Post 1

Friday, January 25, 2019 12:18 AM

INFOSEC NOTES 1: RISK WEIGHING/ASSESSMENT

September 8, 2016 Moo Comments 0 Comment

Risk Assessment: AKA: Risk analysis | Risk Calculation: Deals w/threats/vulns | Impacts loss of info

Vulnerability: Weaknesses that can be exploited by threats

Components of risk-assessments:

Risks an org is exposed to	○ Develop various scenarios that help evaluate how to deal w/risks ○ Assumes "what-if's" & plans on how to deal w/them
Risks that need addressing	▪ Reality checks that provide risks on "likelihoods" of them occurring ▪ It provides an assessment on what's real/unlikely Example: Industrial espionage/theft have a higher chance of occurring ○ A hurricane hitting an area of NYC doesn't
Coordination w/BIA	○ **Business Impact Analysis** ○ Allows an org to make intelligent decisions about various security scenarios

Computing Risk Assessment: You should determine the impact an event could have

SLE x ARO = ALE

ALE: Annual Loss Expectancy value: Monetary measure of how much loss expected in a year.

SLE: Single Loss Expectancy value: How much you expect to lose at any one time

SLE can be divided into 2 parts:

- **AV: Asset Value**
- EF: Exposure Factor

ARO: Annualized Rate of Occurrence: The likelihood of an event happening w/in a year

- Typically drawn from past data

Single Loss Expectancy x Annualized Rate Occurrence = Annual Loss Expectancy

Quantitative vs. Qualitative Risk Assessment

Qualitative: Opinion-based || Subjective	**Quantitative:** Cost-based || Objective

Formulas for calculating **SLE/ALE/ARO** are quantitative b/c they lead to dollar amounts

Example:

Company loses their mission statement: It's sad panda, but it doesn't cost

them money
Company loses important data connected to accounts: Much WORSE because it also costs them

Additional Risk Terminology

Likelihood:

NIST: National Institute of Standards & Technology:

- View it as a score representing the possibility of threat initiation
- Helps explain it in quantitative/qualitative terms
- The scale below helps put it into perspective

Likelihood Assessment Scale

Qualitative Values	Semi-Quantitative Values	Description
Very High	10	Adversary is almost certain to initiate a threat event
High	8	Adversary is highly likely to initiate a threat event
Moderate	5	Adversary is somewhat likely to initiate a threat event
Low	2	Adversary is unlikely to initiate a threat event
Very Low	0	Adversary is highly unlikely to initiate a threat event

Threat Vectors: The way in which an attacker exposes a threat

- Can be a tool used, exploit, or vulnerability scanner etc…
- It can be anything that's exploitable pretty much

MTBF: Mean Time Between Failures: Measures anticipated incidence of failure for a component

- Measures lifetime expectancy
- **If the component CAN be repaired**

Example: You realize a power supply should die w/in a year: The is the MBTF

MTTF: Mean Time To Failure: Average time to failure for non-repairable system

- If the system CANNOT be repaired

MTTR: Mean Time To Restore: How long it takes to repair a system/component once a failure occurs

- Does it take 24 hours to repair that system? Then the MTTR is 24 hours

RTO: Recovery Time Objective: Max amt of time a process/service is allowed to be down while consequences are acceptable

- Beyond this time: Break in business can affect it negatively

RPO: Recovery Point Objective: Defines the point at which the system needs to be restored

- Where the system crashed (needs complete redundancy)
- The closer the RPO matches the item of crash: The more expensive

Acting On Your Risk Assessment

5 possible actions you can choose to follow:

Risk Avoidance	Making a decision not to engage in actions associated with a risk Example: Company feels email attachments are too great a risk: Bans their use
Risk Transference	Sharing the burden of the risk w/someone else Example: An insurance company

Risk Mitigation	Any time you take steps to reduce risk Example: Installing AV SW/educating users/monitoring network traffic/firewalls/etc.. Suggestions for risk mitigation awareness: • Keep sec msgs in circulation • Target new employees/current staff • Set goals to ensure a high percentage of staff is trained on security practices • Repeat the info to raise awareness Can be done through the use of: • Routine audits/addressing user rights/permissions/changing incident management & data loss reports • Theft policies being effectively in place and tech controls that are enforced properly
Risk Deterrence	Understanding the enemy: Letting them know there are repercussions if they cause harm to you Example: Prosecution policies on login pages/MOTD banners/cameras
Risk Acceptance	A choice: When the cost of the other 4 choices exceeds value of harm that would occur • Has to be an identified risk

DLP: Data Loss Prevention: Monitoring contents of systems (workstations/servers/networks) to ensure key content isn't removed

- Also monitor who is using/transmitting the data
- DLP systems share commonality w/network intrusion prevention systems
- MyDLP: Open source solution that runs on most Windows platforms

Risks Associated w/Cloud Computing

Cloud computing: Hosting services/data on the Internet instead of locally | Example: Google Docs or Office 365

3 Ways of implementing cloud computing:

- **PaaS: Platform as a Service: AKA: Cloud platform services**

Vendors allow apps to be created/run on their infrastructure
Two well-known models: Amazon Web Services/Google Code

- **SaaS: Software as a Service: Applications remotely run over Web**

Advantage:

1. No local hardware required
2. No software applications need to be installed on the machine accessing the site
3. Example: Salesforce.com
4. Usually subscription based

- **IaaS: Infrastructure as a Service: Utilizes virtualization | Clients pay outsourcers for resources used**

1. Closely resembles traditional utility models: Electric/gas/water

Example: GoGrid

Risks associated with cloud computers:

Regulatory Compliance: Regulatory agency rules which must be complied with

- Can be difficult when data isn't located on your servers
- Ensure whoever hosts your data takes privacy/security seriously

User Privileges: Enforcing usr privs can be taxing

- Ensure usrs have least privs needed: Escalated privileges could allow unauthorized access to data
- Won't have same control over accounts in cloud as if locally
- If problem occurs: May have to wait on their technical staff

Data Integration/Segregation: Data hosting companies put more than one company's data on a server

- Use encryption to protect data
- Assume client DB is hosted on server that other companies also use
- If that company obtains root access/crashes: Could access your data
- Data segregation: Keep your data on secure servers
- Integration: Ensure data isn't comingled beyond expectation
- Ensure perms set properly

Risks Associated w/Virtualization

Virtualization: Allowing one set of hardware host multiple virtual machines

Breaking Out of the VM: Possible to break out of the virtualization layer/access other VM's

Network/Security Controls Can Intermingle: Tools may not have same granularity as those used to manage network

- Could lead to privilege escalation

Hypervisor: VM monitor: SW that allows the VM's to exist

- If Hypervisor can be successfully attacked: Can gain access to all VM's
- Been demonstrated possible on most systems: VMWare/Xen/MS VM
- Apply patches: Keep system up to date

Developing Policies/Standards/Guidelines

Standards	Tell us what's expected
Guidelines	Provide specific advice on how to accomplish given tasks/activities
Policies	Provide people w/guidance about expected behavior

Scope Statement: Outline of what policy intends to accomplish/which documents/laws/practices it addresses

- Helps readers understand what the policy is about/how it applies

Policy Overview Statement: Goal of policy/why it's impt/how to comply: Single paragraph

Policy Statement: Substance of policy: Clear/unambiguous as possible: Paragraph/bulleted lists/checklists

Accountability Statement: Additional info to reader about who to contact if

problems discovered
- Indicates consequences of not complying w/policy

Incorporating Standards Deals w/specific issues/aspects of business

5 key points of standards documents

• **Scope/Purpose**	Explain/describe intention • SW/updates/add-ins/relevant info that allows carrying out tasks
• **Roles/Responsibilities**	Outlines who is responsible for implementing/monitoring/maintaining standard
• **Reference Docs**	How standard relates to org's diff policies: • Connects standard to underlying policies in place
• **Performance Criteria**	How to accomplish task: Relevant baseline tech standards
• **Maintenance/Admin Reqs**	Outline what's required to manage/admin system/network

Following Guidelines: Help org implement/maintain standards: Provides info on how to accomplish policies/maintain standards
- Can be less formal than policies/standards
- Nature is to help users comply

4 items for a good guidelines doc:
1. **Scope/Purpose:** Overview/statement of guideline's intent
2. **Roles/Responsibilities:** ID's individuals responsible for accomplishing certain tasks
3. **Guideline Statements:** Step-by-step instructions on how to accomplish task
4. **Operational Considerations:** Specify/ID what duties required at what intervals
    - Daily/weekly/monthly tasks

Example: Sys backups

Guidelines help orgs in 3 ways:
1. If a process or set of steps isn't performed routinely: Experience support/security staff will forget how to do them
2. When you're training someone on something new, written guidelines reduce learning curve
3. A crisis or high-stress situation: Guidelines can help keep you from coming unglued

Business Policies to Implement Affect security: Address organizational/departmental business issues
- Not corporate-wide personnel issues

Separation of Duties Policies: Designed to reduce risk/fraud/prevent other losses to org
- Good policies require more than 1 person to finish

Collusion: Agreement between 2/more parties established for the purpose of committing deception/fraud

Privacy Policies: Define what controls required to implement/maintain sanctity of data privacy in work environment
- Restrictions regarding privacy addressed in legislation
- A legal doc that outlines how data is secured

- What info a company collects/privacy choices based on account
- Potential info sharing of data w/other parties/sec measures/enforcement

Last paragraph of policy: Should appear in every policy: Addresses that policy may change

AUP: Acceptable Use Policies Describe how employees can use company sys/resources both SW/HW

- Outline consequences of misuse
- Address installation of personal SW on company PCs/use of personal HW, like USB

Pod slurping: When portable devices are plugged directly into a machine, they bypass network security measures

- Can be done using free cloud drives instead
- Section on smart phone usage/presence workplace

Security Policies: What controls required to implement/maintain sec of systems/usrs/networks

- Should be used as a guide in system implementations/evaluations

Mandatory Vacations

Job Rotation

Least Privilege: Used when assigning perms: Give usrs only perms needed to do their work

Succession Planning: Those internal to org who have ability to step into other positions

- ID'ing key roles that can't be left unfilled | Associating employees who can step into them

Understanding Control Types/False Positives/Negatives

Risk assessment/analysis Calculating potential risks/making decisions based on vars associated w/those risks (likelihood)

- Once you ID risks: Address w/actions other than avoidance: Put controls in place to address those risks
- **NIST: National Institute of Standards and Technology:** Places controls into various types

The control types fall into 3 categories:

1. Management
2. Operational
3. Technical

Control Types/Controls:

Control Type	Controls	Control Type	Controls
Management	**Risk Assessment**	Management	**Planning**
Management	**Sys/Service Acquisition**	Management	**Cert/Accreditation/Sec Assessment**
Operational	**Personnel Sec**	Operational	**Phys/Env Protection**
Operational	**Contingency Planning**	Operational	**Config Mgmt**
Operational	**Maintenance**	Operational	**Sys/Info Integrity**
Operational	**Media Protection**	Operational	**Incident Response**

Operational	**Awareness/Training**	Technical	**ID/Auth**
Technical	**Access Control**	Technical	**Audit/Accountability**
Technical	**Sys/Comm Protection**		

False Positives: Events aren't really incidents: Anomalies: Don't declare emergency until you have one
- Alerted to situation that isn't really threat

False Negatives: You aren't alerted to situation when you should have been
Risk Management: Best Practices | Best Practices: What is known in the industry/consistently show superior results
BIA: Business Impact Analysis: Evaluating critical systems to define impact/recovery plans
- Not concerned w/external threats/vulns
- Focuses on impact loss would have

Identifying Critical Functions
- "What functions are necessary to continue operations until full service can be restored?
- Establishes which systems must be restored for business to continue

Prioritizing Critical Business Functions: Operations must be prioritized as essential/non-essential functions
Calculating Timeframe for Critical Systems Loss:
"How long can they survive without a critical function?"
"Which functions must be reestablished during that time?"

Tangible loss: Loss of sales/production
Intangible loss: Loss of faith in service

A thorough BIA will accomplish 7 org goals:
1. True impact/damage an outage can cause is visible
2. Understanding true loss potential may help in fight for budget
3. Process will doc which business processes being used/impact they have/how to restore them quick

ID'ing Critical Sys/Components
SPOF: Single Point of Failure
HA: High Availability: Five Nines Availability: Measures to keep services/sys operational during outage
Redundancy: Duplicated systems of fail over to other systems in event of malfunction
Clustering: Involves multiple sys connected together cooperatively (load balancing)
- If any of the sys fail: Others pick up the slack

Fault Tolerance: Ability to sustain operations in event of component failure
2 key components of fault tolerance:
1. Spare parts
2. Electrical power

Example: Always have UPS (uninterruptable power supply)/backup generator

RAID: Redundant Array of Independent Disks: Uses multiple disks to provide fault tolerance | Several designations for RAID levels

RAID Level 0	**Disk striping:** Uses multiple drives/maps them together as single physical drive • Primarily done for performance (not fault tolerance) • If any drive in 0 array fails: Entire local drive becomes unusable
RAID Level 1	**Disk mirroring**: Provides 100% redundancy b/c everything stored on 2 disks • If 1 disk fails: Another continues to operate: Failed disk can be replaced/ and 1 array can be regen • Advantage: 100% data redundancy at expense of doubling storage reqs • Each drive keeps exact copy of all info: Reduces effective storage capability to 50% **Disk duplexing**: Some implementations of disk mirroring Mirroring: One controller card writes sequentially to each disk Duplexing: Same data is written to both disks simultaneously
RAID Level 3	**Disk striping w/parity disk:** Uses striping in conjunction w/separate disk that stores parity info Parity information: Value based on value of data stored in each disk location • Sys ensures data can be recovered in event of failure
RAID Level 5	**Disk striping w/parity:** Commonly used: Operates similarly to disk striping as RAID 0 • Parity info is spread across all disks in array instead of single disk • Requires min 3 disks/max 32

Disaster Recovery: Ability to recover operations after disaster

- Plan/design comprehensive backup plan that includes storage/procedures/maintenance

Backups: Duplicate copies of key info stored in location other than where it is currently | Paper/PC records

Disaster Recovery Planning: Keeping backups of key data files/db's/apps/paper records available

- Must develop solid set of procedures to manage/ensure protection

Key paper records that should be archived:

Board minutes	Board resolutions	Papers of incorporation	Critical contracts	Financial statements	Incorporation docs
Local dos	Personal info	Tax records			

Critical files that should be backed up:

Apps	Appt files	Audit files	Cust lists	DB files	Emails	Financial data	OS's	Prospect lists
Transaction files	Usr files	Usr info	Utilities					

Tabletop exercise: Sitting around a table w/a facilitator to discuss situations that could arise and how best to respond to them.

From <https://www.piratemoo.net/moosings/information-security/infosec-notes-1-risk-weighingassessment/>

Post 2

Friday, January 25, 2019 12:18 AM

MONITORING SYS LOGS

September 16, 2016 Moo Comments 0 Comment

Network Monitors: AKA Sniffers: Troubleshoot network problems

- Usually PC w/NIC running promiscuous mode

Promiscuous mode: NIC looks at any packets seen on network: Even if not addressed to that card

Windows Server: Service: Network Monitor: Used to gain basic info about traffic

- More detailed version: Included w/**SMS: Systems Management Server**
- 3rd party programs; *Wireshark: Good for monitoring*

Sniffer: Subject named B/C best monitor: Everyone started calling net monitoring HW sniffers

Monitoring System Logs: Must monitor event logs

Event Logs: Broad category that includes logs irrelevant to security issues

Windows 7 logs: 2 most impt:

Application log	**Various events logged by apps/programs** • Many apps record errors to this log • Useful on servers w/DB like SQL Server • Clues at attempts to compromise DB's
Security Log	**Most impt things in it: Successful/unsuccessful login attempts** • Records events related to resource use • Creating/opening/deleting files/other objects **Admins can specify which events recorded in log** • Logon auditing can be turned off: Shouldn't be **Access log** • Linux: Separate logs for successful/failed login attempts Default: Win doesn't log both successes/failures

Win: Doesn't create audit logs by name: Still useful

- SharePoint/SQL/Other services: Often call app logs audit logs

Linux logs:

/var/log/faillog	**Failed usr logins** • Tracking attempted cracks into system
/var/log/apport.log	**Records app crashes** • Attempts to compromise sys/viruses/spyware

Viewing Event Logs:

1. Start > Control Panel > Admin Tools > Event Viewer
2. Click error msg/read details

Options w/in Event Viewer:

- *Perform actions:* Save logs (.evt/.txt/.csv) | Open/filter/view/change properties in logs
- *Default:* Logs overwritten as space needed | Can config auto archiving

- Logs of services like DNS: Routinely examine

Understanding Hardening

Hardening: Term applied to OS's "*Locking down*" OS as much as possible

- **ALL** unnecessary services **OFF** | **ALL** unneeded SW **GONE** | Patches updated | Accts checked /etc…
- Trying to make sys as secure as possible

Working w/Services:

Services: Programs run when OS boots: Often running in BG w/out usr interaction

- Services: Can provide attack vector: Only use necessary/required
- Control Panel > Admin Tools
- Services not *"bad*": You should KNOW what you're running

File/Print Servers	**Primarily vuln to DoS/access attacks** • Can be targeted at specific protocols/overwhelm ports
Networks w/PC Based Systems	**Make sure NetBIOS disabled on servers** • Effective firewall in place • Popular attacks take place through NetBIOS services **Ports:** 135, 137, 138, 139 **Unix:** Port 111 (**RPC: Remote Procedure Call**)

RPC: Programming int that allows remote computer to run programs on local machine

Directory Sharing:	Limit sharing to what's essential: Root dirs should be hidden from browsing

Protecting Management Ints/Apps

- Good to log in as usr: Run things as Admin
- Access to management should be only for those who need it
- Even remove usr access to utilities like regedit

Performance Monitor (Sys Monitor): Tool for looking at possible illicit activity

- Passwd protection to protect management/functionality/consoles on workstations/servers

Software: Remove unnecessary SW

Patch: Update to sys: Functionality/bug fixes: FIRST to single machine/tested before applying

Service Pack: Periodic update: Corrects problems in vers of product

- Provides tools/drivers/updates/extended product functionality/etc..

Updates: Code fixes for products provided to customers when experiencing problems: No workaround

Security updates: Address vulns: Mandatory

User Account Control: Impt part of hardening: Only active accts should be working/managed

- Disable unnecessary accts (local included)
- Make sure accts all have passwds up to code
- No acct should have privs in excess of job function
- Disable accts of employees who leave: Temp employees too: Default Guest Accts: Disable

Filesystems in Win:

FAT	**File Allocation Table** • MS: Earliest FS: Small drives | Upgraded to FAT-16/then FAT32 **FAT32**: Larger disk systems to be used on Win sys **Only allows 2 types of protection access privs:** 1. Share-level 2. User-level If usr has Write/Change Access perms to drive/dir: Have access to any file in that dir • Very insecure
NTFS	**New Technology Filesystem** • Introduced w/NT to address sec problems ○ Before NT released: MS knew had to create FS w/growing disk sizes/sec concerns/file stability ○ FAT didn't do well when power went out/when sys crashed • **NTFS: Transaction-tracking system:** Made possible for NT to back out of disk ops in progress ○ Specifically when NT crashed/lost power ○ Files/dirs/vol can each have own sec | Flexible/built-in ○ Tracks sec in ACL's: Which can hold perms for usrs/groups ○ Each entry in ACL can specify what type access given (Read-Only/Change/Full Control) ○ File-encryption programs dev to encrypt data while stored on HDD ○ MS recommends network shares be established w/NTFS **To see ver: fsutil fsinfo ntfsinfo C:**

C$/admin$/etc…: Hidden admin shares for use in managing computer on network

- Perm disable only through Regedits
- Temp disable w/Comp Mgmt console: Returns on reboot unless perm disable w/Group Policy

FS Linux: ext3/ext4/XFS | **Macs:** HFS (hierarchical FS) | **HFS Plus: AKA HFS Extended**

Securing The Network: MAC Limiting/Filtering: Limit access to MAC addresses that are known

- Filter those that aren't: Not full proof: **ipconfig /all**

802.1X	Adding port auth to MAC filtering: Takes sec to switch port lvl **IEEE 802.1X:** • Defines port-based sec for wireless net access control • Offers means of auth/defines **EAP: Extensible Authentication Protocol** over IEEE 802 ○ **EAP: AKA EAP over LAN** or **EAPOL** AP's don't need to do auth: Instead rely on auth server
Disable Unused Ports	Ports are connections: Like channels • Example: SMTP: Port 25 Sometimes called app ports **All not in use disabled:** Otherwise possible open door for attacker Can block w/Firewall/disabling service (doing one w/out other can create single point of failure)
Rogue Machine Detection	Someone can add unauth machine: Could be intruder/neighboring office/employee/etc..

Security Posture: Impossible to evaluate sec w/out having baseline config

doc: Must represent secure state
Continuous Sec Monitoring: Once baseline sec config doc'd: Monitor/maintain
"That which gets measured gets improved"
Reg measurements of net traffic lvls/routine evals for regulatory compliance/checks of netsec device configs
Security Audits: Monitoring should take place on 7 lvls
Basic ongoing monitoring: Not labor intensive: SW solutions available | Implement scheduled/in-depth checks of sec
Security audit: Integral part of monitoring: Can be check of any aspect of sec including:

- Review of sec logs/policies/compliance/device configs/Incident response reports

Setting Remediation Policy: When gap in sec posture detected: Should 1st be classified/then remediation plan implemented

- Specifies how you classify/respond to gap

Minor	Doesn't pose immediate threat to sec
Serious	Could pose immediate threat: But unlikely/difficult to exploit
Critical	Immediate threat: Must be addressed ASAP

Reporting Sec Issues

Alarms	**Current ongoing problem** • Conditions to respond to now | Can indicate trends happening • Even after problem solved: Still look for indications condition may not be isolated
Alerts	**Slightly below alarms** • Issues to pay attention to: Not about to bring sys down at any moment
Trends	Trends in threats

Differentiating bet Detection Controls/Prevention Controls: Some sec controls implemented to detect potential threats

- Others designed to prevent/min threats

IDS: Intrusion Detection System: Detecting intrusion
IPS: Intrusion Prevention System: Preventing intrusion

- Various levels: Can be based on host (H-IDS) or network (N-IDS)
- Some tools fall bet 2: Example: Honeypot

Honeypot: Comp that has been designated as target for comp attacks: Allows itself to succumb to them

- Sys can be used to gain info about how attack dev/what methods used to institute it
- Draws attackers away from higher-value systems/allows admins to gain info
- Not normally secured/locked down
    - If misconfig: Can be used to launch attacks on other sys

Honeynet project: Synthetic network that can be run on single sys/attached to network using normal NIC

- Sys looks like entire corporate network: All info fake

Enticement	Process of luring someone into plan/trap • Advertising free SW/brag no one can break into machine | Inviting people to

	try
Entrapment	Process where law enforcement/gov encourages person to commit crime when individual expresses desire not to • Valid legal defense in criminal prosecution

Enticement: Legal in US: Entrapment isn't

From <https://www.piratemoo.net/moosings/information-security/monitoring-sys-logs/>

Post 3

Friday, January 25, 2019 12:19 AM

MASTERING TCP/IP/FIREWALLS/ETC

September 16, 2016 Moo Comments 0 Comment

OSI Relevance: TCP/IP precedes creation of OSI model: Carries same operations w/4L's instead of 7

TCP/IP Broken into 4 layers:

1. Application
2. Transport (Host-to-Host)
3. Internet
4. Network Access (Network Interface/Link Layer)

TCP/IP: Doesn't concern itself w/topology/physical connections:

- Controller that resides in comp/host deals w/physical protocol/topology
- Comms w/controller: Lets it worry about topology/phys connection

Host: Any device connected to network that runs TCP/IP protocol suite/stack

HTTP	SMTP	**Application**
TCP	UDP	**Transport/Host-to-Host**
IP		**Internet**
Network	Topology	**Network Access**

Encapsulation: Process used to pass msgs bet layers in TCP/IP

Application: Highest layer: Allows apps to access services/protocols to exchange data

Example: Web browsers

Most commonly used app layer protocols:

HTTP	**Hypertext Transfer Protocol** • Used for web pages/WWW • HTTP uses **HTML (Hypertext Markup Language)** HTML files: Normal txt files that contain special coding **HTML allows:** • GFX/special fonts/chars to display by browser/web-enabled apps **Port: 80**
HTTPS	**HTTP Secure** • Used for pages usrs should see when entering personal info **Port: 443** • Netscape originally created protocol for use w/browser • Accepted standard: RFC 2818
FTP	**File Transfer Protocol** • Allows connections to FTP servers for file ul's/dl's **Ports: 20, 21** • Transfers files bet hosts (insecure) ○ Options released to try to create more secure protocol **FTPPS: FTP over SS:** Support for SSL crypto

	SFTP: SSH FTP: AKA **Secure FTP** **SCP: Secure Copy: Port: 22:** Combines RCP [Remote Copy Program] w/SSH **TFTP: Trivial FTP:** Can config to xfer files bet hosts w/out usr interaction • Unattended mode: Should avoid/more secure alts
SMTP	**Simple Mail Transfer Protocol:** Email comm **Port: 25**
Telnet	**Interactive Terminal Emulation Protocol** • Interactive session w/Telnet server/Appear to client as local session **Port: 23**
DNS	**Domain Name System:** Allows hosts to resolve hostnames to IP's **Port: 53**
RDP	**Remote Desktop Protocol:** Win-based terminal servers: Port 3389
SNMP	**Simple Network Management Protocol** • Mgmt tool allows comm bet network devices/console • Routers/bridges/intelligent hubs **Port: 161 source, 162 receive**
POP	**Post Office Protocol:** Receiving email • Implementation of advanced features: Standard int **Port: POP3: 110** **IMAP: Internet Message Access Protocol**: **Port: 143** **Difference between POP/IMAP:** **POP:** Created to email client/not keep on server **IMAP:** Intended to store email on server/allow access from there

SSL: Secure Sockets Layer: Establish secure comm connection bet 2 TCP-machines

- Protocol uses handshake method of establishing session
- Originally dev by Netscape: Maintain session using symmetric encryption

TLS: Transport Layer Security: AKA: SSL 3.1: Expands on SSL

- Doesn't interoperate w/SSL: Standard supported by IETF (Internet Engineering Task Force)

Transport or Host-to-Host Layer: Provides app layer w/session/datagram comm services:

TCP	**Transmission Control Protocol** • Reliable 1-to-1 connection oriented session • Establishes connection: Ensures other end receives packets sent • Ensures packets are decoded/sequenced properly • Connection persistent during session: When session ends/bye connection
UDP	**User Datagram Protocol** • Unreliable connectionless comm method between hosts • Best-effort protocol/Faster than TCP • Sessions don't establish • No guarantee error-free Purpose: Send small packets of info • App responsible for ACK correct reception of data

Internet Layer: Routing/IP addressing/packaging: Standard protocols of Internet layer:

IP	**Internet Protocol**

	• Responsible for addressing • Fragments/reassembles msg packets • Only routes info: Doesn't verify for accuracy • Accuracy checking responsibility of TCP • IP determines dest known/routes info to dest ○ If unknown IP sends packet to router/which sends it on
ARP	○ **Address Resolution Protocol** ▪ Resolves IP to Network Int layer address ▪ Includes HW addresses ▪ Can resolve IP to MAC address ▪ MAC's used to ID HW net devices (NICs)
ICMP	**Internet Control Message Protocol** • Provides maintenance/reporting functions • Ping uses it/Tests connectivity • Reports if destination unreachable • Routers/other devices report path info bet hosts w/ICMP

Network Access Layer (Interface layer): Placing/removing packets on phys net through comm w/net adapters in host

- Allows TCP/IP to work w/virtually any type of topology w/little mod

IPv4	32 Bit addresses
IPv6	128 Bit addresses

Encapsulation: Allows transport protocol to be sent across net/utilized by equiv service/protocol at receiving host

Well-Known Ports Ports: ID how comm process occurs: Special addresses that allow comm bet hosts

- # added from originator, indicating which port to comm w/on server
- **IANA: Internet Assigned Numbers Authority:** Defined list of ports: *well-known ports*

Well-Known TCP ports

Service	TCP Port	Service	TCP Port
FTP (data channel)	20	FTP (control channel)	21
SSH/SCP	22	Telnet	23
SMTP	25	TACACS auth service	49
POP3	110	SFTP	115
NNTP	119	NetBIOS name service	137
NetBIOS datagram service	138	NetBIOS session service	139
IMAP	143	LDAP	389
HTTPS	443	FTPS (data channel)	989
FTPS (control channel)	990	MS WBT Server	3389

Well-Known UDP Ports

Service	UDP Port	Service	UDP Port
SSH/SCP	22	TACACS auth service	49
DNS name queries	53	TFTP	69
HTTP	80	NetBIOS name service	137

NetBIOS datagram service	138	NetBIOS session service	139
IMAP	143	SNMP	161
LDAP	389	FTPS (data channel)	989
FTPS (control channel)	990	MS WBT Server	3389

View Active TCP/UDP Ports: **netstat /-a**

3-Way Handshake: TCP establishes session

- Host called client originates connection | Client sends TCP segment/msg to server
- Client segment includes **ISN: Initial Sequence Number** for connection/window size
- Server responds w/TCP segment: Contains its ISN/value indicating buffer/window size
- Client sends back ACK of server's sequence #
- When session over: Similar process occurs

SYN

SYN/ACK

ACK

FIN

Application Programming Interface: API: Allow programmers to create ints to protocol suite

- When programmer needs to create web-enabled app: Call/use API to make connection/send/receive data/end connection
- API's prewritten: MS uses WinSock (Windows Sockets) API to int to protocol suite
- Can access TCP/UDP protocols to accomplish needed task

Other Protocols to know:

iSCSI: Internet Small Computer Systems Interface:

- **Ports: 860, 3260**
- Allows data storage/transfers across existing network
- Enables creation of SANs (Storage Area Networks)

Fibre Channel: Intended to work on fiber based networks

- Fiber fell off: Uses SCSI to create SAN across any existing network
- **FCoE: Fibre Channel over Ethernet:** Not routable at IP layer (iSCSI is) and can't work across large networks

Designing a Secure Network

DMZ's: Demilitarized Zones	**Subnetting**	**VLAN's**	**Remote Access**	**NAT**	**Telephony**	**NAC's**

DMZ: Demilitarized Zones: Area where you can place public server for access by people you might not trust

- Isolates server in DMZ: Hides/removes access to other areas of network
- Others can't access further resources | Using firewalls to isolate network
- Assume person accessing resource isn't trustful

Create DMZ: Use firewall that can transmit in 3 directions

1. Internal network
2. External world
3. Public info you're sharing (DMZ)

Bastion host: Host that exists outside DMZ: Open to public

Subnetting: Using subnet mask value to divide network into smaller parts

- Bits from node portion of host address: Creates additional networks

2 primary reasons for use:

1. IP used more effectively
2. Network more secure/manageable

Confines traffic to network it needs to be on/reducing it/more broadcast domains/Reduces range of broadcast traffic

VLAN: Virtual Local Networks: Create groups of usrs/sys/segment them on network

Let's you hide segments of network from others/control access

- Reduces scope of broadcasts: Improves performance/manageability: Decreased dependence on phys topology

Tunneling protocols: Ability to create tunnels bet networks: More secure/support add protocols/provide virtual paths be sys

Most common protocols for tunneling

Point-to-Point Tunneling Protocol	**PPTP: Supports encapsulation in single p-t-p env** • Encapsulates/encrypts PPP packets • Negotiation bet 2 ends of PPTP connection done in clear • After negotiation: Channel encrypted **Weakness:** Packet capture device/sniffer • Can capture negotiation process/use info **Port: 1723 TCP**
Layer 2 Forwarding L2F	**Cisco: Method of creating tunnels for dial-up** • Similar to PPP: Shouldn't be over WANs • Provides auth: Doesn't encrypt **Port: 1701 TCP**
L2 Tunneling Protocol L2TP	**MS/Cisco combined their tunneling protocol into 1** • Hybrid: PPTP/L2F: Point-to-point • Supports multiple protocols • Works over IPX/SNA/IP: Can be used as bridge across many sys types **Problems:** Doesn't provide data sec/not encrypted • Sec can be provided: Protocols like IPSec **Port: 1701 UDP**
Secure Shell/SSH	**Tunneling protocol originally for Unix** • Encryption establishes secure connection bet 2 sys • Preferred method of sec for telnet/clear text-oriented programs in Unix **Port: 22 TCP**
Internet Protocol Sec/IPSec	**NOT tunneling protocol: Used in conjunction w/them** • IPSec: Primarily LAN-to-LAN connections: Can be used w/dial-up • Provides secure auth/encryption of data/headers **Can work in tunneling/transport mode** **Tunneling:** Data/payload/msg headers encrypted **Transport:** Encrypts only payload

Tunneling: Creating virtual dedicated connection bet 2 sys/networks

- Can create the by encapsulating data in mutually agreed protocol for transmission
- Data passed through tunnel appears at other side as part of network

- Tunneling protocols usually include data sec/encryption: Most popular L2TP

RAS: Remote Access Services: Any service that offers ability to connect remote systems

Windows-based clients: RRAS: Routing and Remote Access Services: Previously known as RAS

- RAS connection accomplished via dial-up (*POTS: Plain-Old Telephone Service* and modem)
- VPN's/ISDN/DSL/Cable modems | May be secure/or in clear
- Depends on protocols used

NAT: Network Address Translation: Originally extended # of usable IP's

- Now allows org to present single address to all Internet for computer connections
- NAT service provides IP addresses to hosts/systems in network/track inbound/outbound traffic
- Only info an intruder will be able to get is connection has single address
- Hides network
- Most NAT implementations assign internal hosts private IP/use public addresses

Private address ranges/all non-routable
10.0.0.0-10.255.255.255
172.16.0.0-172.31.255.255
192.168.0.0-192.168.255.255

NAC: Network Access Control: Op sec issues include NAC auth/sec topologies after installation complete

Issues:

- Daily operations of network
- Connections to other networks
- Backup/Recovery plans

NAC: A set of standards defined by the network for clients attempting to access it

Understanding Various Network Infrastructure Devices:

Firewalls: One of the 1st lines of defense in a network

Function as 1/more of the following:

- Packet filter
- Proxy firewall
- Stateful packet inspection firewall

Packet Filter Firewalls: Passes/blocks traffic to specific addresses based on type of app

- Doesn't analyze data of packet
- Decides whether to pass it based on packet's addressing info

Proxy Firewall: Intermediary bet your/any other network

- Used to process requests from outside network
- Examines data/makes rule-based decisions: Should request be fwded/refused
- Intercepts all packets/reprocesses for use internally: Includes hiding IP addresses

- Requests from internal network usrs/routed through proxy
- Proxy repackages request/sends it along; isolating user from ext network

Typically uses 2 NICs: Dual-homed firewall

- One card connected to outside network: Other connected internally
- Proxy SW manages connection bet 2 NICs/Segregates from each other/improved sec
- Proxy function can occur at either application/circuit lvl

Application-level proxy	Functions read individual cmds of protocols being served
Circuit-level proxy	Creates circuit bet client/server: Doesn't deal w/contents of packets being processed • Unique app level proxy server must exist for each protocol supported • Many proxy servers also provide full auditing/accounting/other usage info/normally not kept

Stateless firewalls: Make decision based on data that comes in: Packet not based on complex decisions

Stateful inspection: AKA: SPI: Stateful Packet Inspection filtering: After a packet is passed, the packet/path are forgotten

- Stateful: Records kept using table that tracks every comm channel
- Remembers where packet came from/where next one should come from

Router: Primary instrument for connectivity bet 2/more networks: Provide path between networks

- Routers, in conjunction w/a **CSU/DSU: Channel Service Unit/Data Service Unit**
- Also used to translate from LAN framing to WAN framing
- Needed: Network protocols diff in LANs/WANs: Border routers

Usually comm routing info by 1 of 3 standard protocols

- **RIP: Routing Info Protocol**
- BGP: Border Gateway Protocol
- OSPF: Open Shortest Path First

Switches: Multiport devices that improve network efficiency

- A table of MAC addresses instead of IP addresses

Load Balancing: Shifting a load from 1 device to another

- Most often device is server/term could be used for any device

Load balancer: SW/HW: Split traffic for website into individual requests: Rotated to redundant servers as they become available

Proxy: A device that acts on behalf of others

Web Security Gateway: Proxy server (proxy/caching functions) w/web protection SW built-in

- Web protection dependent on vendor: Can be virus scanner on incoming packets/etc…
- Red flags can detect/prohibit inappropriate content
- Can config most to block known HTTP/HTML exploits/strip ActiveX tags/strip Java applets/block/strip cookies

VPN: Virtual Private Network: Private connection that occurs through public

network

- Can be used to connect LANs together across internet/public networks
- W/a VPN: Remote end appears connected to network as if local
- Typically use tunneling

VPN concentrator: HW device used to create remote access VPN's: Creates encrypted tunnel sessions bet hosts

- Cisco models often incorporate **SEP: Scalable Encryption Processing** modules to allow for GW based encryption/redundancy

IDS	**Intrusion Detection System:** SW runs on individual workstations/devices to track activity • Reports/monitors intrusion attempts Activity: Element of data source that is of interest to operator • Anything that could be suspicious
Alert	Msg from analyzer indicating event of interest occurred • Contains info about activity/specifies
Analyzer	**Process that analyzes data collected by sensor** • Looks for suspicious activity among all data collected • Monitor events/determine whether unusual things happen • Can use rule-based processed
Data Source	Raw info IDS uses to detect suspicious activity (audit files/sys logs/traffic)
Event	Occurrence in data source that indicates suspicious activity happened
Manager	Process operator uses to manage IDS
Notification	How IDS alerts operator
Operator	User responsible for IDS
Sensor	Collects data from source/passes to analyzer for analysis

4 Primary Approaches to IDS:

Behavior-Based-Detection IDS	Looks for variations in behavior/Unusually high traffic/Policy violations
Signature-Based-Detection IDS	**AKA: Misuse-detection IDS (MD-IDS)** • Evaluates attacks on signatures/audit trails • Generally established methods of attack
Anomaly-Detection IDS	**AD-IDS:** Looks for anomalies: Things out of ordinary
Heuristic IDS	Uses algs to analyze traffic passing through network • Requires tweaking/fine-tuning: To prevent false positives

IPS: Intrusion Prevention System: Reacts to intrusion that's detected: Mostly by blocking comm from offending IP: False positives

Working w/a Network-Based IDS:

NIDS: Attaches the sys to point in network where it can monitor/report all network traffic

Port spanning/mirroring: Copies traffic from all ports to single port/disallows bidirectional traffic on that port

Passive response: Most common type of response:

Logging	Recording event that occurred under what circumstances
Notification	Comm event-related info to appropriate personnel
Shunning	Ignoring attack

Active response: Taking action based on attack/threat/quickest action possible to reduce impact

Terminating Processes/Sessions	Force resets/etc..
Network config changes	
Deception	Fool attacker into thinking attack succeeding while sys monitors

Host-Based IDS: HIDS: Designed to run on host system: Typically as service/BG process

- Examines machine logs/sys events/app interactions
- Normally doesn't monitor incoming traffic to host
- Popular on servers that use encrypted channels/channels to other servers

Problems: If sys compromised: Logs may be inaccurate
Benefit: Potential to keep checksums on file

NIPS: Network Intrusion Prevention System: Focuses on prevention

- Signature matches/then take course of action

Log Files in Linux

/var/log/faillog	/var/log/lastlog	/var/log/messages	/var/log/wtmp

Protocol Analyzers: Process of monitoring data transmitted across network
Spam Filters: Can be added to catch unwanted mail/filter before delivered internally
UTM Appliances: UTM: Unified Threat Management: Standalone appliance
NGFW: Next Generation Firewall: URL filtering/content inspection/malware inspection
Web application firewall: WAF: Real-time appliance of rules to block traffic to/from web servers/to prevent attacks
Application-Aware Devices: Device w/ability to respond to traffic based on what's there

From <https://www.piratemoo.net/moosings/information-security/mastering-tcpipfirewallsetc/>

Post 4

Friday, January 25, 2019 12:19 AM

CH. 4 UNDERSTANDING ACCESS CONTROL BASICS

September 26, 2016 Moo Comments 0 Comment

Access control: Allowing correct usrs into a sys/keeping others out
Identification: Finding out who someone is
Authentication: Mechanism of verifying that identification
Identification = Claiming an identity
Authorization = Proving that identity
Example: People can claim to be anyone (identification) | Proving it (auth) requires evidence (DL/Passport)
Auth systems are based on 1/more of 5 factors:

- **Something you know: Password/PIN**
- Something you have: Smart card/Token/ID device
- Something you are: Fingerprints/retinal pattern/biometrics
- Something you do: Action you must take to complete auth
- Somewhere you are: Geolocation

Sys: Pass priv info bet each other to establish ID: Once auth occurs: 2 sys can comm in manner specified in design

- 7 common methods used for auth

Fall w/in categories of

Single factor	Multifactor

Out-of-band auth: Process where sys you are auth'ing gets info from public records | Asks you questions to help auth
Authentication & Authorization: Single Factor
SFA: Single-factor auth: Most basic form of auth: Only 1 type of auth checked
Traditional username/passwd combo implemented: Unique identifiers for logon process

- Sec sys requires someone to establish who they are through a confirmed logon process
- Most OS's use usernames/passwds to accomplish
- Values can be plain text/encrypted
- Logon process: ID's you to OS/network | OS compares info from sec processor
- Either accepts/denies logon attempt
- Might establish privs/perms based on stored data about user ID

Mutual authentication: 2/more parties auth each other

- Client may auth to server: Server may auth to client when establishing

secure session

May employ encryption:

- Ensures client isn't unwittingly connecting/providing creds to rogue server: Can steal data from real one
- Commonly implemented when data to be sent is critical (financial/medical records)

Multifactor Authentication: 2/more access methods included as part of auth process

- Can consist of 2F/3F (factor) sys and more: As long as more than 1 factor involved in auth process
- Shouldn't be from same category

Example: Using passwd/smart card = 2F auth

Using passwd/PIN 1F auth: Both involve ***"something you know"***

2-factor auth: A sys that uses smart cards/passwds

Example: Requires both smart card/login/password process

Layered Sec/Defense in Depth

Shouldn't rely on single entity for protection: Implement multiple layers of sec

Example: Guard at building door & lock on server door

Firewall: Great to restrict traffic: You also want AV/IDS/etc.. To truly help protect sys

Network Access Control:

Operational Security: *HOW* an org achieves goals

- Part of sec triad that includes phys/mgmt sec

NAC: Network Access Control: AKA: Network Admission Control: Auth/sec topologies after network install complete

Issues include:

Daily operations of network	Connections to other networks
Backup plans	Recovery plans

Op sec: Encompasses everything not related to design/phys sec in network

- Vulns in sys where weak/inadequate policies:
- Example: Implementing passwd expiration policy: Requires change every 30-60 days
- No password rotation is vuln: Can be eliminated
- Most companies use SW developed by 3rd parties: Can have sec/vuln issues

Tokens: Security tokens: Similar to certificates: Used to ID/Auth usr

- Contains rights/access privs of token bearer as part of token
- Small piece of data that holds info about usr
- Many OS's generate token: Applied to every action on sys
- If token doesn't grant access: It won't be displayed/access denied
- Auth sys creates token every time usr connects/session begins
- Token destroyed at end of session

Federations: Collection of networks that agree on standards of operation (sec standards)

- Normally, networks related in some way
- Could be industry association that establishes standards

Example: **IM federation (Instant Messaging):** Multiple IM providers form

comm standards

- Allows usrs on diff platforms w/diff clients to comm freely
- Other situations: Group of partners elects to establish common sec/comm standards forming federation
    - Would facilitate comm bet employees in each of various partners

Federated identity: Means of linking usr's ID w/privs in manner that can be used across business boundaries

Example: MS Passport/Google checkout

- Allows usr to have single ID: Can be used across diff business units/entirely diff businesses

Potential Authentication/Access Problems: 2 problem areas that apply to auth/access issues:

- **Transitive Access**
- Client-Side Attacks

Transitive Access: Transitive: Transition

With transitive access:

Party **A:** Trusts another party: **B**: If party B, trusts another party; C, a relationship can exist where A may trust C

Party A → Trusts Party B

Party B → Trusts Party C

Party A → May form a relationship to trust **party C** because of **B**

Early OS's: Process often exploited

Current OS's: WinServer2012:

- Problems solved creating transitive trust trusts

Transitive trusts: Type of relationship that can exist bet domains (opposite = **nontransitive trusts**)

- When the relationship is transitive: A relationship flows like in the chart above (A trusts C)
- **AD: ALL versions:** Default: Domains in a forest trust each other with 2-way transitive trust relationships
- Makes administration easier when adding new child domain (no admin intervention to establish trusts)
- Possible for a hacker to acquire more trust by virtue of joining the domain

Validating Trust Relationship

1. AD Domains/Trusts > *Right-click domain name* > Properties
2. *Trusts* tab > Select domain/forest to validate > Properties
3. Transitivity of Trust item > Validate > Confirm > OK > Exit

Authentication Issues to Consider: Always better to educate usrs/raise awareness than lower sec

Tips for authentication	**Beware of trends that make passwds predictable** Example: Super Bowl time: Teams/players • ID proof when issue arises **Process starts when ID typed in sign-on screen** • Auth challenges claim of who accesses resource • Claiming specific usr (admin): Auth requires proof

	Incorporate 2nd value to prove usr's ID (mother's maiden name) • Helpful when ID proofing invoked Example: Lost password
Problems w/ID-proofing	Question answers may be something others can guess/learn **Increase difficulty of fraudulent ID proofing:** • Questions: Difficult to guess • Biometrics: Voice ID **Under NO chances:** Person proofing be allowed access immediately: • Info should be sent to email acct of record

Authentication Protocols: Used to aid authenticating usr to system:

PAP	**Password Authentication Protocol** • Older sys: No longer used • Sends usrname/passwd to auth server in plain text
SPAP	**Shiva Password Authentication Protocol** • Replaced PAP • Encrypts usrname/passwd
CHAP	**Challenge Handshake Authentication Protocol** • Designed to stop MITM attacks • During initial auth: Connecting machine asked to generate random number (hash)/send it to server • Periodically: Server will challenge client machine: Demands to see # again **If attacker has taken over session:** • Won't know #/won't be able to auth
TOTP	**Time-Based One-Time Password** • Algorithm that uses time-based factor to create unique passwds
HOTP	**HMAC-Based One-Time Password** • Algorithm based on using: **Hash Message Authentication Code (HMAC)** alg

Account Policy Enforcement:

Account policy: Determines sec params regarding who can/can't access sys

Components of account policy enforcement:

- Password length/complexity
- Password expiration
- Password recovery
- Password disablement/lockout

Password Length/Complexity: Obtain fine balance bet 2 extremes

- 8 chars (upper/lowercase) general min for passwd length
- Most sys today encourage use of at least 1 non-alpha char-punctuation/special chars/numbers etc…

Win: Passwd value set to 0 = No required passwds

Win8/older Win vers: Local Security Policy (overridden by Group Policy values on domain controller):

- Allows choice to enable passwd complexity

Passwd complexity: Usrs can create passwds that meet the following:

- Can't contain usr acct name/part of usr's full name: That exceeds 2 consecutive chars
- Must be at least 8 chars long
- Have to contain chars from at least 3 of the following 4 sets:
  - A-Z

- a-z
- 0-9
- Non-alpha chars (!, $, #, % etc…)

Methods that allow person w/phys access to Win machine to retrieve passwd regardless of complexity:

- Tools like Ophcrack freely avail
- Longer phrases w/complexity

Example: **!L!k3Ch33s3Burg3rsFromBug3rK!ng** (easy to remember/harder to guess)

Password Expiration: Passwds must expire: The longer same value used: More likely to be broken

- **90 days:** Acceptable for many orgs
- MS says: **42 days:** To enforce strong passwd usage through org
- Keep usrs from changing passwd to same value as old one/last time: Enable passwd history
- MS OS: Set this between **0 – 24 (0 = disabled)**
- **Set to 24** = 24 unique passwds must be used by usr before they can reuse

Along w/expiration date: Config min # of days that can exist bet passwd changes

If setting disabled/set too low:

- Usrs can immediately change new passwds to other values
- Recommended: Don't set it lower than 2 days

Password Recovery

Password Disablement/Lockout

Usr gone for while? Acct should be disabled till return: Example: Maternity leave

Usr gone for good? Acct should be removed immediately: Example: Fired

Acct Lockout Duration	**Locked acct's duration before it's unlocked:** Value range: **0 – 99,999 min (Win)** **Setting to 0:** • Doesn't disable feature • Does require admin to unlock acct before it can be used again
Acct Lockout Threshold	**How many incorrect attempts usr can give before acct locks:** Value range: **0 – 999 failed attempts (Win)** **Setting at 0:** • Acct will never be locked out • Attempts to enter values at passwd-protected screensaver: **Count**
Reset Acct Lockout Counter After	**# of min to wait bet counting failed login attempts: Same batch attempts** Example: Threshold set to 3 to lock acct after 3 bad tries: Value set to 5 • Usr tries/fails: Waits 5+ min: Tries again: 3 attempts again before lock Value range: **0 – 99,999 min** • Can set only if threshold set

	• Reset time: Must be less than/equal to acct lockout duration

Usrs w/Multiple Accts/Roles:
Educate: Show why employees should only use elevated accts when necessary
Policies: Make sure they're understood/signed-off by those operating in group
Generic Account Prohibition
Generic account: Any acct that is shared: Allows multiple usrs to log in/use

- Guest/Anon/Temp accts
- Can make it easy to grant access to sys
- Should be avoided/not used when possible

Suffer from 2 significant problems:

1. Passwd shared: Against normal sec procedures
2. Auditing can be a hassle

Group-based/User-assigned Privs: 2 methods of priv assignment prevalent today

1. **Group-based priv**: Acquired as result of belonging to group: Editor's group/Admin group/etc…
2. **User-assigned privs**: Privs that can't be assigned by usr

Example: You create a file in most OS: Can change perms associated w/file: Possible to give others privs
Understanding Remote Access Connectivity

SLIP	**Serial Line Internet Protocol (ancient history)** • Older protocol used in early remote access envs: Serves as starting point for most remote discussions ○ Originally designed to connect Unix sys: In a dial-up env/It's only supported serial comms ○ Was only used to pass TCP/IP traffic: Wasn't secure/efficient ○ Legacy system thing

Point-to-Point Protocol: 1994: Support for multiple protocols: AppleTalk/IPX/DECnet
Works w/POTS, ISDN (Integrated Services Digital Network): Also faster connections like T1

- In case of ISDN: PPP normally used 64Kbps B channel for transmission :
  - Allows many channels in connection to be bonded together to form single virtual connection (like ISDN)
- Doesn't provide data sec: Does provide auth using the **CHAP: Challenge Handshake Authentication Protoco**l
- Encapsulates network traffic in: **NCP: Network Control Protocol**
- Authentication handled by **LCP: Link Control Protocol**

PPP connection: Allows remote usrs to log onto network/have access as though they were local usrs

- Doesn't provide any encryption services for the channel
- Largely insecure: Unsuitable for WAN connections

To counter this: Other protocols created to take advantage of PPP's flexibility/build on it

- Make sure all PPP connections use secure channels/dedicated-high-

speed connections

Remote usrs who connect directly to sys: Don't necessarily need encryption: Likelihood of phone line tap is relatively low

- Make sure connections go through a network that uses encryption-oriented tunneling system

Working w/Tunneling Protocols:

Tunneling protocols: Add ability to create tunnels bet networks

Can be:

- More secure
- Support additional protocols
- Provide virtual paths bet systems

Simply put: Data encapsulated in other packets sent across public network

- Once received at other end: Sensitive data stripped from other packets: Recompiled into original form

Most common tunneling Protocols:

PPTP	**Point-to-Point Tunneling Protocol:** • Supports encapsulation in single point-to-point env • Encrypts/encapsulates PPP packets • Favorite low-end protocol **Negotiation bet 2 ends of PPTP connections done in clear** • **AFTER** negotiation: Channel encrypted **Weakness:** • Packet-capture (sniffer) can see negotiation process • Use info to determine: Connection type | How tunnel works • MS developed PPTP: Supports it **Port 1723 TCP**
L2F	L**ayer 2 Forwarding:** • Created: Cisco: Method of creating tunnels for dial-up connections • Similar to PPP: **DON'T** use over WAN's **Provides auth: Doesn't provide encryption** **Port 1701 TCP**
L2TP	**Layer 2 Tunneling Protocol:** • MS/Cisco combined tunneling protocols into 1 • Hybrid of PPTP/L2F • Primarily point-to-point • Supports multiple network protocols: Can be used beside TCP/IP **Works over:** • IPX • SNA • IP • Can be used as bridge across many types of systems **Weakness:** • Doesn't provide data security: Info NOT encrypted • Can be provided by protocols like IPSec **Port 1701 UDP**
SSH	**Secure Shell:** • Originally designed for Unix systems • Uses encryption to establish secure connection • Provides alternative sec-equivalent programs for such Unix standards as: ○ Telnet

	○ FTP ○ Other • Avail for Win • Preferred method of sec for Telnet/other clear-text programs in Unix env **Port 22 TCP**
IPSec	**Internet Protocol Security:** **NOT tunneling protocol** • Used in conjunction w/tunneling protocols • Oriented towards LAN-to-LAN connections • Can be used w/remote connections • Provides secure auth/encryption of data & headers • Good choice **Can work in Tunneling/Transport modes** **Tunneling:** Data/payload/msg headers encrypted **Transport:** Encrypts only payload Add-on to IPv4 | Built in IPv6

Working w/RADIUS

RADIUS: Remote Authentication Dial-In Service: Mechanism that allows auth of remote/other connections

- Originally for dial-up: Moved beyond that

IETF standard: Implemented by most major OS's

- Can be managed centrally: Servers allow access to network can verify w/RADIUS server whether incoming caller authorized
- Large network? Allows single server to perform all authentications

Caller: WinServer03/08/12: Refer to ability to remotely access sys as dial-in privs. Few people actually "dial" | Term just stuck

Use RADIUS when:

- You want to improve sec w/single service to auth usrs who connect remotely
- Gives single source for authentication to take place
- Can implement auditing/accounting

Difficulties:

- Server malfunctions? Entire network may refuse connections
- Fix? Many RADIUS sys allow multiple servers to be used to increase reliability
- Critical components of infrastructure

TACACS/TACACS+/XTACACS

TACACS	**Terminal Access Controller Access-Control System** • Client/server-oriented env • Operates similarly RADIUS
XTACACS	**Extended Terminal Access Controller Access-Control System** • Replaced original ver • Combined auth/authorization w/logging to enable auditing
TACACS+	**Most current method** • Replaces previous 2 • Allows credentials to be accepted from multiple methods: Like Kerboros • Process occurs in same manner as RADIUS • Cisco: Widely implemented TACACS+ for connections • Widely accepted alt to RADIUS

VLAN Management: VLAN: Virtual Local Area Network

Allows creation of groups of usrs/sys | Segmenting them on network

- Segmentation lets you hide areas of network from others
- Controls paths that data takes to get from 1 point to another
- Contains traffic to certain areas
- Increases security: Allows usrs w/similar data sensitivity lvls to be grouped

SAML: Security Assertion Markup Language

- Open standard based on XML: Used for auth/authorizing data
- Service providers use SAML: Proves ID of someone connecting to service provider

SAML v2.0

Understanding Auth Services:

LDAP	**Lightweight Directory Access Protocol:** Standardized dir access protocol • Allows queries to be made of dirs (pared-down x.500-based dirs) **If dir service supports LDAP: Can query dir w/LDAP client** • Main access protocol used by AD **Port 389** • *Syntax:* Uses commas bet names **LDAPS: Secure LDAP** • All LDAP comms encrypted w/SSL/TLS **Port 636**
Kerboros	**Auth protocol: Named after 3-headed dog: Stood at gates of Hades** • Original design by MIT: Very popular • Allows single sign-on to distributed network **Uses KDC: Key Distribution Center:** To orchestrate process • KDC auth's principal (usr/program/sys) • Provides it w/ticket • After ticket issues: Can be used to auth against other principals Process occurs auto when another principal performs request/service **Weakness:** Single-point-of-failure: • KDC goes down? Auth process stops **When using Kerboros: Usr auth's to the KDC** • **Given TGT: Ticket Granting Ticket** • Ticket encrypted: Has a time limit of up to 10hrs • Ticket lists privs of usr (like token) **Each time usr wishes to access a resource: Usr's comp presets KDC w/TGT** • TGT sends usr's comp service ticket: Grants usr access to that service • **Service tickets:** Usually good for up to 5 min • Usr's comp then sends service ticket to server usr is trying to access **Final auth check:** Server comms w/TGT to confirm/validate service ticket

Single Sign-On Initiatives: Problems larger sys must deal w/: Need for usrs to access multiple systems/apps

- May require usr to remember multiple accts/passwds

SSO: Single Sign-On: Purpose: To give usrs access to all apps/sys needed when log in

Good: Once usr auth: Can access all resources on network/browse multiple dirs

Bad: Removes doors that exist bet usr/resources

In case of Kerboros:

- Single token allows any “Kerberized” apps to accept usr as valid
- Each app that wants to use SSO must be able to accept/process token by Kerboros

AD uses slightly diff method:

- Server that runs AD retains info about all access rights for all usrs/groups in network
- When usr logs onto sys: AD issues usr GUID: Globally Unique Identifier
- Apps that support AD: Use this GUID to provide access control

On decentralized network: SSO passwds stored on each server/can represent risk

Understanding Access Control:

4 Primary Methods: Each methods has advantage/disadvantage

1. **MAC: Mandatory:** All access predefined
2. **DAC: Discretionary:** Some flexibility
3. **RBAC: Role-Based:** Allows usr’s role to dictate access capabilities
4. **RBAC: Rule-Based:** Limits usr to settings in preconfig policies

LBAC: Lattice-Based: Variation of MAC

- Lattice composed of subjects (usrs/sys) | Resources labeled to provide access control

Method chosen will greatly be affected by org philosophy on sharing info

Example: High-sec Env: MAC/RBAC (Role) || Traditional bus/edu: DAC

MAC	**Mandatory Access Control:** Inflexible: How info access permitted • All access capabilities predefined • Usrs can’t share info unless rights to share are established by admins • Enforces rigid model: Also considered most secure **For MAC model to work effectively:** • Admins must think relationships through carefully in advance of implementation **Advantages:** • Security access well established/defined • Security breaches easier to investigate/correct **Disadvantages:** • Lack of flexibility • Requires change over time • Inability to address changes can make it a pain **Environment Type:** • Confidentiality: Driving force • Often gov/mil classifications
DAC	**Discretionary Access Control:** Some flexibility: How info is accessed • Allows usrs to share info dynamically w/other usrs • Increases risk of unauthorized disclosure of info **Disadvantage:** • More difficult time ensuring info access is controlled: Only appropriate access issued Example: Perm structures that exists for “other” w/files in Unix/Linux **All perms in OS fall w/in 3 groups of usrs:** 1. Owner 2. Group

	3. Other **Perms associated w/owner/group belonged to: Based on roles** • All those not owner/member of owner group fall w/in other ○ Perms for this group: Set separately from other 2: Very few exceptions
RBAC: Role	**Role-Based Access Control:** More flexibility: Approaches access control on established roles in org • Implements access by job function/responsibility • Each employee has 1/more roles that allow access to specific info **If person moves from 1 role to another: Access for previous role no longer avail** • More flexibility than MAC || Less flexibility than DAC **Advantage:** • Strictly based on job function not individual needs **Environment Type:** • Ones with high turnover rates **Win OS: Works similarly:** Perms on domain determined by groups usrs placed in
RBAC: Rule	**Rule-Based Access Control**: Settings in preconfig sec policies to make all decisions **These rules can be to:** • Deny all except those who specifically appear in list (allow) • Deny those who specifically appear in list (true deny) **Entries in list may be usrnames/IP's/hostnames/domains** • Often in conjunction w/Role-Based: Adds greater flexibility • Easiest way to implement: ACL's

Implementing Access Controlling Best Practices

Least Privileges: Any given usr/sys given min privs necessary to accomplish their tasks

Example: Sales managers need to run reports from DB: Given privs only allowing running of reports

Any priv could be used to cause harm to sys: More true when usrs have more privs then needed

Privilege Escalation: Common attack: Giving each usr min privs reduces risk

- If no access to given resource ***AND*** something goes bad: Can't be held responsible

Separation of Duties: Make certain to have as many diff lvls of perms/privs as possible

At min:

1. Separate SA (Sys Admin) acct from reg accts: Never log in SA for routine functions
2. SA acct: ONLY for operations that require those privs
3. Limit SA acct to as small a group as possible
4. Separate audit/logging responsibilities from SA

ISO standard: Recommends segregation of duties/separation of envs: Way to reduce misuse of sys/info

Time of Day Restrictions: One of easiest policies to enforce

- Config when an acct can have access to sys

Example: Config access between 7AM-6PM

- Prevents accts from being used by attackers other 113 hours
- Restrictions can be applied: Policies for groups/usrs in AD: Set locally/through add-ons

User Access Review

Access review: Process to determine whether usr's access lvl still appropriate

- People's roles w/in org can change over time
- Usr's shouldn't have "leftover" privs from previous job roles

Continuous monitoring: Ongoing audit of what resources usr accesses

Smart Cards: Generally used for access control/sec

- Card usually contains small amt of mem: Can be used to store perms/access info
- Difficult to counterfeit || Easy to steal
- Many orgs don't put identifying marks on smart cards
- Passwd/PIN required to activate most smart cards: Encryption employed to protect contents
- W/many: If you enter wrong PIN multiple times (like 3): Card will shut down

2 main types of smart cards:

- **CAC: Common Access Cards**
- PIV: Personal Identification Verification Cards

CAC	**Common Access Card** • Issued by DoD: General ID/auth card for military personnel/contractors/non-DoD employees **Front of card:** • Picture • Integrated chip beneath • Barcode **Back of card:** • Magnetic strip • Another barcode **Used for accessing DoD comps/signing email/implementing PKI**
PIV	**Personal Identification Verification Card** • What CAC is for military employees • PIV is to federal employees/contractors **Per HSPD-12: Homeland Security Presidential Directive:** • PIV will eventually be required by all US gov employees/contractors

ACL: Access Control Lists: Enables devices in network to grant/ignore requests from specified usrs/sys

- May find certain IP's constantly scanning network: Can block at router
- Allow stronger set of access controls to be established in network
- Allows admin to design/adapt network to deal w/specific threats

Implicit Deny: Implied at end of each ACL

- If not explicitly granted: Access denied
- Denied B/C it doesn't appear on list (source/dest address/packet type/etc)

Firewall Rules: Used to dictate what traffic can pass bet firewall/internal network

3 possible actions:

1. Block

2. Allow
3. Allow only if secure

Rules can be applied to inbound/outbound traffic: Any type of network (LAN/Wireless/VPN/Remote access)

- Audit rules on reg basis: Verify results: Make mods as necessary

Port Security: L2: Allows admin to config switch ports so only certain MAC addresses can use them

- Cisco's Catalyst | Juniper EX Series switches
- Differentiates dumb switches from managed/intelligent ones

DAI: Dynamic ARP Inspection: Works w/these/other smart switches to protect ports from ARP spoofing

Areas of port sec:

MAC Limiting/Filtering	**Limit access to network:** • MAC addresses known: Filter those that aren't • NOTE: Tools can be used to change MAC/circumvent this control
802.1X	**Add port auth to MAC filtering** • Takes security down to switch port lvl • Increases security
Unused Ports	**All ports not in use should be disabled** • Otherwise: Open door for attacker

Working w/802.1X: Defines port-based sec for wireless network access control

- Offers means of auth: Defines EAP (Extensible Authentication Protocol) over IEEE 802
- **Biggest benefit:** AP's/switches don't need to do the auth: Relies on auth server to do the work

Flood Guards/Loop Protection:

Flood guard: Protection feature built into many firewalls: Allows admin to tweak tolerance for unanswered login attacks

- Reducing tolerance makes it possible to lessen likelihood of successful DoS
- Resource: Inbound/Outbound appears overused: Flood guard kicks in

Loop Protection: L2 switch configs: Intended to prevent broadcast loops

- When config: Can choose to disable broadcast fwding/protect against duplicate ARP requests

STP: Spanning -Tree Protocol: Intended to ensure loop-free bridged eth0 LANs

- Data-Link Layer: Ensures only 1 active path exists bet 2 stations

Preventing Network Bridging:

Network bridging: Occurs when device has more than 1 NIC installed

- Opportunity presents itself for usr on 1 of NICs to which device is attached to jump to other
- Although multiple cards have been used in servers for years (multi-homed hosts):
  - Not uncommon to find multiple cards in laptops
  - Or for bridging to occur w/out usr understanding

Prevention:

- Can config network: When bridging detected: Shut off/disable jack
- Can also create profiles that allow for only 1 int
- Can config workstations to disable unused connections
- Win 8: Network Sharing Center > Change Adapter Settings

Log Analysis: Crucial to ID problems: Have the ability to turn on at many diff locations/lvls

- Programs exist that can automate logging process: ManageEngine

TOS: Trusted OS: Any OS that meets gov requirements for security

CC: Common Criteria: Most common sets of standards for security

- Joint effort among Canada/France/Germany/Netherlands/UK/US
- Standard outlines comprehensive set of evaluation criteria broken down into 7 EALs

EAL: Evaluation Assurance Levels 1-7

EAL1	**Primarily used:** • When usr wants assurance that sys will operate correctly • Threats to security aren't viewed as serious
EAL2	**Requires product devs to use good design practices:** • Sec isn't high priority in EAL 2 certification
EAL 3	**Requires conscientious dev efforts to provide moderate lvls of sec**
EAL 4	**Requires positive security engineering:** • Based on good commercial dev practices • Benchmark for commercial systems
EAL 5	**Intended to ensure security engineering implemented in a product from early design phases:** • Intended for high lvls of security assurance • EAL doc indicates special design considerations likely required to achieve this lvl of certification
EAL 6	**High levels of assurance of specialized security engineering**: • Certification indicates high lvls of protection against significant risks • Systems w/this cert will be highly secure from penetration attackers
EAL 7	**Intended for extremely high lvls of security:** • Certification requires extensive testing/measurement/complete independent testing of every component

EAL:

Replaced **TCSEC: Trusted Computer Systems Evaluation Criteria** sys for certification: Popular in US

Replaced **ITSEC: Information Technology Security Evaluation Criteria:** Popular in Europe

- Recommended lvl: Commercial systems: EAL 4: Few OS's approved
- Just BC OS is capable of being certified at high lvl of security: Doesn't mean implementation is at that level

Secure Router Config: Suggestions:

Change default passwd	**Passwd for admin set before router leaves factory:** • Assume every intruder wants unauth access | Knows default passwds | Can Google them • Good passwd principle: Change it to value known by those needed
Advanced settings	**Differs based on manufacturer/type:**

	• Often settings to block ping requests/perform MAC filtering/etc..
Upgrade firmware	Router manufacturers often issue patches when problems discovered • Apply when needed

From <https://www.piratemoo.net/moosings/information-security/ch-4-understanding-access-control-basics/>

Post 5

Friday, January 25, 2019 12:19 AM

WORKING W/WIRELESS

September 30, 2016 Moo Comments 0 Comment

IEEE 802.11x family of protocols provides for wireless communications using RF transmissions.

Frequencies in use for 802.11:

- 2.4 Ghz
- 5 Ghz

802.11	Defines WLANs transmitting • 1 – 2 Mbps • 2.4 Ghz
802.11a	○ BW up to 54 Mbps ○ 5 Ghz
802.11b	BW up to 11 Mbps (fallback rates: 5.5/2/1 Mbps) • 2.4 Ghz • AKA **Wi-Fi** or **802.11 high rate**
802.11g	Faster speeds: Still interference problem: Has to share spectrum w/others using that frequency (w/802.11b) • BW up to 54 Mbps • 2.4 Ghz
802.11i	Security enhancements to standard • Focus on auth • WPA2: Name given to it by Wi-Fi alliance
802.11n	Most popular: • Can operate at 2.4/5 Ghz ranges • Can reach speeds up to 600 Mbps (actual speed much slower) Advantages: Offers higher speed: A frequency that doesn't have as much interference

Most of time: Wireless AP will work w/more than one 802.11 standard

WEP/WAP/WPA/WPA2

WEP: Wired Equivalent Privacy	Intended: Basic sec for wireless networks Wireless sys frequently use WAP: Wireless Application Protocol for network comms • WPA/WPA2 replaced WEP • Protocol designed to provide priv equiv to wired network • Implemented in # of devices Vulnerable: • Weaknesses in RC4 encryption alg • Allowed alg to be cracked in as little as 5 min

IV: Initialization Vector: IV that WEP uses for encryption: 24-bit: Weak: IV's reused w/same key

IV attack: Attackers can examine repeated results: Easy to crack secret key

- RC4: As alg: Was too small

- IV static/IV part of RC4 encryption key

TKIP: Temporal Key Integrity Protocol: To strengthen WEP: Employed

- Placed 128-bit wrapper around WEP encryption w/key based on things like:
  - MAC address
  - Destination device
  - Serial # of packet
- TKIP: Designed as backward-compatible replacement to WEP: Could work w/existing HW
- W/Out TKIP WEP much weaker: TKIP still crack able

WAP: Wireless Application Protocol: Designed for use w/wireless devices

- Became data transmission standard adopted by many manufacturers (Motorola/Nokia)
- Functions equiv = to TCP/IP functions: Attempts to serve same purpose for wireless devices

WML: Wireless Markup Language: Smaller version of HTML: WAP uses: Used for Internet displays

- WAP-enabled devices can respond to scripts using env called **WMLScript**
- This scripting language: Similar to Java
- Ability to accept web pages/scripts allows malicious code/viruses to be transported to WAP enabled devices
- WAP sys comm using WAP GW sys.
- GW converts info back/forth between HTTP/WAP as well as encodes/decodes bet the protocols

Packet sniffing:

Gap in the WAP: Vuln where interconnection bet WAP server/net no encrypted: Packets can be sniffed

- Concern exists when converting bet WAP and SSL/TLS: Exposing plain text
- Prevalent versions of WAP prior to 2.0

Wi-Fi Protected Access and WPA2: Designed to address core/easy-to-crack problems of WEP

- Created to implement the 802.11i standard

Difference bet WPA/WPA2: WPA implements most (not all) of 802.11i in order to be able to comm w/older wireless devices that might still need an update through firmware to be compliant

- WPA: Uses RC4 encryption alg w/TKIP
- WPA2 implements full standard: Not compatible w/older devices
- WPA mandates use of TKIP
- WPA2: Requires **CCMP: Counter Mode w/Cipher Block Chaining Message Authentication Code Protocol**

CCMP: Cipher Block Chaining Message Authentication Code Protocol:

- Uses 128 bit AES encryption w/48-bit initialization vector
- Increases difficulty in cracking
- Minimizes risk of replay attack

WTLS: Wireless Transport Layer Security: Sec layer of WAP: Provides

auth/encryption/data integrity for wireless devices

- Designed to narrow BW of these types of devices: Moderately secure
- Reasonable sec for mobile
- Encrypted/auth connection between client/server
- Similar in function to TSL BUT: Uses lower BW/less processing power
- Used to support wireless devices that don't have extremely powerful processors

Comm bet a WAP client and WAP server protected by WTLS

Once on the Internet: Connection is typically protected by SSL (Secure Sockets Layer)

SSL: Internet standard for encrypting data bet points on the network

Understanding Wireless Devices: Device uses WAP? Probably doesn't have sec enabled

7 Levels of sec exist in WAP:

Anonymous Auth	Allows almost anyone to connect to wireless portal
Server Auth	Requires workstation to auth against server
Two-Way Auth (Client/Server)	Requires both ends of connection to auth to confirm validity
WSP	**Wireless Session Protocol** • Manages session info/connection bet devices
WTP	**Wireless Transaction Protocol** • Services similar to TCP/UDP for WAP
WTLS	**Wireless Transport Layer Security** • Security layer of WAP
WDP	**Wireless Datagram Protocol** • Provides common int bet devices

Wireless AP's: Primary method to connect wireless device to network via portal

Wireless AP: Low power transmitter/receiver: transceiver strategically placed for access

- Portable device/AP comm using one of 7 comm protocols: Including 802.11
- Uses portion of RF spectrum called microwave
- Can be less secure
- Verify encryption is turned on / change default passwd

Antenna Placement: Can be crucial in allowing clients to reach AP

- Depends on env of placement
- Greater distance signal travels: More it will attenuate
- Can lose signal quickly over short distance if building materials absorb it
- Avoid placing near ground/near metal
- Place in center of area to be served/high enough to get around most obstacles

Power level controls: Allow you to reduce amt of output provided

Antenna Types: Can be internal on AP: Consist of 1/2/3 external poles

Omnidirectional	A 360-degree pattern • Even signal in all directions
Directional	Forces signal in 1 direction

	• Since focused: Can cover greater distance w/stronger signal

Gain value: All antenna rated in these terms: Expressed in dBi numbers
- Antenna advertised w/20 dBi would be 20x stronger than 0 dBi
- Every 3 dBi bles power output

MAC Filtering: Typically off by default: **MAC address:** Unique identifier that exists for each NIC
MAC filtering: When used, admin compiles list of MAC addresses/machines associated w/them
- When client attempts to connect: Additional check of MAC is done after other values match up
- If address appears in list: Client allowed to join

Network lock: Used in place of MAC filtering terminologically speaking

Changing Order of Preferred Networks:
1. Network/Sharing Center > Click any network in list > Drag it up/down to change order of preferred networks

Captive Portals: Requires users to agree to some condition before using network/Internet
- Cisco app in Identity Services Engine: Vulns identified w/it

Working w/VPNs: VPN: Virtual Private Network: Can be config either Network/Data Link of network
- Enhances sec | For tunneling: Use IPSec/SSL

EAP: Extensible Authentication Protocol: Provides framework for auth often used w/wireless networks
5 EAP types adopted by WPA/WPA2

- **EAP-TSL**
- EAP-PSK
- EAP-MD5
- LEAP
- PEAP

By adding tunneling: TTLS adds 1 more layer of sec against MITM attacks/eavesdropping
WPS: Wi-Fi Protected Setup: # of SOHO routers use series of EAP msgs to allow new hosts to join network using WPA/WPA2
Often requires usr to do something in order to complete enrollment process
- Press button on router w/in short period
- Enter PIN
- Bring new device close-by (near-field comm)

NFC: Near Field Communication: Requires usrs to bring client close to AP to verify (RFID/Wi-Fi) it's present
- Can be used to "**bump**" phones/send data from 1 to another
- WPS attacks commonplace: Brute-force used to guess PIN
- Once attacker gains access: On the Wi-Fi
- Suggestion: Disable WPS in devices that allow it

LEAP	**Lightweight Extensible Authentication Protocol** • Cisco extension to EAP • Being phased out in favor of PEAP • Created as quick fix for problems w/WEP • Lacks native Win support • Requires mutual auth: Susceptible to dictionary attacks Considered weak EAP protocol
PEAP	**Protected Extensible Authentication Protocol** • **Cisco/RSA/MS**: Worked together to create • Replaces LEAP • Native support for Win starting w/XP • More secure b/c it establishes an encrypted channel bet server/client

Wireless Vulns to Know: Vuln to all same attacks as wired networks

- RF signals for data emanation = Additional weaknesses
- RF signals can be easily intercepted

Many networks regularly broadcast SSID:

- Disable SSID broadcast: AKA: **Cloaking**
- Considered very weak sec

Config Wireless Connection Not Broadcasting

1. Right click network icon > Connect to a network > Properties > Connection tab
2. Check > Connect Even if network isn't broadcasting > OK

Site Survey: Involves listening in on existing wireless networks

- Intel/data capture to be performed on sys
- Term initially meant determining whether proposed location was free from interference

When used by attacker: Site surveys can determine what types of sys in use/protocols used/critical info

- Virtually all wireless networks vuln to site surveys

Interference	Can be unintentional (other devices)
Jamming	When interference intentional • Intent to jam signal/keep legitimate device from comm
War driving	Driving around w/laptop looking for AP's to comm w/ • NIC's on intruder laptop set to promiscuous: Looks for signals coming from anywhere • Can gain/steal access/corrupt data
War chalking	Those who discover a way into network leave signals (in chalk) on/outside premise • To notify others a vuln exists there • Marks can be on sidewalk/side of building/nearby signpost/etc…

Rogue AP	Any AP added to your network that hasn't been authorized • May be added by attacker: Could be innocently added • Unintentional: If usr doesn't implement same lvl of sec: Can open door for MITM/evil twin
Evil twin attack	Rogue AP poses as legitimate wireless service provider to intercept info usrs transmit • Emissions from wireless portal may be detectable through walls/7 city

	blocks
Bluejacking	Sending of unsolicited msgs (spam) over Bluetooth
Bluesnarfing	Gaining of unauth access through Bluetooth • Can be obtained through smartphone/any Bluetooth device

From <https://www.piratemoo.net/moosings/information-security/working-wwireless/>

Post 6

Friday, January 25, 2019 12:20 AM

HOST/DATA/APP SEC

October 17, 2016 Moo Comments 0 Comment

Database: Primary tool for data mgmt: Complex set of programs that work together to provide access to data

Relational Database: Most common approach to db implementation

- Data can be viewed in dynamic ways: Usr/admin needs

SQL: Structured Query Language: Common language used to speak to db's

- Allows queries to be config'd in real time/passed to db servers
- Don't confuse SQL w/MS SQL Server

Relational db: Could query db to find records that meet criteria

Early DB's: Connected end usr directly to data through app programs

- Intended to allow easy data access/transactions performed against db
- Businesses allowed customer access to data (tracking orders/purchases/etc

3 Diff models:

One-Tier	***AKA Single-tier environment*** • DB/app exist on single sys/desktops running standalone db • Early Unix implementations also worked this way • Each usr would sign onto term/run dedicated app that accessed data
Two-Tier	Client workstation/sys runs an app that comms w/db running on diff server • Common implementation
Three-Tier	Isolates end usr from db **Introduces middle-tier server:** • Server accepts requests from clients/Evaluates them • Sends them to db server for processing • Server sends data back to middle-tier server • Then data to client sys Middle server: Can also control access to db

NoSQL: Most commercial relational db's use SQL (Oracle/MS SQL Server/MySQL/PostGres/etc..)

NoSQL db: Not relational db: Doesn't use SQL

- Less common than relational | Used when scaling impt

NoSQL vs. SQL

Feature	NoSQL DB	SQL DB
DB	Non-Relational/Distributed	Relational
Schema	Dynamic	Pre-defined
Data Storage	Stores everything in single nested doc • Often XML fmt (doc-based)	Individual records stored as rows in tables • Table based

Benefits	Can handle large volumes of: • Structured/semi-structured/Unstructured data	Widely supported • Easy config for structured data
Scaling	**Horizontal:** Add more servers	**Vertical:** Beef up server
Vendors/Implementations	MongoDB, CouchDB	Oracle, MS, MySQL
Susceptible to SQLi attacks	No: Susceptible to similar injection-type attacks	Yes

Big Data: Extremely large amts of data | Normally can't fit on single server |SAN

SAN: Storage Area Network: Separate network set up to appear as server to main org network

- Network isolation

Fuzzing: Most apps written to accept particular type of data: str/num values/etc…

- Sometimes possible to enter unexpected values/causes app to crash
- Usr may be left w/elevated privs/access to values they shouldn't

Fuzzing: Technique of providing unexpected values as input to an app in order to make it crash

- Values can be random/invalid/unexpected
- Can flood input w/stream of random bits

Prevention: Input validation: Ensure input is of expected type

Secure Coding: Only real defense against XSS/SQLi/Buffer Overflows

OWASP: Open Web Apps Sec Project

- Voluntary group dedicated to forming secure coding practices for web-based apps/mobile/client apps/back-end issues
- Range of coding standards: Most fundamental: **Input validation**

Input validation: OWASP recommends all data input by usr be validated before it's processed

2 primary ways to do input validation:

- **Client-side validation**
- Server-side validation

Client-side	**Takes input that usr enters into txt field:** Client side check for invalid chars/input • Can be as simple as verifying input doesn't exceed required length • Or complete check for SQLi chars **Validation:** Accomplished on client page before any data sent to server
Server-side	**Validating data after server received it:** **Process can include:** • Checking logic to see if data sent conforms to expected params • Unusual to have just server-side validation • May have sys w/only client-side validation • Server-side normally done w/client-side

CERT Secure: CERT: Computer Emergency Response Team: Details standards for secure coding

- Covers many same issues as OWASP: Also language-specific standards

Exception handling: Programs encounter errors: How handled is critical

App Config Baselining

Baselining: Always involves comparing performance to metric:

- Can be done w/any metric (network performance/CPU usage)
- Can be done w/apps
- Do w/key apps prior to major config changes

3 Types of OS patches:

Hotfix	**Immediate/urgent patch** • Represent serious sec issues: NOT optional • Must be applied
Patch	**Some additional functionality** • Non-urgent/Sometimes optional
Service Pack	**Assortment of hotfixes/patches to date** • Apply BUT test so no problems caused by update

Perms: Any given usr will be granted only privs necessary to perform job function

Full Control	Read/Write/Exe/Assign perms to others
Modify	Read/Write/Delete
Read & Execute	Not all files docs: Read/execute priv needed to run program
Read	Read file but not mod it
Write	Mod file
List folder contents	See what's in folder: Not read files

ACL's: List of who can access what resource at what lvl

- Can be phy list

White/Black lists: Special types of ACL's

White list: List of items allowed | **Black list:** Lists of things prohibited

Antimalware: Actions you should take:

AV	Keep definitions up to date • Run on server/workstations • Active monitor: Incoming files/scans
Antispam filters	Help keep majority of unwanted mail from usrs
Antispyware SW	Often ID'd by presence of tracking cookies on hosts
Pop-Up Blockers	Pop-ups (pop-under) unwanted programs running
Host-based Firewalls	1st line of defense against attackers/malware
Host-based IDS	Available for individual hosts SNORT installs a host-based IDS

Pop-up blocker: 3 possible settings for blocking levels

Low	Allows pop-ups from sites considered secure
Medium	(Default) Blocks most pop-ups
High	Blocks all (Ctrl + Alt will override)

WAF: Web application firewall: Can look at every request bet web client/server/ID possible attacks

Host SW Baselining

Security baseline: Defines lvl of sec will be implemented/maintained

- Can choose to set a low baseline by implementing next to no sec
- Or high baseline that doesn't allow usrs to make any changes at all
- AKA: *Performance baseline*

Hardening Web Servers: Every service/capability supported on site potential target

2 areas of interest w/web servers

1. **Filters:** Limit traffic allowed
2. **Controlling access to scripts:** Often run at elevated perm lvls
    - User can break out of script while at elevated lvl
    - Verify all scripts on server have been tested/debugged/approved

Hardening FTP Servers: Servers not intended for high-sec apps b/c of inherent weaknesses

- Create separate drive/subdir on sys to allow file transfers
- If possible use VPN/SSH
- Usually unencrypted
- Separate logon accts/passwds
- Disable anon usr acct

Hardening DNS Servers

3 types of attacks:

DNS DoS	Make sure DNS server SW/OS SW up to date • Use 2F auth w/registrar
Network Footprinting	Footprinting: Act of gathering data about network to find ways to intrude • Looking for vulns/any means of entry • Lots of info stored in DNS servers
Compromising Record Integrity	DNS lookup sys usually involve primary/secondary DNS server • If you make change to primary/2ndary change propagates to other trusted DNS servers Bogus record inserted into DNS server: Record will point to location attacker intends Ensure ALL DNS servers require authentication before updates made DNSSEC: DNS Security Extensions: Created by IETF to add sec/maintain backward compatibility • Checks digital sigs • Can protect info by digitally signing records • Designed to protect against forged DNS data

DNS Poisoning: Existed in early implementation of DNS

AKA cache poisoning: Daemon caches DNS reply packets/which sometimes contain other info (data used to fill packets)

- Extra data can be scanned for info useful in break-in/MITM

ARP Poisoning: Tries to convince network the attacker's MAC is the associated w/IP so traffic sent to attacker's machine

Hardening DHCP Services

- Network/segment: Only 1 DHCP server should be running
- More than 1? Can clash w/each other over which 1 provides the address
- Duplication of TCP/IP addresses/conflicts

NAT: Server can service DHCP-enabled clients

3 Primary backup types:

Full	**All changes to data archived** • Full back up done every 2 hours at 2AM • Sys crashes 10:05AM > Restore it from 10AM Time consuming/negative impact on server performance
Differential	**All changes since last full backup archived** • Full back up done every 2 hours at 2AM & Differential every 2 hours after • Sys crashes 10:05AM > Restore full back up from 2AM And differential backup from 10AM Gets larger each time done Time consuming/resource intensive NOT same impact as full backups but will still slow down network
Incremental	**All changes since last backup of any type archived** • Full back up done every 2 hours at 2AM then incremental every 2 hours • System crashes 10:05AM > Restore last full back up at 2AM • Then each incremental back up done since MUST be restored in order • More complex • Smaller/Less time/doesn't consume many resources

HSM: Hierarchical Storage Management:

- Continuous online backup by using optical/tape/jukeboxes
- Appears as infinite disk to sys/can config to provide closest ver of avail real-time backup

RAID: Redundant Array of Independent Disks

RAID levels:

RAID 0	Striped Disks	**Distributes data across multiple disks: Improved speed** • Read/write performance • No fault tolerance • 2 disks
RAID 1	Mirroring	**Fault tolerance as it mirrors contents of disks** • For every disk: Identical disk in sys • 2 disks • 50% total capacity used for data | 50% for mirror • Primary drive fails? Sys keeps running on backup **Duplexing**: If add another controller to sys: Still RAID 1
RAID 3/4	Striped Disks Dedicated Parity	**3/more disks w/data distributed across disks** • Uses 1 dedicated disk to store parity info • Storage capacity of array reduced by 1 disk • Disk fails? Only partial data loss • Data remaining on other disks along w/parity info allows data recovery
RAID 5	Striped Disks Distributed Parity	**3/more disks: Protects data against loss of any 1 disk** • Similar to RAID 3 • Parity distributed across drive array • Doesn't forgo entire disk for storing parity bits
RAID 6	Striped Disks Dual Parity	**4/more disks: Protects data against loss of any 2 disks** • Adds additional parity block to RAID 5 • Each parity block distributed across drive array • Parity not dedicated to any specific drive

RAID 1+0		**Mirrored data set (RAID 1) which is striped (RAID 0)** • Stripe of mirrors • 4 drives: 2 mirrored to hold half striped data • 2 mirrored for the other half of data
RAID 0+1		**Opposite of 1+0** • Striped mirrored (mirror of stripes) • 4 drives • 2 mirror drives to replicate data on RAID 0 array

Data at rest: Data not currently being transmitted
Data in transit: Info being sent over some connection: Active data
Clustering/Load Balancing:
Cluster: Anytime you connect multiple computers to work/act together as a single server

- Utilize parallel processing/add redundancy/costs

Load Balancing: Split workload across multiple computers

- Servers answering HTTP requests (farms) may not be in same geo location
- If locations are split: Mirror site: Mirrored copy can add geographic redundancy (requests answered quicker)

App Sec

Key Mgmt	Increasing impt: PKI services
Credential Mgmt	Usernames/passwds stored in 1 location: Used to access websites/other computers • Newer versions of Win include Credential Manager to simplify mgmt
Authentication	Teach best practices | Never config app to auto login
Geo-Tagging	GPS coordinates accompany file such as img • Common practice w/smartphones
Encryption	
App White-Listing	Approved list of apps
Transitive Trust/Auth	Anytime 1 entity accepts usr w/out requiring additional authentication on behalf of another • More stops requiring auth the better

Best Practices: DLP: Data Loss Prevention: Sys monitor contents (workstations/servers/networks) to make sure it's not del/rem

- Also monitor who uses data looking for unauthorized access/transmitting data
- Share commonality w/network IPS

HW-Based Encryption Devices HW-based encryption can be applied

- Advanced config in BIOS: Can choose to enable/disable TPM

TPM: Trusted Platform Module: Assists w/hash key generation

- Name assigned to chip that can store crypto keys/passwds/certs
- Can be used to protect smartphones/devices too
- BitLocker can be used w/w/out TPM
- More secure w/it but not needed
- TPM chip may be installed on MB: Set to off in BIOS by default

Verify presence of TPM Chip in win7

1. Control Panel > Sec > BitLocker Drive Encryption > TPM Administration

From <https://www.piratemoo.net/moosings/information-security/hostdataapp-sec/>

Post 7

Friday, January 25, 2019 12:20 AM

INFOSEC: CH. 8 CRYPTOGRAPHY

October 22, 2016 Moo Comments 0 Comment

Cryptography: Science of altering info so it can't be decoded w/out key | Study of cryptographic algs
Cryptanalysis: Study of how to break cryptographic algs | **Cryptology:** 2 subjects taken together
Historical Cryptography
1st recorded crypto efforts 4,000 years ago

- Early efforts: Translating msgs from 1 language to another/substituting chars

Old methods didn't depend on math like modern methods

- Used technique for scrambling text

Cipher: Method used to encode chars to hide their value
Ciphering: Process of using a cipher to encode a msg
2 primary types of non-math crypto

- **Substitution**
- Transposition

Substitution Ciphers: Type of coding/ciphering sys that changes 1 char/symbol into another
Caesar cipher: One of oldest known substitution ciphers

- Used by Julius Caesar
- Shifts all letters a certain # of spaces in alphabet
- Julius Caesar supposedly used shift of 3 to right

I will pass the Security plus test
If you shift each letter three to the right, you get the following

(JKL) (WXY)(JKL)(MNO)(MNO) (QRS)(BCD) etc…
L zloo sdvv wkh Vhfxulwb soxv whvw

Substitution ciphers not for modern use:

- Computer would crack instantly
- Letter/word frequency becomes issue
- All languages have certain words/letter combos that appear more often
- English: 3 letter word is most likely the, or, and
- Single letter word is most likely I or A
- Can guess wkh is word the and L is I
- The more cipher text Easier to decrypt

Other examples? Atbash, PlayFair and Scytale
Multi-Alphabet Substitution: Substitution ciphers didn't change underlying

letter/word frequency of text

- Combated by: Having multiple substitutions

Example: Shift 1st letter by 3 to right | 2nd letter by 2 to right | 3rd letter by 1 to left

Vigenere cipher	**Most famous example of multi-alphabet substitution from historical times** • Used keyword to look up cipher txt in table • Usr took 1st letter in txt | Go to Vigenere table | Match letter from keyword to find ciphertext letter • Each letter in the keyword generated diff substitution alphabet

Transposition cipher: Transposes/scrambles letters in certain manner

- Msg broken into blocks of equal size | Each block then scrambled

Rail Fence Cipher: Example of transposition cipher

- Write msg letters out diagonally over number of rows | Read off cipher row by row

For example: You write the message out as

m e m a t r h t g p r y
e r e f e t e o a a t
Ciphertext:
MEMATRHTGPRYETEFETEOAAT

Another example:

Moon	**Beams**	**are**	**Nice**
on Mo	**amsBe**	**re A**	**ce.Ni**

Each char including spaces moved to right 3 positions

ROT13: One of oldest known encoding algs

- Rotate every letter 13 places in alphabet
- A becomes N | B becomes O
- Same rotation of 13 letters used to encrypt message also decrypts

Easiest way to solve:

- Write A to M letters in 1 column | N to Z in another
- Replace letter in msg w/one that appears beside it in other column

Examples:

Neg snve qrohgf urer Fngheqnl
Art fair debuts here Saturday!

Gevcyr pbhcbaf ng Xebtre!
Triple coupons at Kroger!

Gel lbhe unaq ng chmyrf.
Try your hand at puzzles.

Enigma Machine: A typewriter that implemented multi-alphabet substitution cipher

- Key was hit: Diff substitution alphabet used
- Used 26 diff substitution alphabets

Steganography: Process of hiding msg in medium such as digital image/audio/other file

Uses LSB: Least significant bit method

- Everything stored in bits organized into bytes
- Single pixel on screen is stored in 3 bytes/24bits
- If change last bit (least significant in each byte): NOT noticeable change in img
- Can store data by putting it in least significant bits of file
- Also possible with audio/video/any digital file type

Steganography can be used for digital watermarking
Transport Encryption Can be done in tunneling/transport mode
Tunneling: Data or payload/msg headers encrypted
Transport: Encrypts only payload
Modern cryptography is divided into 3 major areas

- **Symmetric crypto**
- Asymmetric crypto
- Hashing algs

Symmetric Algorithms: Require both ends of encrypted msg to have same key/processing algs

- Generates secret key that must be protected

Symmetric key: AKA: Secret/Private key: Key that isn't disclosed to people who aren't authorized

- If key lost/stolen: Process is breached

Symmetric crypto algs:

- Always faster than asymmetric
- Can be just as secure w/a smaller key size

Example: RSA (asymmetric alg) uses keys of a min length of 2048 bits

- AES (symmetric alg) uses key sizes of 128/192/256 bits

Symmetric methods use either block/stream cipher
Block: Alg works on chunks of data: Encrypting 1/moving to next
Stream: Data encrypted 1 bit/byte at time
Common standards that use symmetric algs

DES	**Data Encryption Standard:** Used since mid-70's • Primary standard used in gov/industry until replaced by AES • 56 bit key • 7 modes that offer sec/integrity • Considered insecure bc of small key size
3DES	**Triple DES (Data Encryption Standard)** • Upgrade from DES • Still used even though AES preferred choice for gov apps • Harder to break than many other systems • More secure than DES • Increases key length to 168 bits (using 3-56 bit DES keys)
AES	**Advanced Encryption Standard** • Replaced DES as current standard • Rijndael alg • Developed by Joan Daemen/Vincent Rijmen • Current product used by US gov agencies • Supports key sizes of: 128/192/256 bits w/128 bits default

AES-256	**256 bits instead of 128** • Qualifies for gov classification as Top Secret
CAST	**Alg developed by Carlisle Adams/Stafford Tavares (CA-ST)** • Used some MS/IBM products • 40-bit to 128-bit key • Fast/efficient **2 additional versions** • CAST-128 • CAST-256
RC	**Ron's Cipher/Ron's Code** • RSA laboratories • Author: Ron Rivest **Current levels:** • RC4 • RC5 • RC6 **RC5:** Uses key size up to 2048 • Considered strong system **RC4:** Popular w/wireless/WEP/WPA • Streaming cipher that works w/key sizes bet: 40-2048 bits • Used in SSL/TLS • Popular w/utilities used w/BitTorrent files since providers limit DL **Obfuscates header/stream:** • More difficult to see files being moved
Blowfish/Twofish	**Team led by Bruce Schneier** • 64-bit block cipher at very fast speeds • Symmetric block cipher • Can use variable length keys from 32 bits – 448 bits **Twofish:** Similar/works on 128-bit blocks • Distinctive feature of latter: Complex key schedule
IDEA	**International Data Encryption Alg** • Developed by Swiss consortium • 128-bit key • Similar in speed/capability to DES but more secure • Used in **PGP (Pretty Good Privacy):** Public domain encryption sys used by many for email Ascom AG holds right to market
One-Time Pads	Only truly secure crypto implementation 2 reasons why: 1. Uses a key that's as long as plain-txt msg: No pattern in key app for attacker to use 2. Only used once: Discarded Even if you break 1-time pad cipher: Same key never used again • Knowledge of it would be useless

Key Exchange
2 primary approaches to key exchange

- **In-band key exchange**
- Out-of-band key exchange

In-band key exchange: Key is exchanged w/in same comm channel going to be encrypted (IPSec uses it)
Out-of-band exchange: Some other channel than one going to be secured used to exchange key
Forward secrecy: Property of any key exchange system

- Ensures if 1 key compromised: Subsequent keys won't also be compromised

Perfect forward secrecy: Occurs when this process is unbreakable: Common approach uses ephemeral keys
Asymmetric Algs Uses 2 keys to encrypt/decrypt data
These asymmetric keys are referred to as:

- **Public key**
- Private key

Sender uses public key to encrypt msg

- Receiver uses private key to decrypt msg
- What 1 key does: Other undoes

Symmetrical systems: Require key to be private bet 2 parties
Asymmetrical systems: Each circuit has 1 key

- Public key may be truly public/or a secret bet the 2 parties
- Private key kept private: Only owner (receiver) knows it

If someone wants to send msg: Can use your public key to encrypt msg then send it to you
Can use your private key to decrypt msg: Private key always kept protected

RSA	**Named after: Ron Rivest, Adi Shamir, Leonard Adleman** • Early public-key encryption system uses large integers as basis for this process • Widely implemented: Became standard • Works w/both encryption/digital signatures • Used in many envs including SSL (Secure Sockets Layer) • Can be used for key exchange

Key generation example

1. Generate 2 large random primes: p/q approx. equal size: n=pq is required bit length (2048 bits/4096 bits/etc)

Let n = pq
Let m = (p-1)(q-1)

1. Choose small number e, co-prime to m (note: 2 #'s co-prime if they have no common factors):

Find d, such that
de % m = 1

1. Publish e/n as public key. Keep d/n as secret key

Encrypt as follows: **C=M^e % n**
or put another way: compute ciphertext:
c = m^e mod n

1. Decrypt as follows:

P = C^d % n
or put another way: use this private key (d,n) to compute:

m = c^d mod n

Diffie-Hellman	**Whitfield Diffie/Martin Hellman: Diffie-Hellman key exchange** **Considered founders of public/private key concept:** • Original work envisioned splitting key into 2 parts • Alg used primarily to send keys across public networks • Process not used to encrypt/decrypt msgs • Merely for creation of symmetric key bet 2 parties **Method actually developed a few years earlier:** • Malcolm J. Williamson of British Intelligence Service: Classified **Exam**: If asked about alg for exchanging keys over insecure medium: • Unless context IPSec: Answer Diffie-Hellman
ECC	**Elliptic Curve Cryptography:** • Similar functionality to RSA: Uses smaller key sizes to obtain same lvl of sec • Based on using points on curve combined w/point at infinity/difficulty of solving discrete log problems • NSA recommended 7 implementations of ECC **Many variations of Elliptic Curve:** **ECC-DH: Elliptic Curve Diffie-Hellman** **ECC-DSA: Elliptic Curve Digital Signature Alg**
ElGamal	**Taher ElGamal in 1984** • Asymmetric: 7 variations created including Elliptic Curve ElGamal • ElGamal/related algs use ephemeral key **Ephemeral key:** Key that exists only for that session • Alg creates key to use for single comm session/not used again

Adding ephemeral key to:

- Diffie-Hellman: Turns to **DHE (Ephemeral Diffie-Hellman)**
- Elliptic Curve Diffie-Hellman: Turns to **ECDHE**

Alg	Common Use
Diffie-Hellman	Key agreement
ElGamal	Transmitting digital sigs/key exchanges
ECC: Elliptic Curve	Option to RSA: Uses less computing power than RSA/popular in smaller devices like smartphones
RSA	Most commonly used public-key alg: Used for encryption/digital sigs

What Crypto to Use?

Key principle: Kerckhoffs' principle

1st stated by Auguste Kerckhoffs in 19th century

- Sec of alg should depend only on secrecy of key/not secrecy of alg
  - Algs could be public to examine: Process will still be secure as long as key is secret

Hashing Algs: Hashes used to store data such as hash tables diff from crypto hashes

In crypto hash function may have 3 characteristics:

• **Must be 1-way**	**NOT** reversible: Once hashed: Can't unhash
• **Var-length input produced fixed-length output**	Whether you hash 2 chars/2 million: Hash size same
3. Alg must have few/no collisions	Hashing 2 diff inputs doesn't give same input

Hashing algs to be familiar w/

SHA	**Secure Hash Alg** • Designed to ensure integrity of msg **SHA:** 1-way hash that provides hash value that can be used w/encryption protocol: • Produces 160-bit hash value **SHA-2:** 7 sizes: 224, 256, 334, 512 • Most widely used • No known issues **SHA-3:** Now standard Originally named Keccak: Designed by Guido Bertoni/Joan Daemen/Michael Peeters/Gilles Van Assche
MD	**Message Digest Alg** • Creates hash value: Uses 1-way hash • Hash value used to help maintain integrity • 7 versions • Most common are: MD5/MD4/MD2 **MD4:** Used by NTLM to compute NT hash **MD5:** Newest version: Produces 128-bit hash • More complex than predecessors/offers greater sec **Biggest weakness:** • Doesn't have strong collision resistance: No longer recommended **SHA 1/2:** Recommended alts
RIPEMD	**RACE Integrity Primitives Eval Msg Digest** • Based on MD4 • Questions regarding sec: Replaced by RIPEMD-160 (uses 160-bits) • Ver that exist use 256/320 bits • All versions of RIPEMD remain
GOST	**Symmetric cipher** • Developed in old Soviet Union • Modified to work w/hash functions • Processes var-length msg into fixed-length output of 256 bits
LANMAN	**Prior to NT: MS used LANMAN protocol for authentication** • Used LM hash/2 DES keys • Replaced by NT LAN Manager (NTLM) w/NT
NTLM	**NT LAN Manager** • Replaced LANMAN protocol w/release of Win Nt • Uses MD4/MD5 hashing algs • 7 versions of this protocol exist (NTLMv1, NTLMv2) and is still in widespread use • Despite the fact that MS has pointed to Kerboros as being preferred authentication protocol • Both LANMAN/NTLM used for the purpose of authentication primarily

Rainbow Tables and Salt

Since hashing alg isn't reversible: May think it's impossible to break hash: Not true

Rainbow table: All possible hashes computed in advance

- Creates series of tables
- Each has all possible 2-letter, 3-letter, 4-letter..combos: Hash of that combo using hashing alg like SHA-2
- If you search table for given hash: Letter combo in table has passwd seeking

- Password cracking tools: Example: OphCrack use rainbow tables

Salt: Countermeasure: Addition of bits at key locations: Either before/after hash

- If you type in the passwd letmein: Bits added by OS before hashed
- Using Salt: Should someone apply rainbow table attack: Hash they search for will yield a diff letter combo

Key Stretching: Process used to take key that might be weak/making it stronger (usually by making it longer)

- Key is input into an alg that will strengthen it/make it longer/less susceptible to brute-force attacks
- Many methods

2 methods

PBKDF2	**Passwd-Based Key Derivation Function 2** • Part of PKCS #5 v2.01 • Applies some function (hash/HMAC) to passwd along w/Salt to produce derived key
Bcrypt	**Used w/passwords** • Uses derivation of Blowfish alg, converted to hashing alg to hash passwd and add Salt

Quantum Crypto: Based on smallest particles known: Only method currently practical is **QKE: Quantum Key Exchange**

Cryptanalysis Methods: Common Code-Breaking Techniques:

Frequency Analysis	**Looking at blocks of an encrypted msg to determine any common patterns** • Only works on historical algs: Not modern ones
Chosen Plaintext	**Attacker obtains ciphertxts corresponding to set of plaintexts of their choosing** • Allows attacker to attempt to derive key used/decrypt msgs w/that key • Can be difficult but not impossible • Methods such as differential cryptanalysis chosen plaintext attacks
Related Key Attack	**Like chosen-plaintext attack:** Except attacker can obtain ciphertexts encrypted under 2 diff keys • Very useful if you can obtain plaintext/matching ciphertext
Brute-Force Attacks	
Human Error	**Major cause of encryption vulns:** • If an email sent using encryption scheme: Someone else may send it in clear (unencrypted) • If cryptanalyst gets hold of both msgs: Process of decoding future msgs simplified • Codekey might wind up in wrong hands giving insights into what key consists of

Cryptographic system: Sys/method/process used to provide encryption/decryption

- HW/SW/Manually performed
- Exist for same reasons sec exists: Provide confidentiality/integrity/authentication/nonrepudiation/access control

Confidentiality/Strength :

Strength: Effectiveness of crypto sys in preventing unauthorized decryption

- Strong sys is diff to crack | AKA work factor

Work factor Describes estimate of amt of time/effort needed to break sys

- May be considered weak if uses weak keys/defective design/easily decrypted

Cipher suites work w/SSL/TLS to combine authentication/encryption/msg auth

- Most vendors allow you to set cipher suite preferences on a server to determine lvl of strength
- Sybase: Can set the cipher suite preferences to Weak/Strong/FIPS/All
- Strong: Limits choices to only encrypt algs that use keys of 64 bits/more
- Weak: Adds all encryption algs less than 64 bits
- FIPS requires encryptions/hash/key exchange algs to be FIPS complaint
- AES/3DES/SHA1
- Apache offers similar choices but instead of words strong/weak uses High/Medium/Low

Integrity: Assurance msg wasn't modified during transmission: Modification may render it inaccurate

Common method of verifying integrity involves adding a MAC

MAC: Msg Auth Code to msg: Derived from msg/shared secret key

Ensures integrity of msg:

- MAC would be encrypted w/msg adding another layer of integrity checking
- From the MAC you'd know if it came from originator/hadn't been altered
- Receiver also calcs MAC value/compares it to value sent in msg

HMAC: Hash-Bashed Msg Authentication Code: Uses hashing alg along w/symmetric key

Digital Signatures: Similar in function to standard sig on doc

- Validates integrity of msg/sender
- Msg encrypted using encryption sys/2nd piece of info – digital sig – is added to msg

Message digest: Receiver compares sig area in msg w/calc value

- If values match: Msg hasn't been tampered w/originator verified as person they claim

Authentication: Process of verifying sender is who they say they are

- Common method of verifying authenticity is addition of a digital sig

Nonrepudiation: Prevents 1 party from denying actions they carried out

Example: You're gone – kids break something – they're all innocent – you have a cam though so you can tell who is lying

- Similar type of proof can be achieved in a 2-key system: Anyone can claim to be legitimate sender

Cas: Certificate Authority: 3rd party orgs that manage public keys/issue certs verifying validity of sender's msg

- Verifying aspect serves as nonrepudiation: Respected 3rd party vouches for individual
- Goal of any effective crypto system must include nonrepudiation

Key Features

Key escrow	Addresses possibility a 3rd party may need to access keys • Keys needed to encrypt/decrypt data held in escrow acct/made avail if 3rd party requests them

Key recovery agent	Entity has ability to recover key/key components/plaintext msgs as needed • Unlike escrow: Recovery agents typically used to access info encrypted w/older keys
Key registration	Process of providing certs to usrs/RA • **RA: Registration Authority:** Handles this function when load must be lifted from **CA: Cert Authority**

Issue of keys no longer used: Key may have expired: May have been canceled due to breach of sec/replaced
CRL: Certificate Revocation List: Most widely used: List of certs a CA states should no longer be used
Being replaced by a real time protocol called: **OCSP: Online Cert Status Protocol**
Trust models: Exist in PKI implementations/come in # of types
4 main types of trust models used w/PKI are:

- **Bridge**
- Hierarchical
- Hybrid
- Mesh

Origins of Encryption Standards 7 US gov agencies: Involved in creation of standards for secure sys

NSA	**National Security Agency** • Responsible for: Creating/breaking codes/coding systems for US gov • Chartered 1952 • Obtains foreign intel/supplying to US gov agencies that need it
NSA/CSS	**NSA/Central Security Service** • Independently functioning part of NSA • Created in early 70's • Supports all branches of military • Helps standardize/support DoD
NIST	**National Institute of Standards/Technology** • Formerly known as **NBS: National Bureau Standards** • Developing/supporting standards for US gov for over 100 years • Became very involved in crypto standards/sys/tech in variety of areas • NIST publishes info about known vulns in OS/apps

Industry Associations and the Developmental Process

RFC	**Request for Comments** • Originated in 1969 • Mechanism used to propose standard • Doc-creation process w/set of practices • Processed through designated RFC editor
AMA	**American Bankers Association** • Banking/financial industries
IETF	**Internet Engineering Task Force** • Improving Internet: Computer sec issues • Uses working groups to develop/propose standards
ISOC	**Internet Society**

	• Professional group: Membership consists primarily of Internet experts • Oversees # of committees/groups including IETF
W3C	**World Wide Web Consortium** • Concerned w/interoperability/growth/standardization of WWW
ITU	**International Telecommunications Union** • Responsible for all aspects of telecom/radio comm standards worldwide **3 main groups** **1. ITU-R**: Concerned w/radio comm/spectrum mgmt **2. ITU-T:** Concerned w/telecom standards **3. ITU-D:** Concerned w/expanding telecom in undeveloped countries Headquartered in Switzerland operates as sponsored agency in UN
IEEE	**Institute of Electrical/Electronics Engineers** • Focused on tech/related standards • Actively developed of PKC/wireless/networking protocol standards

Public-Key Infrastructure X.509 (PKIX): Working group formed by IETF to dev standards/models for PKI env

PKCS: Public-Key Crypto Standards: Set of voluntary standards created by RSA/sec leaders

Early members of this group included Apple/MS/DEC(now HP)/Lotus/Sun/MIT

15 published PKCS standards: Coordinated through RSA

PKCS 1	RSA Crypto
PKCS 2	Incorporated in PKCS 1
PKCS 3	Diffie-Hellman Key Agreement
PKCS 4	Incorporated in PKCS 1
PKCS 5	Password-Based Crypto
PKCS 6	Extended-Cert Syntax
PKCS 7	Crypto Message Syntax
PKCS 8	Private-Key Info Syntax
PKCS 9	Selected Attribute Types
PKCS 10	Cert Request Syntax
PKCS 11	Crypto Token Interface
PKCS 12	Personal Info Exchange Syntax
PKCS 13	Elliptic Curve Crypto
PKCS 14	Pseudorandom Number Generators
PKCS 15	Crypto Token Info Fmt

X.509 Standard

- Defines the cert fmts/fields for public keys
- Also defines the procedures that should be used to distribute public keys
- X.509 version 2: Still used primary method of issuing CRL certs

Current version X.509 is version 3

Comes in 2 types

End-Entity Certificate	Most common: Issued by CA to an end entity **End entity:** Sys that doesn't issue certs but merely uses them
CA Certificate	CA cert issued by 1 CA to another CA

	• 2nd CA can then issue certs to end entity

All X.509 certs have the following:

- Signature: Primary purpose of cert
- Version
- Serial number
- Signature alg ID
- Issuer name
- Validity period
- Subject name
- Subject public-key info
- Issuer unique identifier (for versions 2/3 only)
- Subject unique identifier (versions 2/3 only)
- Extensions (version 3 only)

SSL/TLS: Secure Sockets Layer: Used to establish secure comm connection bet 2 TCP-based machines

- Protocol uses handshake method of establishing session
- # of steps in handshake depends on whether steps combined and/or mutual auth is included
- # of steps is always bet 4-9 inclusive: Based on who is doing documentation
- One of early steps will always be to select appropriate cipher suite to use

Cipher suite: Combo of methods such as an auth/encryption/msg auth code (MAC) algs used together

- When connection request made to server > Server sends msg back to client indicating secure connection needed
- Client sends server cert indicating capabilities of client
- Sever evals cert/response w/session/encrypted key
- Session secure at end of process
- Session will stay open until 1 end/other issues cmd to close

In order for SSL to work properly: Clients must be able to accept the lvl of encryption you apply

TLS: Transport Layer Security: Sec protocol that expands on SSL AKA SSL 3.1: Doesn't interoperate w/SSL

Certificate Management Protocols: CMP: Messaging protocol used bet PKI entities/used in some PKI envs

XKMS: XML Key Management Specification: Designed to allow XML-based programs access to PKI services

Secure Multipurpose Internet Mail Extensions

S/MIME: Secure Multipurpose Internet Mail Extensions: Standard used for encrypting email

- S/=Secure version of MIME: Published to Internet as standard by RSA

SET: Secure Electronic Transaction:

- Provides encryption for credit card numbers that can be transmitted over Internet
- Visa/MC developed
- Works in conjunction w/electronic wallet that must be set up in advance of transaction

Electronic wallet: Device that ID's electronically in same way as cards carried in wallet

SSH: Secure Shell: Tunneling protocol originally used on Unix sys

- Avail both Unix/Win
- Handshake process bet client/server similar to process described in SSL
- Primarily intended for interactive term sessions

Process:

Phase 1: A secure channel to negotiate the channel connection

Phase 2: A secure channel used to establish the connection

PGP: Pretty Good Privacy: Freeware email encryption sys introduced in early 90's

- Uses both symmetrical/asymmetrical sys as a part of process
- This combo of processes makes it so competent

During encryption process: Doc is encrypted w/public and session key

- 1-use random # to create ciphertext
- Session key encrypted into public key/sent w/ciphertext
- On receiving end: Private key used to ascertain session key
- Session/private key are then used to ascertain session key
- Session/private key then used to decrypt ciphertext back into original doc

Alternative to PGP: Freeware: GPG: GNU Privacy Guard

- Interoperable w/PGP
- Considered hybrid program since also uses combo of symmetric/public-key crypto

Secure HTTP: S-HTTP: HTTP w/msg sec (added by using RSA/digital cert)

HTTPS creates sec chan || S-HTTP creates sec msg

- Can use multiple protocols/mechanisms to protect the message

IP Sec: Sec protocol provides auth/encryption across Internet

- Becoming standard for encrypting VPN chan/built into IPv6
- Primary use: To create VPNs
- IPSec in conj. w/L2 Tunneling Protocol (L2TP)/L2 Fwding (L2F) creates packets difficult to read if intercepted by 3rd party
- Works at L3 of OSI

2 primary protocols used by IPSec are:

AH	Authentication Header | Protocol 51
ESP	Encapsulating Security Payload | Protocol 50

Both operate in either the transport/tunnel mode

Federal Info Processing Standard: FIPS: Set of guidelines for US federal gov info sys

- Used when existing commercial/gov doesn't meet federal sec requirements | Issued by NIST

Using Public-Key Infrastructure

PKI: Public-Key Infrastructure: Intended to offer a means of providing security to messages/transactions on a grand scale

PKI: 2 key asymmetric systems w/4 main components:

1. CA: Certificate Authority
2. RA: Registration Authority

3. RSA
4. Digital certs

Msgs encrypted w/public key and decrypted w/private key

Main goal: To define infrastructure that should work across multiple vendors/sys/networks

- Framework not specific technology
- Dependent on perspective of SW vendors that implement it
- Major difficulties? Each vendor can interpret doc about this however they choose
- Major functions/components of PKI infrastructure and how they work in relationship to entire model

CA: Certificate Authority	**Org responsible for issuing/revoking/distributing certs** • Mechanism that associates public key w/individual • Contains lots of info about usr • Each usr of PKI sys has cert that can be used to verify authenticity **Steps in getting cert:** • Submit CSR: Certificate Signing Request • Request fmt for CA • Can be either priv/public w/VeriSign being 1 of best known of public variety

Working w/RA's/Local Registration Authorities

Registration Authority	**Offloads some of work from CA** • RA sys operates as middleman in process: • Can distribute keys/accept registrations for CA/validate ID's • RA doesn't issue certs: That's CA
Local Registration Authority	**LRA: Takes process a step further** • Can be used to ID/establish the ID of an individual for cert issuance

Implementing Certs: Provide primary method of ID'ing a given usr is valid

- **Can also be used to store authorization info**

X.509: Most popular cert used is version 3

- Standard cert fmt supported by ITU
- Purpose of cert: Bind public key to usr's ID
- When authenticating: Certs can be used to auth only the client (single sided) or both parties (dual sided)

Certificate Policies: Define what certs do

Cross certification: Process of requiring interoperability

- Orgs using the cert also have right to decide which types of certs used/for what purposes

Certificate Practice Statements: CPS: Detailed statement CA uses to issue certs/implement policies

- CA provides the CPS to users of its services
- How certs issued/what measures taken to protect certs/Rules CA usrs must follow to maintain eligibility
- If CA unwilling to provide info to usr: CA may be untrustworthy

Certificate Revocation: Process of revoking cert before it expires

- Handled either through **CRL: Certificate Revocation List**
- or using **OCSP: Online Certificate Status Protocol**

Repository: DB/DB server where certs stored

- Once certs revoked: Can't be used/trusted again

Latency: Gap bet when CRL issued/when it reaches usrs may be too long for some apps

Implementing Trust Models: For PKI to work: Capabilities of CAs must be readily avail to usrs

4 main types of trust models used w/PKI's

1. **Hierarchical**
2. **Bridge**
3. **Mesh**
4. **Hybrid**

Granularity: Ability to manage individual resources in CA network

Hierarchical trust: AKA Tree	**Root CA at top provides all info** • Intermediate CAs next in line: Trust only info provided by root CA • Root CA: Also trusts intermediate CA's that are in their lvl hierarchy/none that aren't • Arrangement allows high lvl of control at all lvls of tree **Most common implementation:** • Large org that wants to extend its cert processing capabilities **Leaf CA's:** Any CA that is at the end of a CA network/chain
Bridge Trust	**Peer-to-peer relationship exists among root CA's** • Root CA's can comm w/one another allowing cross certification • Allows cert process to be established bet orgs/depts Each intermediate CA trusts only CA's above/below it • CA structure can be expanded w/out creating additional layers of CA's Advantages: • Flexible/interoperability bet orgs Disadvantage: • Lack of trustworthiness of root CA's • Illegitimate cert could become avail to all usrs in bridge structure/intermediate CA • Large geographically dispersed org/when you have 2 orgs working together
Mesh Trust	**Expands concepts of bridge model by supporting multiple paths/multiple root CA's** • Each root CA can cross-certify w/other root CA's in mesh as web structure • Useful in situation where 7 orgs must cross-certify certs Advantage: • More flexibility when config CA structure Disadvantage: • Each root CA must be trustworthy in order to maintain security
Hybrid Trust	**Can use capabilities of any/all structures above** • Flexibility of this model also allows you to create hybrid envs • Disadvantage: Can become complex/confusing • A user can unintentionally acquire trusts that they shouldn't have obtained

From <https://www.piratemoo.net/moosings/information-security/infosec-ch-8-cryptography/>

Post 8

Friday, January 25, 2019 12:20 AM

MALWARE/VULNS/THREATS

October 28, 2016 Moo Comments 0 Comment

Understanding Malware: SW exploitation: Attacks launched against apps/higher-lvl services
Includes: Gaining access to data using weaknesses in the data access objects of a DB | Flaw in service/app

Viruses/access attacks:

Spyware	**Works on behalf of 3rd party:** Differs from other malware: • Spread to machines by usrs who inadvertently ask for it • DL'ing programs/infected sites/etc… **What it does:** • Monitors usr activity/reports to another party w/out informing usr • Gathers info about usr to pass onto marketers • Intercepts personal data **What separates spyware from malware?** • Commercial gain
Adware	**Purpose:** Deliver ads/generate revenue for creator • Can have same qualities/similar to spyware
Rootkits	**Programs that have ability to hide things from OS** • May be # of processes running on sys that don't show up in Task Manager • May have established connections that don't appear in netstat • Masks presence of these items **They do this by:** • Manipulating function calls to OS • Filtering out info that would normally appear • Can theoretically hide anywhere there is enough mem to reside (video/PCI cards/etc) • Many written to get around AV/Antispyware programs
Trojan Horses	**Programs that enter sys/network under guise of another program** • Attachment/part of an install • Can create backdoors/replace valid programs during install • Always back up after install new SW/OS's Things to look for? If app opens TCP/UDP ports it shouldn't
Logic Bomb	**Programs/code snippets that execute when certain predefined event occurs** • Bomb may send note to attacker when usr logged on/using word processor • Msg informs attacker usr is ready for attack • Doesn't begin attack: Tells attacker victim has met criteria
Backdoor	**2 diff meanings:** 1. **Original:** Troubleshooting/dev hooks into sys: Went around normal auth ○ Allowed operation examination inside program while code running ○ Stripped out code when moved into production • **Gaining access/inserting program/utility that creates entrance for attacker** ○ May allow certain UID to log on w/out passwd: Admin privs ○ Back Orifice/NetBus: Remote admin tools

	▪ Used by attackers to take control of Win based sys ▪ Typically installed using Trojan ▪ Allow remote user to take full control of sys which are installed
Botnets	**SW running on infected computers called zombies** Bots: Form of SW that runs auto/autonomously Botnet: Malicious SW running on zombie under control of bot-herder
DoS/DDoS	**Denial-of-Service/Distributed-Denial-of-Service attacks:** Can be launched by botnets • Also many forms of adware/spyware/spam (spambots) • Most bots written to run in BG w/no visible evidence of presence • Many malware kits can be used to create botnets/mod existing ones
Ransomware	**SW, often delivered through Trojan: Takes control of sys: Demands 3rd party be paid** • Control can be accomplished by encrypting HDD's/changing passwd info, etc… • Usr usually assured by paying amt: They'll be given code to revert sys

Surviving Viruses

Virus: Piece of SW designed to infect sys

3 ways:

1. Contaminated media (DVD/USB/CD)
2. Email/social networking sites
3. As part of another program

Can be classified as:

Polymorphic	Change form in order to avoid detection
Stealth	Attempt to avoid detection by masking themselves from apps
Retroviruses	Attack/bypass AV installed on machine
Multipartite	Attack sys in multiple ways
Armored	Designed to make itself difficult to detect/analyze
Companion	Attaches self to legitimate program then creates program w/diff file extension
Phage	Modifies/alters other programs/DB's
Macro	Exploits enhancements made to apps, which are used by programmers to expand capability

Each virus type has a diff attack strategy/diff consequences

Symptoms:

- Sys start loads more slowly (spreading to other files in sys/resources)
- Unusual files appear on HDD/disappear from sys (many del key files)
- Program sizes change from installed vers (attaching itself to programs on disk)
- Browser/word processing/SW exhibits unusual chars (screens/menus change)
- Sys shuts down/starts/does great deal of disk activity
- Lose access to disk/other resources (changed settings on device to make unusable)
- Sys doesn't reboot/gives unexpected error msgs during startup

Armored	**Makes self diff to detect/analyze:** • Cover w/protective code: Stops debuggers/disassemblers from examining critical elements • Some aspects of code as decoy to distract from analysis while actual code hides other areas in programs

	• More time takes to deconstruct: Longer it can live: More it can replicate/spreads
Companion	**Attaches self to legitimate programs: Creates program w/diff file extension** • May reside in temp • Usr tries to type legitimate program companion executes instead • Makes changes to pointers in Registry so they point to infected program
Macro	**Exploits enhancements made to apps used by programmers to expand capability of app (word/excel)** • Word supports mini-BASIC: Allows files to be manipulated auto • Programs in doc are called macros • Can tell word processer/spell check doc auto when open • Can infect all docs on sys/spread to other sys via email/other
Multipartite	**Attacks in multiple ways** • Boot sector/exe files/destroy apps/etc.. • Hope: Usr won't be able to correct all problems
Phage	**Mods/alters other programs/DB's** • Infects all of these files • Only way to remove is to reinstall • If a single incident of the virus is missed: it will start again
Polymorphic	Change form in order to avoid detection • Virus/malware • Mutation: Will encrypt parts of itself, hide from AV SW, delete files on a sys, etc…
Retrovirus	**Attacks/bypasses AV** • Can directly attack AV potentially destroy virus def db file • Leaves usr w/false sense of sec
Stealth	**Attempts to avoid detection by masking itself from apps** • May attach to boot sector of HDD • When sys util/program runs/redirects cmds around self to avoid detection • May move from A to B during virus scan

Managing Spam to Avoid Viruses

Spam: Not truly virus/hoax: **Any unwanted/unsolicited email:** Can open door to other problems

- Issues with antispam programs: False positives (can flag legitimate emails): Always check spam folder

Spam found its way into other forms:

SPIM: Spam over Instant Messaging

SPIT: Spam over Internet telephony

AV Software:

- Should be at gateways, servers and desktops
- Can use SW at each location from diff vendors

Understanding Various Types of Attacks

Attack: Occurs when an unauthorized individual/group of, attempts to access/mod/damage sys's or envs.

Attackers have various reasons:

- Fun
- Criminals attempting to steal
- Political statement/terrorism

ID'ing DoS/DDoS Attacks

DoS: Denial-of-Service: Prevent access to resources by usrs authorized to use them

- Occur from single system: Specific server/org may be target

Types of attacks occur in this category:

1. Deny access: Info/apps/sys/comm
2. Bring down site while comm/sys continue to operate
3. Crash OS
4. Fill comms channel of network/prevent access by auth usrs
5. Open as many TCP sessions as possible; TCP SYN flood DoS attacks

2 most common DoS attacks:

1. **Ping of death:** Crashes sys by sending ICMP (Internet Control Msg Protocol) packets larger than sys can handle
    1. Example: sPing
2. **Buffer overflow:** Attempts to insert more data (usually long input strs) into buffers then can hold
    1. Examples: Code Red, Slapper, Slammer

DDoS: Distributed-Denial-of-Service:

- Amplifies concepts of DoS by using multiple systems (botnets) to conduct attack
- Exploit inherent weaknesses of dedicated networks like DSL/Cable

Master controller: Orchestrates attack/may be another unsuspecting usr

Zombies/Nodes: Sys taking direction from master control computer

- Merely carries out the instructions they've been given

Difference bet DoS/DDoS? DDoS's use multiple computers to attack target, DoS don't

Spoofing Attacks:

Spoofing Attack	**Attempt by someone/something to masquerade as someone else (considered access attack)** **Most popular today**: **IP spoofing:** Goal: Make data look as if it came from trusted host **ARP spoofing: AKA ARP poisoning**: MAC (Media Access Control) address data faked • By faking value: Possible to make look as if came from network didn't • Can be used to gain access/fool router into sending data intended for another host • Or to launch DoS attack • Address being faked is address of legitimate user **DNS spoofing: AKA: DNS poisoning:** DNS server given info about name server that it thinks is legitimate • Can send usrs to website other than 1 to wanted • Reroute mail/redirection where data from DNS server used to determine destination **Domain name kiting:** When new domain issues 5-day grace period before you must pay • Those engaged in kiting delete acct w/in 5 days/re-register it • Allows to have accts never paid for **NOTE:** Spoofing tricks something/someone into think something legitimate is occurring

Pharming: Form of redirection in which traffic intended for 1 host is sent to another

- Small scale: Changing entries in hosts file
- Large scale: Changing entries in DNS server (poisoning)
- Either case: When usr attempts to go to site: Redirected to another

Phishing, Spear Phishing, Vishing

Phishing: A form of SE where you ask someone for piece of info you're missing by making it look like legitimate request

Spear phishing: Unique since msg made to look as if it came from someone you know/trust as opposed to 3rd party

Vishing: Combining phishing w/VoIP: Elevated form of social engineering

- Makes possible for someone to call you from almost anywhere in world w/out worry of tracing, caller ID/etc..

Xmas Attack: Nmap (network mapping): Known as Xmas attack/Xmas scan/ Christmas Tree attack

- Advanced scan that tries to get around firewall detection/looks for open ports
- Accomplishes this by setting 3 flags **(FIN, PSH,** and **URG)**

MiTM: Man-in-the-Middle: Place something (such as SW/rogue router) bet server/usr: Neither server admin/usr aware

- Intercepts data/sends info to server as if nothing wrong
- Server responds to SW thinking it's comm w/legit usr
- Attacking SW continues sending info on to server etc…

TCP/IP hijacking: Involves attacker gaining access to host in network/logically dc'ing it

- Attacker then inserts another machine w/same IP
- Happens quickly: Gives attacker access to session/all info on original sys
- Server won't know: Will respond as if client trusted

Replay Attacks: When info captured over network: Kind of access/mod attack

- In distributed env: Logon/passwd info sent bet client/auth sys
- Attacker can capture info/replay later
- Can also occur w/sec certs from sys such as Kerberos:
- Attacker resubmits cert, hoping to be validated by auth sys/circumvent time sensitivity
- If successful: Attacker will have all rights/privs from original cert

Smurf attack: Spoofing target machine's IP/broadcasting that machine's routers so routers think target is sending out broadcast

- Causes every machine on network to respond to attack
- Overloads target system
- Eliminating smurf attacks: **Prohibit ICMP traffic through router**
- If router blocks ICMP traffic, smurf attacks from an external attacker aren't possible

Password Attacks: Occur when acct attacked repeatedly: Passwd crackers

Brute-Force	Attempt to guess passwds until successful guess
Dictionary	Uses dictionary of common words to find usr's passwd
Hybrid	Uses combo of dictionary/brute force

	Example: You know likelihood of employee passwds: Can seed values
Birthday	Based on probability: If 25 ppl in room: Probability 2 ppl have same bday • Increases as additional people enter room • Probability doesn't mean it WILL occur, only that it's more likely
Rainbow Table	Focuses on ID'ing stored value • Values: Hashed phrases/words: Compares them to values found on table • Can reduce amt of time needed to crack a passwd • Salt: Greatly reduces each which rainbow tables can be used

Privilege Escalation: Involves usr gaining more privs than should have

- Often associated w/bugs left in SW | Devs occasionally leave backdoors

Malicious Insider Threats: Most dangerous threats to any network is an insider intent on doing harm

White box testing: Testing systems from premise of knowing something about network

Full Disclosure testing: Trying to find info about weaknesses/source code/routing/etc…

Transitive Access: Party A trusts Party B || Party B trusts Party C || A may trust C

Server 2008**: Transitive trusts:** Type of relationship that can exist bet domains

Client-Side Attacks: Targets vulns in client apps that interact w/malicious server

- Usr accesses trusted site and unknowingly dl's rogue code
- Rogue code: Allows attacker to install/exe programs on affected machine remotely
- Newly installed programs run w/priv lvl of individual who accessed server
- If usr had elevated privs: Malware runs at that lvl
- Often data accessed along way pushed out across net using HTTPS to encrypt/make less likely detected

Typo squatting/URL hijacking: Registering domains similar in name to those of known entity based on typo

- Example: Sybex.com//Sybecks.com
- Hope usr will misspell word: Go to their site: Could dl Trojans/worms/virus

Watering Hole Attack: When attacker takes checks a target by seeing looking/poisoning/waiting affiliates for results

- May attack site through 3rd party company that the main target usrs

Types of Application Attacks:

Cross-site scripting: Using client-side script lang: Attacker can trick usr who visits site into having code execute locally

- Attacker finds some place on site where they can interact w/others
- Types in some script (JS) || Filter inputs

XSRF: Cross-Site Request Forgery: AKA session riding/1-click attack

- Unauthorized cmds coming from trusted usr/website
- Often done w/out usr's knowledge: Employs some type of social networking to pull off
- Disable running of scripts (and browser profiles)

SQLi: SQL Injection: AKA Insertion: Structured Query Lang: Used for

comm w/online db's

- Attacker manipulates DB code to take advantage of weakness in it

Various types of exploits use SQLi: Most common fall into these categories:

- Escape chars not filtered correctly
- Type handling not properly done
- Conditional errors
- Time delays

Always filter input: Site code should be checked to see if certain chars are in text fields and to reject that input

LDAP Injection: Attack exploits weaknesses in LDAP (Lightweight Directory Access Protocol) implementation

- Can occur when usr input not properly filtered: Result: Executed cmds/mod content/results returned to unauth queries
- Filter usr input/validation scheme to make certain queries don't contain exploits

XML Injection: When usr enters values that query XML (Xpath) w/values that take advantage of exploits

- Xpath: Works in similar manner to SQL: Doesn't have same lvls of access control | Weaknesses w/in can return entire docs
- Filter usr input/sanitize to ensure data doesn't cause Xpath to return more than it should

Directory Traversal/Cmd Injection: When attacker able to gain access to restricted dirs (such as root) through HTTP

- If attacker can gain access to root dir of sys (limited to admin usrs: Can gain access to everything on sys
- If attacker can get out of this dir and get to C:\Windows, possibility for inflicting harm increased

Simplest ways to perform directory traversal: Using cmd injection attack that carries out action

- Example: Exploiting a weak IIS implementation by calling up a web page along w/the param cmd.exe?**/c+dir+c:**
- Would call the cmd shell/execute a dir listing of the root drive (C:\)
- W/Unicode support, entries such as **%c%^1c and %c^0%af** can be translated into / and \ respectively
- Most vuln scanners will check for weaknesses w/dir traversal/cmd injection

Buffer Overflows: When an app receives more data than programmed to accept

- Can cause an app to terminate/write data beyond end of allocated space
- Termination may leave the sys sending data w/temp access to priv lvls in the sys
- While overwriting can cause impt data to be lost
- Usually the result of programming error in dev of SW

Integer Overflow: Like buffer overflow: Involves putting too much info into a small of a space (set aside for numbers)

Cookies: Text files that a browser maintains on the usr's HDD in order to

provide a persistent, customized experience each visit
Evercookie: A breed of cookie that writes data to multiple locations to make it next to impossible to rem completely
Locally Shared Objects/Flash Cookies: AKA Flash cookie: Data stored on a usr's computer by Adobe Flash

- Can represent a sec/privacy threat

Malicious add-ons: Add-ons w/the potential and intent to harm a system
ActiveX: A tech that was implemented by MS to customize controls/icons/features which increases the usability of web-enabled sys.

- Runs on client
- Uses a method called Authenticode for security

Authenticode: Type of cert tech that allows ActiveX components to be validated by a server

- Components are dl'd to the client HDD, potentially allowing sec breaches
- Browsers should require confirmation to accept ActiveX controls

Session hijacking: When item used to validate usr's session (cookie) stolen/used by another to establish session

- Host thinks it's still comm w/1st party || MiTM/Sidejacking use session hijacking || Firesheep

Header manipulation: Attack uses other methods (hijacking/xsrf/etc..) to change values in HTTP headers/falsify access

- When used w/XSRF, the attacker can even change a usr's cookie
- IE 8+ include *InPrivate Filtering*: To help prevent this (config browser not to share info that can be captured/manipulated)

Arbitrary Code/Remote Code Execution: Possible for programmer to create means by which program they write can remotely accept cmds/execute the cmds

- Can be unrelated to the action program accepting them
- Can run on the host machine w/in a shell, cmd interpreter, etc…
- Host program can be running w/elevated privs/capable of doing more harm than what usr might be limited to

Tools for Finding Threats
Active response: Any that allows SW to manage resource in the network if an incident occurs
Passive response: Notification/reporting of attacks or suspicious activities
Protocol analyzer/packet sniffer: Interchangeable: Tools used in process of monitoring data transmitted across a network
Vulnerability Scanner: Examples: Retina, Nessus, SAINT, OpenVAS

Passively Testing Sec Controls	Vuln scanner can test sec controls w/out doing any actual harm • Only looks for openings there/reports them back to you • Passive
Interpreting Results	Most vuln scanning programs, interpret results of findings/deliver report can be shared w/mgmt
ID'ing Vuln	Just knowing a port is open means little unless you associate it w/vuln
ID'ing Lack of Sec Controls	Looking for weaknesses in sec controls: Impt as ID'ing areas where no controls in place • Want to know what's missing altogether, not just what's weak

ID'ing Common Misconfigs	

Honeypot: A computer that has been designated as a target for attacks

- When larger referred to as **honeynets**

Banner Grabbing: Looks at banner/header info msgs sent w/data to find out about sys

- Banners often ID host/OS/other info that can be useful
- Banners can be snagged w/Telnet/tools like netcat/Nmap

Risk Calculations and Assessment Types

Risk	Actual danger under consideration? Likelihood of attack being successful
Threat	Likely dangers associated w/risk? Means/source of potential attack? Weighed against the likelihood of attack
Vuln	Where is sys weak? ID flaws/holes/areas of exposure/perils

Baseline Reporting: Makes sure things operating status quo: Change detection used to alert admins/mods when changes made

- Often combined w/gap analysis to measure controls at particular company against industry standards

Code Review: Looks at all custom written code for holes: Often conducted as part of gray box testing

- **Manual assessment:** Reading code
- **Automated assessment:** Using tools to scan code

Attack surface: Area of app that avail to usrs: Those who are/aren't auth

- Can include services/protocols/ints/code
- Smaller attack surface: Less visible app is to attack
- Larger attack surface: More likely to become target

ASR: Attack Surface Reduction: Min possibility of exploitation by reducing amt of code

Architectural approach: Using control framework to focus on foundational infrastructure

- Popular w/sec regulatory standards/compliance standards (ISO)

Design Review: Assessment examines ports/protocols used/rules/segmentation/access control

- More granular than architectural assessments | Should be done more often

From <https://www.piratemoo.net/moosings/hacking/malwarevulnsthreats/>