gift.
• People tend to follow requests to abide by public commitments.
  – Asked to abide by security policy & demonstrate compliance by disclosing their password is secure:
    ▪ Were talking about cybersecurity today and how safe peoples passwords are, what is your online password?
Six Tendencies

- People tend to follow group norms.
  - Attacker mentions names of others who have “complied” with the request, and will the subject comply as well.
- People tend to follow requests under time commitment.
  - First 10 callers get some benefit.

Steps of Social Engineering

- Conduct research
  - Get information from public records, company phone books, company web site, checking the trash.
- Developing rapport with subject
  - Use information from research phase. Cite common acquaintances, why the subjects help is important.
- Exploiting trust
  - Asking subject to take an action. Manipulate subject to contact attacker (e.g. phishing).
- Utilize information obtained from attack
  - Repeating the cycle.

Phishing

- A website (or other form of interaction) where the user believes they are communicating with an entity they trust but are actually communicating with the attacker.
  - Could be a phone call claiming to be from your bank.
  - Could be a paper letter that appears genuine.
    - Most commonly it is a link in an email message
    - Or a search result
    - Or a link on a web page
    - Mistyped domain name (typosquatting)
      - Visible text of link might show name or even URL of legitimate site, but target of link is different
      - Sometimes subtly different letter or character or prefix

Other Redirections

- Man in the Middle
  - At free hotspots
  - Through hacks such as superfish
- Domain name hijacking
  - Simple malware
  - Cache poisoning
    - (in a couple of slides)
Context Sensitive Certificate Verification and Specific Password Warnings

- Work out of University of Pittsburgh
- Changes dialogue for accepting signatures by unknown CAs.
- Changes dialogue to prompt user about situation where password are sent unprotected.
- Does reduce man in the middle attacks and phishing
  - By preventing easy acceptance of CA certs
  - Requires specific action to retrieve cert
  - Would users find a way around this?

CSci530: Computer Security Systems
Lecture 11 – 4 November 2016
Review of mid-term exam

Dr. Clifford Neuman
University of Southern California
Information Sciences Institute

2016 Mid-term Exam Question 1

1. Code sent as SMS Text Message
2. Smart Card
3. Password
4. Kerberos
5. Shibboleth
6. Secure-ID
7. Fingerprint

- a. Something you know
- b. Something you Have
- c. Something about you
- d. Relies on possession of encryption key
- e. Vulnerable to replay attack (in any form)
- f. Vulnerable to man in the middle attack

2016 Mid-term Exam Question 2

a. What are the strengths and weaknesses of mandatory access policies as compared with discretionary policies? Why are both kinds of policies important? List at least two examples each of mandatory access control policy models and representations or implementations of discretionary policies. (10 points)

b. Provide an example of how existing network, mobile, cloud, or commercial services today could benefit from technologies supporting mandatory access controls. What would such policies mean for the users of today’s computer systems (what might be different from what they expect of their computers today). (10 points)

c. Explain the steps used by a web browser to exchange an encryption key for use in encrypting the channel of an SSL (or TLS) protected session and to ascertain that it is communicating with the server to which it is trying to connect. (15 points)

d. List as many ways as you can think of for an attacker to obtain access to the information that has been communicated through the SSL (or TLS) connection described in part c. (5 points)
2016 Mid-term Exam Question 3

Cryptographic File Access Control (40 points)

• There has been a lot of recent talk about how state sponsored cyber-criminals are trying to influence our upcoming election through attacks on voter registration systems, possibly voting tabulation systems, and the campaign and personal systems of various candidates for office. You have been hired as an intern at the federal election commission to present proposed measures that can be taken to protect the integrity of our voting systems. Because you have not yet completed CSci530 there are many things that you haven’t been taught yet about security, but your report is to focus on recommendations based on topics from the first half of the course, in particular on the application of cryptography, key management, identity management, and policy and access control.

• In the discussion that you provide you may think long term and consider technologies that require deployment of infrastructure not yet universally available so long as you state your assumption and why you believe that infrastructure will be deployed (separate from the voting systems you are discussing). You should not assume that we are looking for a completely on-line voting system. Instead, it will still be the case that the majority of voters will go to a polling place to cast their votes which will then be tallied locally, with results communicated to higher level centers (e.g. local polls to county registrars to the state level who will communicate results nationally and publish the results). Vote by mail (which can possibly include email) can be supported for absentee voters and others.

• There will be tradeoffs to consider, balancing the ability to ascertain the integrity of the results (that all votes were cast by authorized voters and correctly counted), secrecy of ballots (that the votes of an individual cannot be determined by others), and convenience and functionality (e.g. ease with which absentee votes can be cast, early voting, etc). No solution is perfect. I am more interested in your reasoning and the effectiveness of your solutions for specific situations.
The Domain Name System Evolution

- Historically the host file is located on the local machine
  - E.g., c:\windows\system32\drivers\etc\hosts
  - Need to be maintained and updated by an administrator
- Maintaining the hosts files for all Internet domain names and sub domains is not feasible
  - Hence the birth of a distributed database that is called DNS
  - A service run by a myriad of organizations, ISP’s and Internet authorities (ICANN)
  - To facilitate the mapping of URL to IP addresses
  - (rewrite)

Disadvantages of centralized DNS

- Single point of failure: poor reliability
- RTT is long for a distant centralized database
- Difficult to maintain and update
- Poor performance

DNS (Domain Name System)

- Internet host’s and router interfaces:
  - IPv4 address (32 bit): used for addressing datagrams
  - “name”, e.g., www.cnn.com, used by humans
- Translating from www.cnn.com to 64.236.16.20 in order to deliver a datagram

Domain Name System (DNS)

- An infrastructure for translating between IP addresses and a name
- DNS is an application-layer protocol used by hosts to query DNS servers
- A distributed database implemented in a hierarchy of name servers
- Distributed DNS is more reliable and faster than centralized DNS
- DNS is defined in RFC 1034 and RFC 1035

Distributed, Hierarchical Database for DNS

- A host needs the IP address of www.cnn.com and queries the local DNS server
  - If the local DNS server does not have the RR (resource record) for www.cnn.com in its cache, then the local DNS server
    - Queries a root server to find .com DNS server
    - Queries .com DNS server to get cnn.com DNS server
    - Queries cnn.com DNS server to get IP address for www.cnn.com
DNS: Root name servers

- Contacted by local DNS name server that cannot resolve a name
- A local DNS system is pre-configured with known addresses of the root servers in a file called root hints
  - This file must be updated periodically by the local administrator
- Root name servers:
  - The root name servers know which servers are responsible for the top-level domains (TLD), such as .edu
  - Each top-level domain (such as .edu) has its own set of servers
    - TLD servers in turn delegate to the name servers responsible for individual domain names (such as ns.mit.edu)
  - Two of the root servers are managed by US
- 13 organizations manage the root DNS servers
- The locations of the root servers are listed in the following table: (source www.root-servers.org)

TLD Types

- Country-code TLDs (ccTLDs)
  - There are more than 240 ccTLDs. Examples include .uk, .in, and .jp.
- Sponsored generic TLDs (gTLDs)—specialized domains with a sponsor representing a community of interest
  - These TLDs include .edu, .gov, .int, .mil, .aero, .coop, and .museum.
- Unsponsored generic TLDs (gTLDs)—domains without a sponsoring organization
  - The list of unsponsored gTLDs includes .com, .net, .org, .biz, .info, .name, and .pro.

TLD name servers

- .edu TLD
  - a.gtld-servers.net. 192.5.6.30 2001:503:a83e:0:0:0:2:30
  - c.gtld-servers.net. 192.26.92.30
  - d.gtld-servers.net. 192.26.92.30

- .com TLD
  - a.gtld-servers.net. 192.5.6.30 2001:503:a83e:0:0:0:2:30
  - b.gtld-servers.net. 192.33.14.30 2001:503:231d:0:0:0:2:30
  - c.gtld-servers.net. 192.26.92.30
  - d.gtld-servers.net. 192.26.92.30

- .org
  - a.org.afilias-nst.info. 199.19.56.1 2001:500:e:0:0:0:0:1
  - b.org.afilias-nst.org. 199.19.54.1 2001:500:c:0:0:0:0:1

- .fr TLD
  - a.nic.fr. 192.93.0.129 2001:660:3005:3::1:1

Source: http://iana.org/domains/root/db/
A host at auburn.edu wants an IP address for mit.edu.

- If the RR is not in the cache of the local DNS server, then the local DNS server will carry out the recursive query for the local client.

### Recursive query

- Arrow 1: auburn.edu performs the recursive query for the host.
- edu TLD DNS server
- Root DNS server

### Iterated (non-recursive) query

- Arrows 2, 3, and 4: E.g., root DNS replies to dns.auburn.edu to contact .edu TLD DNS.
- ns.mit.edu plays the role of authoritative name server.
- Root DNS says, "I do not know the IP address, but ask .edu TLD DNS server who will help you out!"

---

### Authoritative DNS servers (1)

- An authoritative name server provides an authoritative answer to a DNS query.
- Primary DNS server, also known as a master server, contains the original set of data.
- Secondary or slave name server contains copies usually obtained from synchronization with the master server.
- It is recommended that three servers be provided for most organizations (in RFC 2182).
- The IP addresses of authoritative DNS servers are maintained by ICANN and kept in the TLD DNS servers.
- All authoritative name servers are initially treated equally.
- Resolvers often measure the performance of the various servers, choose the server with the best performance for most queries.

---

### Authoritative DNS servers (2)

- Each ISP, company, university, organization has at least one authoritative name server.
- When a host makes a DNS query, the query is sent to its local authoritative DNS server or a recursive server.
- DNS server acts as proxy, forwards query into the DNS hierarchy: recursive query.
- The DNS information for one domain name is stored as a resource record(s) (RR's).
- A DNS zone is a portion of the global Domain Name System (DNS) namespace for which administrative responsibility has been delegated.

---

### DNS Resolver

- Inside a host, a process called DNS resolver obtains the mapping from name to IP address.
- RESOLVERS are programs that obtain information from name servers in response to client requests.
- A cache preserves a mapping for a certain amount of time.
- A DNS resolver can be running inside a computer that is:
  - A client computer
  - A web server, mail server, etc.
  - A DNS server.
- Resolvers must have access to at least one name server.
  - Use that name server's information to answer a query directly.
  - Perform the query using referrals to other name servers.
Caching Name Server/Recursive Name Server

- The terms recursive server and caching server are often used synonymously as in BIND (Berkeley Internet Name Domain).
- Typical implementation:
  - Move the resolver function out of the local machine and into a name server which supports recursive queries.
  - Produces an easy method of providing domain service in a PC which lacks the resources to perform the resolver function.
  - Centralizes the cache for a whole local network.
  - Each PC must have a list of name server addresses that will perform the recursive requests.
  - A router that connects a home network to DSL/cable modem provides caching/recursive name service.
  - E.g. 192.168.x.1 is the LAN interface that provides caching DNS.
  - Some vendors refer to it as DNS relay.

Forwarder and Firewall

- A caching name server does not necessarily perform the complete recursive lookup itself.
  - Instead, it can forward some or all of the queries that it cannot find from its cache to another caching name server, commonly referred to as a forwarder.
  - Caching servers unable to pass packets through the firewall would forward to the server that can traverse the firewall, and that server would query the Internet DNS servers on the internal server's behalf.

DNS Hierarchy in a zone

- PA: primary authoritative
- SA: secondary authoritative
- R: recursive
- Internet
- Internal network
- Subnets in area 1
- Subnets in area 2
- Subnets in area 3

DNS Reliability and Security

- Mitigate risk by duplicating the DNS function of an organization onto two servers, one primary and one secondary, so that if the one goes down, DNS is still available.
- Split DNS:
  - Only let local users that are part of the domain query a private DNS server (or stealth DNS server) to ensure confidentiality of resource naming conventions and other sensitive information only available to internal hosts.
  - Set up a public DNS server outside the firewall or in a DMZ for outsiders to learn the IP addresses for a web server or mail server.
  - From a security perspective, only publish a mapping to the public domain when it is necessary.
  - Split Horizon is normally used to describe a DNS server that will give different responses based on the source IP address, or some other characteristic.
**DNS protocol and port**

- DNS uses UDP port 53 for lookups and transfers
  - This port must be opened to the VPN through the firewall if a remote user needs to use the internal/private DNS for lookups
  - Note: this decision will be defined in the planning phase and should be used with a virtual private network (VPN)
- TCP port 53 comes into play only when the response data size exceeds 512 bytes, or for tasks such as a zone transfer
  - Zone transfer: from primary authoritative to secondary authoritative server

**DNS: caching and updating records (1)**

- Once a name server learns mapping, it caches mapping
  - Cache entries timeout after some time
  - TLD servers are typically cached in a local name server (such as dns.auburn.edu)
    - A local name server can be an authoritative or recursive name server
    - Thus root and TLD name servers not often visited
  - Update/notify mechanisms in RFC 2136
- Caching/Recursive Servers:
  - If a server is going to provide caching services, then it must provide recursive queries
  - Recursive queries need access to the root servers which is provided via the 'type hint' statement: root servers' IP addresses are in a file
    - A caching server using BIND will typically have a named.conf file which includes type hint, file 'root.servers';

**DNS: caching and updating records (2)**

- For Windows DNS server:
  - A root hints file, Cache.dns, that is stored in the systemroot\System32\Dns folder on the server computer
  - The contents of this file are preloaded into server memory when the service is started and contain pointer information to root servers for the DNS namespace
- Caching/Recursive Servers:
  - To create a caching-only name server, install the DNS service but do not configure any zones
  - Configure client computer's TCP/IP properties to use the caching-only DNS server for name resolution
  - Provide DNS name resolution for computers in the same domain
  - Cache the result to answer potential future queries within a certain expiration (time-to-live) period
  - Servers with Recursion Access Control provide control over which hosts are permitted to use DNS recursive lookups

**RR Format**

- (www.auburn.edu, 131.204.2.251, A)
- Type (16 bit): A

\[
\text{RR format: } (\text{name, [pref.], value, type, [ttl]})
\]

- www.auburn.edu IN A 131.204.2.251
- www IN A 131.204.2.251
- Class 16 bit: IN which identifies a protocol family or instance of a protocol is the Internet system

\[
\text{RR format: } \text{name [ttl] [Class] Type [pref.] value}
\]
DNS resource record (RR) Type (1)

RR format: (name, [pref.], value, type, [ttl])

- Type=A
  - Name is host’s name
  - Value is IP address
- Type=NS
  - Name is domain name (e.g. auburn.edu)
  - Value is name of authoritative name server for this domain (e.g. dns.auburn.edu)
- Type=MX
  - Name is domain name (e.g. auburn.edu)
  - Value is name of authoritative name server for this domain (e.g. aumail.duc.auburn.edu)
  - A preference value is designated for each mail server if there are multiple MX RR’s in a domain

DNS resource record (RR) Type (2)

RR format: (name, [pref.], value, type, [ttl])

- Type=CNAME
  - Name (such as www.ibm.com) is alias name for “canonical” (real) name
  - Value is canonical name (such as servereast.backup2.ibm.com)
  - www.ibm.com (name) is really servereast.backup2.ibm.com (value)
- Type=AAAA
  - IPv6 host address (AAAA) resource record
  - Maps a DNS domain name to an Internet Protocol (IP) version 6 128-bit address
- TTL: time to live in cache
  - 32 bit integer for the number of seconds

RR example

- Company x has a webserver w.x.com with IP address 131.204.2.5
- General public uses www.x.com or x.com to access the website
- One Type A RR for the host (w.x.com, 131.204.2.5, A, 3 hours)
- One Type CNAME RR for aliasing (www.x.com, w.x.com, CNAME, 3 hours)
- One Type CNAME RR for aliasing (x.com, w.x.com, CNAME, 3 hours)
How to make usc.edu work

e. edu TLD server contains
- 3 NS RR’s
- 3 A RR’s

ns.usc.edu: authoritative name server contains
- 1 A RR for web server
- 1 MX RR for mail server
- 1 A RR for mail server

Example: usc.edu

USC registers name usc.edu at DNS registrar (e.g., Educause)

Registrar inserts six RRs (3 authoritative DNS servers) into edu TLD server:
- (usc.edu, ns.usc.edu, NS)
- (ns.usc.edu, 128.125.253.172, A)
- (usc.edu, ns1.usc.edu, NS)
- (ns1.usc.edu, 128.125.252.173, A)
- (usc.edu, ns2.usc.edu, NS)
- (ns2.usc.edu, 128.125.251.174, A)

Inserting records into DNS (2)

USC created additional RRs in authoritative server ns.usc.edu inside usc.edu domain
- Type A RR for www.usc.edu
  - (www.auburn.edu, 128.125.253.146, A)
- Type MX (mail exchange) RR for usc.edu
  - (usc.edu, mailserver.usc.edu, MX)
- When multiple mail servers are available, each server has a type MX RR and one type A RR
- Preference = 10 as default value for mail server
- When multiple MX RR available, mail server with smallest Preference value is used
- No CNAME RR for multiple mail servers

For a small organization, one can put RRs for authoritative servers in an ISP’s DNS server hosting the web and mail service for the organization

Zone file

Zone file for auburn.edu

$ORIGIN auburn.edu
$TTL 2d
@ IN SOA dns.auburn.edu. master.auburn.edu. (2003080800 ; serial number
12h ; refresh (h: hour)
15m ; retry (m: minute)
3w ; expiry (w: week)
3h ; minimum)

IN NS ns.auburn.edu.
IN MX 10 mailserver.auburn.edu.
ns IN A 128.125.253.172
webserver IN A 128.125.253.146
mailserver IN A 128.125.253.78
www IN CNAME webserver.auburn.edu.

$ORIGIN webserver.auburn.edu.

IN CNAME webserver.auburn.edu.
SOA: Start of Authority (1)

- The first Resource Record must be the SOA (Start of Authority) record
- The SOA defines global parameters for the zone (domain)
- There is only one SOA record allowed in a zone file
- The master.auburn.edu. represents the email address of master@auburn.edu
- The generic format is described below:
  - The serial number
    - An unsigned 32 bit value in range 1 to 4294967295 with a maximum increment of 2147483647
    - In BIND implementations this is defined to be a 10 digit field
    - This value must increment when any resource record in the zone file is updated

BIND 9 named.conf

- Bind is a DNS Nameserver implementation
- BIND uses the following configuration items:
  - A standard resolver (Caching-only DNS Server) config. file: named.conf
  - Zone file: master.localhost,
  - Other files: localhost.rev and root.servers
- Listen-on defines the port and IP address(es) on which BIND will listen for incoming queries
  - The default is port 53 on all server interfaces

named.conf file

options {
  directory "C:\Windows\system32\dns\etc";
  version "BIND 9";
  recursion yes;
  allow-recursion {131.204.0.0/16};
  listen-on {131.204.10.13};
};
zone "." {
  type hint;
  file "root.servers";
};
zone "auburn.edu" in {
  type master;
  file "master.localhost";
  allow-update {none};
};
zone "0.0.127.in-addr.arpa" in {
  type master;
  file "localhost.rev";
  allow-update {none};
};

Hint and root.servers

- When a name server cannot resolve a query it uses the file root.servers
  - The file root.servers defines a list of name servers (a.root-servers.net - m.root-servers.net) where BIND can get a list of TLD servers for the particular TLD e.g. .com
  - The root.servers file can be obtained from ICANN
    - Using anonymous FTP for file /domain/named.root on server ftp.internic.net or rs.internic.net
  - The root server file is defined using a normal zone clause with type hint as outlined in the following example:
    - The dot (".") zone identifies the DNS server as a root server

  
  3600000 IN NS A.ROOT-SERVERS.NET. A.ROOT-SERVERS.NET. 3600000 A 198.41.0.4
  3600000 IN NS B.ROOT-SERVERS.NET. B.ROOT-SERVERS.NET. 3600000 A 192.228.79.201
  3600000 IN NS C.ROOT-SERVERS.NET. C.ROOT-SERVERS.NET. 3600000 A 199.7.91.1
  3600000 IN NS D.ROOT-SERVERS.NET. D.ROOT-SERVERS.NET. 3600000 A 199.7.91.2
  3600000 IN NS E.ROOT-SERVERS.NET. E.ROOT-SERVERS.NET. 3600000 A 198.41.0.5

12/1/2016 Chapter 2 DNS and AD 209

12/1/2016 Chapter 2 DNS and AD 210

12/1/2016 Chapter 2 DNS and AD 211

12/1/2016 Chapter 2 DNS and AD 212
The localhost.rev file maps the IP address 127.0.0.1 to the name 'localhost'. This special zone allows reverse mapping of the loopback address 127.0.0.1 to satisfy applications which do reverse or double lookups. Any request for the address 127.0.0.1 using this name server will return the name localhost. The 0.0.127.IN-ADDR.ARPA zone is defined as shown below.

* This file should not require modification.

```plaintext
$TTL 86400 ; could use SOA IN-ADDR.ARPA.
root.localhost. (1997022700 ; Serial
15 ; Retry
1w ; Expire
3h ; Minimum
1 IN PTR localhost.
```

**DNS Message format**

- **Header**: 32 bits
- **Question**: The question for the name server
- **Answer**: RRs answering the question
- **Authority**: RRs pointing toward an authority
- **Additional**: RRs holding additional information

### DNS protocol and message format (1)

- **ID**: 16 bit # for query and reply
- **Number of questions**: Number of questions in question section
- **Number of RRs in authority records section**: Number of RRs in authority records section
- **Number of RRs in answer section**: Number of RRs in answer section
- **Number of RRs in additional section**: Number of RRs in additional section
- **Question section**: (variable number of questions)
- **Answer section**: (variable number of resource records)
- **Authority records section**: (variable number of resource records)
- **Additional section**: (variable number of resource records)
**DNS protocol and message format (2)**

<table>
<thead>
<tr>
<th>ID</th>
<th>Flags</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of questions</td>
<td>Number of RRs in answer section</td>
</tr>
<tr>
<td>Number of RRs in authority records section</td>
<td>Number of RRs in additional section</td>
</tr>
<tr>
<td>Question section (variable number of questions)</td>
<td>Answer section (variable number of resource records)</td>
</tr>
<tr>
<td>Authority records section (variable number of resource records)</td>
<td>Additional section (variable number of resource records)</td>
</tr>
</tbody>
</table>

**ID Flags**

- Name and type for a query
- RRs as the answer
- Resource records point toward another authoritative name server
- Non-recursive reply contains no answer and delegates to another DNS server
- Additional: "helpful" RR, e.g., suggestion to ask another DNS server (plus server's IP address) that may have answer

**Question section** (variable number of questions)

**Answer section** (variable number of resource records)

**Authority records section** (variable number of resource records)

**Additional section** (variable number of resource records)

---

**Find the IP address of a name server**

**Terminal — bash — 80x10**

---

**Query auburn.edu's mail RR**

---

**DNS for load balancing**

- **Load balancing**
  - Replicated Web/mail or other servers
  - A set of IP addresses for one name
  - Balance the load of each replicated server: Round robin DNS
    - When the request comes to the DNS server to resolve the domain name, it provides one of the several canonical names in a rotated order
    - This redirects the request to one of the several servers in a server group
    - Once the BIND feature of DNS resolves the domain to one of the servers, subsequent requests from the same client are sent to the same server

**Name:** google.com
- Address: 74.125.67.100 (name = gw-in-f100.google.com)
- Address: 74.125.45.100 (yx-in-f100.google.com)
- Address: 209.85.171.100 (cg-in-f100.google.com)

---
How to get an IP address of www.cnn.com using recursive mode

Step 1. From Client to Local DNS Server
- Data Format
  * Header Section's Flag: QR - 0 (Query), RD - 1 (Recursive Query Desired)
  * Questions Section: QNAME- www.cnn.com, QTYPE - A

Step 2~3. Between Local and Root DNS Server
- Step 2's Data Format
  * Header Section's Flag: QR - 0 (Query), RD - 0 (Non-Recursive Query Desired)
  * Question Section: QNAME- www.cnn.com, QTYPE - A
- Step 3's Data Format
  * Header Section's Flag: QR - 1 (Response)
  * Authority Section: (com, a.gtld-servers.net, NS)
  * Additional Section: (a.gtld-servers.net, 192.5.6.30, A)

Step 4~5. Between Local and TLD DNS Server
- Step 4's Data Format
  * Header Section's Flag: QR - 0 (Query), RD - 0 (Non-Recursive Query Desired)
  * Question Section: QNAME- www.cnn.com, QTYPE - A
- Step 5's Data Format
  * Header Section's Flag: QR - 1 (Response)
  * Authority Section: (cnn.com, ns1.timewarner.net, NS)
  * Additional Section: (ns1.timewarner.net, 204.74.108.238, A)
Step 6~7. Between Local and Authoritative DNS Server
- Step 6’s Data Format
  * Header Section’s Flag: QR - 0 (Query), RD - 0 (Non-Recursive Query Desired)
  * Question Section: QNAME: www.cnn.com, QTYPE: A
  * Additional Section: (ns1.timewarner.net, 204.74.108.238, A)

Step 8. From Local DNS Server to Client
- Step 8’s Data Format
  * Header Section’s Flag: QR - 1 (Response), RA - 1 (Recursive Query Available)
  * Authority Section: (www.cnn.com, (ns1.timewarner.net, NS)
  * Additional Section: (ns1.timewarner.net, 204.74.108.238, A)
DNS Caching

- DNS responses are cached
  - Quick response for repeated translations
- DNS negative queries are also cached
  - For example, misspellings
- Cached data periodically times out
- Cache poisoning for phishing
  - Redirect website's traffic to bogus website by forging DNS mapping
  - An attacker attempts to insert a fake address record for an Internet domain into the DNS
  - If the server accepts the fake record, the cache is poisoned and subsequent requests for the address of the domain are answered with the address of a server controlled by the attacker
  - For as long as the fake entry is cached by the server (entries usually have a time to live (TTL) of a couple of hours) subscriber's browsers or e-mail servers will automatically go to the address provided by the compromised DNS server

Drive-By Pharming

- Alice is visiting a malicious site
- Malicious scripts are loaded to Alice's computer
- Malicious scripts discover router
- Crack the password of the router and login
  - Most home routers have default passwords
- Modify DNS setting in the router to a name server controlled by attacker
  - Alice will be visiting bogus sites since DNS provides mappings to sites forged by attacker
  - Capture critical information by bogus sites
Operation Ghost Click

Two-year FBI investigation called Operation Ghost Click
- Beginning in 2007, the cyber thieves used malware known as DNSChanger to infect computers worldwide
- DNSChanger redirected unsuspecting users to rogue servers controlled by the cyber thieves
- More than four million computers infected in over 100 countries while generating $14 million in illegitimate income
- Of the computers infected with malware, at least 500,000 were in the United States, including computers belonging to U.S. government agencies, such as NASA


DNS Vulnerabilities

- Deployed DNS may include no authentication
  - Any DNS response is generally believed
  - No validating mechanism for the authenticity of information
- When a DNS caching server gets a query from a subscriber for a domain, it looks to see if it has an entry cached
  - If it does not, it asks authoritative DNS servers (run by domain owners) and waits for their responses
  - First response wins the cache acceptance

DNS cache poisoning

- Prior to Dan Kaminsky's discovery in 2008, attackers could only exploit this narrow opening
  - They had to beat legitimate authoritative DNS servers by sending a fake query response, hoping they arrive at the caching server first with the correct query parameter value
    - The same IP address it was sent from
    - The same port number is was sent from
    - The answer matches the question asked
    - A unique ID number matches what was sent
  - These races typically only lasted a fraction of a second, making it difficult for an attacker to succeed

Best DNS cache poisoning

- Dan Kaminsky discovered this new vulnerability because a security researcher figured out a way to eliminate the narrow time window
  - The ID that the attacker needs to guess is not fully random (or not random at all)
  - Attacker rapidly firing questions at the caching server that an attacker knows the server will not be able to answer
    - E.g., an attacker can ask where x1y2z3.amazon.com is, knowing a caching server is unlikely to have such an entry
    - That provokes subsequent questions from the caching server and creates millions of opportunities to send fake answers by attacker
**Best Cache Poisoning attack**

Bank A's DNS server

Query by Attacker

Response from ns.devil.com

AR: authority record section

Add: additional section

ns.devil.com

www.devil.com

1.1.1.2

pdns1.amazon.com 1.1.1.2

Bank A's DNS server

Query by Attacker

Response from ns.devil.com

AR: authority record section

Add: additional section

ns.devil.com

www.devil.com

1.1.1.2

pdns1.amazon.com 1.1.1.2

Attacker needs to access the private network of Bank A by other means/attacks

Bank A's DNS server

Query by Attacker

Response from ns.devil.com

AR: authority record section

Add: additional section

ns.devil.com

www.devil.com

1.1.1.2

pdns1.amazon.com 1.1.1.2

Attacker needs to access the private network of Bank A by other means/attacks

**RFC 2308**

When a nonexistent domain name is queried, name errors (NXDOMAIN) are indicated by the presence of "Name Error" in the RCODE field.

NXDOMAIN responses may provide some assistance in the authority section and the additional section.

- Authority section:
  - SOA (Start of Authority, RFC 1035) RR: defines the zone name, an e-mail contact and various time and refresh values applicable to the zone
  - NS RR's of the domain, such as:
    - Amazon.com NS pdns1.amazon.com
    - .com NS a.gtld-servers.net

- Additional section:
  - A RR's of the name servers
    - pdns1.amazon.com A 1.1.1.2
    - a.gtld-servers.net A 1.1.1.2

**Cache Poisoning one TLD**

Bank A's DNS server

Query by Attacker

Response from ns.devil.com

AR: authority record section

Add: additional section

ns.devil.com

www.devil.com

1.1.1.2

pdns1.amazon.com 1.1.1.2

Attacker needs to access the private network of Bank A by other means/attacks

Bank A's DNS server

Query by Attacker

Response from ns.devil.com

AR: authority record section

Add: additional section

ns.devil.com

www.devil.com

1.1.1.2

pdns1.amazon.com 1.1.1.2

Attacker needs to access the private network of Bank A by other means/attacks

Bank A's DNS server

Query by Attacker

Response from ns.devil.com

AR: authority record section

Add: additional section

ns.devil.com

www.devil.com

1.1.1.2

pdns1.amazon.com 1.1.1.2

Attacker needs to access the private network of Bank A by other means/attacks

Bank A's DNS server

Query by Attacker

Response from ns.devil.com

AR: authority record section

Add: additional section

ns.devil.com

www.devil.com

1.1.1.2

pdns1.amazon.com 1.1.1.2

Attacker needs to access the private network of Bank A by other means/attacks

12/1/2016 Chapter 26 Cyber Threats and Defense 237

12/1/2016 Chapter 26 Cyber Threats and Defense 238

12/1/2016 Chapter 26 Cyber Threats and Defense 239

12/1/2016 Chapter 26 Cyber Threats and Defense 240
Short-term Defense

- The patches that have been released in 2008 randomize the source port for the recursive server.
  - UDP port used for a query should no longer be the default port 53, but rather a port randomly chosen from the entire range of UDP ports (not including the reserved ports).
  - Microsoft’s updated DNS server is said to use 11 bits for randomizing about 2,500 UDP ports.
- Makes it harder for an attacker to guess query parameters.
  - Both the 16-bit query ID and as many as 11 additional bits for the UDP port must be correct, for a total of up to 134 million combinations.
- 2^{16} * 2^{11} = 2^{27} = 1.34 \times 10^8
- DNS servers behind network address translation (NAT): most NATs de-randomized the UDP ports used by the DNS server, rendering the new fix less effective.
- Another security researcher demonstrated that it was still possible to poison a DNS server even with the protection afforded by randomization across 64,000 UDP ports.

Source: http://www.caminista.net/tech/ip술ip-dns-vuln.html

Disable open recursive name servers

- The attack is not effective if the attacker cannot send query packets to the name server.
- If you must run a recursive name server, limit access to only those computers that need it (e.g., your hosts).

Long-term solution: authentication

- Resolver cannot distinguish between valid and invalid data in a response.
  - Idea is to add source authentication.
  - Verify the data received in a response is the same as that entered by the zone administrator.
- DNSSEC (DNS Security Extensions) protects against data spoofing and corruption.
- DNSSEC also provides mechanisms to authenticate servers and requests.
- DNSSEC provides mechanisms to establish authenticity and integrity.
**Authenticating DNS Responses**

- Each DNS zone signs its data using a private key
  - Recommend signing done offline in advance
- Query for a particular record returns:
  - The requested resource record set
  - A signature (RRSIG) of the requested resource record set (RRset)
- Resolver authenticates response using public key
  - Public key is pre-configured or learned via a sequence of key records in the DNS hierarchy

**DNSSEC Standards (1)**

- Goals: provides authentication and integrity of DNS responses
  - No confidentiality
  - No DoS protection
  - PKI-based
  - Authoritative DNS server signs its data in the zone
  - Signature can be signed in advance
- IETF RFC 3757, 4033, 4034, 4035, 4509, 4641, 5155
  - RFC 3757: Domain Name System Key (DNSKEY) Resource Record (RR) Secure Entry Point (SEP) Flag
  - RFC 4033: introduces DNSSEC and describes capabilities and limitations
  - RFC 4034: defines Resource Records for the DNSSEC

**DNSSEC Standards (2)**

- RFC 4035:
  - Describes the DNSSEC protocol
  - Defines the concept of a signed zone to authenticate both DNS resource records and authoritative DNS error indications
- RFC 4509:
  - Use of SHA-256 in DNSSEC Delegation Signer (DS) Resource Records (RRs)
- RFC 4641:
  - DNSSEC Operational Practices
  - Obsoletes RFC 2541
  - Gives more up-to-date requirements with respect to key sizes and the new DNSSEC specification
- RFC 5155
  - DNS Security (DNSSEC) Hashed Authenticated Denial of Existence
- NIST SP 800-81r1
  - Secure Domain Name System (DNS) Deployment Guide

**DNS Security Extensions (1)**

- DNSSEC allows RRs and zones to have origin authentication and integrity
  - One private key signs one zone
  - Use this case as the example since it is simple to understand
  - It is possible to use multiple private keys for signing a zone
- The Zone Signing Key (ZSK) can be used to sign all the data in a zone on a regular basis
  - When a Zone Signing Key is to be rolled, no interaction with the parent is needed
  - This allows for signature validity periods on the order of days
- The Key Signing Key (KSK) is only to be used to sign the DNSKEY RRs, containing ZSK, in a zone
  - If a Key Signing Key is to be rolled over, there will be interactions with parties other than the zone administrator
DNS Security Extensions (2)

- New types of RRs for DNSSEC
  - DNSKEY RR: Public key resource record
    - Contains the public key
  - RRSIG: Signature resource record
    - Each RRset has its corresponding RRSIG
  - DS: Delegation Signer (optional)
    - A parent domain can optionally delegate to a new key pair for signing RRs in the child domain
    - Containing a digest
  - NSEC: Next resource record
    - Enables the DNS server to inform the client that a particular domain or type does not exist

RRset Example

- RRset: RRs with same name, class and type
  - One RRset
    - auburn.edu. 3600 IN NS dns.auburn.edu
    - auburn.edu. 3600 IN NS dns.eng.auburn.edu
    - auburn.edu. 3600 IN NS dns.duc.auburn.edu
  - Another RRset
    - dns.auburn.edu. 3600 IN A 131.204.41.3
    - dns.eng.auburn.edu. 3600 IN A 131.204.10.13
    - dns.duc.auburn.edu. 3600 IN A 131.204.2.110

- RRsets are signed, not the individual RRs

DNS Security Extensions (3)

- DNSKEY: Public key resource record
  - A zone signs its authoritative resource record sets (RRsets) by using a private key and stores the corresponding public key in a DNSKEY RR
  - A resolver can then use the public key to validate signatures covering the RRsets in the zone, and thus to authenticate them

- RRSIG: Signature resource record
  - Each RRset has its corresponding RRSIG, containing a public-key signature which is stored as a resource record
    - E.g., www.x.com RR (type A) has a RRSIG RR containing the signature
    - The algorithm used (RSA/SHA1) to create the signature is contained in the RRSIG
    - The valid period of the RRSIG is also contained in RRSIG
    - RRSIG’s are computed for every RRset in a zone file and stored
    - Add the corresponding pre-calculated signature for each RRset in answers to queries

RRSIG Example

- RRSIG for the RRset containing 3 NS RRs of auburn.edu
  - auburn.edu. 3600 IN RRSIG A 5 2 3600 (20120101120000 20110101120000 0001 auburn.edu. MQJ+8… )
    - 5: RSA/SHA-1
    - 2: Labels, (the number of labels in the FQDN)
      - Hostnames are composed of series of labels concatenated with dots, as are all domain names
      - For example, "auburn.edu" is a hostname with 2 labels
      - For example, "eng.auburn.edu" is a hostname with 3 labels
    - 3600: TTL
    - 20120101120000: signature expiration
    - 20110101120000: signature inception
    - 0001: key tag
    - auburn.edu.: signer's name
    - MQJ+: signature
DNS Security Extensions (3)

- **DS**: Delegation Signer (optional)
  - When the parent zone delegates the name resolution to a child zone, the private key for signing is usually changed
  - E.g., .com DNS server has a pair of keys for signing and verifying .com zone
  - x.com has its own key pair for signing and verifying x.com zone
  - www.x.com RR is signed by x.com’s private key
  - Each DNSKEY of a zone has a corresponding DS RR
  - DS RR contains the digest of the corresponding DNSKEY
  - E.g., SHA-1 is the algorithm to generate the digest
  - RRs in the zone x.com is verified using public key in DNSKEY(x.com)

- **NSEC**: Next resource record
  - Enables the DNS server to inform the client that a particular domain or type does not exist

NSEC RR (1)

- Provides authenticated denial of existence for DNS data
- Providing negative responses with the same level of authentication and integrity
- Defeat the attack discovered by Kaminsky
- The NSEC record allows resolver to authenticate a negative reply for either name or type non-existence with the same mechanisms used to authenticate other DNS replies
- **NSEC3 RR**
  - Format and use the same as the NSEC Record
  - Uses hashed names instead of cleartext
  - Use of NSEC records requires a canonical representation and ordering for domain names in zones
  - Chains of NSEC records explicitly describe the gaps, or “empty space”, between domain names in a zone and list the types of RRsets present at existing names

NSEC RR (2)

- NSEC points to the next domain name in the zone
- Also lists all the existing RRs for “name”
- NSEC record for last name “wraps around” to first name in the zone
- Following names are sorted in canonical DNS name order
  - x.com
  - p.x.com
  - s.x.com
  - www.x.com
  - z.com
  - y.z.com
  - www.z.com

NSEC RR (3): zone file in pseudo format

- The canonical order of the unique domain names in the zone x.com:
  - x.com. IN SOA ns.x.com. master.x.com. (12985 3600 8000 3600)
  - IN RRSIG (SOA)
  - IN NS ns.x.com.
  - IN RRSG (NS)
  - IN MX mail.x.com.
  - IN RRSG (MX)
  - mail.x.com. IN A 131.204.101.8
  - IN RRSG (A)
  - ns.x.com. IN A 131.204.101.7
  - IN RRSG (A)
  - p.x.com. IN A 131.204.101.9
  - IN RRSG (A)
  - s.x.com. IN NS ns.x.com.
  - IN RRSG (NS)
  - www.x.com. IN A 131.204.101.10
  - IN RRSG (A)
NSEC RR (4)

The pseudo format (containing only the important fields) of NSEC RRs covering the gaps in the namespace relating to domain names and RR types found at each name:

- x.com:
  - IN NSEC mail.x.com. (NS SOA MX RRSIG NSEC)
  - IN A 131.204.101.8

- mail.x.com:
  - IN NSEC ns.x.com. (A RRSIG NSEC)
  - IN A 131.204.101.9

- ns.x.com:
  - IN NSEC p.x.com. (A RRSIG NSEC)
  - IN NS s.x.com. (A RRSIG NSEC)
  - IN A 131.204.101.10

- p.x.com:
  - IN A 131.204.101.10
  - IN RRSIG (NS)

- s.x.com:
  - IN NS www.x.com. (NS RRSIG NSEC)
  - IN RRSIG (NS)

- www.x.com:
  - IN RRSIG (NS)

Zone file in pseudo format (1)

x.com.
IN SOA ns.x.com. master.x.com. (12985 3600 2700 8000 3600)
IN RRSIG (SOA)
IN NS ns.x.com.
IN RRSIG (NS)
IN MX mail.x.com.
IN RRSIG (MX)
IN NSEC mail.x.com. (NS SOA MX RRSIG NSEC)
IN RRSIG (NSEC)

mail.x.com.
IN A 131.204.101.8
IN RRSIG (A)
IN NSEC ns.x.com. (A RRSIG NSEC)
IN RRSIG (NSEC)

Zone file in pseudo format (2)

ns.x.com.
IN A 131.204.101.7
IN RRSIG (A)
IN NSEC p.x.com. (A RRSIG NSEC)
IN RRSIG (NSEC)

p.x.com.
IN A 131.204.101.9
IN RR SIG (A)
IN NS s.x.com. (A RRSIG NSEC)
IN RRSIG (NS)

s.x.com.
IN NS www.x.com. (NS RRSIG NSEC)
IN RRSIG (NS)

IN A 131.204.101.10
IN RR SIG (A)
IN NS s.x.com. (A RRSIG NSEC)
IN RRSIG (NS)

Example: the use of NSEC RR

When a query for "p.x.com IN A" arrives (which does not exist in the zone), the authoritative server replies with the NSEC RRSIGSet that proves that the name does not exist in the zone:

- In this case, the response from the server will consist of the normal DNS reply indicating that the name does not exist:

- p.x.com. NSEC RR indicating there are no authoritative names between "p.x.com." and "s.x.com."
- www.x.com. NSEC RR (the last domain in the zone) proving that there are no wildcard names in the zone that could have been expanded to match the query
- Accompanying RRSIG RRs for each of the foregoing NSEC records for authentication
PKI: chain of trust (1)

- By using the hierarchical property of the DNS, DNSSEC can verify signatures without configuring the public keys of every single domain
- PKI allows a DNS cache server/resolver to verify signatures by tracing from a trusted anchor’s key down the DNS delegation chain
- Each level of the DNS must deploy DNSSEC
- Resolver can learn a zone’s public key by having a trust anchor configured into the resolver

Trusted anchor:
- Forming an authentication chain from a newly learned public key back to a previously known authenticated public key, which in turn either has been configured into the resolver or must have been learned and verified previously
- Therefore, a resolver must be configured with at least one trust anchor’s public key initially
  - The KSK of the root server published by ICANN

12/1/2016 | Chapter 26 Cyber Threats and Defense | 261

PKI: chain of trust (2)

- DNS query:
  - Public keys are stored in a new type of resource record, the DNSKEY RR
  - The private keys used to sign zone data must be kept secure
  - The target key has to be signed by either a configured authentication key or another key that has been authenticated previously
  - The target key: the public key is being used for authentication
- DS RRs used to link parent and child
- DS points to a Key Signing Key (KSK) of a child zone
  - Signature from that KSK over a DNSKEY RRset transfers trust to all keys in DNSKEY RRset
  - Key that DS points to, a KSK, only signs a DNSKEY RRset containing both KSK and ZSK
- Zone Signing Key (ZSK) in a DNSKEY RR sign entire zone’s RRs

12/1/2016 | Chapter 26 Cyber Threats and Defense | 262

DNSKEY and DS

- KSK serves as the “anchor” of the authentication chain to a child zone
- Need to install at least one public key in a recursive server/resolver to anchor the authentication chain

12/1/2016 | Chapter 26 Cyber Threats and Defense | 263

PKI: chain of trust (3)

- An alternating sequence consisting of DNS public key (DNSKEY) RRsets and Delegation Signer (DS) RRsets forms a chain of signed data
  - DS RR’s are used to link parent and child zones and a DS RR of the parent zone (e.g., .com) points to a Key Signing Key (KSK) of a child zone (e.g., amazon.com)
  - The parent zone creates a hash of the public key of KSK of its child and stores it in the parent zone in a RR called a DS RR
  - The parent zone signs this DS RR by generating a RRSIG RR using parent zone’s ZSK
  - The DS RR contains a hash/digest of a child zone’s DNSKEY RR (a KSK), and this new DNSKEY RR is authenticated by matching the hash in the DS RR using the parent zone’s public key (ZSK of the parent zone) and the RRSIG of the DS RR
  - In essence, a DNSKEY RR (ZSK) of the parent zone is used to verify the signature covering a DS RR and allows the DS RR to be authenticated

12/1/2016 | Chapter 26 Cyber Threats and Defense | 264
PKI: chain of trust (4)

* A KSK serves as the “anchor” of the authentication chain to a child zone
* A successful signature verification from that KSK over a DNSKEY RRset in a child zone transfers trust to all keys in the DNSKEY RRset
  * Then the DNSKEY RR in this set, containing the ZSK of the zone, may be used to authenticate another DS RR, and so forth until the chain finally ends with a DNSKEY RR whose corresponding private key signs the desired DNS data
  * The DNSKEY RR in this set, containing the ZSK of the zone, can also be used to authenticate other RRSIG’s in this zone

DS RR in the parent zone

* If the zone administrator intends to sign a zone, the zone apex must contain at least one DNSKEY RR to act as a secure entry point (SEP) from parent zone into the zone
  * This secure entry point (SEP) could then be used as the target of a secure delegation via a corresponding DS RR in the parent zone
  * The child zone’s SEP public key should be signed by the corresponding private key of the parent zone as a DS RR
  * This child’s SEP key is called a KSK contained in a DNSKEY RR in a child zone
  * Successful verification of DS RR in the parent zone is the authentication of the public key of child zone’s KSK
  * If the SEP Flags’ value is 257, then the DNSKEY record holds a key intended for use as a secure entry point
    * This flag is only intended to be a hint to zone signing or debugging software

DS RR and RRSIG RR in parent zone (1)

* As part of the chain of trust, the zone has to inform its parent of its public key, KSK, securely through out-of-DNS channel means
  * The parent creates a hash of the public key of its child zone’s KSK and stores it in the parent zone in a RR called a DS RR
  * It also signs this DS RR by generating a RRSIG RR
    * The keys periodically have to be changed because any key can be broken with sufficient computing power, aided by the volume of signature data generated
    * In a chained secure zone, whenever a zone changes its KSK, its parent has to be notified of the new key
    * The parent then has to generate a new DS RR and sign it again
* To reduce the administrative burden involved, a common strategy is to use another key pair, the ZSK for signing the child zone

DS RR and RRSIG RR in parent zone (2)

* The KSK is used for signing only the DNSKEY RRSet; all of the other authoritative RRsets in the zone file are signed with the ZSK
  * The KSK is the key that is published to the parent
  * The parent will generate the DS RR and a RRSIG RR using the parent’s own ZSK
  * The KSK is used less frequently to sign the DNSKEY RRset and hence needs to be changed less frequently
  * There may be situations, in which either because of the manageable frequency of key rollovers (key change) or the criticality of DNS information served by the zone, administrators may not use two distinct key pairs for the ZSK and KSK
Separating the functions of KSK and ZSK

- No parent/child interaction is required when ZSKs are updated
- The KSK can be made stronger (i.e., using more bits in the key material)
- This has little operational impact since it is only used to sign a small fraction of the zone data
- The KSK is only used to verify the zone's key set, not for other RRSets in the zone
- As the KSK is only used to sign a key set, which is most probably updated less frequently than other data in the zone, it can be stored separately from, and in a safer location, than the ZSK
- A KSK can have a longer key effective period

DS Example (1)

- a DS RR in .com zone:
dkey.x.com. 10000 IN DS 2000 5 1 (6BB183AF...)
  - KEY ID = 10000
  - Value 2000 is the key tag for the corresponding dskey.x.com. DNSKEY RR
  - Value 5 denotes the algorithm RSA/SHA-1 used by this dskey.x.com. DNSKEY RR
  - The value 1 is the algorithm, SHA-1, used to construct the digest
  - 6BB183AF...: KSK's hash
- a child zone's (x.com's) DNSKEY RR (a KSK):
dkey.x.com. 10000 IN DNSKEY 257 3 5 (AQOe...)
  - Value 257 indicates that the Zone Key bit (bit 7) in the SEP Flags field has value 1
  - Value 3 is the fixed Protocol value
  - Value 5 indicates the public key algorithm: RSA/SHA-1
  - AQOe...: Base64 encoded public key string

DS Example (2)

- DS digest = digest_algorithm (DNSKEY owner name || DNSKEY RDATA)
- DNSKEY RDATA = Flags || Protocol || Algorithm || Public Key
- The DNSKEY RR referred to in the DS RR must be a DNSSEC zone key using KEY ID
- The DNSKEY RR Flags (16 bits) must have Flag bit 7 set
  - If the DNSKEY flags do not indicate a DNSSEC zone key, the DS RR (and the DNSKEY RR it references) must not be used in the validation process
  - Key Signing Keys (KSKs) has SEP Flags = 257
  - Zone Signing Keys (ZSKs) has SEP Flags = 256

ZSK's DNSKEY RR

- An example for a ZSK's DNSKEY RR:
x.com IN DNSKEY 256 3 5 (AQF+KGJ7......)
  - Zone Signing Keys (ZSKs) has SEP Flags = 256
  - Value 3 is the fixed Protocol value
  - The Protocol Field must have value 3
  - Value 5 indicates the public key algorithm: RSA/SHA-1
  - AQF.....: Base64 encoded public key string
Authentication Chain (1)

- A sequence of a ZSK in a DNSKEY RR and Delegation Signer (DS) RR in a parent zone, as well as the KSK in a child zone certified by the corresponding DS RR, forms an authentication chain of signed data.
- A DNSKEY RR (ZSK) is used to verify the signature covering a DS RR and allows the DS RR to be authenticated in a parent zone.
- The DS RR contains a hash of the KSK of a child zone and this KSK's DNSKEY RR is authenticated by matching the hash in the DS RR in the parent zone.
- This child zone KSK authenticates the DNSKEY RRset, which contains a ZSK, which in turn authenticates another DS RR, and so forth until the chain finally ends with a DNSKEY RR whose corresponding private key signs the desired DNS RR data.

Example
- The root ZSK in a DNSKEY RR of the root zone is used to sign the DS RR for ".com".
- This KSK signs the DNSKEY RRset of ".com", containing ZSK.
- The ZSK's private key signs the DNSKEY RRset of "amazon.com", containing ZSK.
- The "amazon.com" KSK signs the DNSKEY RRset of "www.amazon.com".

Authentication Chain using KSK(root)

- Signatures are pre-generated using private keys.
- Authentication using public keys, starting from the anchor KSK(root).
- The public key of KSK(.com) is obtained using DNSKEY(.com) RR.
- The authentication for the public key of KSK(.com) uses the RRSIG(DNSKEY(.com))

Example (1)
- dns.auburn.edu receives a recursive request for address mapping "www.amazon.com" from a client host.
- A DNS server (dns.auburn.edu) was configured to have the public key of the root DNS server and the root KSK in a DNSKEY(root) RR.
- The TLD .com name server RR, contained in the root server, is signed using the ZSK of a DNSKEY(root) RR to generate RRSIG(.com)

Authentication Chain (2)

Example
- The root ZSK in a DNSKEY RR of the root zone is used to sign the DS RR for ".com".
- The "amazon.com" DS RR contains a hash that matches "amazon.com" KSK.
- This KSK signs the DNSKEY RRset of "amazon.com", containing ZSK.
- The "amazon.com" KSK signs the DNSKEY RRset of "www.amazon.com".
- The "www.amazon.com" KSK signs the DNSKEY RRset of "amazon.com".
- The "amazon.com" ZSK signs the DNSKEY RRset of "amazon.com".
- The root KSK is published for verifying the root ZSK.

Example (2)
- The root ZSK in a DNSKEY RR of the root zone is used to sign the DS RR for ".com".
- The "amazon.com" DS RR contains a hash that matches "amazon.com" KSK.
- This KSK signs the DNSKEY RRset of "amazon.com", containing ZSK.
- The "amazon.com" ZSK signs the DNSKEY RRset of "www.amazon.com".
- The "www.amazon.com" KSK signs the DNSKEY RRset of "amazon.com".
- The "amazon.com" ZSK signs the DNSKEY RRset of "amazon.com".
- The root KSK is published for verifying the root ZSK.

Example (3)
- dns.auburn.edu receives a recursive request for address mapping "www.amazon.com" from a client host.
- A DNS server (dns.auburn.edu) was configured to have the public key of the root DNS server and the root KSK in a DNSKEY(root) RR.
- The TLD .com name server RR, contained in the root server, is signed using the ZSK of a DNSKEY(root) RR to generate RRSIG(.com)
Example (2)

- The DS(.com) RR points to the KSK of a DNSKEY(.com) that is used to sign the DNSKEY(.com) RRset.
- The ZSK in a DNSKEY(.com) RR signs all RR's contained in .com TLD server.
- When the signature of DNSKEY(.com) RRset is verified successfully, dns.auburn.edu trusts the DNSKEY(.com).

Example (3)

- The ZSK in a DNSKEY(.com) is used to verify RRSIG(amazon.com NS), DS(amazon.com)… by dns.auburn.edu.
- The KSK of amazon.com pointed by DS(amazon.com) is used to verify the ZSK in a DNSKEY(amazon.com).
- The ZSK of amazon.com is used to verify www.amazon.com RR.
- After successful verification, dns.auburn.edu accepts the www.amazon.com RR and delivers to the client.

Authentication Chain

<table>
<thead>
<tr>
<th>Root</th>
<th>.COM</th>
<th>.COM</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://www.amazon.com">www.amazon.com</a></td>
<td>dns.auburn.edu</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>72.21.207.65</td>
<td>pdns1.amazon.com</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>root</td>
<td>.COM</td>
<td>.COM</td>
</tr>
</tbody>
</table>

DNSSEC Deployment (1)

- Feb 28, 2009:
  - The US government has digitally signed the .gov TLD, effectively implementing the Domain Name System Security Extensions (DNSSEC) protocols throughout the top tier of the federal Internet space.
- On 5/5/2010:
  - The 13 authoritative root servers for the domain name system have switched to the DNS Security Extensions (DNSSEC) security protocol. All 13 root servers are now serving a signed version of the root zone.
  - DNSSEC in the .org TLD registry in June 2010.
  - DNSSEC in the .edu TLD registry in June 2010.
DNSSEC Deployment (2)

- The .com domain's DNSSEC became operational 3/31/2011
- The three largest zones are .com, .net, .org
  - The .com domain: the Internet's most popular top-level domain with more than 80 million registered names
  - The .org space has more than 7.5 million domains registered in it
  - The .gov top-level domain has about 3,700 domains

Root Zone Signing (1)

- VeriSign is the Root Zone Maintainer
  - Manages the Root Zone Signing Key (ZSK)
    - 1024 bits
    - ZSK is replaced four times a year (1-3 months)
    - US Government
      - RSA-SHA1 or RSA-SHA-256 until 2015
      - ECDSA after 2015
    - Incorporates NTIA-authorized changes
      - US Department of Commerce (DoC)
      - National Telecommunications and Information Administration (NTIA)
  - Signs the root zone with the ZSK
  - Distributes the signed zone to the root server operators

Root Zone Signing (2)

- Key Signing Key (KSK) is used to sign ZSK
  - 2048 bits
  - KSK is replaced one time a year (1-2 years)
  - US Government
    - RSA-SHA1 or RSA-SHA-256 until 2015
    - ECDSA after 2015
- ICANN publishes the public part of the KSK
- IANA Functions Operator
  - Manages the Key Signing Key (KSK)
  - Accepts DS records from TLD operators
  - Verifies and processes request
  - Sends update requests to DoC for authorization and to VeriSign for implementation
Trusted vs. Trustworthy

- We trust our computers
  - We depend upon them.
  - We are vulnerable to breaches of security.
- Our computer systems today are not worthy of trust.
  - We have buggy software
  - We configure the systems incorrectly
  - Our user interfaces are ambiguous regarding the parts of the system with which we communicate.

A Controversial Issue

- Many individuals distrust trusted computing.
- One view can be found at http://www.lafkon.net/tc/
  - An animated short film by Benjamin Stephan and Lutz Vogel
What is Trusted Computing

- Attestation
  - Includes Trusted path
- Separation
  - Secure storage (data/keys)
  - Protection of processes
- The rest is policy
  - That’s the hard part
  - And the controversial part

Separation of Security Domains

- Need to delineation between domains
  - Old Concept:
    - Rings in Multics
    - System/Privileged vs. User mode
  - But who decides what is trusted
    - User in some cases
    - Third parties in others
  - Trusted computing provides the basis for making the assessment.

Trusted Path

- We need a “trusted path”
  - For user to communicate with a domain that is trustworthy.
    - Usually initiated by escape sequence that application can not intercept: e.g. CTL-ALT-DEL
  - Could be direct interface to trusted device:
    - Display and keypad on smartcard

Communicated Assurance

- We need a “trusted path” across the network.
- Provides authentication of the software components with which one communicates.
The Landscape – Early Work

- Multics System in late 1960s.
  - Trusted path, isolation.
  - Described early need for remote attestation and how accomplished.

The Landscape – Industry

- Industry interest in the late 1990s.
- Consortia formed such as the Trusted Computing Group.
- Standards specifications, starting with specs for hardware with goal of eventual inclusion in all new computer systems.
  - Current results centered around attestation and secure storage.

The Landscape – Applications

- Digital Rights Management
- Network Admission Control
  - PC Health Monitoring
  - Malware detection
- Virtualization of world view
  - VPN Segregation
  - Process control / SCADA systems
- Many other users

Discussion - Risks

- Trusted computing is a tool that can be misused.
  - If one party has too much market power, it can dictate unreasonable terms and enforce them.
- Too much trust in trusted computing.
  - Attestation does not make a component trustworthy.
  - Some will rely too much on certifications.
**Discussion - Benefits**

- Allows systems to be developed that require trustworthy remote components.
  - Provides protection of data when out of the hands of its owner.
- Can provide isolation and virtualization beyond local system.
  - Provides containment of compromise.

**Discussion – What’s missing**

- Tools to manage policy
  - Managing policy was limitation for TC support in Vista
- Applications that protect the end user
  - We need more than DRM and tools to limit what users run.
- New architectures and ways of thinking about security.

**Trusted Baggage**

- So why all the concerns in the open source community regarding trusted computing.
  - Does it really discriminate against open source software.
  - Can it be used to spy on users.

**Equal Opportunity for Discrimination**

- Trusted computing means that the entities that interact with one another can be more certain about their counterparts.
- This gives all entities the ability to discriminate based on trust.
- Trust is not global – instead one is trusted “to act a certain way”.
Equal Opportunity for Discrimination(2)

• Parties can impose limits on what the software they trust will do.
• That can leave less trusted entities at a disadvantage.
• Open source has fewer opportunities to become “trusted”.

Is Trusted Computing Evil

• Trusted computing is not evil
  – It is the policies that companies use trusted computing to enforce that are in question.
  – Do some policies violate intrinsic rights or fair competition?
  – That is for the courts to decide.

What can we do with TC?

• Clearer delineation of security domains
  – We can run untrusted programs safely.
    • Run in domain with no access to sensitive resources
      – Such as most of your filesystem
      – Requests to resources require mediation by TCB, with possible queries user through trusted path.

Mediating Programs Today

• Why are we so vulnerable to malicious code today?
  – Running programs have full access to system files.
  – Why? NTFS and XP provide separation.
    • But many applications won’t install, or even run, unless users have administrator access.
  – So we run in “System High”
Corporate IT Departments Solve this

- Users don’t have administrator access even on their own laptops.
  - This keeps end users from installing their own software, and keeps IT staff in control.
  - IT staff select only software for end users that will run without administrator privileges.
  - But systems still vulnerable to exploits in programs that cause access to private data.
  - Effects of “Plugins” can persist across sessions.

The next step

- But, what if programs were accompanied by third party certificates that said what they should be able access.
  - IT department can issues the certificates for new applications.
  - Access beyond what is expected results in system dialogue with user over the trusted path.

Red / Green Networks (1)

- Butler Lampson of Microsoft and MIT suggests we need two computers (or two domains within our computers).
  - Red network provides for open interaction with anyone, and low confidence in who we talk with.
  - We are prepared to reload from scratch and lose our state in the red system.

Red / Green Networks (2)

- The Green system is the one where we store our important information, and from which we communicate to our banks, and perform other sensitive functions.
  - The Green network provides high accountability, no anonymity, and we are safe because of the accountability.
  - But this green system requires professional administration.
  - My concern is that a breach anywhere destroys the accountability for all.
Somewhere over the Rainbow

- But what if we could define these systems on an application by application basis.
  - There must be a barrier to creating new virtual systems, so that users don’t become accustomed to clicking “OK”.
  - But once created, the TCB prevents the unauthorized retrieval of information from outside this virtual system, or the import of untrusted code into this system.
  - Question is who sets the rules for information flow, and do we allow overrides (to allow the creation of third party applications that do need access to the information so protected).

A Financial Virtual System

- I might have my financial virtual system. When asked for financially sensitive data, I hit CTL-ALT-DEL to see which virtual system is asking for the data.
- I create a new virtual systems from trusted media provided by my bank.
- I can add applications, like quicken, and new participant’s, like my stock broker, to a virtual system only if they have credentials signed by a trusted third party.
  - Perhaps my bank, perhaps some other entity.

How Many Virtual Systems

- Some examples:
  - My open, untrusted, wild Internet.
  - My financial virtual system
  - My employer’s virtual system.
  - Virtual systems for collaborations
    - Virtual Organizations
  - Virtual systems that protect others
    - Might run inside VM’s that protect me
      - Resolve conflicting policies
      - DRM vs. Privacy, etc

Digital Rights Management

- Strong DRM systems require trust in the systems that receive and process protected content.
  - Trust is decided by the provider of the content.
  - This requires that the system provides assurance that the software running on the accessing system is software trusted by the provider.
Privacy and Anti-Trust Concerns

- The provider decides its basis for trust.
  - Trusted software may have features that are counter to the interests of the customer.
    - Imposed limits on fair use.
    - Collection and transmission of data the customer considers private.
    - Inability to access the content on alternative platforms, or within an open source O/S.

Trusted Computing Cuts Both Ways

- The provider-trusted application might be running in a protected environment that doesn’t have access to the user’s private data.
  - Attempts to access the private data would thus be brought to the user’s attention and mediate through the trusted path.
  - The provider still has the right not to provide the content, but at least the surreptitious snooping on the user is exposed.

What do we need for TC

- Trust must be grounded
  - Hardware support
    - How do we trust the hardware
    - Tamper resistance
      - Embedded encryption key for signing next level certificates.
    - Trusted HW generates signed checksum of the OS and provides new private key to the OS

Privacy of Trusted Hardware

- Consider the processor serial number debate over Intel chips.
  - Many considered it a violation of privacy for software to have ability to uniquely identify the process on which it runs, since this data could be embedded in protocols to track user’s movements and associations.
  - But Ethernet address is similar, although software allows one to use a different MAC address.
  - Ethernet addresses are often used in deriving unique identifiers.
PRIVACY ISSUES SURROUNDING HTML5 BATTERY STATUS API

Presented By Kalhan Koul

HTML5 BATTERY STATUS API

It enables websites to access the battery state of a mobile device or a laptop.

Websites use this information to switch between energy-saving or high-performance modes.

 Doesn't require user to grant permissions.

EXPLOIT

The capacity of the battery, charging Time and discharging Time expose a finger printable surface that can be used to track web users in short time intervals.

A third-party script that is present across multiple websites can link users' visits in a short time interval by exploiting the battery information.

EXPLOIT CONTD.

Users who try to re-visit a website with a new identity may use browsers' private mode or clear cookies and other client side identifiers. When consecutive visits are made within a short interval, the website can link users' new and old identities by exploiting battery level and charge/discharge times.

The website can then reinstantiate users' cookies and other client side identifiers, a method known as respawning.

Sources:
http://lukaszolejnik.com/battery.pdf
https://www.theguardian.com/technology/2016/aug/02/battery-status-indicators-tracking-online
**FIX**

Firefox removed support for Battery API as of Firefox 52.
Apple expected to do the same with Safari.
Changes being made to Battery Status API W3C specifications.

---

**The Key to your Trusted Hardware**

- Does not have to be unique per machine, but uniqueness allows revocation if hardware is known to be compromised.
  - But what if a whole class of hardware is compromised, if the machine no longer useful for a whole class of applications. Who pays to replace it.
- A unique key identifies specific machine in use.
  - Can a signature use a series of unique keys that are not linkable, yet which can be revoked (research problem).

---

**Non-Maskable Interrupts**

- We must have hardware support for a non-maskable interrupt that will transfer program execution to the Trusted Computing Base (TCB).
  - This invokes the trusted path

---

**The Hardware Basis**

- Trusted computing is proof by induction
  - Each attestation stage says something about the next level
  - Just like PKI Certification hierarchy
- One needs a basis step
  - On which one relies
  - Hardware is that step
  - (well, second step anyway)
Hardware Topics

• Trusted Platform Module
• Discussion of Secure Storage
• Boot process

Trusted Platform Module

• Basically a Key Storage and Generation Device
• Capabilities:
  – Generation of new keys
  – Storage and management of keys
    ▪ Uses keys without releasing

Trusted Platform Module (TPM)?

Smartcard-like module on the motherboard that:
• Performs cryptographic functions
  – RSA, SHA-1, RNG
  – Meets encryption export requirements
• Can create, store and manage keys
  – Provides a unique Endorsement Key (EK)
  – Provides a unique Storage Root Key (SRK)
• Performs digital signature operations
• Holds Platform Measurements (hashes)
• Anchors chain of trust for keys and credentials
• Protects itself against attacks

TPM 1.2 spec:
www.trustedcomputinggroup.org

Why Use A TPM?

• Trusted Platforms use Roots-of-Trust
  – A TPM is an implementation of a Root-of-Trust
• A hardware Root-of-Trust has distinct advantages
  – Software can be hacked by Software
    ▪ Difficult to root trust in software that has to validate itself
  – Hardware can be made to be robust against attacks
    ▪ Certified to be tamper resistant
    ▪ Hardware and software combined can protect root secrets better than software alone
• A TPM can ensure that keys and secrets are only available for use when the environment is appropriate
  – Security can be tied to specific hardware and software configurations
Endorsement Key

- Every TPM has unique Endorsement key
  - Semi-root of trust for system
  - Generated and installed during manufacture
    - Issues
    - Real root is CA that signs public key associated with Endorsement key

Using Encryption for Atestation

- Extend
  - Add data to a PCR
  - 20 byte hash hashed into current PCR
  - As each module loaded its hash extends the PCR

- Quote
  - Sign current value of PCR

Secure Storage

- Full Disk Encryption
  - Key in register in disk
  - Or key in TPM and data encrypted/decrypted by TPM
- Seagate Drive uses register in Disk
  - Key must be loaded
  - User prompt at BIOS
  - Or managed by TPM
    - But OS image maybe on disk, how to get

OS Support for Trusted Computing (1)

- Separation of address space
  - So running processes don’t interfere with one another.
- Key and certificate management for processes
  - Process tables contain keys or key identifiers needed by application, and keys must be protected against access by others.
  - Processes need ability to use the keys.
OS Support for Trusted Computing (2)

- Fine grained access controls on persistent resources.
  - Protects such resources from untrusted applications.
- The system must protect against actions by the owner of the system.

Disk Layout & Key Storage

Windows Partition Contains
- Encrypted OS
- Encrypted Page File
- Encrypted Temp Files
- Encrypted Data
- Encrypted Hibernation File

Where's the Encryption Key?
1. SRK (Storage Root Key) contained in TPM
2. SRK encrypts VEK (Volume Encryption Key) protected by TPM/PIN/Dongle
3. VEK stored (encrypted by SRK) on hard drive in Boot Partition

Vista co-existence

- BitLocker encrypts Windows partition only
- You won’t be able to dual-boot another OS on the same partition
- OSes on other partitions will work fine
- Attempts to modify the protected Windows partition will render it unbootable
  - Replacing MBR
  - Modifying even a single bit
Outline of Discussion

- Introduction – security vs privacy
- You are being tracked
- Aggregation of data
- Traffic analysis and onion routing
- P3P and Privacy Statements
- Protecting data on personal laptops/desktops
- Forensics
- Retention/Destruction Policies
- Who’s data is it anyway

What is Privacy?

- Privacy is about Personally Identifiable Information
- It is primarily a policy issue
  - Policy as a system issue
    - Specifying what the system should allow
  - Policy as in public policy
    - Same idea but less precise and must be mapped
- Privacy is an issue of user education
  - Make sure users are aware of the potential use of the information they provide
  - Give the user control
- Privacy is a Security Issue
  - Security is needed to implement the policy

Security v. Privacy

- Sometimes conflicting
  - Many security technologies depend on identification.
  - Many approaches to privacy depend on hiding ones identity.
- Sometime supportive
  - Privacy depends on protecting PII (personally identifiable information).
  - Poor security makes it more difficult to protect such information.
Major Debate on Attribution

- How much low level information should be kept to help track down cyber attacks.
  - Such information can be used to breach privacy assurances.
  - How long can such data be kept.

Privacy not Only About Privacy

- Business Concerns
  - Disclosing information we think of as privacy related can divulge business plans.
    - Mergers
    - Product plans
    - Investigations
  - Some “private” information is used for authentication.
    - SSN
    - Credit card numbers

You Are Being Tracked

- Location
  - From IP address
  - From Cell Phones
  - From RFID
- Interests, Purchase History, Political/Religious Affiliations
  - From RFID
  - From Transaction Details
  - From network and server traces
- Associates
  - From network, phone, email records
  - From location based information
- Health Information
  - From Purchases
  - From Location based information
  - From web history

2009 current event


- SAN FRANCISCO — There is a new common symptom of the flu, in addition to the usual aches, coughs, fevers and sore throats. Turns out a lot of ailing Americans enter phrases like “flu symptoms” into Google and other search engines before they call their doctors.
  - link
More news - FOIA docs show feds can lojack mobiles without telco help –
Ars Technica - Julian Sanchez 10/16/2008

• Triggerfish, also known as cell-site simulators or digital analyzers, are nothing new: the technology was used in the 1990s to hunt down renowned hacker Kevin Mitnick. By posing as a cell tower, triggerfish trick nearby cell phones into transmitting their serial numbers, phone numbers, and other data to law enforcement.

CSci530: Security Systems
Lecture 14 – December 2, 2016
Privacy, Cyber-Physical, Review

Dr. Clifford Neuman
University of Southern California
Information Sciences Institute

Announcements

• Research paper due today
  – Can turn in now or later
  – Instructions on paper assignment
  – Can take a week extra with small penalty
• Course evaluation to be performed online
  – Extra 5 min at break to do this
• Final Exam
  – Monday 12/12 11AM SGM 124

Why Should you Care?

• Aren’t the only ones that need to be concerned about privacy the ones that are doing things that they shouldn’t?
• Consider the following:
  – Use of information outside original context
    • Certain information may be omitted
  – Implications may be mis-represented.
  – Inference of data that is sensitive.
  – Such data is often not protected.
  – Data can be used for manipulation.
Old News - Shopper's Suit Thrown Out
Los Angeles Times – 2/11/1999

• Shopper's Suit Thrown Out
• By Stuart Silverstein, Staff Reporter

A Vons shopper's lawsuit that raised questions about the privacy of information that supermarkets collect on their customers' purchases has been thrown out of court. Los Angeles Superior Court Judge David Horowitz tossed out the civil suit by plaintiff Robert Rivera of Los Angeles, declaring that the evidence never established that Vons was liable for damages.

The central issue in the case was a negligence claim Rivera made against Vons. It stemmed from an accident at the Lincoln Heights' Vons in 1996 in which Rivera slipped on spilled yogurt and smashed his kneecap.

Although that issue was a routine legal matter, the case drew attention because Rivera raised the privacy issue in the pretrial phase. Rivera claimed that he learned that Vons looked up computer records of alcohol purchases he made while using his club discount card and threatened to use the information against him at trial.

Vons, however, denied looking up Rivera's purchase records and the issue never came up in the trial, which lasted two weeks before being thrown out by the judge Tuesday.

A Vons spokesman said the company was “gratified by the judge’s decision.” M. Edward Franklin, a Century City lawyer representing Rivera, said he would seek a new trial for his client.

Aggregation of Data

• Consider whether it is safe to release information in aggregate.
  – Such information is presumably no longer personally identifiable
  – But given partial information, it is sometimes possible to derive other information by combining it with the aggregated data.

Anonymization of Data

• Consider whether it is safe to release information that has been stripped of so called personal identifiers.
  – Such information is presumably no longer personally identifiable
  – But is it. Consider the release of AOL search data that had been stripped of information identifying the individual performing the search.
    • What is important is not just anonymity, but likability.
    • If I can link multiple queries, I might be able to infer the identity of the person issuing the query through one query, at which point, all anonymity is lost.

Traffic Analysis

• Even when specifics of communication are hidden, the mere knowledge of communication between parties provides useful information to an adversary.
  – E.g. pending mergers or acquisitions
  – Relationships between entities
  – Created visibility of the structure of an organizations.
  – Allows some inference about your interests.
Information Useful for TA

- Lists of the web sites you visit
- Email logs
- Phone records
- Perhaps you expose the linkages through web sites like linked in.
- Consider what information remains in the clear when you design security protocols.

Linkages – The Trail We Leave

- Identifiers
  - IP Address
  - Cookies
  - Login IDs
  - MAC Address and other unique IDs
  - Document meta-data
  - Printer microdots
- Where saved
  - Log files
- Persistence
  - How often does IP address change
  - How can it be mapped to user identification

Unlinking the Trail

- Blind Signatures
  - Enable proof of some attribute without identifying the prover.
  - Application in anonymous currency.
  - Useful in voting.

Obama's cell phone records breached
Washington (CNN) 11/21/2008

- Records from a cell phone used by President-elect Obama were improperly breached, apparently by employees of the cell phone company, Verizon Wireless said Thursday.
- "This week we learned that a number of Verizon Wireless employees have, without authorization, accessed and viewed President-Elect Barack Obama's personal cell phone account," Lowell McAdam, Verizon Wireless president and CEO, said in a statement.
- McAdam said the device on the account was a simple voice flip-phone, not a BlackBerry or other smartphone designed for e-mail or other data services, so none of Obama's e-mail could have been accessed.
- Gibbs said that anyone viewing the records likely would have been able to see phone numbers and the frequency of calls Obama made, but that "nobody was monitoring voicemail or anything like that."
Unlinking the Trail

- **Anonymizers**
  - A remote web proxy.
  - Hides originators IP address from sites that are visited.
  - Usually strips off cookies and other identifying information.

- **Limitations**
  - You are dependent on the privacy protections of the anonymizer itself.
  - All your activities are now visible at this single point of compromise.
  - Use of the anonymizer may highlight exactly those activities that you want to go unnoticed.

Onion Routing

- **Layers of peer-to-peer anonymization.**
  - You contact some node in the onion routing network.
  - Your traffic is forwarded to other nodes in the network.
  - Random delays and reordering is applied.
  - With fixed probability, it is forwarded on to its destination.

- **TA requires linking packets through the full chain of participants.**
  - And may be different for each association.

Privacy in Deep Web

- **Tor** – The Onion Routing
  Tor emphasizes anonymity, and a VPN emphasizes privacy.

- **Recent News** –
  - Wikipedia Editors now use Tor after Receiving life threats – 11th Nov, 2016

- **Tor over VPN**

Presented by –
Shashwat Ajmani
Tor over VPN

• Make sure the VPN does not keep logs
• More the encryption of the VPN, more the safety
• VPN only sees the connection to TOR nodes and encrypted data
• Do not enter any personal data over VPN

P3P and Privacy Statements

• Most commercial web sites provide a privacy statement.
  – Most are not worth the paper they are printed on
  – You probably view it on your screen
  – Many actually are illustrative, as they are written to say that “we can’t control what happens to you data
    – so don’t blame us”
  – Who reads them anyway
  – How are they enforced
    – Some are certified by outside endorsers

P3P was a protocol that was designed to allow users to specify their preferences, and to have these preferences negotiated by a browser when connecting to a site.
  – But it still doesn’t provide any enforcement that the site follows it stated policy.
  – It doesn’t ensure that the data held by the site is not compromised by outsiders.
  – You may still see support in some browsers, but it saw only brief adoption by web sites.

Protecting Data in Place

• Many compromises of privacy are due to security compromised on the machines holding private data.
  – Your personal computer or PDAs
  – Due to malware or physical device theft
• Countermeasures
  – For device theft, encryption is helpful
  – For malware, all the techniques for defending against malicious code are important.
  – Live malware has the same access to data as you do when running processes, so encryption might not be sufficient.
Forensics

- Tools are available to recover supposedly deleted data from disks.
  - Similar tools can reconstruct network sessions.
  - Old computers must be disposed of properly to protect any data that was previously stored.
    - Many levels of destruction
  - Tools like whole disk encryption are useful if applied properly and if the keys are suitably destroyed.

Privacy – Retention Policies

- PII (personally identifiable information)
  - Is like toxic waste
  - Don't keep it if you can avoid it
- Regulations
  - Vary by Jurisdiction
  - But if you keep it, it is “discoverable”

The future of Privacy

- Who’s data is it anyway
  - Should PII carry tags that limit its use.
  - How do we enforce that such tagged policies are actually followed.

CSci530: Security Systems

Lecture 14 – December 2nd, 2016
Security for Critical Infrastructure and Cyber-Physical systems

Dr. Clifford Neuman
University of Southern California
Information Sciences Institute
Critical Infrastructure

- Critical
  - Compromise can be catastrophic
  - Existing approaches to protection often based on isolation.
- Infrastructure
  - It touches everything
  - It can’t be isolated by definition
- Smart (or i- or e-)
  - We can’t understand it
  - The Cyber Components can’t be isolated.

Outline

- The Power Grid is Federated
  - Even more so for the Smart Grid
- Security Depends on Defining Boundaries
  - Cyber and Physical Domains
- Resiliency of the Power Grid
  - Operational Resiliency
  - Resiliency of Individual Functions
- Using Smarts to Improve Resiliency
  - Redefining Boundaries

Trends in Power Systems

- Evolution of power distribution
  - Local power systems
  - Interconnected
  - More centralized control
  - Automated reaction to events
  - Reaching into the neighborhoods
  - Encompassing the home
Federated Systems

• Characteristics
  – Parts of the system managed by different parties.
  – Lack of physical control in placement of components
  – Lack of a common set of security policies
  – The “administrative” dimension of scalability

• The Power Grid is Naturally Federated
  – Other Federated Systems
    • The Financial System
    • Cloud Computing
    • The Internet in general

Federation in Power Systems

• Power systems span large geographic areas and multiple organizations.
  – Such systems are naturally federated

• Avoiding cascading blackouts requires increasingly faster response in distant regions.
  – Such response is dependent on network communication.

• Regulatory, oversight, and “operator” organizations exert control over what once were local management issues.
  – Staged power alerts and rolling blackouts

• Even more players as the network extends to the home.
  – Customers
  – Information Providers

Meaning of Security for C-P Systems

• In traditional cyber systems, emphasis on:
  – Confidentiality and Integrity

• In Cyber-Physical systems much greater emphasis on:
  – Resiliency (one example of “availability”)
  – The consequences of failure are greater.

• Interrelation of Integrity with Resilience

Understanding Securability

• Security is About Boundaries
  – We must understand the boundaries
  – Containment of compromise is based on those boundaries

• Federated Systems Cross Boundaries
  – Federation is about control
    • And the lack of central coordinated control
    • By definition, we can’t control parts of the system.
  – Protecting such systems requires constraints at the boundaries.
Securing the Power Grid

- **Traditional Security**
  - It’s about protecting the perimeter.
  - Imposing policy on ability to access protected resources.

- **In Federated Systems**
  - The adversary is within the perimeter.
  - There are conflicting policies.

- The failure lies in not defining the perimeter
  - Or more precisely, in choosing the wrong one
  - Allowing the boundaries to change
  - Not implementing correct containment at the boundary

Threat Propagation

- Modeling can help us understand how threats propagate across domains.
  - There are several classes of propagation to be considered, based on the domains that are crossed.
    - Cyber-Cyber
    - Cyber-Physical
    - Physical-Cyber
    - Physical-Physical
    - And transitive combinations.

Cyber-Cyber Threats

- Cyber-Cyber threats (traditional cyber security)
  - Easily scaled (scripts and programs)
  - Propagate freely in undefended domains
  - We understand basic defenses (best practices)

Cyber-Physical Threats

- Cyber-Physical threats (physical impact of cyber activity)
  - Implemented through PLC
    - or by PHC (social engineering)
    - or less direct means (computing power consumption)
  - Physical impact from programmed action
  - But which domain is affected (containment)
Physical-Cyber Threats

- Physical-Cyber threats (impact to computing)
  - For example, causing loss of power to or destruction of computing or communication equipment.
    - A physical action impacts the computation or communication activities in a system.
  - Containment through redundancy or reconfiguration
    - Standard disaster recovery techniques including off-site backup, and even cloud computing.
  - Still need to expect
    - Computing supply chain issues and hardware provenance (counterfeit products, or changes during fabrication).

Physical-Physical Threats

- Physical-Physical threats (propagation of impact)
  - Traditionally how major blackouts occur
    - Cascading failure across domains
    - System follows physics, and effects propagate.
  - Containment is often unidirectional
    - A breaker keeps threat from propagating upward
    - But it explicitly imposes the impact downward
    - Firewalls and circuit breakers have analogies in many problem domains (including the financial sector)
    - Reserves often necessary for containment
  - Such containment in problem specific areas often protects against only known threats.

Transitive Threats (example)

- Dependence on unsecure web sites as control channels.
  - End customer smart devices (including hybrid vehicles) will make decisions based on power pricing data.
    - Or worse – based on an iPhone app
  - What if the this hidden control channel is not secure
    - Such as a third party web site or Smart Phone viruses
  - An attack such control channels could, for example, set pricing data arbitrarily high or low, increase or decrease demand, or directly controlling end devices.
    - Effectively cycling large number of end devices almost simultaneously.

Transitive Threats

- More interesting real-world threats combine the binary threats for greater impact.
  - Cyber-Physical-Physical
    - Multiple Chevy Volts’s controlled from hacked smart phones.
  - Cyber-Physical-Cyber (CPC)
    - Controlling device on HAN that causes meter to generate alerts creating DOS on AMI network.
  - Physical-Cyber-Physical (PCP)
    - Leverage Cyber response, e.g. 3 Sensor Threshold for fire suppression system.
The Correct Perimeters

- Systems can be secure for a particular function
  - We need to define perimeters for particular functions
- In the Power Grid
  - Billing and Business operations are one function
  - SCADA and infrastructure control are another.
  - In the smart grid, customer access and HAN control a third

Changing Boundaries

- Federated systems change over time
  - They evolve with new kinds of participants
    - E.g. Power grid → Smart Grid
    - Now the customer is part of the control loop
  - New peers join the federation
    - Not all my be as trusted
    - An adversary could acquire an existing participant
  - Mis-guided public policy could require expansion of protection domains.
  - This is why a monolithic security domain will not work.

Containment

- Containment techniques must be appropriate to the boundary and the function to be protected.
  - Firewalls, Application Proxies, Tunnels (VPN’s) suitable in the Cyber Domain.
  - Cyber-Physical boundaries require different techniques.
    - We must understand cyber and physical paths
    - We must understand the coupled systems of systems impact of faults originating in single domain.
    - We must understand the C-P impact of Cyber attack automation
  - We need to group similar, yet distinct protection domains.

Understanding Resilience

- Operational Resilience is the capability of a system to fulfill its mission in a timely manner, even in the presence of attacks or failures.
  - The definition also usually includes the ability of the system to restore such capability, once it has been interrupted.
  - A system performs many functions and operational resilience is a function of functional resilience of different aspects of the system.
  - The function depends on domain understanding (especially time-scales)
Resilience of the Smart Grid

Resiliency is the ability of a system to operate in spite of attacks and failures
- Operational resiliency concerns a system’s ability to meet its established service capabilities – for power grids, this is the ability to continue to deliver power.
  - Operational resiliency depends on resiliency of particular underlying functions.
  - Balancing generation/supply with load (within boundaries)
  - Communication and control.
  - Billing
- The Smart Grid provides both challenges and opportunities for resiliency.

Smart Grid Communication

- SG functions dependent on communication
  - Meter reads for billing
  - Meter reads for situational awareness
  - Remote connect/disconnect
  - Load management (curtailment, etc.)
  - Detection, Diagnosis, and Remediation
- In one SG project communication may be over
  - An AMI communication path, or
  - An Internet communication path

Meeting Resiliency requirements

Resiliency for a particular function – Dependencies are Domain Specific
- Demand Response
  - Ability to curtail required KWh of load in bounded time
- Meter reads
  - Ability to receive meter data within a larger bounded time (on the order of days, to months)
- Meter Outage Management
  - Ability to identify and diagnose outages as they occur
- Broader Situational Awareness and Attack Detection
  - Ability of attacker to block communication of events
- Operational Resiliency
  - Providing sustainable power delivery

Smart Grids are Less Resilient

- Dependence on less reliable parts of system
  - Efficiencies and new capabilities are enabled through reliance on parts of the system outside the utilities control.
    - Internet communication for demand response
    - External sites for energy forecasts, pricing
    - Remote control of EV Charging, remote disconnect
    - Billing
- Failures or compromise of new capabilities can impact Operational Resilience
  - Providing sustainable power delivery
Smart Grids are More Resilient

- Automation and redundancy can mitigate impacts of failures within the system.
  - Multiple communications paths through Internet, and AMI.
  - Demand response can provide new “reserve” capacity.
  - “Distributed Generation” may be closer to loads.
  - Improvements in energy storage can increase timescales over which load and supply must be balanced.
- But reliance on these technologies makes the system more dependent on the communication and IT infrastructure.
  - Which becomes a point of attack on the system.

Functional Resilience

- In the Smart Grid functions include:
  - SCADA
    - Impact of failure in seconds
  - Demand Response
    - Impact of failure depends on reserves, but order of minutes to hours
  - Billing and administrative
    - Impact of failure on order of months.
  - Home automation, customer “features”
    - Little impact of failure

How resilience used to be achieved

- Availability has always been a critical service for power control networks and C-P systems.
  - The control network for interconnects was managed separately.
    - Sole purpose was to exchange commands and information needed to keep the system functional.
  - Integrity and confidentiality was provided through limited physical access.

Securing the Smart Grid

- We must recognize that complete physical separation is no longer possible.
  - Because the Smart Grid extends into physically unsecure areas.
- Thus we must provide isolation through technical means.
  - We must define protection domains.
  - Improve support in the hardware, OS, and middleware to achieve isolation.
  - Design the system to identify policy on control flows so that Smart Grid components enforce it.
**Modeling a Secure Smart Grid**

- As a security problem, we need to model Smart Grid security using an adversarial model.
  - Traditional security limits information and control flow within the cyber realm.
  - For the Smart Grid we must model physical pathways.
    - E.g. effects of tripping a breaker in one part of a system will have effects in another part, independent of the cyber communication between them.
    - These causal relationships should be modeled as information and control channels.
  - Procedures and processes in the physical realm convert information channels into control channels.

**Securing the Smart Grid**

- Domain and security experts should identify all classes of sensors, actuators, and potential measurement points in the system.
  - Decide how each is associated with control and information channels.
  - Identify the other parties on the channel.
  - Identify security services needed for the channel.
    - Confidentiality
    - Integrity
    - Availability / Performance Isolation
    - Access Control
    - Anomaly Detection / Intrusion Detection
    - Trust Management

**Summary**

- The Smart Grid extends to homes & businesses
  - New security implications for such connections.
  - Hidden control channels.
- Critical and non-critical functions will not be separate
  - Availability is critical – Defined as Resilience
  - Performance isolation needed for critical communication.
- The federated nature of the smart grid demands:
  - Federated architectures to secure it.
  - Federated systems to model it
- Existing security for the power grid does not address the implications of the new architecture.
  - Containment Architecture is Needed
  - Many domains based on participants, and physical structure of the system.
- Resiliency is Key
  - Reconfigurability and Islanding can help
Defining The Cloud

- The cloud is many things to many people
  - Software as a service and hosted applications
  - Processing as a utility
  - Storage as a utility
  - Remotely hosted servers
  - Anything beyond the network card
- Clouds are hosted in different ways
  - Private Clouds
  - Public Clouds
  - Hosted Private Clouds
  - Hybrid Clouds
  - Clouds for federated enterprises

Risks of Cloud Computing

- Reliability
  - Must ensure provider’s ability to meet demand and to run reliably
- Confidentiality and Integrity
  - Service provider must have their own mechanisms in place to protect data.
  - The physical machines are not under your control.
- Back channel into own systems
  - Hybrid clouds provide a channel into ones own enterprise
- Less control over software stack
  - Software on cloud may not be under your enterprise control
- Harder to enforce policy
  - Once data leaves your hands

Defining Policy

- Characterize Risk
  - What are the consequences of failure for different functions
- Characterize Data
  - What are the consequences of integrity and confidentiality breaches
- Mitigate Risks
  - Can the problem be recast so that some data is less critical.
    - Redundancy
    - De-identification
  - Control data migration within the cloud

Controlling Migration

- Characterize Node Capabilities
  - Security Characteristics
    - Accreditation of the software for managing nodes and data
  - Legal and Geographic Characteristics
    - Includes data on managing organizations and contractors
  - Need language to characterize
  - Need endorsers to certify
- Define Migration Policies
  - Who is authorized to handle data
  - Any geographic constraints
  - Necessary accreditation for servers and software
    - Each node that accepts data must be capable for enforcing policy before data can be redistributed.
  - Languages needed to describe
Enforcing Constraints

- With accredited participants
  - Tag data and service requests with constraints
  - Each component must apply constraints when selecting partners
    - Sort of inverting the typical access control model
- When not all participants are accredited
  - Callbacks for tracking compliance
  - Trusted computing to create safe containers within unaccredited systems.

Cloud Security Summary

- Great potential for cloud computing
  - Economies of scale for managing servers
  - Computation and storage can be distributed along lines of a virtual enterprise.
  - Ability to pay for normal capacity, with short term capacity purchases to handle peak needs.
- What needs to be addressed
  - Forces better assessment of security requirements for process and data.
  - Accreditation of providers and systems is a must.
  - Our models of the above must support automated resolution of the two.

Review - Topics

- Cryptography
- Key Management
- Identity Management (and Authentication)
- Policy (and Authorization)
- Attacks
  - Classic
  - The human element
- Defenses
  - Firewalls, Intrusion Detection and Response, Encryption, Tunnels, Defenses to Malware
- Architectures and Trusted Computing
- Cyber-Physical and Cloud Computing
Glossary of Attacks
This is not a complete list
• Availability
  – Denial of Service (DoS AND DDoS)
  – Over consumption of resources
    – Network, ports, etc
    – Take down name servers, other critical components
  – Exploits to crash system
  – Cache poisoning

• Confidentiality
  – Eavesdropping
  – Key Cracking
  – Exploiting Key Mismanagement
  – Impersonation
    – Exploiting protocol weakness
    – Discovered passwords
    – Social Engineering
    – Exploiting mis-configurations

• Integrity
  – Breaking Hash Algorithms
  – Exploiting Key Mismanagement
  – Impersonation
    – Exploiting protocol weakness
    – Discovered passwords
    – Social Engineering
  – Exploiting mis-configurations
  – Cache Poisoning

• Miscellaneous
  – Spam
  – Phishing
  – Malware attacks
    – Spyware
    – Viruses
    – Worms
    – Trojan Horse
  – Man in the middle
  – SQL Injection
  – Cross Site Scripting
Intrusion Detection and Trusted Computing

a) Why is signature-based intrusion detection poorly suited for the detection of zero-day attacks? (10 points)

b) What are the strengths of a network-based collector in an intrusion detection system? (5 points)

c) What are the strengths of a host- or application-based collector in an intrusion detection system? (5 points)

d) What is the difference between attestation and accreditation? (10 points)

e) Explain what it means to “Extend a PCR”? (5 points)

f) What is the function of the “endorsement key”, and how do we know that the correct endorsement key was used for the claimed function? (5 points)

Privacy and user Tracking

For each of the following techniques used to protect privacy or to breach user's privacy, match them with relevant terms or approaches used to either implement or defend against the technique. This is not a one-to-one mapping; more than one term may be relevant to a technique, and more than one technique may use the same term in its implementation or description. If you list a match that we are not looking for, but which is still correct, while you will not lose credit, you will not get credit either. You will lose a point if you associated an approach or technique with a threat that it is not effective against. There are more blanks in the page below than actual correct answers, so you do not need to fill in all the blanks.

1. Traffic Analysis
2. User tracking
3. Data mining / inference
4. Spyware (including unexpected functions in installed software)
5. Linkability
6. P3P, DoNotTrack, and Privacy Policies

i. Cookie
ii. Anonymization
iii. Onion Routing
iv. User Education
v. Personally Identifiable Information
vi. Encryption
vii. Aggregation
viii. User location

Securing your own IT Infrastructure (40 points)

- You have a paranoid streak and have gotten tired of relying on service providers to secure your information. You are no longer willing to depend on someone else the cloud for backup and storage and you are determined to set up your own IT infrastructure to manage your own data. Fortunately, there are now a large number of products available that can assist you in doing just that. Unfortunately, many of these products leave some inherent vulnerability in your resulting system. In this problem, you are going to explore those issues and begin to understand just how hard it is to make your system truly secure.

The requirements for your system are:

i. You will support a file system (or file systems) capable for storing at least 2 TB of data. Some of this data you consider to be highly sensitive (e.g. tax returns, credit card statements), some is critically sensitive such as passwords and encryption keys, while other data is less sensitive, and you will want to be able to access such less sensitive information with other users on the Internet. There will be data of intermediate sensitivity, which you want to be able to access while away from your home, but which you do not plan to share with others.

ii. You require the ability to backup your data, including support for periodic off-site backup of data.

iii. Your home network supports many “appliances” including security cameras, DVR systems such as Tivo, Televisions, Entertainment systems, and home automation systems capable of controlling lights and unlocking doors.

iv. Your network supports multiple home computers, including tablets, smartphones, laptop computers, and desktop computers.

v. You have a single connection into your network through a cable modem, DSL, or FiOS or similar capability, and you will deploy a router and wireless system for your network. At this point, I could ask the single question, how will you secure this system, and you could write 200 pages and the question would be impossible for us to grade. As such, I can't ask such an open ended question and instead ask a few specific questions which by no means cover the entire space of options.
2013 Final Exam – Q3.3

a. In designing the network that will meet the requirements about file systems above, how will you protect the critically sensitive information differently than the other classes of information? How will you share the less sensitive information with other users on the internet? How might you support your own personal access to data of intermediate sensitivity, which you need to access when traveling? How will you protect the highly sensitive data, which needs to be readily accessible from your computers when you are at home? (10 points)

b. I mentioned that your network will have a router, most likely at the point of connection to the internet, but which is also responsible for forwarding packets among the other devices on your network. Tell me what capabilities you will require on this device, in order to improve the security of your home network as a whole. Please be sure to note that while it will obviously have firewall functionality, there should be a lot more that it does too. (10 points)

c. Defense in Depth – There will inevitably be security vulnerabilities on the devices in your home network. Group the devices into classes based on the impact of a device vulnerability on the security of the system as a whole, explain the impact, and describe how you can reduce the impact (or if aspects of your design above already reduce that impact, explain how it does so). (10 points)

d. System Updates – With all the devices listed above, you are certain to require software updates for many of these devices. Discuss for which devices you are likely to enable automatic software updates, explain any vulnerabilities created by said choice, and how the impact of those vulnerabilities might be mitigated elsewhere in the system. Understand that for some of these devices, you will not be able to change the way updates are processed or validated, but can only enable or disable automatic updates. (10 points)
Security in Routing

• Routing is a peer to peer system
• Topology is dynamic
  – (otherwise we would not need routing protocols)
• Routing is Transitive
• Security through Signing updates
• Policy is the hard part
• Systems SIDR, SBGP, etc

IP Traceback

• IP Addresses are spoofable
  – Difficulty depends on next level protocol
• How can we mitigate this effect
  – Ingress filtering
  – IP Traceback techniques
  – Only effects certain address spoofing, not relays

Mobile Devices

• Characteristics
  – Resource limited
  – Intermittent connectivity
    • Offline operation

Battling Bot-nets

• Detection
  – Finding the control panel
  – Learning what they do
• Response
  – Isolation/quarantine
Security For Middleware

- DCOM, CORBA, RPC, etc
- Issues
  - Authentication in underlying protocols
  - Confidentiality and integrity
  - Delegation
  - Management

Honey

- Honeypots
  - Looks like interesting system
- Honeynets
  - Dynamic Virtualization
- Honeytokens
  - Setting a trap

Outside Looking In

- How do we get out from an infected system.
  - Boot off CD
  - Mount drive on analyzer, etc.

Ecommerce Security

- Security of Trading Platform
  - Protecting the user
  - Protecting the company
  - The Untrusted Merchant
- Auctions
  - Fairness
- Payment Security
Ecommerce: Trading Platform

• Traditional platform security
  – Move critical data off server
• Use third parties to avoid need to collect critical customer data.

Ecommerce: Fraud

• Often external to system
  – Use of stolen credit cards
  – Drop locations for shipping
• Advertising fraud
  – Pay-per impression/click/action
  – Commission hijacking

Ecommerce: Auctions

• Typical real-world auction fraud techniques apply.
• Online issues
  – Denial of service
  – Visibility of proxy bids

Ecommerce: Payment

• Secure, reliable, flexible, scalable, efficient, and unobtrusive payment methods are required as a basic service of the Internet and must be integrated with existing and evolving applications.
Reliability

- Commerce will depend on the availability of the billing infrastructure.
- The infrastructure may be a target of attack for vandals.
- The infrastructure must be highly available and should not present a single point of failure.

Scalability

- The payment infrastructure should support multiple independent accounting servers and should avoid central bottlenecks.
- Users of different accounting servers must be able to transact business with one another and the funds must be automatically cleared between servers.

Efficiency

- Frequent payments for small amounts must be supported (micropayments).
- Performance must be acceptable, even when multiple payments are required.
- Merchants and payment servers must be able to handle the load.
- Per transaction cost must also allow small payment amounts.

Unobtrusiveness

- Users should not be constantly interrupted to provide payment information.
- However, users do want to control when, to whom, and how much is paid.
- Users must be able to monitor their spending.
Integration

- Payment systems must be tied to the existing financial infrastructure.
- Applications must be modified to use the payment infrastructure.
- Payments should be supported by common protocols that underlie applications.
- A common framework should support integration of multiple payment methods.

Multiple forms of payment

- Secure presentation
- Customer registration
- Credit-debit instruments
- Electronic currency
- Server scrip
- Direct transfer
- Collection agent

Secure presentation (and non-secure variant)

Uses traditional credit card numbers
  - As safe as the phone (cordless?)
  - Potentially huge customer base
  - Little need for infrastructure

Examples - products based on:
  - Secure Sockets Layer
  - SHTTP

Issues
  - No customer signature
  - Legitimacy of merchant
  - Real time authorization
  - Transaction cost

Customer registration

- Customers register and receive passwords, keys, or new account identifiers
  - Transactions clear through financial service provider who gateways to existing financial system (credit cards or checking accounts)
  - Protects external account information

Examples:
- First Virtual
- CyberCash
- SET

Issues:
- Security of system specific credentials
- Real time authorization
- Transaction cost
Credit-debit instruments

Financial service provider maintains accounts for customers
- Authorized individuals spend from account.
- Payment instrument authorizes transfer.
- Modes: credit like credit card, debit like checks
- Requires new infrastructure

Examples:
- USC’s NetCheque
- CMU’s NetBill
- FSTC Electronic Check Project

Issues
- Security of system specific credentials and instruments
- Aggregation and tie to financial system
- Durability of account information and of provider

Electronic currency

Users purchase currency from currency servers. Currency is presented to merchant who turns it in to currency server.
- Potential for anonymity
- Possible off line operation

Examples:
- Mondex
- NetCash
- DigiCash
- Various stored value cards

Issues
- Backing of the currency
- Level of anonymity
- Tamper resistance of hardware
- On-line vs. off-line
- Who’s at fault for counterfeiting
- Storage requirements
- Extensive matching capabilities required

Server scrip

• Payment instrument spendable with individual merchants.
  - Verification of scrip is a local issue
  - Requires a market and other forms of payment to enable purchase of merchant script.

• Examples:
  - Millicent
  - Payword

• Issues:
  - Aggregation of purchases improves performance
  - But must manage many kinds of currency

Direct transfer

• Customer initiates transfer of funds to account of merchant
  - May result in instrument sent externally

• Examples:
  - Most on-line bill payment mechanisms

• Issues
  - Matching of payment to customer or transaction
  - Account management similar to credit-debit model
Collection agent

- Merchant refers customer to third party who collects payment and provides receipt.
  - Receipt is presented to merchant who then provides the goods or services.
- Examples:
  - OpenMarket payment switch
- Issues
  - Third party implements the payment methods
  - Issues are the same as for methods supported

Some representative systems

Available today
- Secure Socket Layer
- CyberCash
- SET
- Open Market

Trials
- Mondex

Demonstrated, Research
- FSTC Electronic Check
- NetCheque
- NetCash
- NetBill

No longer with us
- First Virtual
- DigiCash

Secure socket layer (secure presentation)

- Merchant has certified public key
- Client submits form with credit card information to merchant encrypted
- Merchant obtains authorization for credit card in same manner as for phone order
- Availability: NetScape Commerce Server, IE, Apache, OpenMarket, Others, (Verifone)

First Virtual (customer registration)

- Customer establishes First Virtual account
  - Customer sends account ID to merchant
  - Merchant forwards to FV server
  - FV server verifies through e-mail to customer
    - Customer can refuse payment to merchant
      - If too frequent, customer loses account
- Issues:
  - Does not use encryption
    - No changes to client software
    - Minimal changes needed for merchant
    - Known compromise scenario, but of limited use
  - Exposure limited by delaying payment to merchant (waived for vetted merchants)
- Availability: FV (now MAIL) no longer does payments,
  Customer base sent to CyberCash
CyberCash (customer registration)

- Customer registers credit card with CyberCash and selects signature key
  - Special software running on client encrypts and signs credit card number and transaction amount and sends to merchant.
  - Merchant forwards to CyberCash server which obtains authorization and responds to merchant.

- Issues:
  - Credit card number not exposed to merchant
  - Payment clears through credit card system
  - Will adopt SET for credit card payment
  - CyberCoin for “micropayments”

- Availability: http://www.cybercash.com
  Core commercial product is different than described here; does credit card authorizations for merchants.

DigiCash (electronic currency)

- Software implementation of electronic currency providing unconditional anonymity
  - Special software on client implements electronic wallet to store and retrieve currency.
  - Online detection of double spending
  - Post-fact tracking of double spending

- Availability: http://WWW.DigiCash.COM
  - In Chapter 11 reorganization (11/4/98)

Secure Electronic Transactions (SET)

- Customer obtains signature key from card issuer
  - Special software running on client encrypts and signs credit card number and transaction amount and sends to merchant.
  - Merchant forwards to acquirer which processes transaction through credit card system and responds to merchant with authorization.

- Advantages
  - Certification of customer and merchant
  - Credit card number not exposed to merchant

- Disadvantages
  - Slow encryption
  - In practice, many are dropping the customer registration requirement

- Availability: Part of product offerings by others

Open Market (collection agent)

Provides multi-mechanism collection services for web browsers.
- Payment is made to Open Market payment switch.
- Switch authorizes delivery of goods.
- Added value provided to customer through “smart statement”.

Availability: http://www.openmarket.com

OM

M

B
Mondex (electronic currency)
- Provides smart-card based electronic currency for point of sale and card to card transactions
  - Currency can be accepted off-line
  - Uses a tamper resistant smart card
  - Card signs transactions, so no anonymity
  - Card-to-card transactions using “wallet”
  - Smartcard reader needed to use on network
- Availability: several pilots underway, not available yet for Internet transactions

Electronic Check (Credit-debit)
- Electronic check provides credit-debit payment instruments that can be sent across the Internet, but which clear through existing banking networks (e.g., ACH)
  - Instrument authenticated using public key cryptography and digital signatures
  - PCMCIA “electronic checkbook” protects keys
  - Trial expected in 1997.

USC/ISI NetCheque® (credit-debit)
- Implements on-line “checking-account” against which payments are authorized.
  - No prior arrangement between customer and merchant.
  - A check authorizes the payee to transfer funds from the payor’s account.
  - Multiple currencies per account.
  - Payments clear through multiple payment servers.
- Availability as research prototype: http://www.netcheque.org

Flow of NetCheque Payment Instrument
NetCheque representation

- Internal representation is opaque
- Important fields:
  - Account and accounting server
  - Amount, payee, expires
  - Customer and merchant info
  - Signatures and endorsements
- MIME encoded for use by applications
- Applications display checks according to their own requirements.
  - Display check makes it look like check
  - Statement displays one line per check
- Statement API returns entire check with endorsement
  - Allows easy import of information from check into users financial applications.

NetCheque Payment Instrument

```
--NetCheque(SM)V1.0
Content-Type: APPLICATION/X-NETCHEQUE
Content-Transfer-Encoding: BASE64
Content-Description: Pay 10.00 NCU to marketplace@NETCHEQUE.ISI.EDU
AAAAAQAAAA5OZXRDaGVxdWVfVjEuMAAAAA1TT0ZUV0FSRV9HMS64xAAAAAQED
NTE4AzI2N2GCAQcwggEDoAMCAQWhExsRTkVUQ0hFUVVFVRS5JU0k=
S3RNZ10kKQf1eJLo29MlU5G2EAAAA==
--NetCheque(SM)V1.0--
```

NetCheque security

- Check has plaintext part and signature
- Endorsements are separately signed and linked to a particular check
- Signature component is modular
  - Current implementation is Kerberos proxy
    - Signature verifiable by customer’s bank
  - Can accommodate RSA or DSS signatures

Clearing funds through multiple servers
USC/ISI's NetCash

- Users purchase currency from currency server using NetCheque - deposits to currency server's account back the currency
- Supports weakly anonymous payment
  - Cash can be exchanged for new cash anonymously
  - Customer chooses the currency server
- Multiple currency servers, the NetCheque system is used to clear cross-server payments

Offloading the risks

- Limiting exposure to risk
  - Credit vs. debit model for accounts
  - Deferring payment to merchants
- Shifting risk to other parties
  - Agreements shifting risk to merchant
  - Regulations protecting the consumer
  - Insurance - perhaps as higher transaction fees

Technical solutions

- Protecting payment credentials
  - Token cards
  - Smart cards
- On-line authorization
  - Detects double spending
  - Checks for sufficient funds
  - Enables checks for spending patterns
- Tagging documents

Hypothetical Case Studies

- Past exams
  - Electronic voting (Fall 2004)
  - Medical records (Fall 2003)
  - Security for the DMV (Fall 2008)
  - Cloud Based File Store (Fall 2011)
Electronic Voting

You have been asked to design a system to support the collection and counting of votes for the next election. In particular, you have been asked to design a system that will accurately tabulate votes entered by voters at polling places throughout the state and to transmit those votes to the county clerk of each county where the totals will be tabulated.

(a) Threats. What are the threats in such a system? What can go wrong?
(b) Requirements. What are the requirements for authentication, authorization, assurance, audit, and privacy? Explain who and what must be authenticated, what authorizations are required, what assurance is needed for the software, and what kind of records must be maintained (as well as what kinds of records should not be maintained).
(c) Considering the requirements listed above, and how they relate to the assurance problem, i.e. how can steps taken for authentication, authorization and audit be used to ensure that the software has not been modified to improperly record or transmit votes?
(d) What technologies proposed for digital rights management be used to provide stronger assurance that the system’s integrity has not been compromised. What is similar about the two problems, and how would such technologies be applied to the voting problem.

Medical Records

You have been hired as a consultant to advise on the design of a security mechanism that will be used to protect patient data in a new medical records system. This system will manage and support the transmission of patient records, including very large images files for X-rays, MRI, CAT-scans and other procedures. The system must provide appropriate levels of protection to meet HIPAA privacy regulations, and it must allow the access to records needed by physicians and specialists to which patients are referred.

(a) Describe appropriate requirements for confidentiality, integrity, accountability, and reliability/availability in such a system.
(b) In what part(s) of the system (e.g., where in the protocol stack would you include support for each of the requirements identified in (a)? Why would you place mechanisms where you suggested; what were the issues you considered?
(c) What security mechanisms and approaches to implement those mechanisms would you use to meet the requirements in (a) as implemented in the parts of the system you identified in (b)?

Security for the DMV - 2008

(30 points) Design Question – You have been hired by the state of California to improve the security of the computer systems at the department of motor vehicles. Much of the information in the system is sensitive and it will be important to limit access to this data, not just by the general public, but also to maintain strict accountability for access by DMV and law enforcement employees themselves.

Given the large number of terminals throughout the state (including those in patrol cars) from which such data is accessible, you have been asked to consider approaches that will prevent data from being downloaded and then transferred to other computer systems outside of the state’s network.

a) Describe the data to be protected in such a system and suggest the policy that should be applied for each class of data i.e. who can view it and who can modify it. (10 points)
b) Suggest techniques that can be applied to prevent mis-use of the data by insiders, i.e. those that might have authorization to access the data according to the policies implemented by the computer systems, but who might not have legitimate need to access the data. (5 points)
c) Suggest techniques that could prevent the data from being accessed by malicious code that might end up installed on, and having infected, terminals in the system. (10 points)
d) Suggest techniques that would prevent data from being downloaded from the system and then transferred to other external systems over which the access controls to the data might not be enforced. (10 points)

2011 Final Design Problem

(40 points) – Security in a Cloud Based File Store

You have been hired to redesign the security mechanisms for a cloud based file service (similar to Dropbox). Your main concern is ensuring the confidentiality and integrity of data stored in the cloud. Ideally, files stored in the cloud will only be readable to authorized users, and not accessible to others including employees of the cloud storage company itself.

Files stored in the cloud will be accessed by their owner on various devices, including desktop and laptop computers, smartphones, and from the web. Certain “shared” directories (and the files they contain) may be accessible to selected other users with whom the owner has chosen to share a directory. Files should remain accessible to authorized users on their devices even when the users are disconnected from the network. The owner of a shared file or directory must be able to revoke access to other users that were previously authorized.