Table of Contents

CompTIA Security+ Certification Exam Objectives

EXAM NUMBER: SY0-601

Revised 5/9/2022

About the Exam

Candidates are encouraged to use this document to help prepare for the CompTIA Security+ (SY0-601) certification exam. The CompTIA Security+ certification exam will verify the successful candidate has the knowledge and skills required to:

- **Assess the security posture of an enterprise environment and recommend and implement appropriate security solutions**
- **Monitor and secure hybrid environments, including cloud, mobile, and IoT**
- **Operate with an awareness of applicable laws and policies, including principles of governance, risk, and compliance**
- **Identify, analyze, and respond to security events and incidents**

This is equivalent to two years of hands-on experience working in a security/systems administrator job role.

These content examples are meant to clarify the test objectives and should not be construed as a comprehensive listing of all the content of this examination.

EXAM DEVELOPMENT

CompTIA exams result from subject matter expert workshops and industry-wide survey results regarding the skills and knowledge required of an IT professional.

CompTIA AUTHORIZED MATERIALS USE POLICY

CompTIA Certifications, LLC is not affiliated with and does not authorize, endorse or condone utilizing any content provided by unauthorized third-party training sites (aka "brain dumps"). Individuals who utilize such materials in preparation for any CompTIA examination will have their certifications revoked and be suspended from future testing in accordance with the CompTIA Candidate Agreement. In an effort to more clearly communicate CompTIA's exam policies on use of unauthorized study materials, CompTIA directs all certification candidates to the **CompTIA Certification Exam Policies**. Please review all CompTIA policies before beginning the study process for any CompTIA exam. Candidates will be required to abide by the **CompTIA Candidate Agreement**. If a candidate has a question as to whether study materials are considered unauthorized (aka "brain dumps"), he/she should contact CompTIA at examsecurity@comptia.org to confirm.

PLEASE NOTE

The lists of examples provided in bulleted format are not exhaustive lists. Other examples of technologies, processes, or tasks pertaining to each objective may also be included on the exam although not listed or covered in this objectives document. CompTIA is constantly reviewing the content of our exams and updating test questions to be sure our exams are current, and the security of the questions is protected. When necessary, we will publish updated exams based on testing exam objectives. Please know that all related exam preparation materials will still be valid.

TEST DETAILS

Required exam	SY0-601
Number of questions	Maximum of 90
Types of questions	Multiple-choice and performance-based
Length of test	90 minutes
Recommended experience	• At least 2 years of work experience in IT systems administration with a focus on security • Hands-on technical information security experience • Broad knowledge of security concepts
Passing score	750 (on a scale of 100–900)

EXAM OBJECTIVES (DOMAINS)

The table below lists the domains measured by this examination and the extent to which they are represented:

DOMAIN	PERCENTAGE OF EXAMINATION
1.0 Attacks, Threats, and Vulnerabilities	24%
2.0 Architecture and Design	21%
3.0 Implementation	25%
4.0 Operations and Incident Response	16%
5.0 Governance, Risk, and Compliance	14%
Total	**100%**

1.0 Threats, Attacks, and Vulnerabilities

1.1 Compare and contrast different types of social engineering techniques.

- **Phishing**
- **Smishing**
- **Vishing**
- **Spam**
- **Spam over instant messaging (SPIM)**
- **Spear phishing**
- **Dumpster diving**
- **Shoulder surfing**
- **Pharming**
- **Tailgating**
- **Eliciting information**
- **Whaling**
- **Prepending**
- **Identity fraud**
- **Invoice scams**
- **Credential harvesting**
- **Reconnaissance**
- **Hoax**
- **Impersonation**
- **Watering hole attack**
- **Typosquatting**
- **Pretexting**
- **Influence campaigns**
  - Hybrid warfare
  - Social media
- **Principles (reasons for effectiveness)**
  - Authority
  - Intimidation
  - Consensus
  - Scarcity
  - Familiarity
  - Trust
  - Urgency

1.2 Given a scenario, analyze potential indicators to determine the type of attack.

- **Malware**
  - Ransomware
  - Trojans
  - Worms
  - Potentially unwanted programs (PUPs)
  - Fileless virus
  - Command and control
  - Bots
  - Cryptomalware
  - Logic bombs
  - Spyware
  - Keyloggers
  - Remote access Trojan (RAT)
  - Rootkit
  - Backdoor
- **Password attacks**
  - Spraying
  - Dictionary
  - Brute force
    - Offline
    - Online
  - Rainbow table
  - Plaintext/unencrypted
- **Physical attacks**
  - Malicious Universal Serial Bus (USB) cable
  - Malicious flash drive
  - Card cloning
  - Skimming
- **Adversarial artificial intelligence (AI)**
  - Tainted training data for machine learning (ML)
  - Security of machine learning algorithms
- **Supply-chain attacks**
- **Cloud-based vs. on-premises attacks**
- **Cryptographic attacks**
  - Birthday
  - Collision
  - Downgrade

1.3 Given a scenario, analyze potential indicators associated with application attacks.

- **Privilege escalation**
- **Cross-site scripting**
- **Injections**
  - Structured query language (SQL)
  - Dynamic-link library (DLL)
  - Lightweight Directory Access Protocol (LDAP)
  - Extensible Markup Language (XML)
- **Pointer/object dereference**
- **Directory traversal**
- **Buffer overflows**
- **Race conditions**
  - Time of check/time of use
- **Error handling**
- **Improper input handling**
- **Replay attack**
  - Session replays
- **Integer overflow**
- **Request forgeries**
  - Server-side
  - Cross-site
- **Application programming interface (API) attacks**
- **Resource exhaustion**
- **Memory leak**
- **Secure Sockets Layer (SSL) stripping**
- **Driver manipulation**
  - Shimming
  - Refactoring
- **Pass the hash**

1.4 Given a scenario, analyze potential indicators associated with network attacks.

- **Wireless**
  - Evil twin
  - Rogue access point
  - Bluesnarfing
  - Bluejacking
  - Disassociation
  - Jamming
  - Radio frequency identification (RFID)
  - Near-field communication (NFC)
  - Initialization vector (IV)
- **On-path attack (previously known as man-in-the-middle attack/ man-in-the-browser attack)**
- **Layer 2 attacks**
  - Address Resolution Protocol (ARP) poisoning
  - Media access control (MAC) flooding
  - MAC cloning
- **Domain name system (DNS)**
  - Domain hijacking
  - DNS poisoning
  - Uniform Resource Locator (URL) redirection
  - Domain reputation
- **Distributed denial-of-service (DDoS)**
  - Network
  - Application
  - Operational technology (OT)
- **Malicious code or script execution**
  - PowerShell
  - Python
  - Bash
  - Macros
  - Visual Basic for Applications (VBA)

1.5 Explain different threat actors, vectors, and intelligence sources.

- **Actors and threats**
  - Advanced persistent threat (APT)
  - Insider threats
  - State actors
  - Hacktivists
  - Script kiddies
  - Criminal syndicates
  - Hackers
    - Authorized
    - Unauthorized
    - Semi-authorized
  - Shadow IT
  - Competitors
- **Attributes of actors**
  - Internal/external
  - Level of sophistication/capability
  - Resources/funding
  - Intent/motivation
- **Vectors**
  - Direct access
  - Wireless
  - Email
  - Supply chain
  - Social media
  - Removable media
  - Cloud
- **Threat intelligence sources**
  - Open-source intelligence (OSINT)
  - Closed/proprietary
  - Vulnerability databases
  - Public/private information-sharing centers
  - Dark web
  - Indicators of compromise
  - Automated Indicator Sharing (AIS)
    - Structured Threat Information eXpression (STIX)/Trusted Automated eXchange of Intelligence Information (TAXII)
  - Predictive analysis
  - Threat maps
  - File/code repositories
- **Research sources**
  - Vendor websites
  - Vulnerability feeds
  - Conferences
  - Academic journals
  - Request for comments (RFC)
  - Local industry groups
  - Social media
  - Threat feeds
  - Adversary tactics, techniques, and procedures (TTP)

1.6 Explain the security concerns associated with various types of vulnerabilities.

- **Cloud-based vs. on-premises vulnerabilities**
- **Zero-day**
- **Weak configurations**
  - Open permissions
  - Unsecure root accounts
  - Errors
  - Weak encryption
  - Unsecure protocols
  - Default settings
  - Open ports and services
- **Third-party risks**
  - Vendor management
    - System integration
    - Lack of vendor support
  - Supply chain
  - Outsourced code development
  - Data storage
- **Improper or weak patch management**
  - Firmware
  - Operating system (OS)
  - Applications
- **Legacy platforms**
- **Impacts**
  - Data loss
  - Data breaches
  - Data exfiltration
  - Identity theft
  - Financial
  - Reputation
  - Availability loss

1.7 Summarize the techniques used in security assessments.

- **Threat hunting**
  - Intelligence fusion
  - Threat feeds
  - Advisories and bulletins
  - Maneuver
- **Vulnerability scans**
  - False positives
  - False negatives
  - Log reviews
  - Credentialed vs. non-credentialed
  - Intrusive vs. non-intrusive
  - Application
  - Web application
  - Network
  - Common Vulnerabilities and Exposures (CVE)/Common Vulnerability Scoring System (CVSS)
  - Configuration review
- **Syslog/Security information and event management (SIEM)**
  - Review reports
  - Packet capture
  - Data inputs
  - User behavior analysis
  - Sentiment analysis
  - Security monitoring
  - Log aggregation
  - Log collectors
- **Security orchestration, automation, and response (SOAR)**

1.8 Explain the techniques used in penetration testing.

- **Penetration testing**
  - Known environment
  - Unknown environment
  - Partially known environment
  - Rules of engagement
  - Lateral movement
  - Privilege escalation
  - Persistence
  - Cleanup
  - Bug bounty
  - Pivoting
- **Passive and active reconnaissance**
  - Drones
  - War flying
  - War driving
  - Footprinting
  - OSINT
- **Exercise types**
  - Red-team
  - Blue-team
  - White-team
  - Purple-team

2.0 Architecture and Design

2.1 Explain the importance of security concepts in an enterprise environment.

- **Configuration management**
  - Diagrams
  - Baseline configuration
  - Standard naming conventions
  - Internet protocol (IP) schema
- **Data sovereignty**
- **Data protection**
  - Data loss prevention (DLP)
  - Masking
  - Encryption
  - At rest
  - In transit/motion
  - In processing
  - Tokenization
  - Rights management
- **Geographical considerations**
- **Response and recovery controls**
- **Secure Sockets Layer (SSL)/Transport Layer Security (TLS) inspection**
- **Hashing**
- **API considerations**
- **Site resiliency**
  - Hot site
  - Cold site
  - Warm site
- **Deception and disruption**
  - Honeypots
  - Honeyfiles
  - Honeynets
  - Fake telemetry
  - DNS sinkhole

2.2 Summarize virtualization and cloud computing concepts.

- **Cloud models**
  - Infrastructure as a service (IaaS)
  - Platform as a service (PaaS)
  - Software as a service (SaaS)
  - Anything as a service (XaaS)
  - Public
  - Community
  - Private
  - Hybrid
- **Cloud service providers**
- **Managed service provider (MSP)/ managed security service provider (MSSP)**
- **On-premises vs. off-premises**
- **Fog computing**
- **Edge computing**
- **Thin client**
- **Containers**
- **Microservices/API**
- **Infrastructure as code**
  - Software-defined networking (SDN)
  - Software-defined visibility (SDV)
- **Serverless architecture**
- **Services integration**
- **Resource policies**
- **Transit gateway**
- **Virtualization**
  - Virtual machine (VM) sprawl avoidance
  - VM escape protection

2.3 Summarize secure application development, deployment, and automation concepts.

- **Environment**
  - Development
  - Test
  - Staging
  - Production
  - Quality assurance (QA)
- **Provisioning and deprovisioning**
- **Integrity measurement**
- **Secure coding techniques**
  - Normalization
  - Stored procedures
  - Obfuscation/camouflage
  - Code reuse/dead code
  - Server-side vs. client-side execution and validation
  - Memory management
  - Use of third-party libraries and software development kits (SDKs)
  - Data exposure
- **Open Web Application Security Project (OWASP)**
- **Software diversity**
  - Compiler
  - Binary
- **Automation/scripting**
  - Automated courses of action
  - Continuous monitoring
  - Continuous validation
  - Continuous integration
  - Continuous delivery
  - Continuous deployment
- **Elasticity**
- **Scalability**
- **Version control**

2.4 Summarize authentication and authorization design concepts.

- **Authentication methods**
  - Directory services
  - Federation
  - Attestation
  - Technologies
    - Time-based one-time password (TOTP)
    - HMAC-based one-time password (HOTP)
    - Short message service (SMS)
    - Token key
    - Static codes
    - Authentication applications
    - Push notifications
    - Phone call
  - Smart card authentication
- **Biometrics**
  - Fingerprint
  - Retina
  - Iris
  - Facial
  - Voice
  - Vein
  - Gait analysis
  - Efficacy rates
  - False acceptance
  - False rejection
  - Crossover error rate
- **Multifactor authentication (MFA) factors and attributes**
  - Factors
    - Something you know
    - Something you have
    - Something you are
  - Attributes
    - Somewhere you are
    - Something you can do
    - Something you exhibit
    - Someone you know
- **Authentication, authorization, and accounting (AAA)**
- **Cloud vs. on-premises requirements**

2.5 Given a scenario, implement cybersecurity resilience.

- **Redundancy**
  - Geographic dispersal
  - Disk
    - Redundant array of inexpensive disks (RAID) levels
    - Multipath
  - Network
    - Load balancers
    - Network interface card (NIC) teaming
  - Power
    - Uninterruptible power supply (UPS)
    - Generator
    - Dual supply
    - Managed power distribution units (PDUs)
- **Replication**
  - Storage area network
  - VM
- **On-premises vs. cloud**
- **Backup types**
  - Full
  - Incremental
  - Snapshot
  - Differential
  - Tape
  - Disk
  - Copy
  - Network-attached storage (NAS)
  - Storage area network
  - Cloud
  - Image
  - Online vs. offline
  - Offsite storage
    - Distance considerations
- **Non-persistence**
  - Revert to known state
  - Last known-good configuration
  - Live boot media
- **High availability**
  - Scalability
- **Restoration order**
- **Diversity**
  - Technologies
  - Vendors
  - Crypto
  - Controls

2.6 Explain the security implications of embedded and specialized systems.

- **Embedded systems**
  - Raspberry Pi
  - Field-programmable gate array (FPGA)
  - Arduino
- **Supervisory control and data acquisition (SCADA)/industrial control system (ICS)**
  - Facilities
  - Industrial
  - Manufacturing
  - Energy
  - Logistics
- **Internet of Things (IoT)**
  - Sensors
  - Smart devices
  - Wearables
  - Facility automation
  - Weak defaults
- **Specialized**
  - Medical systems
  - Vehicles
  - Aircraft
  - Smart meters
- **Voice over IP (VoIP)**
- **Heating, ventilation, air conditioning (HVAC)**
- **Drones**
- **Multifunction printer (MFP)**
- **Real-time operating system (RTOS)**
- **Surveillance systems**
- **System on chip (SoC)**
- **Communication considerations**
  - 5G
  - Narrow-band
  - Baseband radio
  - Subscriber identity module (SIM) cards
  - Zigbee
- **Constraints**
  - Power
  - Compute
  - Network
  - Crypto
  - Inability to patch
  - Authentication
  - Range
  - Cost
  - Implied trust

2.7 Explain the importance of physical security controls.

- **Bollards/barricades**
- **Access control vestibules**
- **Badges**
- **Alarms**
- **Signage**
- **Cameras**
  - Motion recognition
  - Object detection
- **Closed-circuit television (CCTV)**
- **Industrial camouflage**
- **Personnel**
  - Guards
  - Robot sentries
  - Reception
  - Two-person integrity/control
- **Locks**
  - Biometrics
  - Electronic
  - Physical
  - Cable locks
- **USB data blocker**
- **Lighting**
- **Fencing**
- **Fire suppression**
- **Sensors**
  - Motion detection
  - Noise detection
  - Proximity reader
  - Moisture detection
  - Cards
  - Temperature
- **Drones**
- **Visitor logs**
- **Faraday cages**
- **Air gap**
- **Screened subnet (previously known as demilitarized zone)**
- **Protected cable distribution**
- **Secure areas**
  - Air gap
  - Vault
  - Safe
  - Hot aisle
  - Cold aisle
- **Secure data destruction**
  - Burning
  - Shredding
  - Pulping
  - Pulverizing
  - Degaussing
  - Third-party solutions

2.8 Summarize the basics of cryptographic concepts.

- **Digital signatures**
- **Key length**
- **Key stretching**
- **Salting**
- **Hashing**
- **Key exchange**
- **Elliptic-curve cryptography**
- **Perfect forward secrecy**
- **Quantum**
  - Communications
  - Computing
- **Post-quantum**
- **Ephemeral**
- **Modes of operation**
  - Authenticated
  - Unauthenticated
  - Counter
- **Blockchain**
  - Public ledgers
- **Cipher suites**
  - Stream
  - Block
- **Symmetric vs. asymmetric**
- **Lightweight cryptography**
- **Steganography**
  - Audio
  - Video
  - Image
- **Homomorphic encryption**
- **Common use cases**
  - Low power devices
  - Low latency
  - High resiliency
  - Supporting confidentiality
  - Supporting integrity
  - Supporting obfuscation
  - Supporting authentication
  - Supporting non-repudiation
- **Limitations**
  - Speed
  - Size
  - Weak keys
  - Time
  - Longevity
  - Predictability
  - Reuse
  - Entropy
  - Computational overheads
  - Resource vs. security constraints

3.0 Implementation

3.1 Given a scenario, implement secure protocols.

- **Protocols**
  - Domain Name System Security Extensions (DNSSEC)
  - SSH
  - Secure/Multipurpose Internet Mail Extensions (S/MIME)
  - Secure Real-time Transport Protocol (SRTP)
  - Lightweight Directory Access Protocol Over SSL (LDAPS)
  - File Transfer Protocol, Secure (FTPS)
  - SSH File Transfer Protocol (SFTP)
  - Simple Network Management Protocol, version 3 (SNMPv3)
  - Hypertext transfer protocol over SSL/TLS (HTTPS)
  - IPSec
    - Authentication header (AH)/ Encapsulating Security Payloads (ESP)
    - Tunnel/transport
  - Post Office Protocol (POP)/ Internet Message Access Protocol (IMAP)
- **Use cases**
  - Voice and video
  - Time synchronization
  - Email and web
  - File transfer
  - Directory services
  - Remote access
  - Domain name resolution
  - Routing and switching
  - Network address allocation
  - Subscription services

3.2 Given a scenario, implement host or application security solutions.

- **Endpoint protection**
  - Antivirus
  - Anti-malware
  - Endpoint detection and response (EDR)
  - DLP
  - Next-generation firewall (NGFW)
  - Host-based intrusion prevention system (HIPS)
  - Host-based intrusion detection system (HIDS)
  - Host-based firewall
- **Boot integrity**
  - Boot security/Unified Extensible Firmware Interface (UEFI)
  - Measured boot
  - Boot attestation
- **Database**
  - Tokenization
  - Salting
  - Hashing
- **Application security**
  - Input validations
  - Secure cookies
  - Hypertext Transfer Protocol (HTTP) headers
  - Code signing
  - Allow list
  - Block list/deny list
  - Secure coding practices
  - Static code analysis
    - Manual code review
  - Dynamic code analysis
  - Fuzzing
- **Hardening**
  - Open ports and services
  - Registry
  - Disk encryption
  - OS
  - Patch management
    - Third-party updates
    - Auto-update
- **Self-encrypting drive (SED)/ full-disk encryption (FDE)**
  - Opal
- **Hardware root of trust**
- **Trusted Platform Module (TPM)**
- **Sandboxing**

3.3 Given a scenario, implement secure network designs.

- **Load balancing**
  - Active/active
  - Active/passive
  - Scheduling
  - Virtual IP
  - Persistence
- **Network segmentation**
  - Virtual local area network (VLAN)
  - Screened subnet (previously known as demilitarized zone)
  - East-west traffic
  - Extranet
  - Intranet
  - Zero Trust
- **Virtual private network (VPN)**
  - Always-on
  - Split tunnel vs. full tunnel
  - Remote access vs. site-to-site
  - IPSec
  - SSL/TLS
  - HTML5
  - Layer 2 tunneling protocol (L2TP)
- **DNS**
- **Network access control (NAC)**
  - Agent and agentless
- **Out-of-band management**
- **Port security**
  - Broadcast storm prevention
  - Bridge Protocol Data Unit (BPDU) guard
  - Loop prevention
  - Dynamic Host Configuration Protocol (DHCP) snooping
  - Media access control (MAC) filtering
- **Network appliances**
  - Jump servers
  - Proxy servers
    - Forward
    - Reverse
  - Network-based intrusion detection system (NIDS)/network-based intrusion prevention system (NIPS)
    - Signature-based
    - Heuristic/behavior
    - Anomaly
    - Inline vs. passive
  - HSM
  - Sensors
  - Collectors
  - Aggregators
  - Firewalls
    - Web application firewall (WAF)
    - NGFW
    - Stateful
    - Stateless
    - Unified threat management (UTM)
    - Network address translation (NAT) gateway
    - Content/URL filter
    - Open-source vs. proprietary
    - Hardware vs. software
    - Appliance vs. host-based vs. virtual
- **Access control list (ACL)**
- **Route security**
- **Quality of service (QoS)**
- **Implications of IPv6**
- **Port spanning/port mirroring**
  - Port taps
- **Monitoring services**
- **File integrity monitors**

3.4 Given a scenario, install and configure wireless security settings.

- **Cryptographic protocols**
  - WiFi Protected Access 2 (WPA2)
  - WiFi Protected Access 3 (WPA3)
  - Counter-mode/CBC-MAC Protocol (CCMP)
  - Simultaneous Authentication of Equals (SAE)
- **Authentication protocols**
  - Extensible Authentication Protocol (EAP)
  - Protected Extensible Authentication Protocol (PEAP)
  - EAP-FAST
  - EAP-TLS
  - EAP-TTLS
  - IEEE 802.1X
  - Remote Authentication Dial-in User Service (RADIUS) Federation
- **Methods**
  - Pre-shared key (PSK) vs. Enterprise vs. Open
  - WiFi Protected Setup (WPS)
  - Captive portals
- **Installation considerations**
  - Site surveys
  - Heat maps
  - WiFi analyzers
  - Channel overlaps
  - Wireless access point (WAP) placement
  - Controller and access point security

3.5 Given a scenario, implement secure mobile solutions.

- **Connection methods and receivers**
  - Cellular
  - WiFi
  - Bluetooth
  - NFC
  - Infrared
  - USB
  - Point-to-point
  - Point-to-multipoint
  - Global Positioning System (GPS)
  - RFID
- **Mobile device management (MDM)**
  - Application management
  - Content management
  - Remote wipe
  - Geofencing
  - Geolocation
  - Screen locks
  - Push notifications
  - Passwords and PINs
  - Biometrics
  - Context-aware authentication
  - Containerization
  - Storage segmentation
  - Full device encryption
- **Mobile devices**
  - MicroSD hardware security module (HSM)
  - MDM/Unified Endpoint Management (UEM)
  - Mobile application management (MAM)
  - SEAndroid
- **Enforcement and monitoring of:**
  - Third-party application stores
  - Rooting/jailbreaking
  - Sideloading
  - Custom firmware
  - Carrier unlocking
  - Firmware over-the-air (OTA) updates
  - Camera use
  - SMS/Multimedia Messaging Service (MMS)/Rich Communication Services (RCS)
  - External media
  - USB On-The-Go (USB OTG)
  - Recording microphone
  - GPS tagging
  - WiFi direct/ad hoc
  - Tethering
  - Hotspot
  - Payment methods
- **Deployment models**
  - Bring your own device (BYOD)
  - Corporate-owned personally enabled (COPE)
  - Choose your own device (CYOD)
  - Corporate-owned
  - Virtual desktop infrastructure (VDI)

3.6 Given a scenario, apply cybersecurity solutions to the cloud.

- **Cloud security controls**
  - High availability across zones
  - Resource policies
  - Secrets management
  - Integration and auditing
  - Storage
    - Permissions
    - Encryption
    - Replication
    - High availability
  - Network
    - Virtual networks
    - Public and private subnets
    - Segmentation
    - API inspection and integration
  - Compute
    - Security groups
    - Dynamic resource allocation
    - Instance awareness
    - Virtual private cloud (VPC) endpoint
    - Container security
- **Solutions**
  - CASB
  - Application security
  - Next-generation secure web gateway (SWG)
  - Firewall considerations in a cloud environment
    - Cost
    - Need for segmentation
    - Open Systems Interconnection (OSI) layers
- **Cloud native controls vs. third-party solutions**

3.7 Given a scenario, implement identity and account management controls.

- **Identity**
  - Identity provider (IdP)
  - Attributes
  - Certificates
  - Tokens
  - SSH keys
  - Smart cards
- **Account types**
  - User account
  - Shared and generic accounts/credentials
  - Guest accounts
  - Service accounts
- **Account policies**
  - Password complexity
  - Password history
  - Password reuse
  - Network location
  - Geofencing
  - Geotagging
  - Geolocation
  - Time-based logins
  - Access policies
  - Account permissions
  - Account audits
  - Impossible travel time/risky login
  - Lockout
  - Disablement

3.8 Given a scenario, implement authentication and authorization solutions.

- **Authentication management**
  - Password keys
  - Password vaults
  - TPM
  - HSM
  - Knowledge-based authentication
- **Authentication/authorization**
  - EAP
  - Challenge-Handshake Authentication Protocol (CHAP)
  - Password Authentication Protocol (PAP)
  - 802.1X
  - RADIUS
  - Single sign-on (SSO)
  - Security Assertion Markup Language (SAML)
  - Terminal Access Controller Access Control System Plus (TACACS+)
  - OAuth
  - OpenID
  - Kerberos
- **Access control schemes**
  - Attribute-based access control (ABAC)
  - Role-based access control
  - Rule-based access control
  - MAC
  - Discretionary access control (DAC)
  - Conditional access
  - Privileged access management
  - Filesystem permissions

3.9 Given a scenario, implement public key infrastructure.

- **Public key infrastructure (PKI)**
  - Key management
  - Certificate authority (CA)
  - Intermediate CA
  - Registration authority (RA)
  - Certificate revocation list (CRL)
  - Certificate attributes
  - Online Certificate Status Protocol (OCSP)
  - Certificate signing request (CSR)
  - CN
  - Subject alternative name
  - Expiration
- **Types of certificates**
  - Wildcard
  - Subject alternative name
  - Code signing
  - Self-signed
  - Machine/computer
  - Email
  - User
  - Root
  - Domain validation
  - Extended validation
- **Certificate formats**
  - Distinguished encoding rules (DER)
  - Privacy enhanced mail (PEM)
  - Personal information exchange (PFX)
  - .cer
  - P12
  - P7B
- **Concepts**
  - Online vs. offline CA
  - Stapling
  - Pinning
  - Trust model
  - Key escrow
  - Certificate chaining

4.0 Operations and Incident Response

4.1 Given a scenario, use the appropriate tool to assess organizational security.

- **Network reconnaissance and discovery**
  - tracert/traceroute
  - nslookup/dig
  - ipconfig/ifconfig
  - nmap
  - ping/pathping
  - hping
  - netstat
  - netcat
  - IP scanners
  - arp
  - route
  - curl
  - theHarvester
  - sn1per
  - scanless
  - dnsenum
  - Nessus
  - Cuckoo
- **File manipulation**
  - head
  - tail
  - cat
  - grep
  - chmod
  - logger
- **Shell and script environments**
  - SSH
  - PowerShell
  - Python
  - OpenSSL
- **Packet capture and replay**
  - Tcpreplay
  - Tcpdump
  - Wireshark
- **Forensics**
  - dd
  - Memdump
  - WinHex
  - FTK imager
  - Autopsy
- **Exploitation frameworks**
- **Password crackers**
- **Data sanitization**

4.2 Summarize the importance of policies, processes, and procedures for incident response.

- **Incident response plans**
- **Incident response process**
  - Preparation
  - Identification
  - Containment
  - Eradication
  - Recovery
  - Lessons learned
- **Exercises**
  - Tabletop
  - Walkthroughs
  - Simulations
- **Attack frameworks**
  - MITRE ATT&CK
  - The Diamond Model of Intrusion Analysis
  - Cyber Kill Chain
- **Stakeholder management**
- **Communication plan**
- **Disaster recovery plan**
- **Business continuity plan**
- **Continuity of operations planning (COOP)**
- **Incident response team**
- **Retention policies**

4.3 Given an incident, utilize appropriate data sources to support an investigation.

- **Vulnerability scan output**
- **SIEM dashboards**
  - Sensor
  - Sensitivity
  - Trends
  - Alerts
  - Correlation
- **Log files**
  - Network
  - System
  - Application
  - Security
  - Web
  - DNS
  - Authentication
  - Dump files
  - VoIP and call managers
  - Session Initiation Protocol (SIP) traffic
- **syslog/rsyslog/syslog-ng**
- **journalctl**
- **NXLog**
- **Bandwidth monitors**
- **Metadata**
  - Email
  - Mobile
  - Web
  - File
- **Netflow/sFlow**
  - Netflow
  - sFlow
  - IPFIX
- **Protocol analyzer output**

4.4 Given an incident, apply mitigation techniques or controls to secure an environment.

- **Reconfigure endpoint security solutions**
  - Application approved list
  - Application blocklist/deny list
  - Quarantine
- **Configuration changes**
  - Firewall rules
  - MDM
  - DLP
  - Content filter/URL filter
  - Update or revoke certificates
- **Isolation**
- **Containment**
- **Segmentation**
- **SOAR**
  - Runbooks
  - Playbooks

4.5 Explain the key aspects of digital forensics.

- **Documentation/evidence**
  - Legal hold
  - Video
  - Admissibility
  - Chain of custody
  - Timelines of sequence of events
    - Time stamps
    - Time offset
  - Tags
  - Reports
  - Event logs
  - Interviews
- **Acquisition**
  - Order of volatility
  - Disk
  - Random-access memory (RAM)
  - Swap/pagefile
  - OS
  - Device
  - Firmware
  - Snapshot
  - Cache
  - Network
  - Artifacts
- **On-premises vs. cloud**
  - Right-to-audit clauses
  - Regulatory/jurisdiction
  - Data breach notification laws
- **Integrity**
  - Hashing
  - Checksums
  - Provenance
- **Preservation**
- **E-discovery**
- **Data recovery**
- **Non-repudiation**
- **Strategic intelligence/ counterintelligence**

5.0 Governance, Risk, and Compliance

5.1 Compare and contrast various types of controls.

- **Category**
  - Managerial
  - Operational
  - Technical
- **Control type**
  - Preventive
  - Detective
  - Corrective
  - Deterrent
  - Compensating
  - Physical

5.2 Explain the importance of applicable regulations, standards, or frameworks that impact organizational security posture.

- **Regulations, standards, and legislation**
  - General Data Protection Regulation (GDPR)
  - National, territory, or state laws
  - Payment Card Industry Data Security Standard (PCI DSS)
- **Key frameworks**
  - Center for Internet Security (CIS)
  - National Institute of Standards and Technology (NIST) Risk Management Framework (RMF)/ Cybersecurity Framework (CSF)
  - International Organization for Standardization (ISO) 27001/27002/27701/31000
  - SSAE SOC 2 Type I/II
  - Cloud security alliance
  - Cloud control matrix
  - Reference architecture
- **Benchmarks /secure configuration guides**
  - Platform/vendor-specific guides
    - Web server
    - OS
    - Application server
    - Network infrastructure devices

5.3 Explain the importance of policies to organizational security.

- **Personnel**
  - Acceptable use policy
  - Job rotation
  - Mandatory vacation
  - Separation of duties
  - Least privilege
  - Clean desk space
  - Background checks
  - Non-disclosure agreement (NDA)
  - Social media analysis
  - Onboarding
  - Offboarding
  - User training
    - Gamification
    - Capture the flag
    - Phishing campaigns
      - Phishing simulations
    - Computer-based training (CBT)
    - Role-based training
- **Diversity of training techniques**
- **Third-party risk management**
  - Vendors
  - Supply chain
  - Business partners
  - Service level agreement (SLA)
  - Memorandum of understanding (MOU)
  - Measurement systems analysis (MSA)
  - Business partnership agreement (BPA)
  - End of life (EOL)
  - End of service life (EOSL)
  - NDA
- **Data**
  - Classification
  - Governance
  - Retention
- **Credential policies**
  - Personnel
  - Third-party
  - Devices
  - Service accounts
  - Administrator/root accounts
- **Organizational policies**
  - Change management
  - Change control
  - Asset management

5.4 Summarize risk management processes and concepts.

- **Risk types**
  - External
  - Internal
  - Legacy systems
  - Multiparty
  - IP theft
  - Software compliance/licensing
- **Risk management strategies**
  - Acceptance
  - Avoidance
  - Transference
    - Cybersecurity insurance
  - Mitigation
- **Risk analysis**
  - Risk register
  - Risk matrix/heat map
  - Risk control assessment
  - Risk control self-assessment
  - Risk awareness
  - Inherent risk
  - Residual risk
  - Control risk
  - Risk appetite
  - Regulations that affect risk posture
  - Risk assessment types
    - Qualitative
    - Quantitative
  - Likelihood of occurrence
  - Impact
  - Asset value
  - Single-loss expectancy (SLE)
  - Annualized loss expectancy (ALE)
  - Annualized rate of occurrence (ARO)
- **Disasters**
  - Environmental
  - Person-made
  - Internal vs. external
- **Business impact analysis**
  - Recovery time objective (RTO)
  - Recovery point objective (RPO)
  - Mean time to repair (MTTR)
  - Mean time between failures (MTBF)
  - Functional recovery plans
  - Single point of failure
  - Disaster recovery plan (DRP)
  - Mission essential functions
  - Identification of critical systems
  - Site risk assessment

5.5 Explain privacy and sensitive data concepts in relation to security.

- **Organizational consequences of privacy and data breaches**
  - Reputation damage
  - Identity theft
  - Fines
  - IP theft
- **Notifications of breaches**
  - Escalation
  - Public notifications and disclosures
- **Data types**
  - Classifications
    - Public
    - Private
    - Sensitive
    - Confidential
    - Critical
    - Proprietary
  - Personally identifiable information (PII)
  - Health information
  - Financial information
  - Government data
  - Customer data
- **Privacy enhancing technologies**
  - Data minimization
  - Data masking
  - Tokenization
  - Anonymization
  - Pseudo-anonymization
- **Roles and responsibilities**
  - Data owners
  - Data controller
  - Data processor
  - Data custodian/steward
  - Data protection officer (DPO)
- **Information life cycle**
- **Impact assessment**
- **Terms of agreement**
- **Privacy notice**

Security+ (SY0-601) Acronym List

The following is a list of acronyms that appear on the CompTIA Security+ exam. Candidates are encouraged to review the complete list and attain a working knowledge of all listed acronyms as part of a comprehensive exam preparation program.

ACRONYM	DEFINITION
3DES	Triple Data Encryption Standard
AAA	Authentication, Authorization, and Accounting
ABAC	Attribute-based Access Control
ACL	Access Control List
AD	Active Directory
AES	Advanced Encryption Standard
AES256	Advanced Encryption Standards 256bit
AH	Authentication Header
AI	Artificial Intelligence
AIS	Automated Indicator Sharing
ALE	Annualized Loss Expectancy
AP	Access Point
API	Application Programming Interface
APT	Advanced Persistent Threat
ARO	Annualized Rate of Occurrence
ARP	Address Resolution Protocol
ASLR	Address Space Layout Randomization
ASP	Active Server Pages
ATT&CK	Adversarial Tactics, Techniques, and Common Knowledge
AUP	Acceptable Use Policy
AV	Antivirus
BASH	Bourne Again Shell
BCP	Business Continuity Planning
BGP	Border Gateway Protocol
BIA	Business Impact Analysis
BIOS	Basic Input/Output System
BPA	Business Partnership Agreement
BPDU	Bridge Protocol Data Unit
BSSID	Basic Service Set Identifier
BYOD	Bring Your Own Device
CA	Certificate Authority
CAPTCHA	Completely Automated Public Turing Test to Tell Computers and Humans Apart
CAR	Corrective Action Report
CASB	Cloud Access Security Broker
CBC	Cipher Block Chaining
CBT	Computer-based Training
CCMP	Counter-Mode/CBC-MAC Protocol
CCTV	Closed-Circuit Television
CERT	Computer Emergency Response Team
CFB	Cipher Feedback
CHAP	Challenge-Handshake Authentication Protocol
CIO	Chief Information Officer
CIRT	Computer Incident Response Team
CIS	Center for Internet Security
CMS	Content Management System
CN	Common Name
COOP	Continuity of Operations Planning
COPE	Corporate-owned Personally Enabled
CP	Contingency Planning
CRC	Cyclic Redundancy Check
CRL	Certificate Revocation List
CSA	Cloud Security Alliance
CSIRT	Computer Security Incident Response Team
CSO	Chief Security Officer
CSP	Cloud Service Provider
CSR	Certificate Signing Request
CSRF	Cross-Site Request Forgery
CSU	Channel Service Unit
CTM	Counter-Mode
CTO	Chief Technology Officer
CVE	Common Vulnerabilities and Exposures
CVSS	Common Vulnerability Scoring System
CYOD	Choose Your Own Device
DAC	Discretionary Access Control
DBA	Database Administrator
DDoS	Distributed Denial-of-Service
DEP	Data Execution Prevention

ACRONYM	DEFINITION
DER	Distinguished Encoding Rules
DES	Data Encryption Standard
DHCP	Dynamic Host Configuration Protocol
DHE	Diffie-Hellman Ephemeral
DKIM	Domain Keys Identified Mail
DLL	Dynamic-link Library
DLP	Data Loss Prevention
DMARC	Domain Message Authentication Reporting and Conformance
DNAT	Destination Network Address Transaction
DNS	Domain Name System
DNSSEC	Domain Name System Security Extensions
DoS	Denial-of-Service
DPO	Data Protection Officer
DRP	Disaster Recovery Plan
DSA	Digital Signature Algorithm
DSL	Digital Subscriber Line
EAP	Extensible Authentication Protocol
ECB	Electronic Code Book
ECC	Elliptic-curve Cryptography
ECDHE	Elliptic-curve Diffie-Hellman Ephemeral
ECDSA	Elliptic-curve Digital Signature Algorithm
EDR	Endpoint Detection and Response
EFS	Encrypted File System
EIP	Extended Instruction Pointer
EOL	End of Life
EOS	End of Service
ERP	Enterprise Resource Planning
ESN	Electronic Serial Number
ESP	Encapsulating Security Payload
ESSID	Extended Service Set Identifier
FACL	File System Access Control List
FDE	Full Disk Encryption
FIM	File Integrity Monitoring
FPGA	Field Programmable Gate Array
FRR	False Rejection Rate
FTP	File Transfer Protocol
FTPS	Secured File Transfer Protocol
GCM	Galois/Counter Mode
GDPR	General Data Protection Regulation
GPG	GNU Privacy Guard
GPO	Group Policy Object
GPS	Global Positioning System
GPU	Graphics Processing Unit
GRE	Generic Routing Encapsulation
HA	High Availability
HDD	Hard Disk Drive
HIDS	Host-based Intrusion Detection System
HIPS	Host-based Intrusion Prevention System
HMAC	Hash-based Message Authentication Code
HOTP	HMAC-based One-time Password

ACRONYM	DEFINITION
HSM	Hardware Security Module
HSMaaS	Hardware Security Module as a Service
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
HVAC	Heating, Ventilation, Air Conditioning
IaaS	Infrastructure as a Service
IAM	Identity and Access Management
ICMP	Internet Control Message Protocol
ICS	Industrial Control Systems
IDEA	International Data Encryption Algorithm
IDF	Intermediate Distribution Frame
IdP	Identity Provider
IDS	Intrusion Detection System
IEEE	Institute of Electrical and Electronics Engineers
IKE	Internet Key Exchange
IM	Instant Messaging
IMAP4	Internet Message Access Protocol v4
IoC	Indicators of Compromise
IoT	Internet of Things
IP	Internet Protocol
IPS	Intrusion Prevention System
IPSec	Internet Protocol Security
IR	Incident Response
IRC	Internet Relay Chat
IRP	Incident Response Plan
ISA	Interconnection Security Agreement
ISFW	Internal Segmentation Firewall
ISO	International Organization for Standardization
ISP	Internet Service Provider
ISSO	Information Systems Security Officer
ITCP	IT Contingency Plan
IV	Initialization Vector
KDC	Key Distribution Center
KEK	Key Encryption Key
L2TP	Layer 2 Tunneling Protocol
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
LEAP	Lightweight Extensible Authentication Protocol
MaaS	Monitoring as a Service
MAC	Media Access Control
MAM	Mobile Application Management
MAN	Metropolitan Area Network
MBR	Master Boot Record
MD5	Message Digest 5
MDF	Main Distribution Frame
MDM	Mobile Device Management
MFA	Multifactor Authentication
MFD	Multifunction Device
MFP	Multifunction Printer
ML	Machine Learning

ACRONYM	DEFINITION
MMS	Multimedia Message Service
MOA	Memorandum of Agreement
MOU	Memorandum of Understanding
MPLS	Multiprotocol Label Switching
MSA	Measurement Systems Analysis
MS-CHAP	Microsoft Challenge-Handshake Authentication Protocol
MSP	Managed Service Provider
MSSP	Managed Security Service Provider
MTBF	Mean Time Between Failures
MTTF	Mean Time to Failure
MTTR	Mean Time to Repair
MTU	Maximum Transmission Unit
NAC	Network Access Control
NAS	Network-attached Storage
NAT	Network Address Translation
NDA	Non-disclosure Agreement
NFC	Near-field Communication
NFV	Network Function Virtualization
NGFW	Next-generation Firewall
NG-SWG	Next-generation Secure Web Gateway
NIC	Network Interface Card
NIDS	Network-based Intrusion Detection System
NIPS	Network-based Intrusion Prevention System
NIST	National Institute of Standards & Technology
NOC	Network Operations Center
NTFS	New Technology File System
NTLM	New Technology LAN Manager
NTP	Network Time Protocol
OCSP	Online Certificate Status Protocol
OID	Object Identifier
OS	Operating System
OSI	Open Systems Interconnection
OSINT	Open-source Intelligence
OSPF	Open Shortest Path First
OT	Operational Technology
OTA	Over-The-Air
OTG	On-The-Go
OVAL	Open Vulnerability and Assessment Language
OWASP	Open Web Application Security Project
P12	PKCS #12
P2P	Peer-to-Peer
PaaS	Platform as a Service
PAC	Proxy Auto Configuration
PAM	Privileged Access Management
PAM	Pluggable Authentication Modules
PAP	Password Authentication Protocol
PAT	Port Address Translation
PBKDF2	Password-based Key Derivation Function 2
PBX	Private Branch Exchange
PCAP	Packet Capture

ACRONYM	DEFINITION
PCI DSS	Payment Card Industry Data Security Standard
PDU	Power Distribution Unit
PE	Portable Executable
PEAP	Protected Extensible Authentication Protocol
PED	Portable Electronic Device
PEM	Privacy Enhanced Mail
PFS	Perfect Forward Secrecy
PGP	Pretty Good Privacy
PHI	Personal Health Information
PII	Personally Identifiable Information
PIN	Personal Identification Number
PIV	Personal Identity Verification
PKCS	Public Key Cryptography Standards
PKI	Public Key Infrastructure
PoC	Proof of Concept
POP	Post Office Protocol
POTS	Plain Old Telephone Service
PPP	Point-to-Point Protocol
PPTP	Point-to-Point Tunneling Protocol
PSK	Preshared Key
PTZ	Pan-Tilt-Zoom
PUP	Potentially Unwanted Program
QA	Quality Assurance
QoS	Quality of Service
PUP	Potentially Unwanted Program
RA	Registration Authority
RAD	Rapid Application Development
RADIUS	Remote Authentication Dial-in User Service
RAID	Redundant Array of Inexpensive Disks
RAM	Random Access Memory
RAS	Remote Access Server
RAT	Remote Access Trojan
RC4	Rivest Cipher version 4
RCS	Rich Communication Services
RFC	Request for Comments
RFID	Radio Frequency Identification
RIPEMD	RACE Integrity Primitives Evaluation Message Digest
ROI	Return on Investment
RPO	Recovery Point Objective
RSA	Rivest, Shamir, & Adleman
RTBH	Remotely Triggered Black Hole
RTO	Recovery Time Objective
RTOS	Real-time Operating System
RTP	Real-time Transport Protocol
S/MIME	Secure/Multipurpose Internet Mail Extensions
SaaS	Software as a Service
SAE	Simultaneous Authentication of Equals
SAML	Security Assertions Markup Language
SCADA	Supervisory Control and Data Acquisition
SCAP	Security Content Automation Protocol

ACRONYM	DEFINITION
SCEP	Simple Certificate Enrollment Protocol
SDK	Software Development Kit
SDLC	Software Development Life Cycle
SDLM	Software Development Life-cycle Methodology
SDN	Software-defined Networking
SDP	Service Delivery Platform
SDV	Software-defined Visibility
SED	Self-Encrypting Drives
SEH	Structured Exception Handling
SFTP	SSH File Transfer Protocol
SHA	Secure Hashing Algorithm
SIEM	Security Information and Event Management
SIM	Subscriber Identity Module
SIP	Session Initiation Protocol
SLA	Service-level Agreement
SLE	Single Loss Expectancy
SMB	Server Message Block
S/MIME	Secure/Multipurpose Internet Mail Extensions
SMS	Short Message Service
SMTP	Simple Mail Transfer Protocol
SMTPS	Simple Mail Transfer Protocol Secure
SNMP	Simple Network Management Protocol
SOAP	Simple Object Access Protocol
SOAR	Security Orchestration, Automation, Response
SoC	System on Chip
SOC	Security Operations Center
SPF	Sender Policy Framework
SPIM	Spam over Instant Messaging
SQL	Structured Query Language
SQLi	SQL Injection
SRTP	Secure Real-time Transport Protocol
SSD	Solid State Drive
SSH	Secure Shell
SSID	Service Set Identifier
SSL	Secure Sockets Layer
SSO	Single Sign-on
STIX	Structured Threat Information eXpression
STP	Shielded Twisted Pair
SWG	Secure Web Gateway
TACACS+	Terminal Access Controller Access Control System
TAXII	Trusted Automated eXchange of Intelligence Information
TCP/IP	Transmission Control Protocol/Internet Protocol
TGT	Ticket Granting Ticket
TKIP	Temporal Key Integrity Protocol
TLS	Transport Layer Security
TOTP	Time-based One Time Password
TPM	Trusted Platform Module
TSIG	Transaction Signature
TTP	Tactics, Techniques, and Procedures

ACRONYM	DEFINITION
UAT	User Acceptance Testing
UDP	User Datagram Protocol
UEBA	User and Entity Behavior Analytics
UEFI	Unified Extensible Firmware Interface
UEM	Unified Endpoint Management
UPS	Uninterruptible Power Supply
URI	Uniform Resource Identifier
URL	Universal Resource Locator
USB	Universal Serial Bus
USB OTG	USB On-The-Go
UTM	Unified Threat Management
UTP	Unshielded Twisted Pair
VBA	Visual Basic for Applications
VDE	Virtual Desktop Environment
VDI	Virtual Desktop Infrastructure
VLAN	Virtual Local Area Network
VLSM	Variable-length Subnet Masking
VM	Virtual Machine
VoIP	Voice over IP
VPC	Virtual Private Cloud
VPN	Virtual Private Network
VTC	Video Teleconferencing
WAF	Web Application Firewall
WAP	Wireless Access Point
WEP	Wired Equivalent Privacy
WIDS	Wireless Intrusion Detection System
WIPS	Wireless Intrusion Prevention System
WORM	Write Once Read Many
WPA	WiFi Protected Access
WPS	WiFi Protected Setup
XaaS	Anything as a Service
XML	Extensible Markup Language
XOR	Exclusive OR
XSRF	Cross-site Request Forgery
XSS	Cross-site Scripting

Security+ Proposed Hardware and Software List

CompTIA has included this sample list of hardware and software to assist candidates as they prepare for the Security+ exam. This list may also be helpful for training companies that wish to create a lab component for their training offering. The bulleted lists below each topic are sample lists and are not exhaustive.

HARDWARE

- Laptop with Internet access
- Separate wireless NIC
- WAP
- Firewall
- UTM
- Mobile device
- Server/cloud server
- IoT devices

SOFTWARE

- Virtualization software
- Penetration testing OS/distributions (e.g., Kali Linux, Parrot OS)
- SIEM
- Wireshark
- Metasploit
- tcpdump

OTHER

- Access to a CSP

CompTIA

Revised 5/9/2022

IT Governance

Security Standards Organizations[1]

1. CIS – Center for Internet Security
    a. https://cisecurity.org
    b. The CIS is a non-profit organization formed by a large number of commercial, academic, and government bodies. The CIS's mission is to identify, develop, and promote best practices in cybersecurity. To this end, it develops security benchmarks and assessment tools for a wide variety of operating systems and network applications. CIS Controls is a general purpose guide having a common set of 20 security controls. The project began as a consensus project out of the US DoD, and has morphed into the de facto standard for selecting an effective set of security controls. The framework is now maintained by the Center for Internet Security and can be found at https://www.cisecurity.org/controls/.
2. IEC – International Electrotechnical Commission
    a. https://www.iec.ch
    b. Founded in 1906, the IEC is the world's leading organization for the preparation and publication of International Standards for all electrical, electronic and related technologies. These are known collectively as "electrotechnology". The IEC provides a platform to companies, industries and governments for meeting, discussing and developing the International Standards they require. Close to 20 000 experts from industry, commerce, government, test and research labs, academia and consumer groups participate in IEC Standardization work.
3. IEEE – Institute of Electrical and Electronics Engineers
    a. https://www.ieee.org
    b. The IEEE is a professional association of engineers and scientists of many disciplines. Its mission is to advance technological innovation or all sorts; one of the most visible aspects of their work is the global IEEE standards published in a number of technological fields, such as the IEEE 802 networking standards: IEEE 802.3 Ethernet standards and the IEEE 802.11 Wi-Fi standards.
4. IETF – Internet Engineering Task Force
    a. https://www.ietf.org
    b. The IETF began with the US government agencies that first developed TCP/IP, but it is now an open-standards organization under the management of the Internet Society (ISOC). The IETF develops Internet standards by consensus, distributing numbered Request for Comments (RFC) documents via its internal mailing lists. A specification that advances through the review process is classified as a Proposed Standard, and finally, an Internet Standard. Not all common protocols used on the Internet are IETF standards, but a great many are.

[1] Most of the information regarding standards organizations was obtained from CompTIA Security+ Certification SY0-501 Instructor Edition, 30 BIRD Media, 2017. ISBN 978-1-947914-94-0. The information contained throughout this document was also obtained from the various websites, links of which are included here.

5. ISACA – Information Systems Audit and Control Association
    a. https://www.isaca.org
    b. ISACA is an international professional association focused on IT governance. ISACA currently serves more than 140,000 constituents (members and professionals holding ISACA certifications) in more than 180 countries. There is a network of ISACA chapters with more than 200 chapters established in over 80 countries. Chapters provide education, resource sharing, advocacy, networking and other benefits. Some of its major publications include COBIT and Standards, Guidelines and Procedures for Information System Auditing.
6. ISO – International Organization for Standardization
    a. https://www.iso.org
    b. The ISO comprises the standards bodies of over 160 member nations. ISO standards include everything from the OSI network model (ISO/IEC 7498-1) to twist direction in yard (ISO 2), and many involve information technology or security standards and practices.
7. ISOC – Internet Society
    a. https://www.internetsociety.org
    b. The ISOC is the parent organization of the IETF and several other organizations and committees involved in Internet development. The ISOC doesn't directly develop standards itself; instead, it focuses primarily on providing corporate management services for its member organizations, to speak on their behalf in Internet governance discussions, and to organize Internet-related seminars, conferences, and training programs.
8. ITU – International Telecommunications Union
    a. https://www.itu.int/en/Pages/default.aspx
    b. The ITU is a UN agency charged with global tasks related to telecommunications. It allocates shared global use of the radio spectrum, coordinates national governments in assigning satellite orbits, and promotes global technical standards related to networking and communication. Many ITU standards can be recognized by their "letter-period-number" format such as the X.509 (Digital certificates) used by secure websites, and H.264 (MPEG-4) used for digital video encoding both on the Internet and by television providers.
9. NIST – National Institute of Standards and Technology
    a. https://www.nist.gov
    b. NIST is a US government agency charged with developing and supporting standards used by other government organizations. While it primarily promotes standards for use by the US government, they are frequently used by others with similar technology needs. In recent years, computer society standards have become a major part of its mission. NIST shares most of its findings with the security community in general, and regularly publishes information about known software vulnerabilities and security best practices.

10. NSA – National Security Agency
    a. https://www.nsa.gov
    b. The NSA is a US signals intelligence organization whose responsibilities include information gathering, codebreaking, and codemaking. The NSA develops cryptographic standards and secures government information against attack. While much of its work is classified, the NSA has had a role in designing and standardizing some of the most widely used cryptographic standards, such as DES, AES, and SHA.
11. W3C – World Wide Web Consortium
    a. https://www.w3.org
    b. The W3C is a standards organization founded to develop and maintain interoperable standards for the World Wide Web (WWW) used by web browsers and servers as well as other technologies. W3C standards include HTML, XML, CSS, and many others used for web-based communications. While the W3C doesn't focus on security technologies per se, security of web standards is a major topic in the wider field of information security.

Frameworks and Reference Architectures[2]

1. COBIT – Control Objectives for Information and Related Technologies
    a. https://apmg-international.com/product/cobit-5
    b. COBIT 5 is the latest edition of ISACA's globally accepted framework. It provides an end-to-end business view of the governance of enterprise IT, reflecting the central role of information and technology in creating value for enterprises of all sizes. The principles, practices, analytical tools and models found in COBIT 5 embody thought leadership and guidance from business, IT and governance experts around the world. COBIT 5 provides guidance to executives and those charged with making decisions concerning the use of technology in support of organizational objectives. COBIT 5 helps business leaders address the needs of all stakeholders across the enterprise and ultimately maximize the value from information and technology.
2. SABSA – Sherwood Applied Business Security Architecture
    a. https://sabsa.org
    b. SABSA is a proven methodology for developing business-driven, risk and opportunity focused Security Architectures at both enterprise and solutions level that traceably support business objectives. It is also widely used for Information Assurance Architectures, Risk Management Frameworks, and to align and seamlessly integrate security and risk management into IT Architecture methods and frameworks.

       The SABSA framework and methodology is used successfully around the globe to meet a wide variety of Enterprise needs including Risk Management, Information Assurance, Governance, and Continuity Management. SABSA has evolved since 1995 to be the 'approach of choice' for organizations in 50 countries and in sectors as diverse as Banking, Homeless Management, Nuclear Power, Information Services, Communications Technology, Manufacturing and Government.

[2] Information about industry-specific frameworks, such as HIPAA, NERC CIP, and HITRUST Common Security Framework (CSF), are not included in this document.

SABSA is comprised of a series of integrated frameworks, models, methods and processes, used independently or as an holistic integrated enterprise solution, including:

- Business Requirements Engineering Framework (known as Attributes Profiling)
- Risk and Opportunity Management Framework
- Policy Architecture Framework
- Security Services-Oriented Architecture Framework
- Governance Framework
- Security Domain Framework
- Through-life Security Service Management & Performance Management Framework
- The SABSA Institute develops and maintains the method and certifies and accredits the professional Architects who use it in approximately 50 countries around the world.

3. NIST Cybersecurity Framework (CSF) and Risk Management Framework (RMF)
    a. https://nist.gov/cyberframework
        i. The Framework is voluntary guidance, based on existing standards, guidelines, and practices for organizations to better manage and reduce cybersecurity risk. In addition to helping organizations manage and reduce risks, it was designed to foster risk and cybersecurity management communications amongst both internal and external organizational stakeholders.
    b. https://www.nist.gov/cyberframework/risk-management-framework
        i. This has actually been incorporated into the CSF.
4. OWASP – Open Web Application Security Project
    a. https://owasp.org
    b. The Open Web Application Security Project® (OWASP) is a nonprofit foundation that works to improve the security of software. Through community-led open source software projects, hundreds of local chapters worldwide, tens of thousands of members, and leading educational and training conferences, the OWASP Foundation is the source for developers and technologists to secure the web.
5. ISO 27001
    a. https://iso.rg/standard/54534.html
    b. ISO/IEC 27001:2013 specifies the requirements for establishing, implementing, maintaining and continually improving an information security management system within the context of the organization. It also includes requirements for the assessment and treatment of information security risks tailored to the needs of the organization. The requirements set out in ISO/IEC 27001:2013 are generic and are intended to be applicable to all organizations, regardless of type, size or nature.

6. International Organization for Standardization (ISO) 27K
    a. The International Organization for Standardization (ISO) has produced a cybersecurity framework in conjunction with the International Electrotechnical Commission (IEC). The framework was established in 2005 and revised in 2013. Unlike the NIST framework, the ISO 27001 Information Security Management standard must be purchased (iso.org/standard/54534.html). ISO 27001 is part of an overall 27000 series of information security standards, also known as 27K. Of these, 27002 classifies security

controls, 27017 and 27018 reference cloud security, and 27701 focuses on personal data and privacy.

7. ISO 31K
    a. Where ISO 27K is a cybersecurity framework, **ISO 31K** (iso.org/iso-31000-risk-management.html) is an overall framework for enterprise risk management (ERM). ERM considers risks and opportunities beyond cybersecurity by including financial, customer service, competition, and legal liability factors. ISO 31K establishes best practices for performing risk assessments.

8. FedRAMP – Federal Risk and Authorization Management Program
    a. https://www.fedramp.gov/
    b. This process is a government-wide program that provides a standardized approach to security assessment, authorization, and continuous monitoring for systems using cloud products and services
9. EU-U.S. Privacy Shield Framework
    a. https://www.privacyshield.gov/eu-us-framework
    b. https://www.privacyshield.gov/servlet/servlet.FileDownload?file=015t0000000QJdgx

10. OCI – Open Container Initiative
    a. https://opencontainers.org/
    b. Multiple major container platforms exist, but the industry has coalesced around this standard form, which enables standardization and the market stability of the environment. Different vendors in the Container space have slightly different terminologies, so individual implementations need to be checked by vendor to understand the exact definition of a container and cell in their environment.
11. Cloud Security Alliance
    a. The not-for-profit organization **Cloud Security Alliance (CSA)** produces various resources to assist cloud service providers (CSP) in setting up and delivering secure cloud platforms. These resources can also be useful for cloud consumers in evaluating and selecting cloud services.
    b. Security Guidance (cloudsecurityalliance.org/research/guidance)—a best practice summary analyzing the unique challenges of cloud environments and how on-premises controls can be adapted to them.
    c. Enterprise reference architecture (ea.cloudsecurityalliance.org)—best practice methodology and tools for CSPs to use in architecting cloud solutions. The solutions are divided across a number of domains, such as risk management and infrastructure, application, and presentation services.
    d. Cloud controls matrix (cloudsecurityalliance.org/research/working-groups/cloud-controls-matrix)—lists specific controls and assessment guidelines that should be implemented by CSPs. For cloud consumers, the matrix acts as a starting point for cloud contracts and agreements as it provides a baseline level of security competency that the CSP should meet.
12. Statements on Standards for Attestation Engagements (SSAE) Service Organization Control (SOC)
    a. The Statements on Standards for Attestation Engagements (SSAE) are audit specifications developed by the American Institute of Certified Public Accountants (AICPA). These audits are designed to assure consumers that service providers—notably cloud providers, but including any type of hosted or third-party service—meet professional standards

(aicpa.org/interestareas/frc/assuranceadvisoryservices/serviceorganization-smanagement.html). Within SSAE No. 18 (the current specification), there are several levels of reporting:

b. Service Organization Control (SOC2)—evaluates the internal controls implemented by the service provider to ensure compliance with Trust Services Criteria (TSC) when storing and processing customer data. TSC refers to security, confidentiality, integrity, availability, and privacy properties. An SOC2 Type I report assesses the system design, while a Type II report assesses the ongoing effectiveness of the security architecture over a period of 6-12 months. SOC2 reports are highly detailed and designed to be restricted. They should only be shared with the auditor and regulators, and with important partners under non-disclosure agreement (NDA) terms.
c. SOC3—a less detailed report certifying compliance with SOC2. SOC3 reports can be freely distributed.

Benchmarks and Secure Configuration Guides[3]

1. STIGs – Security Technical Configuration Guides
    a. https://iase.disa.mil/stigs/Pages/index.aspx
    b. DoD DISA STIGs program provides comprehensive, proscriptive configuration guides for all major operating systems. (Department of Defense Information Security Agency (DISA) Security Technical Implementation Guides (STIG).
2. NCP - National Checklist Program
    a. https://nvd.nist.gov/ncp/repository
    b. The National Checklist Program (NCP), defined by the NIST SP 800-70, is the U.S. government repository of publicly available security checklists (or benchmarks) that provide detailed low level guidance on setting the security configuration of operating systems and applications.
3. The SANS Institute
    a. https://sans.org
    b. The SANS Institute was established in 1989 as a cooperative research and education organization. Its programs now reach more than 165,000 security professionals around the world. A range of individuals from auditors and network administrators, to chief information security officers are sharing the lessons they learn and are jointly finding solutions to the challenges they face. At the heart of SANS are the many security practitioners in varied global organizations from corporations to universities working together to help the entire information security community.
4. Center for Internet Security (CIS)
    a. The Center for Internet Security (cisecurity.org) is a not-for-profit organization (founded partly by The SANS Institute). It publishes the well-known "The 20 CIS Controls." The CIS-RAM (Risk Assessment Method) can be used to perform an overall evaluation of security posture (learn.cisecurity.org/cis-ram).
    b. CIS also produces benchmarks for different aspects of cybersecurity. For example, there are benchmarks for compliance with IT frameworks and compliance programs, such as **PCI** DSS, NIST 800-53, SOX, and ISO 27000. There are also product-focused benchmarks, such as for Windows Desktop, Windows Server, macOS, Linux, Cisco, web browsers, web servers, database and email servers, and VMware ESXi. The CIS-CAT

[3] Note these there are also vendor-specific guides that are not listed here.

(Configuration Access Tool) can be used with automated vulnerability scanners to test compliance against these benchmarks (cisecurity.org/cybersecurity-tools/cis-cat-pro/cis-cat-faq).

Miscellaneous Guidelines and Resources

1. NSRL – The National Software Reference Library (NSRL)
    a. https://www.nist.gov/itl/ssd/software-quality-group/national-software-reference-library-nsrl
    b. The collects software from various sources and incorporates file profiles into a Reference Data Set (RDS) available for download as a service (www.nsrl.nist.gov)
2. NVD- NISTS's National Vulnerability Database
    a. https://nvd.nist.gov/
    b. The NVD is the U.S. government repository of standards based vulnerability management data represented using the Security Content Automation Protocol (SCAP). This data enables automation of vulnerability management, security measurement, and compliance. The NVD includes databases of security checklist references, security-related software flaws, misconfigurations, product names, and impact metrics.
3. OECD – Organization for Economic Co-operation and Development
    a. http://www.oecd.org/
    b. A primary source of intellectual and political thought on privacy; this multinational entity has, for decades, conducted multilateral discussions and policy formation on a wide range of topics, including privacy.
4. NFPA – National Fire Protection Association
    a. https://www.nfpa.org/codes-and-standards/all-codes-and-standards/list-of-codes-and-standards/detail?code=75
    b. NFPA 75 is the Standard for the Protection of Information Technology Equipment; outlines measures that can be taken to minimize the damage to electronic equipment exposed to water. (See Fire Suppression.)
5. OWASP – Open Web Application Security Project
    a. The Open Web Application Security Project (OWASP) is a not-for-profit, online community that publishes several secure application development resources, such as the Top 10 list of the most critical application security risks (owasp.org/www-project-top-ten). OWASP has also developed resources, such as the Zed Attack Proxy and Juice Shop (a deliberately unsecure web application), to help investigate and understand penetration testing and application security issues.

Data Privacy Laws

We're going to spend a few minutes reviewing some of the laws that have been enacted to protect the privacy of electronic data.

The United States has passed a significant number of laws regarding information security, especially the privacy of electronic data. Prior to the year 2000, there weren't many laws that addressed this particular area of information security. However, due to a rash of break-ins, exploits, and another abuses, many states and the federal government enacted laws to protect the privacy of electronic data.

A key thing to remember is that you need to be very familiar with the requirements of the laws that apply to your organization. To protect your organization and to protect yourself, you need to be aware of how these laws apply to your organization.

This lesson addresses some of the higher profile data privacy laws, but it doesn't identify all the laws that apply to your organization, nor does it provide minute details of each law that it does cover. This lesson provides only a simple, high-level overview of these laws. You are responsible to research whether these laws apply to your organization, and if so, how. You are also responsible to be aware of other federal, state, and even local laws that effect your organization.

Let's begin with the Health Insurance Portability and Accountability Act of 1996 called HIPAA.

HIPPA

HIPPA specifies that all organizations having anything to do with healthcare must protect the health information that they maintain.

These organizations must implement policies and procedures to protect this information, regardless of the storage medium, such as paper in a filing cabinet or electronic digital format. HIPPA also establishes national standards for transferring electronic healthcare information.

The next act you need to be familiar with is the Sarbanes-Oxley Act of 2002.

Sarbox

The Sarbanes-Oxley Act of 2002, sometimes referred to as Sarbox, is the result of a flood of corporate fraud in the late 1990s and the early 2000s. Sarbox requires publically traded companies to adhere to very stringent reporting requirements and implement strong controls on electronic financial reporting systems.

A key point in Sarbox is that organizations have to keep information for a certain time. This especially relates to email. In fact, Sarbox created an entire industry of email archiving companies so that organizations can comply with its regulations.

Next we need to look at the Gramm-Leach-Bliley Act, which we refer to as GLBA.

GLBA

GLBA is designed to protect private data much like HIPAA does; however, GLBA applies to private information held at financial institutions.

There are two main functions of GLBA. First, it requires all banks and any other financial institutions to alert their customers as to that organization's privacy policies. If you've received a little pamphlet from your financial institution that contained that institution's privacy statement, that pamphlet was in response to the Gramm-Leach-Bliley Act.

In addition, all personally identifiable financial information (PII) within a financial institution—"either electronic or paper formats—"has to be protected. Essentially this act specifies that a policy must be in place to protect private information from foreseeable threats and to maintain data integrity. In order to implement this, GLBA requires financial institutions to put in place three main components.

The first one is the Financial Privacy Rule. The second one is the Safeguards Rule. And the third one is called Pretexting Protection. Essentially, the Financial Privacy Rule requires all financial institutions to provide each customer

with the privacy notice that we talked about earlier. It must be provided at the time the relationship is established. That is, when you go in to open a new banking account. It also has to be provided every year thereafter.

The Safeguards Rule requires financial institutions to develop a written information security plan, which describes in detail how the company plans to protect clients' personal information. Finally, Pretexting Protection encourages financial institutions to train their staff how to recognize social engineering exploits, which they call pretexting. The reason is that social engineering exploits use some type of pretext.

You should also be familiar with the Patriot Act of 2001.

Patriot Act

This act enables law enforcement agencies to detect and suppress terrorism by giving law enforcement the authority to request information from organizations. All organizations, public or private, must provide the requested information to the appropriate law enforcement agencies under the authority of a valid court order or a subpoena.

California Database Security Breach Act

The next law you need to be familiar with is the California Database Security Breach Act of 2003. This law specifies that any agency, person, government entity, or company that does business in the state of California must inform California residents within 48 hours if a database breach or other security breach occurs in which personal information has been stolen or is believed to have been stolen.

Most other states have similar state laws modeled on the California Database Security Breach Act of 2003, making it a significant act. Therefore, use your favorite search engine to search for similar acts in the states in which you do business. If you do business in those states and your organization has a breach of personal information, then you are under a regulatory compliance requirement to inform anyone who may be affected.

The last act we're going to look at is the Children's Online Privacy Protection Act of 1998, which is referred to as COPPA.

COPPA

COPPA requires organizations that provide online services designed for kids below the age of 13—"such as websites and gaming sites—"to obtain parental consent prior to collecting a child's personal information and using it, such as displaying it on the website, selling it to a marketing company, and so on.

In addition, COPPA also specifies that such online services cannot require kids to provide more information than what they determine as reasonable in order to participate.

As I stated before, this is not an all-inclusive list of laws and acts regarding the privacy of electronic data with which you need to be familiar.

Summary

You need to research laws that apply to your organization. It's also probably a good idea to use the services of a lawyer who is familiar with this field just to make sure that your organization stays in compliance. created and are subject to change. If the URLs no longer work, use a search engine to find the act or law we talked about.

Security standards organizations

- CIS – Center for Internet security
- IEEE – Institute of Electrical and Electronics Engineers
- IETF – Internet Engineering Task Force
- ISO – International Organization for Standardization
- ISOC – Internet Society
- ITU – International Telecommunication Union
- NIST – National Institute of Standards and Technology
- NSA – National Security Agency
- W3C – World Wide Web Consortium

Data Roles and Responsibilities

A **data governance** policy describes the security controls that will be applied to protect data at each stage of its life cycle. There are important institutional governance roles for oversight and management of information assets within the life cycle:

- **Data owner**—a senior (executive) role with ultimate responsibility for maintaining the confidentiality, integrity, and availability of the information asset. The owner is responsible for labeling the asset (such as determining who should have access and determining the asset's criticality and sensitivity) and ensuring that it is protected with appropriate controls (access control, backup, retention, and so forth). The owner also typically selects a steward and custodian and directs their actions and sets the budget and resource allocation for sufficient controls.
- **Data steward**—this role is primarily responsible for data quality. This involves tasks such as ensuring data is labeled and identified with appropriate metadata and that data is collected and stored in a format and with values that comply with applicable laws and regulations.
- **Data custodian**—this role handles managing the system on which the data assets are stored. This includes responsibility for enforcing access control, encryption, and backup/recovery measures.
- **Data Privacy Officer (DPO)**—this role is responsible for oversight of any personally identifiable information (PII) assets managed by the company. The privacy officer ensures that the processing, disclosure, and retention of PII complies with legal and regulatory frameworks.

In the context of legislation and regulations protecting personal privacy, the following two institutional roles are important:

- **Data controller**—the entity responsible for determining why and how data is stored, collected, and used and for ensuring that these purposes and means are lawful. The data controller has ultimate responsibility for privacy breaches, and is not permitted to transfer that responsibility.
- **Data processor**—an entity engaged by the data controller to assist with technical collection, storage, or analysis tasks. A data processor follows the instructions of a data controller with regard to collection or processing.

Data controller and processor tend to be organizational roles rather than individual ones. For example, if Widget.foo collects personal data to operate a webstore on its own cloud, it is a data collector and data processor. If Widget.foo passes aggregate data to Grommet.foo asking them to run profitability analytics for different customer segments on its AI-backed cloud, Grommet.foo is a data processor acting under the instruction of Widget.foo. Within the Grommet.foo and Widget.foo companies, the data owner might take personal responsibility for the lawful performance of data controller and processor functions.

Data Classifications

Data classification and typing schemas tag data assets so that they can be managed through the information life cycle. A data classification schema is a decision tree for applying one or more tags or labels to each data asset. Many data classification schemas are based on the degree of confidentiality required:

- Public (unclassified)—there are no restrictions on viewing the data. Public information presents no risk to an organization if it is disclosed but does present a risk if it is modified or not available.
- Confidential (secret)—the information is highly sensitive, for viewing only by approved persons within the owner organization, and possibly by trusted third parties under NDA.
- Critical (top secret)—the information is too valuable to allow any risk of its capture. Viewing is severely restricted.

Another type of classification schema identifies the kind of information asset:

- Proprietary—**Proprietary information** or **intellectual property (IP)** is information created and owned by the company, typically about the products or services that they make or perform. IP is an obvious target for a company's competitors, and IP in some industries (such as defense or energy) is of interest to foreign governments. IP may also represent a counterfeiting opportunity (movies, music, and books, for instance).
- Private/personal data—Information that relates to an individual identity.
- Sensitive—This label is usually used in the context of personal data. Privacy-sensitive information about a person could harm them if made public and could prejudice decisions made about them if referred to by internal procedures. As defined by the EU's General Data Protection Regulations (GDPR), sensitive personal data includes religious beliefs, political opinions, trade union membership, gender, sexual orientation, racial or ethnic origin, genetic data, and health information (ec.europa.eu/info/law/law-topic/data-protection/reform/rules-business-and-organisations/legal-grounds-processing-data/sensitive-data/what-personal-data-considered-sensitive_en).

Information Security Business Units

The following units are ofen used to represent the security function within the organizational hierarchy.

Security Operations Center (SOC)

A **security operations center (SOC)** is a location where security professionals monitor and protect critical information assets across other business functions, such as finance, operations, sales/marketing, and so on. Because SOCs can be difficult to establish, maintain, and finance, they are usually employed by larger corporations, like a government agency or a healthcare company.

Risk Management

The Process of identifying, assessing, and mitigating vulnerabilities and threats to the essential functions that a business must perform to serve its customers. This includes five phases: identifying mission-essential functions, identifying vulnerabilities, identifying threats, analyzing business impact, and identifying risk responses

Acronym	Name	Description
MEF	Mission Essential Function	One that cannot be deferred; the organization must be able to perform the function as close to continually as possible, and if there is any service disruption, the mission essential functions must be restored first.
MTD	Maximum Tolerable Downtime	The longest period of time that a business function outage may occur for without causing irrevocable business failure. Each business process can have its own MTD.
RTO	Recovery-time Objective	The period following a disaster that an individual IT system may remain offline. This represents the amount of time it takes to identify that there is a problem and then perform recovery,
WRT	Work Recovery Time	Following systems recovery, there may be additional work to reintegrate different systems, test overall functionality, and brief system users on any changes or different working practices so that the business function is again fully supported.
RPO	Recovery Point Objective	The amount of data loss that a system can sustain, measured in time. That is, if a database is destroyed by a virus, an RPO of 24 hours means that the data can be recovered (from a backup) to a point not more than 24 hours before the database was infected.
BPA	Business Process Analysis	For mission essential functions, it is important to reduce the number of dependencies between components. The BPA identifies inputs, hardware, staff, outputs, and process flows.
SPoF	Single Point of Failure	Reducing dependencies helps to prevent these.
Quantitative Risk Assessments	SLE – Single Loss Expectancy	
	ALE – Annual Loss Expectance	
	ARO – Annual Rate of Occurrence	
	AV – Asset Value	
	EF – Exposure Factor	The measurement of the magnitude of the loss. In other words, what percentage of the asset is lost.
Qualitative Risk Assessments	Categorized	Irreplaceable, High, Medium, Low; Critical, High, Medium, Low probability; One-off, recurring
	Impact Grid	Red, yellow, green
Formulas	ALE=SLE*ARO / SLE=AV*EF	

Risk Management

KPI	Key Performance Indicators	Used to determine the reliability of each asset, such as servers, disk arrays, switches, routers, etc.
MTTF	Mean Time to Failure	Used for non-repairable assets, like a hard drive. The MTTF for the below example is the total time divided by the number of devices, so MTTF=(10*50)/10 with the result being 50 hours/failure
MTBF	Mean Time Between Failures	MTBF= total time/the number of failures (for example, if you have 10 devices that run for 50 hours and two of them fail, the MTBF is 250 hours/failure, (10*50)/2
MTTR	Mean Time to Repair	A measure of the time taken to correct a fault so that a system is returned to full operation. This can also be described as mean time to "replace" or "recover'" This metric is important in determining the RTO.
Notes		

Risk Matrix / Heat Map

Risk Matrix/Heat Map: A risk matrix is used to get a visual representation of the risks affecting a company. The heat map shows the severity of the situation, with the most severe risks being in red:

The areas in red would cause severe damage to the company, where pink would still mean a high risk. The lighter pink and green would mean a medium risk. The darker green and the very dark green would mean a low risk. This is a good way to present a risk analysis to senior management.

Incident Response Procedures

Before we start making incident response plans, we need to have a process in place, and the process we are going to use is as shown in *Figure 12.1*:

The incident response process must be carried out in order, starting with stage 1, which is the preparation phase. Let's look at these stages in order:

- **Preparation**: The preparation phase is where the different incident response plans are written and kept up to date. System configurations are documented as well.
- **Identification**: Once an incident has occurred, it is important that the appropriate incident response plan is invoked, and that stakeholders and the incident response team for that particular incident are notified.
- **Containment**: At this stage, we will isolate or quarantine computers, to prevent the attack from spreading any further and collect the volatile evidence.
- **Eradication**: In the eradication phase, we want to destroy the source of the incident. For example, if it is a virus, we want it totally removed. We will remove the virus or delete infected files, patch the system, and turn off any services that we don't need, so that it is hardened.
- **Recovery**: In the recovery phase, we are getting the company back to an operational state, hopefully within the **Recovery Point Objective** (**RPO**). For example, imaging machines, restoring data, or putting domain controllers or infected machines back online after cleansing.

- **Lessons Learned**: Lessons learned is a detective phase where we pull together all of the facts and plan to prevent a re-occurrence in the future. Failure to carry this out will lead to a re-occurrence.

Example: A domain controller is infected with a virus. The first stage is containment, where we take it off the network. The next stage is eradication, where we remove the virus and patch the server. The last stage is recovery, where the clean server is put back online. After the incident has been dealt with and we are back up and running, we carry out lessons learned, where we look at how the domain controller got the virus in the first place and prevent it from happening again.

Attack Frameworks

There have been different attack frameworks developed to help cybersecurity teams to better prepare themselves for cyber attacks. We are going to look at three different models, so let's start with the MITRE ATT&CK Framework.

MITRE ATT&CK Framework

Mitre is a US Government-sponsored company whose aim is to help prevent cyber attacks. They developed an online framework that can be used by the general public and they have many matrices. They give information about adversaries and their attack methods. They use the acronym *ATT&CK* to help you understand better the attack vectors used by the attackers. If you go on to the *Mitre* website (`https://attack.mitre.org`), you will find a huge spreadsheet that you can use to find information on adversaries, their attack methods, and how to mitigate these attacks. This aids everyone from cybersecurity teams to threat hunters, so let's look at each of these in turn. Let's look at the breakdown of the acronym:

- **Adversarial**: This looks at the behavior of potential attackers who are put into different groups. An example would be *APT28*, a Russian group who allegedly interfered with the US election in 2016.
- **Tactics**: This is the medium by which the attack will be carried out. We could look at a phishing attack from which we can drill down, and it will explain how phishing attacks are launched.
- **Techniques**: These are a breakdown of the processes of how an attack will be launched.

  More information on drive-by compromise can be found at the following link: `https://attack.mitre.org/techniques/T1189/`.
- **Common Knowledge**: This is the documentation relating to the adversaries' tactics and techniques.

Cyber Kill Chain

Lockhead Martin originally developed the *kill chain*, a military model to identify the steps an enemy would take to attack you. It was then adapted to become the *cyber kill chain*, a framework to aid cybersecurity teams in terms of becoming more aware of potential cyber attacks (`https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html`). Refer to *Figure 12.2*.

Stages of the Cyber Kill Chain	
Reconnaissance	Calling employees, sending emails, social engineering, dumpster diving
Weaponization	Create malware payload
Delivery	Delivery medium such as USB, email, web page
Exploitation	Executing code via a vulnerability
Installation	Installing malware on the asset
Command and Control	Infected system sends back information to the attacker
Action on Objectives	'Hands-on keyboard' – attack complete

Figure 12.2 – Cyber Kill Chain

The idea behind that was to give cybersecurity teams an awareness so that they could identify and prevent attacks at an early stage. For example, we could create a security awareness program, warning employees against phishing, and also to report unusual calls from outside agencies. The attacker might then be stopped at the reconnaissance phase.

Stages in the kill chain.

The Lockheed Martin kill chain identifies the following phases:

1. Reconnaissance—in this stage the attacker determines what methods to use to complete the phases of the attack and gathers information about the target's personnel, computer systems, and supply chain.
2. Weaponization—the attacker couples payload code that will enable access with exploit code that will use a vulnerability to execute on the target system.
3. Delivery—the attacker identifies a vector by which to transmit the weaponized code to the target environment, such as via an email attachment or on a USB drive.
4. Exploitation—the weaponized code is executed on the target system by this mechanism. For example, a phishing email may trick the user into running the code, while a drive-by-download would execute on a vulnerable system without user intervention.
5. Installation—this mechanism enables the weaponized code to run a remote access tool and achieve persistence on the target system.
6. Command and control (C2 or C&C)—the weaponized code establishes an outbound channel to a remote server that can then be used to control the remote access tool and possibly download additional tools to progress the attack.
7. Actions on objectives—in this phase, the attacker typically uses the access he has achieved to covertly collect information from target systems and transfer it to a remote system (data exfiltration). An attacker may have other goals or motives, however.

The Diamond Model of Intrusion Analysis

This model is a framework for gathering intelligence on network intrusion attacks. This comprises four key elements: adversary, capabilities, infrastructure, and victims, and these are interconnected:

Figure 12.3 – Diamond Model of Intrusion Analysis

This model was used by the intelligence community until it was declassified in 2013. More information can be found at `https://apps.dtic.mil/sti/pdfs/ADA586960.pdf`. A breakdown of the preceding model follows:

- **Adversary**: This is the threat actor group, and we can use the MITRE ATT&CK model to identify who they are and what attacks they use.
- **Capabilities**: This is where the adversary develops an exploit that they use to carry out the attack. These are also laid out in the MITRE ATT&CK model.
- **Infrastructure**: This is how the attacker can get to the victim. This could be via email, IP address, or remote access.
- **Victim**: This is the person targeted by the adversary.

Example: If we look at the *Stuxnet* virus, we know the capabilities were four zero-day viruses targeting the Siemens **Industrial Control System** (**ICS**). Secondly, we know that the infrastructure used was USB sticks, and that the victim was the *Iran Nuclear Enrichment Facility*. All of this information has been discovered a piece at a time and then when we have this information, we now search for an adversary. The attack is very sophisticated. Therefore, we can narrow down the search for an adversary to someone who is well funded and capable of this sophisticated attack. You can see from this example how we can narrow down who the adversary is. Siemens, China, India, the US, and Israel were all considered. The hardest part of the diamond is to find the adversary.

We can combine the Diamond model for every step of the kill chain to detect adversaries. We can also use the MITRE ATT&CK model to find other information in the Diamond model.

Security Orchestration, Automation, and Response (SOAR)

Orchestrations are the process of running multiple automations to perform complex tasks. Automations are the process of scripting a single activity. These systems are used to collect threat-related data from multiple sources and use playbooks and runbooks. Let's look at each of these in turn:

- **Playbooks**: Playbooks contain a set of rules and actions to enable the SOAR to identify incidents and take preventative action.
- **Runbooks**: These are automated routines to automate many phases of the playbook and so can respond to different types of events.

Security orchestration, automation, and response (SOAR) is designed as a solution to the problem of the volume of alerts overwhelming analysts' ability to respond. A SOAR may be implemented as a standalone technology or integrated with a SIEM—often referred to as a next-gen SIEM. The basis of SOAR is to scan the organization's store of security and threat intelligence, analyze it using machine/deep learning techniques, and then use that data to automate and provide data enrichment for the workflows that drive incident response and threat hunting.

THREATS and MALWARE

Adware - any software application in which advertising banners are displayed while the program is running. Usually includes code that tracks a user's personal information and passes it on to third parties, without the user's authorization or knowledge.

Armored virus,– A virus that is difficult to reverse engineer.

Backdoors - installed by attackers who have compromised a system to ease their subsequent return to the system. Allows ttackers to bypass authentication. They are undocumented and usually illegal.

Boot sector virus: These viruses infect floppy and hard drives. The virus program will load first, before the operating system.

Botnets - Botnet (also known as a zombie army) is a number of Internet computers that, although their owners are unaware of it, have been set up to forward transmissions (including spam or viruses) to other computers on the Internet; Any such computer is referred to as a zombie or bot

Crypto malware – a type of ransomware that encrypts any or all files until payment has been made. Crypto malware is very difficult to remove so restoring from backups is best. Do NOT pay the fee!

File virus: Most viruses fall into this category. A virus attaches itself to a file, usually a program file.

Keylogger – a **type of spyware that captures all of a user's keystrokes.**

Logic bomb - a program, or portion of a program, which lies dormant until a specific piece of program logic is activated; The most common activator for a logic bomb is a date; A logic bomb could also be programmed to wait for a certain message from the programmer;

Macro Virus: This is a new type of virus that use an application's own macro programming feature to distribute themselves. Unlike other viruses, macro viruses do not infect programs; they infect documents.

Ransomware- Unable to open a file that is grayed-out/locked. The user receives a popup mssage that payment must be made.

Remote Access Trojan ("RAT") – A RAT is a type of trojan that gives cybercriminals complete, unlimited, and remote access to a victim's computer. Once activated, it can hide within the system for many months and remain undetected. It connects the victim's device to a command and control (C&C) server controlled by hackers. Port 137 (Netbios) is vulnerable to RATs.

Rootkits - a collection of tools (programs) that enable administrator-level access to a computer or computer network; a cracker installs a rootkit on a computer after first obtaining user-level access; Once the rootkit is installed, it allows the attacker to mask intrusion and gain root or privileged access to the computer and, possibly, other machines on the network. UEFI Secure Boot can identify a rootkit because it prevents unsigned code from executing.

Spyware - advertising companies also install additional tracking software on your system, which is continuously "calling home", using your Internet connection and reports statistical data to the "mothership"; used to monitor all kinds of activity on a computer, ranging from keystroke capture, snapshots, email logging, chat logging and just about everything else.

Trojan - a network software application designed to remain hidden on an installed computer; accesses personal information stored locally on home or business computers then send these data to a remote party via the Internet; may serve merely as a "backdoor" application, opening network ports to allow other network applications access to that computer. Trojans are also capable of launching Denial of Service (DoS) attacks.

Virus - an executable program; may cause damage of your hard disk contents, and/or interfere with the normal operation of your computer; A virus is dependent upon the host file or boot sector, and the transfer of files between computers to spread. Requires human interaction in order to spread.

Virus Hoax: Only exist in the imaginations of the public and the press - known as virus hoaxes. These viruses' hoaxes do not exist, despite rumor of their creation and distribution.

Virus Mutations:

- **Metamorphic virus- can actually rewrite its own code and thus appear different every time it is executed by creating a logical equivalent of its code whenever it is run.**
- **Oligomorphic virus –** changes its internal code to one of a set number of predefined mutations whenever it is executed
- **Polymorphic virus – completely changes from its original form whenever it is executed.**

Worms - Worms are programs that reproduce, execute independently (without user intervention) and travel across the network connections; a computer worm can execute completely independently and spread on its own accord through network connections. Ports 135, 445, 1900, and 5000 are vulnerable to worms.

ATTACKS

Application Programming Interface Attacks - Web applications and cloud services implement application program interfaces (APIs) to allow consumers to automate services. If the API isn't secure, threat actors can easily take advantage of it to compromise the services and data stored on the web application. An API must only be used over an encrypted channel (HTTPS). API calls over plain HTTP are not secure and could easily be impersonated or modified by a third party.

ARP poisoning - ARP (address resolution protocol) operates by broadcasting a message across a network, to determine the Layer 2 address (MAC address) of a host with a predefined Layer 3 address (IP address). Poisoned ARP messages contain the IP address of a default gateway, or a DNS server, and replace the MAC address for the corresponding network resource with its own MAC address. Used to perform a MITM or DoS attack. Mitigation- Mutual Authentication (Kerberos)

Birthday - A birthday attack is a type of cryptographic attack that exploits the mathematics behind the birthday problem in probability theory. This attack can be used to abuse communication between two or more parties. The attack depends on the higher likelihood of collisions found between random attack attempts and a fixed degree of permutations. The birthday paradox states that in a room with 23 people or more, the odds are greater than 50% that two will share the same birthday.

BlueBorne Attack - an attack virus that spreads through air and gets into a device via bluetooth and can then take full control of the device. The targeted device does not need to be paired to the attacker's device or even to be set on discoverable mode. If your bluetooth is on and you are in vicinity of already infected device, then the attack virus will get easily transferred to your device without asking for any permission. Thus, it needs zero human interaction and no internet connection

Bluebugging – the attacker is able to completely take over the mobile device.

Bluejacking – the attacker is able to send unauthorized text messages from the Bluetooth-enabled device.

Bluesnarfing – unauthorized access to leading to theft of information from a Bluetooth-enabled device.

Brute Force/Dictionary- password attack. In cryptography, a brute-force attack consists of an attacker submitting many passwords or passphrases with the hope of eventually guessing a combination correctly. The attacker systematically checks all possible passwords and passphrases until the correct one is found. Alternatively, the attacker can attempt to guess the key which is typically created from the password using a key derivation function. Mitigation- complex password policy; lock-out policy.

Clickjacking - tricking the user into clicking something that they hadn't intended. It uses HTML frames to mask what the user is clicking on.

Collision attack- A collision is where a function produces the same hash value for two different plaintexts. This type of attack can be used for the purpose of forging a digital signature.

Command Injection - an attack in which the goal is execution of arbitrary commands on the host operating system via a vulnerable application. Command injection attacks are possible when an application passes unsafe user supplied data (forms, cookies, HTTP headers etc.) to a system shell.

Denial-of-Service Attacks - the attacker attempts to deny authorized users access either to specific information or to the computer system or network itself; can be accomplished by crashing the system—taking it offline—or by sending so many requests that the machine is overwhelmed. DDoS attack, service is denied by overwhelming the target with traffic from many different systems. All DDoS attacks use spoofed IP addresses and the end result is always resource exhaustion.

1. **Ping of Death** – A DoS attack which involves sending a malformed ping to a computer. Sending a packet larger than 65,535 bytes violates Internet Protocol (IP), the attacker breaks down the packets into fragments, which when recombined, are greater than 65, 535. This can result in a buffer overflow which could crash the system.
2. **SYN Flood-** Disrupts the TCP handshake by sending a barrage of SYN packets resulting in a DoS. Mitigation- flood guards; NIPS
3. **ICMP Flood** – the attacker overloads the victim with a huge number of ICMP echo requests (ping).
4. **UDP Flood**- the attacker overloads the victim with a huge number of UDP packets.
5. **Land Attack**- initiates a SYN Flood attacker by spoofing the IP address of the victim. The victim ends up trying to establish a connection with itself resulting in an endless loop that persists until the timeout value is reached.
6. **Amplification Attack**- All amplification attacks exploit a disparity in bandwidth consumption between an attacker and the targeted web resource. When the disparity in cost is magnified across many requests, the resulting volume of traffic can disrupt network infrastructure. In other words, an Amplification Attack is any attack where an attacker is able to use an amplification factor to multiply its power.
    a. **Smurf attack -** A SMURF attack (named after the program used to perform the attack) is a method by which an attacker can send a moderate amount of traffic and cause a virtual explosion of traffic at the intended target. The attacker sends ICMP Echo Request packets where the source IP address has been forged to be that of the target of the attack. Mitigation- disable ICMP, Chargen; configure routers to not forward packets to broadcast addresses; flood guards; NIPS
    b. **DNS Flood attack** - A **DNS flood** is a type of distributed denial-of-service **attack** (DDoS) where an attacker **floods** a particular domain's **DNS** servers in an attempt to disrupt **DNS** resolution for that domain. If a user is unable to find the phonebook, it cannot lookup the address in order to make the call for a particular resource.

Dictionary attack - a type of Brute Force attack that tries to determine a decryption key or passphrase by trying thousands or millions of likely possibilities, such as words in a dictionary or previously used passwords, often from lists obtained from past security breaches.

Disassociation/Deauthentication Attack – A wireless attack that involves jamming the wireless access point (WAP) forcing the victim to reauthenticate enabling the attacker to sniff and capture credentials that will enable impersonating the victim **(Wireless Replay Attack).**

Domain hijacking – an attacker changes the domain name registration, typically using social engineering. The site may be held for ransom or used for malicious purposes.

DNS poisoning/Zone Transfer (Pharming) - an attacker attempts to insert a fake address record for an Internet domain into the DNS. If the server accepts the fake record, the cache is poisoned and subsequent requests for the address of the domain are answered with the address of a server controlled by the attacker. This kind of attack is often categorized as a "pharming" attack. Mitigation- DNSSec (DNS Security Extensions).

Fraggle- DoS using Echo or Chargen[1]. Mitigation- Disable Echo, Chargen; Configure routers to not forward packets to broadcast addresses; flood guards; NIPS

Golden Ticket Attack – is when the attacker has complete and unrestricted access to an entire domain — all computers, files, folders, and most importantly, the access control system itself.

Initialization Vector (IV[2]) attacks - an attack on wireless networks. The attacker modifies the IV of an encrypted wireless packet during transmission. Once an attacker learns the plaintext of one packet, the attacker can compute the RC4 key stream generated by the IV used**.**

IP Address Spoofing - IP is designed to work so that the originators of any IP packet include their own IP address in the From portion of the packet; nothing prevents a system from inserting a different address in the From portion of the packet. This is known as IP address spoofing.

Juice Jacking - Jacking is the process of hacking into smart devices by way of a USB port that acts as a conduit for power supply and data transfer. Fraudsters illegally obtain access to a smart device and its data, share screens, and inject harmful malware into the device via this process. In layman's terms – juice jacking is the process of hacking into a smartphone or other smart devices via a compromised USB charging port. This could happen when you are charging your smartphone at a public charging booth located at airports, hospitals, coffee shops, or any of several public places.

KRACK attack - uses a replay mechanism that targets the 4-way handshake. KRACK is effective regardless of whether the authentication mechanism is personal or enterprise. It is important to ensure both clients and access points are fully patched against such attacks.

LDAP injection- Directory attack that uses code like BIND, SEARCH, MODIFY, CN, DC, OU (AD uses o, c). Mitigation- Input Validation; WAF, stored procedures. Look for a lot of "&" signs and parentheses.

MAC Flooding- Sending packets to all ports on a switch. Overwhelming the switch causes it to behave like a HUB. Mitigation- VLAN.

Man-in-the-Browser – *takes place at the Application Layer of the OSI model because it uses the functions and features of the browser.* This attack intercepts browsing data using a trojan (malicious add-in). Mitigation: TLS and authentication certificates.

[1] Chargen - character generator; a service of the IP protocol suite intended for testing, debugging, & measurement purposes. It can be abused.

[2] The **Initialization Vector (IV)** is an unpredictable random number used to make sure that when the same message is encrypted twice, the ciphertext always different. It should be exchanged, in public, as part of the ciphertext.

Man-in-the-middle – *takes place at the network layer of the OSI model.* It occurs when attackers are able to place themselves in the middle of two other hosts that are communicating; attacker can then observe all traffic before relaying it and can actually modify or block traffic. Mitigation- Mutual authentication (Kerberos).

Near-field (NFC) Communication Attack – Using an NFC reader to capture data. NFC does not provide encryption, so eavesdropping and man-in-the-middle attacks are possible if the attacker can find some way of intercepting the communication and the software services are not encrypting the data.

Pass-the-Hash Attack - Exploiting cached credentials to perform lateral movement. It is a technique whereby an attacker captures a password hash (as opposed to the password characters) and then simply passes it through for authentication and potentially lateral access to other networked systems. The threat actor doesn't need to decrypt the hash to obtain a plain text password.

Password Spraying – a type of horizontal brute force online[3] attack where the attacker chooses one or more common passwords (for example, password or 123456) and tries them in conjunction with multiple usernames. Mitigation: The best defence against password crackers is to ensure the use of strong passwords (and not to use clear-text protocols, of course). You must also restrict access to password databases carefully to try to prevent any sort of eavesdropper from running on your networks.

Pharming- a cyberattack intended to redirect a website's traffic to another, fake site. Pharming can be conducted by changing the hosts file on the victim's computer. Pharming requires unprotected access to a target computer. The computer hosts file is an operating system file that maps host names to IP addresses. It is a plain text file. It is a common part of an operating system's IP implementation. Mitigation: Group Policy Objects (GPO) and Intrusion Prevention Systems (IPS).

Phishing - Phishing (pronounced —fishing) is a type of social engineering in which an individual attempts to obtain sensitive information from a user by masquerading as a trusted entity in an e-mail or instant message sent to the user.

Poodle Attack- A Poodle Attack is a man-in-the-middle attack that takes advantage of Internet and security software clients' fall-back to SSl 3.0. (Padded Oracle on Downgrade on Legacy encryption.) On average, they only need to make 256 SSL 3.0 requests to reveal one byte encrypted messages. It is a protocol downgrade that allows exploits on an outdated for of encryption.

Rainbow Table- An attack that uses a table of plaintext permutations of encrypted passwords specific to a hash algorithm. Mitigation- Salting; key stretching; use of complex passwords

Replay Attacks/Session Hijacking - A replay attack occurs when the attacker captures a portion of a communication between two parties and retransmits it at a later time, often with changes. Mitigation- Timestamps, sequence numbers, TTL (Time-to-Live)

Session Hijacking – bypassing a user's or computer's browser privacy settings and then impersonating the user by using their session ID, which is stored in a cookie.

SQL Injection- Database attack that uses code like, "OR," "1=1," "SELECT*FROM," "union all select." A log file might also indicate an OWASP code of A1, which signifies an injection attack. Mitigation- Input validation; Stored Procedures; WAF.

---

[3] An ***online*** *password attack* is where the threat actor interacts with the authentication service directly—a web login form or VPN gateway, for instance. The attacker submits passwords using either a database of known passwords (and variations) or a list of passwords that have been cracked offline. *Also, be aware that there are databases of username and password/password hash combinations for multiple accounts stored across the Internet. These details derive from successful hacks of various companies' systems. These databases can be searched using a site such as haveibeenpwned.com.* An ***offline*** *attack* means that the attacker has managed to obtain a database of password hashes, such as %SystemRoot%\System32\config\SAM, %SystemRoot%\NTDS\NTDS.DIT (the Active Directory credential store), or /etc/shadow. Once the password database has been obtained, the password cracker does not interact with the authentication system. The only indicator of this type of attack (other than misuse of the account in the event of a successful attack) is a file system audit log that records the malicious account accessing one of these files. Threat actors can also read credentials from host memory, in which case the only reliable indicator might be the presence of attack tools on a host.

Teardrop- DoS using MSB. Mitigation- Disable MSB; Configure routers to not forward packets to broadcast addresses; flood guards; NIPS

URL Hijacking (Typo squatting) – registering a domain name similar to a legitimate site and waiting for a user to misspell a domain name. Often used is drive-by downloads[4] to install malware on the user's computer.

Vishing - a variation of phishing that uses voice communication technology to obtain the information the attacker is seeking.

XMAS- A port scan that reveals details of the OS and open ports; uses PSH, URG and FIN flags. Mitigation- IDS AND IPS

XML Injection- uses code like, <name>John Doe</name>. Mitigation- Input validation; WAF.

XSRF/CSRF- (Cross-site Request Forgery) steals cookies, makes purchases, and harvests passwords. Whereas in XSS the user is not required to be logged into a secure host, for XSRF, the user MUST be logged in to a secure host. because this attack attempts to take advantage of the user's active session. Mitigation- Mutual authentication; expire the cookie; uncheck "Remember Me."

SSRF - server-side request forgery (SSRF) causes the server application to process an arbitrary request that targets another service, either on the same host or a different one. SSRF attacks are often targeted against cloud infrastructure where the web server is only the public-facing component of a deeper processing chain.

XSS/Session Hijacking- Injecting malicious code into another user's browser. Client-side injected scripts using HTML or JavaScript; uses characters like: <and>. Stored XXS is when the script is stored on the server and executed by a browser. These scripts often make their way to the server by means of a message posting. Mitigation- Input validation on the server side; escaping; WAF.

1. **DOM Based XSS** (or as it is called in some texts, "type-0 **XSS**") is an **XSS** attack wherein the attack payload is executed as a result of modifying the **DOM** "environment" in the victim's browser used by the original client side script, so that the client side code runs in an "unexpected" manner.
2. **Stored XSS - Stored** cross-site scripting (also known as second-order or persistent **XSS**) arises when an application receives data from an untrusted source and includes that data within its later HTTP responses in an unsafe way.
3. **Reflect XSS** - Reflected cross-site scripting (or **XSS**) arises when an application receives data in an HTTP request and includes that data within the immediate response in an unsafe way.

VULNERABILITIES

Buffer Overflow- AKA "Buffer Overrun," is an anomaly where a program, while writing data to a buffer, overruns the buffers boundary and overwrites adjacent memory. . In an overflow attack, the threat actor deliberately submits input that is too large to be stored in a variable assigned by the application. Mitigation- Input Validation

DLL Injection – DLLs are libraries used by applications. Attackers put their own libraries in place so that when the application references the library, they are effectively referencing the bad guys' code.

[4] A **drive-by download** attack refers to the unintentional **download** of malicious code to your computer or mobile device that leaves you open to a cyberattack. You don't have to click on anything, press **download**, or open a malicious email attachment to become infected.

Embedded Systems vulnerabilities- These devices are very often connected to the internet, which makes it really convenient if somebody wants to gain access to these systems. Usually, they're running an outdated version of software, because embedded systems aren't usually upgraded. The system is self-contained. It's one that rarely changes, and therefore upgrades aren't always necessary. But of course, if these embedded systems aren't updated with the latest security patches, they could be vulnerable to new exploits.

End-of-Life vulnerability- This is when a device or a component or a piece of software is no longer under support from the vendor. And usually when this happens, the vendor stops providing any type of security patches.

Integer Overflow- Same as Buffer Overflow, but integers only. Mitigation- Input Validation.

Memory Leak – This occurs when an application fails to properly release memory allocated to it or continuously requests more memory than it needs.

Null Pointer Dereference - A NULL pointer points to memory that doesn't exist.
A NULL pointer dereference exception occurs when an application dereferences an object that is expected to be valid but is NULL (missing data array).

Race Conditions- Where two or more modules use a resource at the same time. A race condition is a coding problem. And that's because on the systems we use these days, there are usually multiple users performing multiple functions all at the same time. And if your coding has not taken into account that these multiple things could happen simultaneously, you will run into a race condition. Mitigation- Code review!

Runtime Errors- An error that occurs during the execution of a program. Mitigation- Exception handling; trapping; logs

Switching Loops/Trunking Loops- Creating Loops on a switch which causes a network to crash. Mitigation- Implement Spanning-tree Protocol (STP); 802.1x for authentication; Rapid STP (RSTP); Time-to-Live (TTL).

Zero-day Vulnerability – a vulnerability that is unknown by the vendor. A zero-day vulnerability is considered to be "zero day" until a patch is released, however long that takes.

Vulnerability Scanners

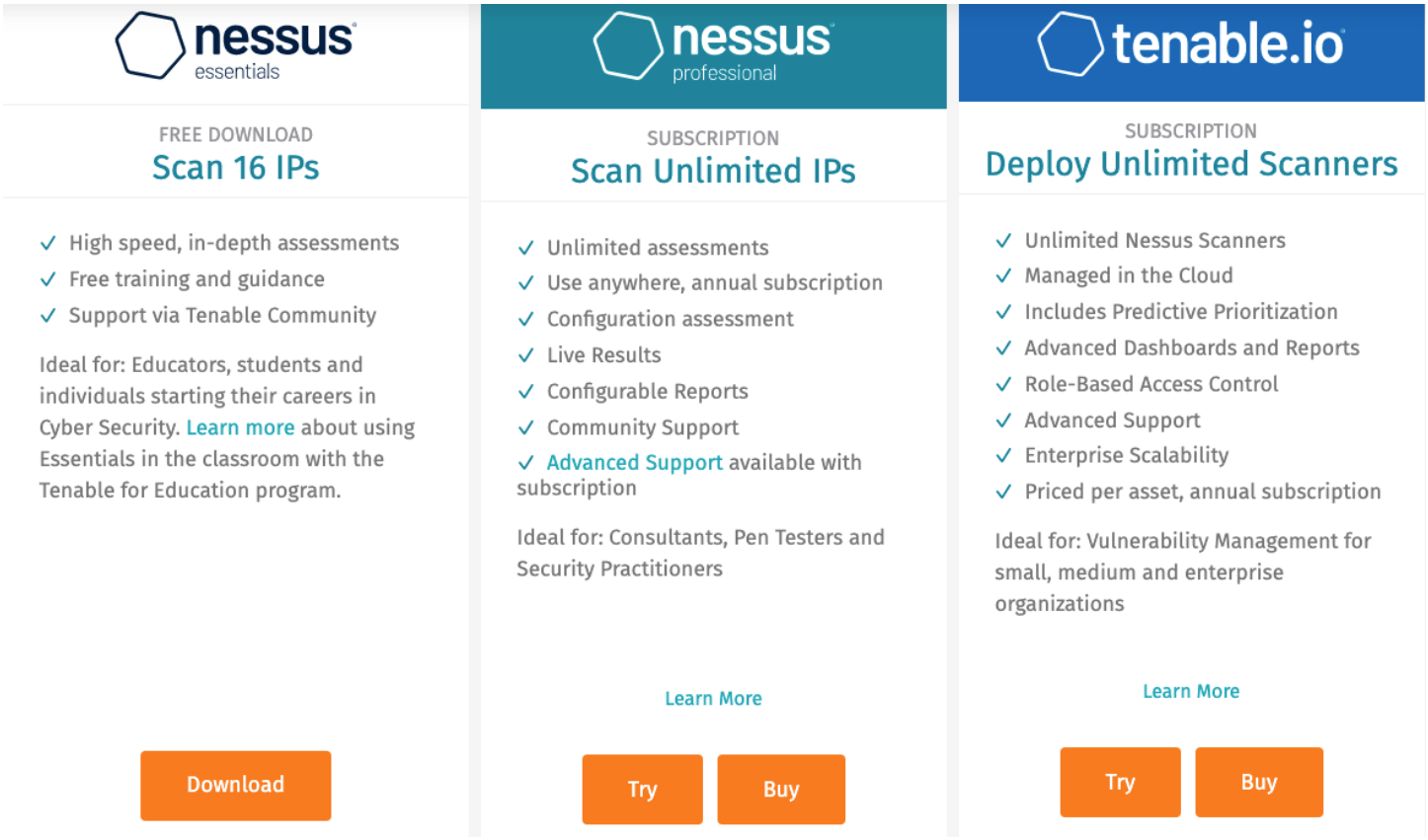
- Passive and do not exploit vulnerabilities

- Nmap performs:
  - Performs scan through target enumeration also known as host specifiers
    - Port Scan
    - Service Scan
    - OS or software scan
    - Host/server discovery

- Protocol Analyzer
  - Protocol scanner
  - Packet Analyzer
  - Packet Sniffer
  - Sniffer

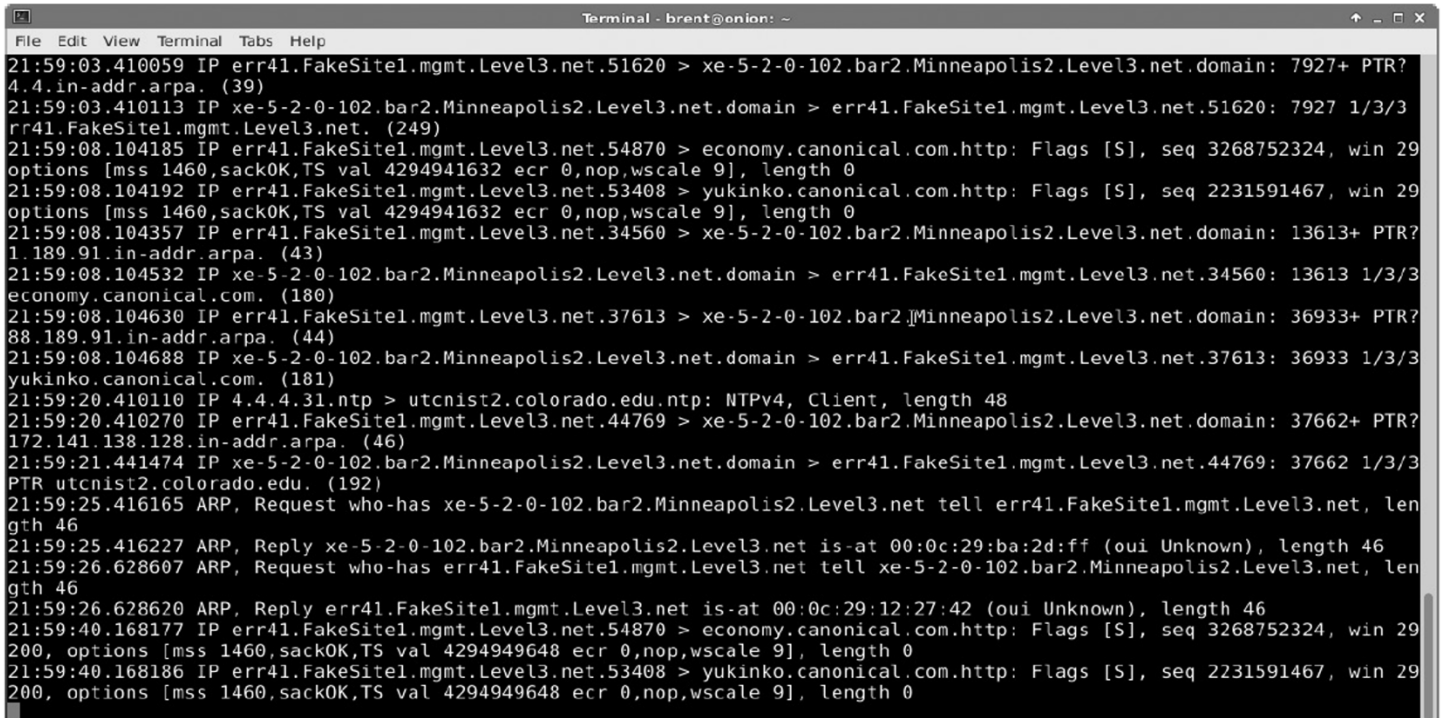
→ All the same thing

- Protocol Analyzers
  - Performs scan by examining data packets that flow between computers on a network as well as between networked computers and the larger internet
    - Port Scan
    - Service Scan
    - OS or software scan
    - Host/server detection
  - TCPdump
    - Command-line packet sniffer
  - WireShark
    - GUI packet sniffer



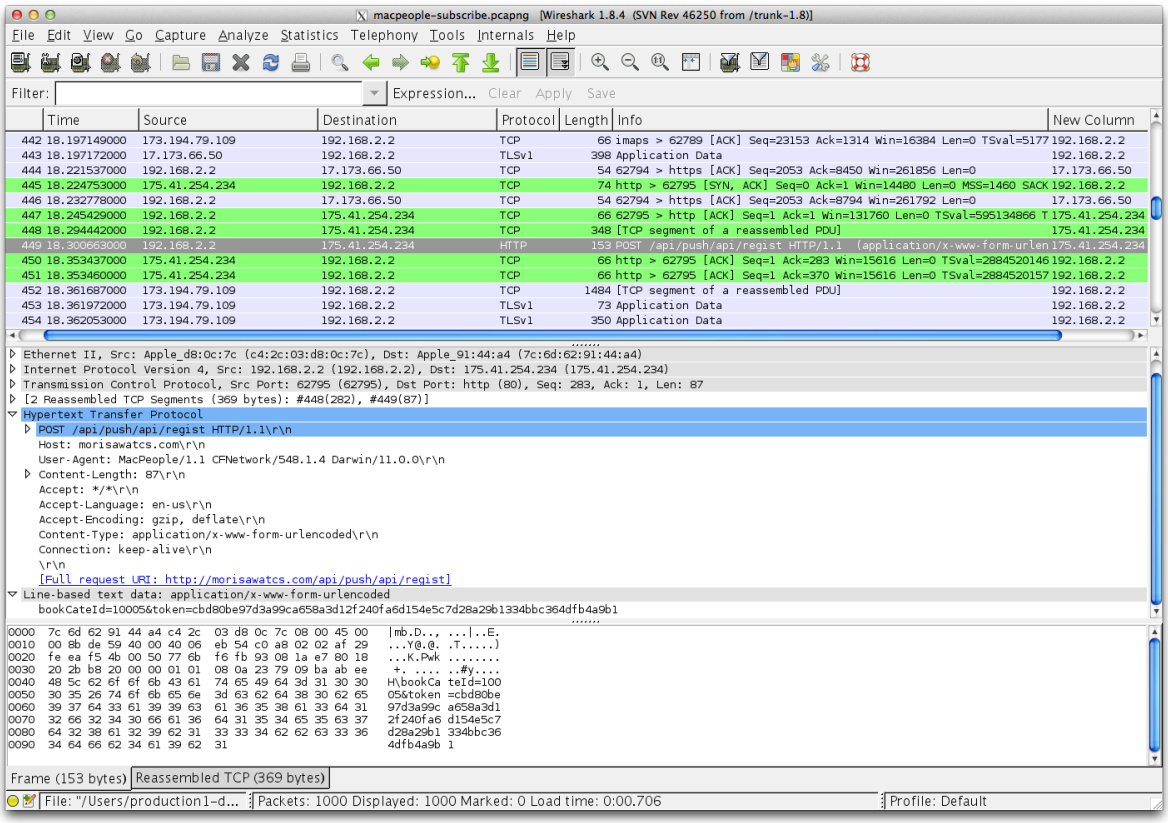
https://www.tenable.com/products/nessus

tcpdump



https://www.tcpdump.org/

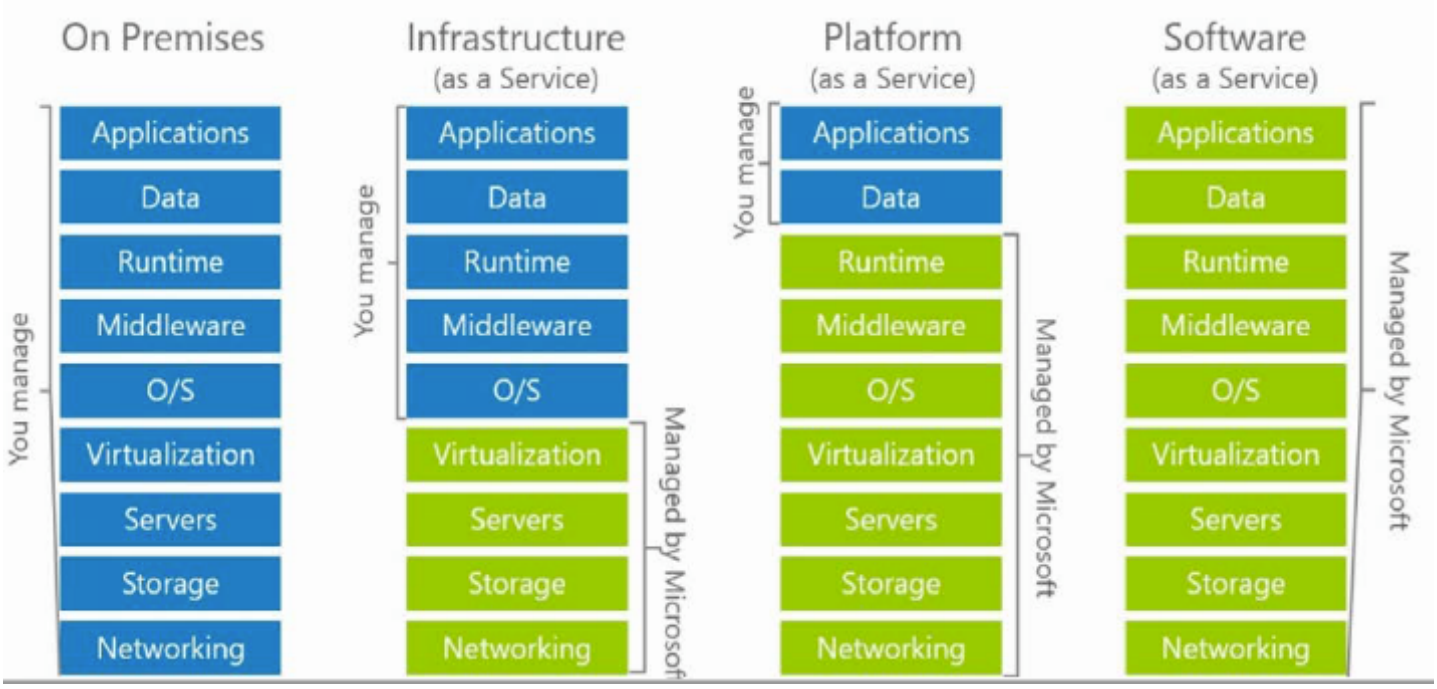
Wireshark

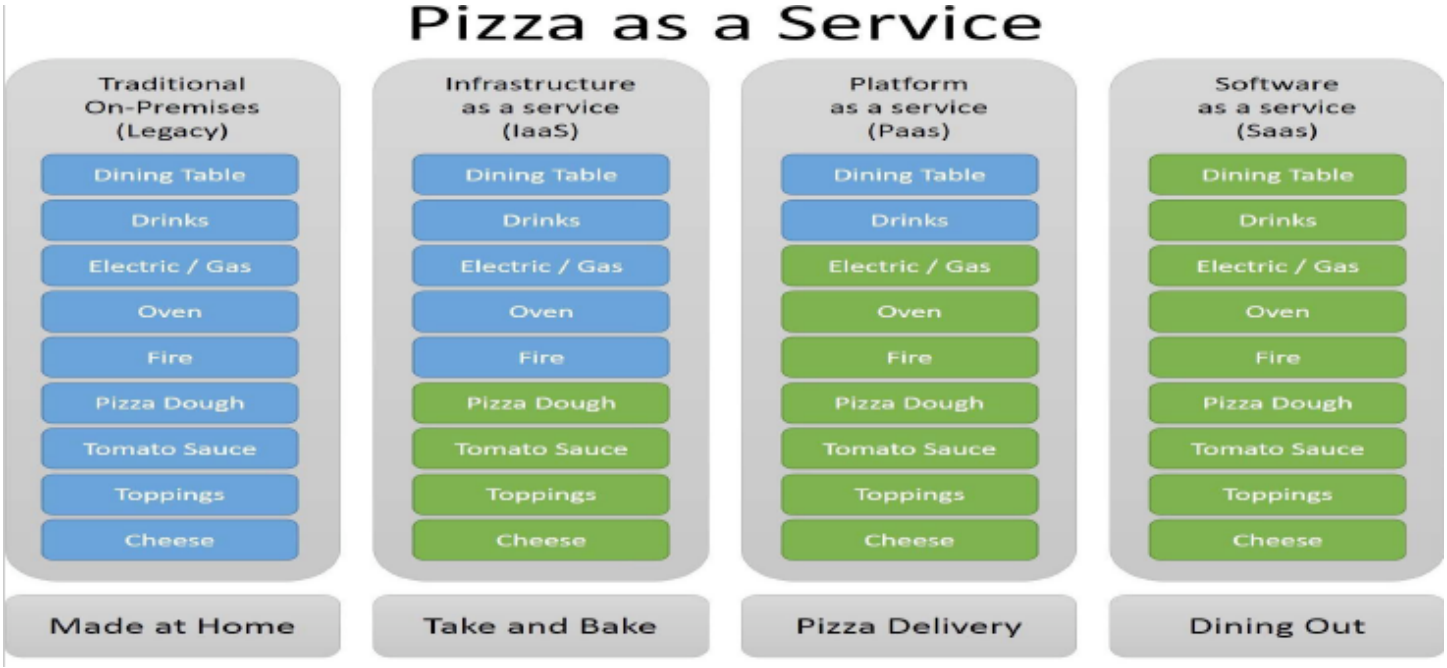


https://www.wireshark.org/

Acrylic (Wi-Fi)

https://www.acrylicwifi.com/en/downloads-free-license-wifi-wireless-network-software-tools/download-wifi-scanner-windows/


On Premises
Infrastructure
(as a Service)
Platform
(as a Service)
Software
(as a Service)
You manage
Managed by Microsoft
Applications
Data
Runtime
Middleware
O/S
Virtualization
Servers
Storage
Networking


Pizza as a Service
Traditional
On-Premises
(Legacy)
Infrastructure
as a service
(IaaS)
Platform
as a service
(Paas)
Software
as a service
(Saas)
Dining Table
Drinks
Electric / Gas
Oven
Fire
Pizza Dough
Tomato Sauce
Toppings
Cheese
Made at Home
Take and Bake
Pizza Delivery
Dining Out

Cloud Deployment Models

Public	• Third-party solution open to everyone • Can be paid or free
Private	• Only accessible only to a single organization • May be on premises or off (Virtual Private Cloud)*
Community clouds	• Private or Public cloud shared between several organizations which have common concerns and needs**
Hybrid	• Combination of models • A company uses a private cloud FTP solution while also utilizing a public cloud service to provide file backup capabilities.

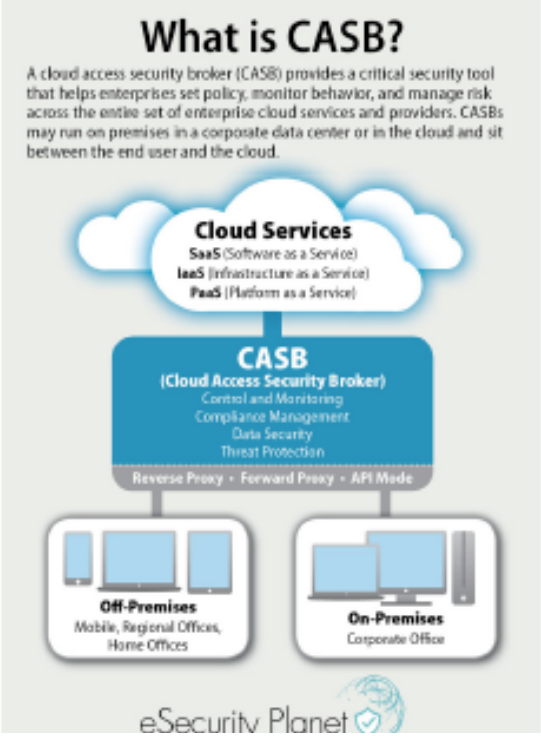
- Hosted Private—hosted by a third-party for the exclusive use of the organization. This is more secure and can guarantee a better level of performance but is correspondingly more expensive.
- **Private**—cloud infrastructure that is completely private to and owned by the organization. In this case, there is likely to be one business unit dedicated to managing the cloud while other business units make use of it. With private **cloud computing**, organizations can exercise greater control over the privacy and security of their services. This type of delivery method is geared more toward banking and governmental services that require strict access control in their operations.

  This type of cloud could be on-premise or offsite relative to the other business units. An onsite link can obviously deliver better performance and is less likely to be subject to outages (loss of an Internet link, for instance). On the other hand, a dedicated offsite facility may provide better shared access for multiple users in different locations.

Exam tip

Private cloud = single tenant
Public cloud = multitenant
Community cloud = same industry, and sharing resources

- **Community Cloud**: The community cloud is where companies from the same industry collectively pay for a bespoke application to be written, and the cloud provider manufacturers host it:



Overview of Cloud Computing

The demand for cloud computing has risen over the last few years as the workforce has been more mobile; the cloud solution is very cost-effective and maintains the high availability of systems. Before you decide to move to a **Cloud Service Provider** (**CSP**), you need to ensure that you trust them 100%.

There are many good reasons why cloud computing has become popular:

- **Elasticity**: The cloud is like a pay-as-you-go model where one day you can increase resources and then the next day you can scale down the resources. You can even add more processor power, faster disks, more memory, or dual network cards whenever you want – there's no need to wait for delivery times, but the cost increases:

  Example 1: A toy firm is hiring 50 temporary workers from October onward to deal with the rush for toys at Christmas. If the toy company were not on the cloud, they would have to purchase another 50 desktops, but instead, they lease **Virtual Machines** (**VMs**) from a CSP. Once the Christmas rush has ended, the lease of their machines ends. You only pay for what you need.

- **Scalability**: Scalability is the ability of a company to grow while maintaining a resilient infrastructure. The cloud enables a company to do so and grow without the worry of needing to make capital expenditure while doing so. It enables the company to grow faster than an on-premises company that needs to invest more money into bricks and mortar. As the cloud allows elasticity, it goes hand in hand with becoming scalable. As your company grows, the cloud provider can allow you to lease more resources. If at any time you want to reduce the amount of resources needed, the cloud provider can do that too.

- **No Capital Expenditure** (**CAPEX**): When you move your infrastructure to the cloud, there is no capital expenditure; normally, IT resources have a maximum lifespan of 3–5 years. As technology keeps moving and hardware becomes obsolete, this means they may have to find $75–300,000 every five years just for hardware.

- **Location-Independent**: As you are accessing the cloud through a browser, it is location-independent, therefore it offers faster recovery if your premises have a disaster.

  Example: One of your company offices is located in Northern California and recently was burned down by a wildfire; however, since your data and infrastructure are cloud-based, you can operate quickly from another location as long as you have internet access. If you had a traditional network, the infrastructure would have been burned down, your desktops would have been gone, and it could take a week or two to get back to an operational state.

- **Regional Storage of Data**: The cloud is regulated, therefore data from a country must be stored within that region as laws on data compliance can change from region to region.
- **No Maintenance Fees**: The CSP provides ongoing maintenance, so when the cloud contract is signed there are no hidden costs.
- **No Disaster Recovery Site Required**: The CSP provides 99.999% availability of its IT systems, therefore, once your data is in the cloud, there is no requirement for a disaster recovery site as the CSP provides that as part of the contract.

Infrastructure as a Service (IaaS)

If you think of a network infrastructure, you think of desktops, servers, firewalls, routers, and switches – the hardware devices for a network. When you purchase these devices, they have a default factory setting and these settings need to be configured. Desktops are bare-bones, meaning that they have no operating system installed. IaaS is the same; you need to preconfigure these devices, install an operating system, and maintain the patch management. See the pricing (as of writing this book) for IaaS in the screenshot that follows:

Microsoft's IaaS offering (July 2018)

Software as a Service (SaaS)

This is where the CSP hosts a bespoke software application that is accessed through a web server. Let's look at three examples of this: Goldmine, Salesforce, and Office 365.

Example 1: GoldMine is a SaaS package, that is, a **Customer Relationship Management** (**CRM**) package, which is used by companies that sell products and services. It will host lists of their customers, with contact numbers and addresses:

> **Exam tip**
>
> SaaS is a bespoke vendor application that cannot be modified and you use it with a pay-per-use model, as a subscription, and you cannot migrate any applications or services to any SaaS environment.

Platform as a Service (PaaS)

This provides the environment for developers to create applications; an example of this is Microsoft Azure. The platform provides a set of services to support the development and operation of applications, rolling them out to iOS, Android devices, as well as Windows devices. You could migrate your bespoke software applications under PaaS. Bespoke means customized.

Security as a Service (SECaaS)

SECaaS provides **Identity and Access Management** (**IAM**), which provides identity management that allows people to have secure access to applications from anywhere at any time. The following screenshot shows Okta providing secure web authentication into Google Apps:

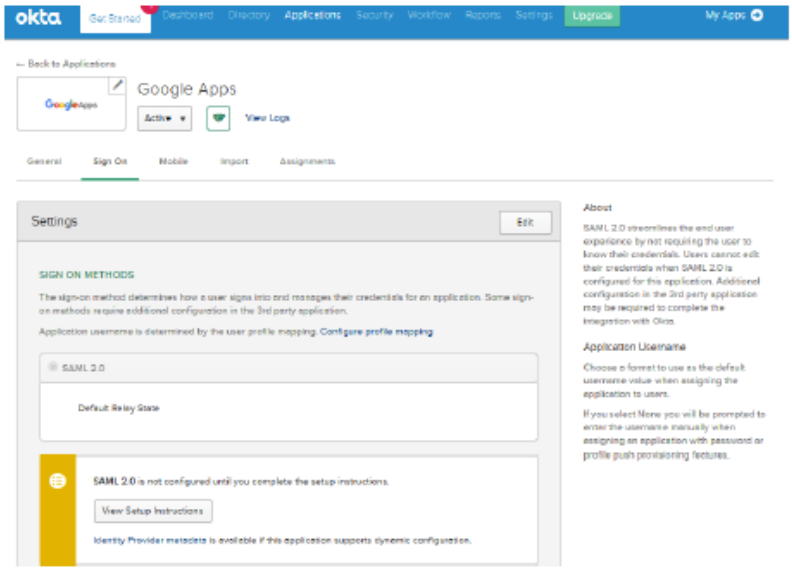


Figure 4.9 – Okta security as a service (SECaaS) for Google Apps

The user in the preceding screenshot needs to validate their identity and has presented a SAML token from Okta, the **identity provider** (**IdP**).

The breadth of technologies requiring specialist security knowledge and configuration makes it likely that companies will need to depend on third-party support at some point. You can classify such support in three general "tiers":

- Consultants—the experience and perspective of a third-party professional can be hugely useful in improving security awareness and capabilities in any type of organization (small to large). Consultants could be used for "big picture" framework analysis and alignment or for more specific or product-focused projects (pen testing, SIEM rollout, and so on). It is also fairly simple to control costs when using consultants if they are used to develop capabilities rather than implement them. Where consultants come to "own" the security function, it can be difficult to change or sever the relationship.

- **Managed Security Services Provider (MSSP)**—a means of fully outsourcing responsibility for information assurance to a third party. This type of solution is expensive but can be a good fit for an SME that has experienced rapid growth and has no in-house security capability. Of course, this type of outsourcing places a huge amount of trust in the MSSP. Maintaining effective oversight of the MSSP requires a good degree of internal security awareness and expertise. There could also be significant challenges in industries exposed to high degrees of regulation in terms of information processing.

- **Security as a Service (SECaaS)**—can mean lots of different things, but is typically distinguished from an MSSP as being a means of implementing a particular security control, such as virus scanning or SIEM-like functionality, in the cloud. Typically, there would be a connector to the cloud service installed locally. For example, an antivirus agent would scan files locally but be managed and updated from the cloud provider; similarly a log collector would submit events to the cloud service for aggregation and correlation. Examples include Cloudflare (cloudflare.com/saas), Mandiant/FireEye (fireeye.com/mandiant/managed-detection-and-response.html), and SonicWall (sonicwall.com/solutions/service-provider/security-as-a-service).

Anything as a Service (XaaS)

Anything as a Service (**XaaS**) describes a multitude of other cloud services that are available, such as **Network as a Service** (**NaaS**), providing network resources; **Desktop as a Service** (**DaaS**); **Backup as a Service** (**BaaS**); and many more. As new services appear, they will fall under the category of XaaS.

There are many other examples of XaaS, reflecting the idea that anything can be provisioned as a cloud service. For example, database as a service and network as a service can be distinguished as more specific types of platform as a service. The key security consideration with all these models is identifying where responsibilities lie. This is often referred to as security in the cloud versus security of the cloud. Security in the cloud is the things you must take responsibility for; security of the cloud is the things the CSP manages. These responsibilities vary according to the service type:

Responsibility	IaaS	PaaS	SaaS
IAM	You	You	You (using CSP toolset)
Data security (CIA attributes/backup)	You	You	You/CSP/Both
Data privacy	You/CSP/Both	You/CSP/Both	You/CSP/Both
Application code/ configuration	You	You	CSP
Virtual network/firewall	You	You/CSP	CSP
Middleware (database) code/configuration	You	CSP	CSP
Virtual Guest OS	You	CSP	CSP
Virtualization layer	CSP	CSP	CSP
Hardware layer (compute, storage, networking)	CSP	CSP	CSP

In this section, we are going to look at different cloud computing concepts that may appear in the CompTIA Security+ exam. Make sure that you are familiar with them:

- **Cloud Service Provider** (**CSP**): CSPs are entities that resell cloud services to customers. They can provide infrastructure, software, VMs, and other services that a customer needs. **Managed Cloud Service Providers** (**MCSP**) will also take over the day-to-day running of your cloud as they have the expertise to do so.
- **Managed Security Service Provider** (**MSSP**): An MSSP will maintain the security environment for companies that will include enterprise firewalls, intrusion prevention and detection systems, and SIEM systems. They have a very highly skilled workforce who will take this headache away from a company. At `https://wizardcyber.com/blog/managed-security-service-provider/` is an article about choosing an MSSP.
- **Fog Computing**: Fog computing complements cloud computing by processing data from IoT devices. It allows you to analyze the data before committing it to the cloud. The data is put in a location between the device and the cloud. It brings cloud computing nearer to the sensor; it also reduces the cost of data moving back and forth between the device and the cloud. We can use 4G/5G, Wi-Fi, or Zigbee. It also reduces latency as it prioritizes the traffic, and this can be important for life support systems. This can help healthcare applications process data more quickly as it is much closer to the device, and they are not transmitting data back and forth between the device and cloud.

Example: An alert from the sensor of a life support system is sent to the cloud and then to the clinician, but with fog computing, which is closer to the sensor, it can reduce the latency as the clinician is alerted much more quickly:

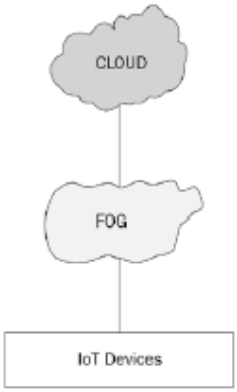


Figure 4.10 – Fog computing

- **Edge Computing**: All the processing of data storage is closer to the sensors rather than being thousands of miles away on a server at a data center.
- **Thin Client**: A thin client is a client that has limited resources that are insufficient to run applications. It connects to a server and processes the application on its resources.
- **Containers**: A container allows the isolation of an application and its files and libraries so that they are not dependent on anything else. It allows software developers to deploy applications seamlessly across various environments. Containers are used by **Platform as a Service** (**PaaS**) products.

  Example: Microsoft's version of Docker runs on Linux but allows application containers on Linux, Windows, and "macOS".

- **Infrastructure as Code**: This is where you manage your computer infrastructure with configuration files rather than by a physical method. This is very common with cloud technologies making it easier to set up computers and roll out patches. This ensures that each computer has the same setup, in contrast with the human errors that may be encountered when setting up a computer manually. An example would be setting up your infrastructure using PowerShell and using **Desired State Configuration** (**DSC**) to ensure that there is no deviation from the required setting. You can use PowerShell scripts to create VMs, firewalls, and load balancers. Let's look at a few more examples:

  a. **Software-Defined Network (SDN)**: Traditional networks route packets via a hardware router and are decentralized; however, in today's networks, more and more people are using virtualization, including cloud providers. A SDN is where packets are routed through a controller rather than traditional routers, which improves performance. It has three different planes: the control plane prioritizes the traffic, the data plane does switching and routing, and the management plane deals with monitoring the traffic. An overview of SDN can be found at `https://www.cisco.com/c/en_au/solutions/software-defined-networking/overview.html`.

  b. **Software-Defined Visibility (SDV)**: This gives you visibility of the network traffic use. It can collect and aggregate the data on the network traffic and provide good reports to the network administrators.

- **Serverless Architecture**: This is where you will use the Backend as a Service, where a third-party vendor hosts your applications as a pay-as-you-go model based on the compute time that you use. You will lease servers or data storage from them.
- **Services Integration**: This is where the provision of several business services is combined with different IT services and are integrated to provide a single solution for a business.

Network Topology and Zones

Given the ability to create segregated segments with the network, you can begin to define a topology of different network zones. A topology is a description of how a computer network is physically or logically organized. The logical and physical network topology should be analyzed to identify points of vulnerability and to ensure that the goals of confidentiality, integrity, and availability are met by the design.

The main building block of a security topology is the zone. A **zone** is an area of the network where the security configuration is the same for all hosts within it. Zones should be segregated from one another by physical and/or logical segmentation, using VLANs and subnets. Traffic between zones should be strictly controlled using a security device, typically a firewall.

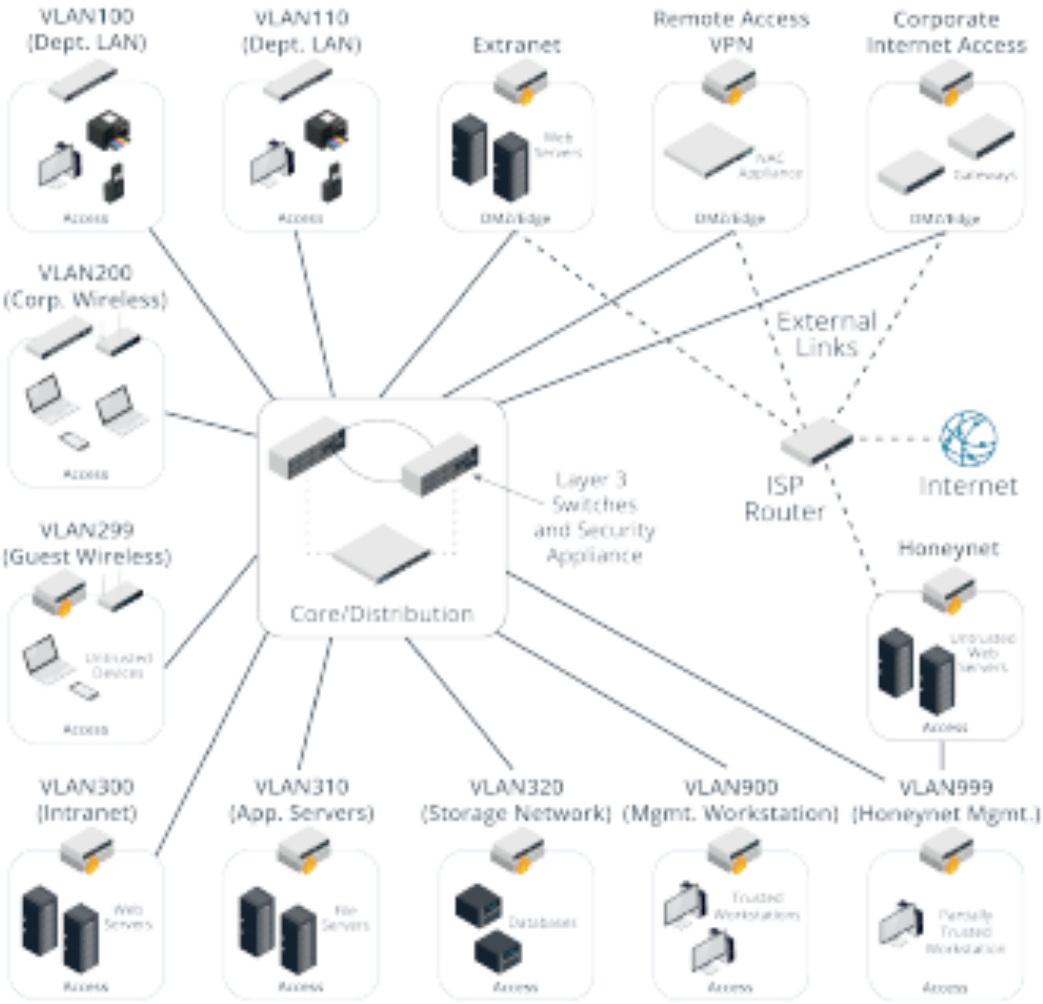
Dividing a campus network or data center into zones implies that each zone has a different security configuration. The main zones are as follows:

- **Intranet (private network)**—this is a network of trusted hosts owned and controlled by the organization. Within the intranet, there may be sub-zones for different host groups, such as servers, employee workstations, VoIP handsets, and management workstations.

Hosts are trusted in the sense that they are under your administrative control and subject to the security mechanisms (anti-virus software, user rights, software updating, and so on) that you have set up to defend the network.

- Extranet—this is a network of semi-trusted hosts, typically representing business partners, suppliers, or customers. Hosts must authenticate to join the extranet.
- Internet/guest—this is a zone permitting anonymous access (or perhaps a mix of anonymous and authenticated access) by untrusted hosts over the Internet.

A large network may need more zones to represent different host groups, such as separating wireless stations from desktop workstations, and putting servers in their own groups. Cisco's enterprise security architecture uses core and distribution layers to interconnect access blocks, with each access block representing a different zone and business function.



Enterprise security architecture. (Images © 123RF.com.)

Demilitarized Zones

The most important distinction between different security zones is whether a host is Internet-facing. An Internet-facing host accepts inbound connections from and makes connections to hosts on the Internet. Internet-facing hosts are placed in one or more **demilitarized zones (DMZs)**. A DMZ is also referred to as a perimeter or edge network. The basic principle of a DMZ is that traffic cannot pass directly through it. A DMZ enables external clients to access data on private systems, such as web servers, without compromising the security of the internal network as a whole. If communication is required between hosts on either side of a DMZ, a host within the DMZ acts as a proxy. For example, if an intranet host requests a connection with a web server on the Internet, a proxy in the DMZ takes the request and checks it. If the request is valid, it retransmits it to the destination. External hosts have no idea about what (if anything) is behind the DMZ.

Both **extranet** and Internet services are likely to be Internet-facing. The hosts that provide the extranet or public access services should be placed in one or more demilitarized zones. These would typically include web servers, mail and other communications servers, proxy servers, and remote access servers. The hosts in a DMZ are not fully trusted by the internal network because of the possibility that they could be compromised from the Internet. They are referred to as **bastion hosts** and run minimal services to reduce the attack surface as much as possible. A bastion host would not be configured with any data that could be a security risk to the internal network, such as user account credentials.

It is quite likely that more than one DMZ will be required as the services that run in them may have different security requirements:

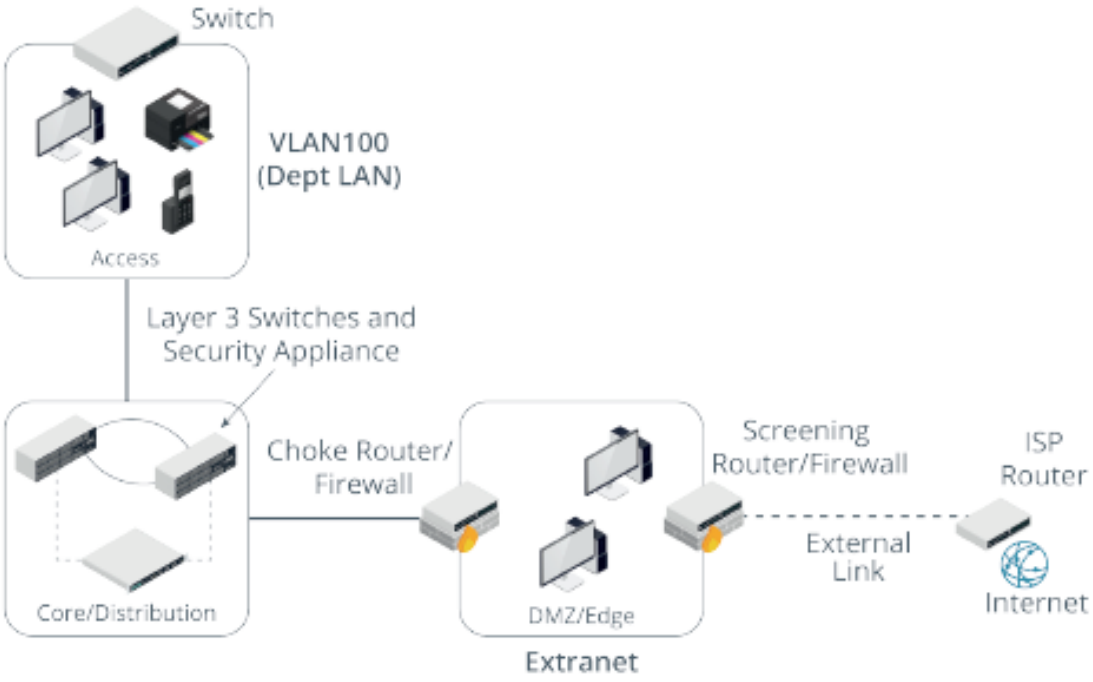
- A DMZ hosting proxies or secure web gateways to allow employees access to web browsing and other Internet services.
- A DMZ hosting communication servers, such as email, VoIP, and conferencing.
- A DMZ for servers providing remote access to the local network via a Virtual Private Network (VPN).
- A DMZ hosting traffic for authorized cloud applications.
- A multi-tier DMZ to isolate front-end, middleware, and backend servers.

Demilitarized Zone Topologies

To configure a DMZ, two different security configurations must be enabled: one on the external interface and one on the internal interface. A DMZ and intranet are on different subnets, so communications between them need to be routed.

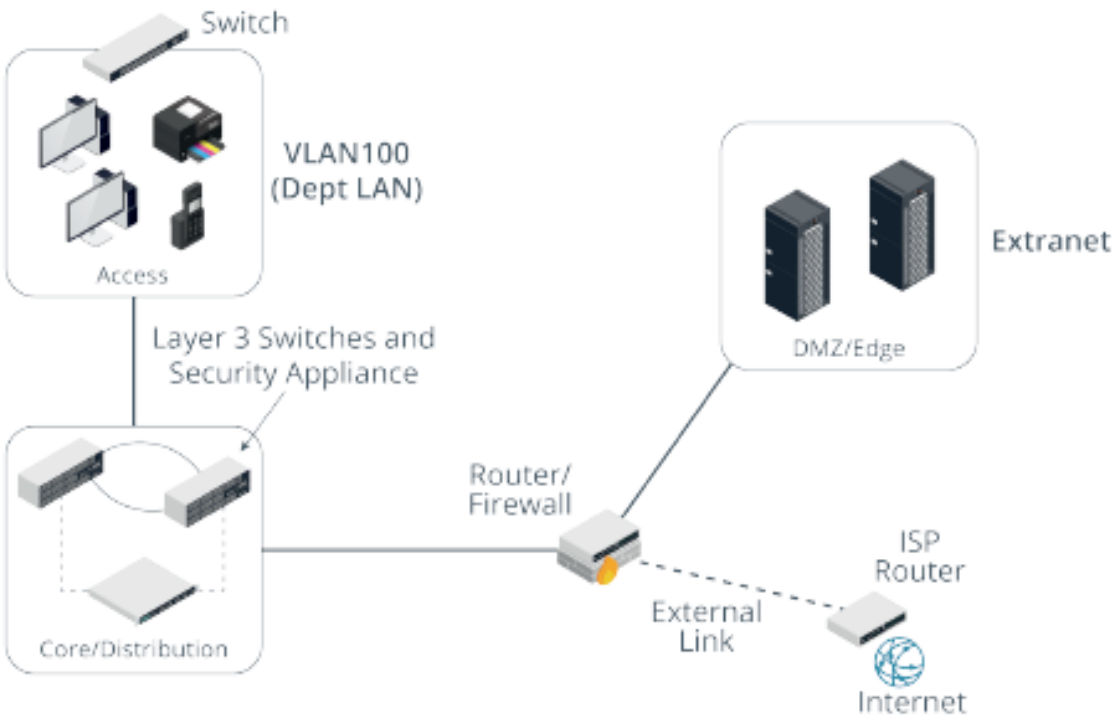
Screened Subnet

A screened subnet uses two firewalls placed on either side of the DMZ. The edge firewall restricts traffic on the external/public interface and allows permitted traffic to the hosts in the DMZ. The edge firewall can be referred to as the screening firewall or router. The internal firewall filters communications between hosts in the DMZ and hosts on the LAN. This firewall is often described as the choke firewall. A choke point is a purposefully narrow gateway that facilitates better access control and easier monitoring.



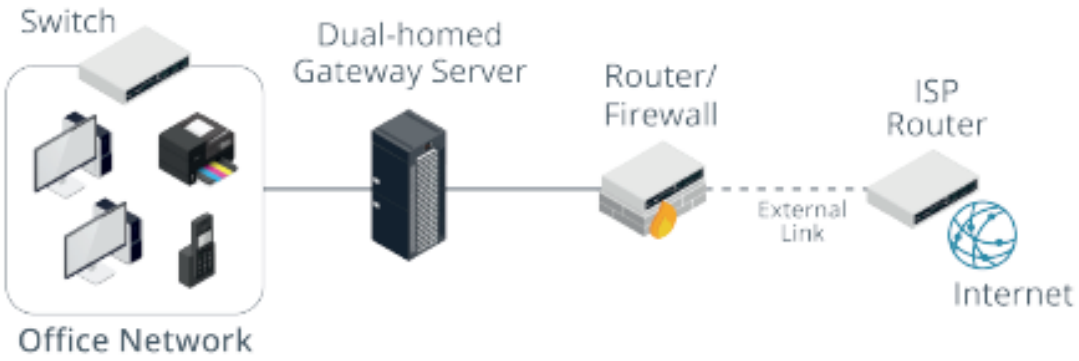
Triple-Homed Firewall

A DMZ can also be established using one router/firewall appliance with three network interfaces, referred to as triple-homed. One interface is the public one, another is the DMZ, and the third connects to the LAN. Routing and filtering rules determine what forwarding is allowed between these interfaces. This can achieve the same sort of configuration as a screened subnet.



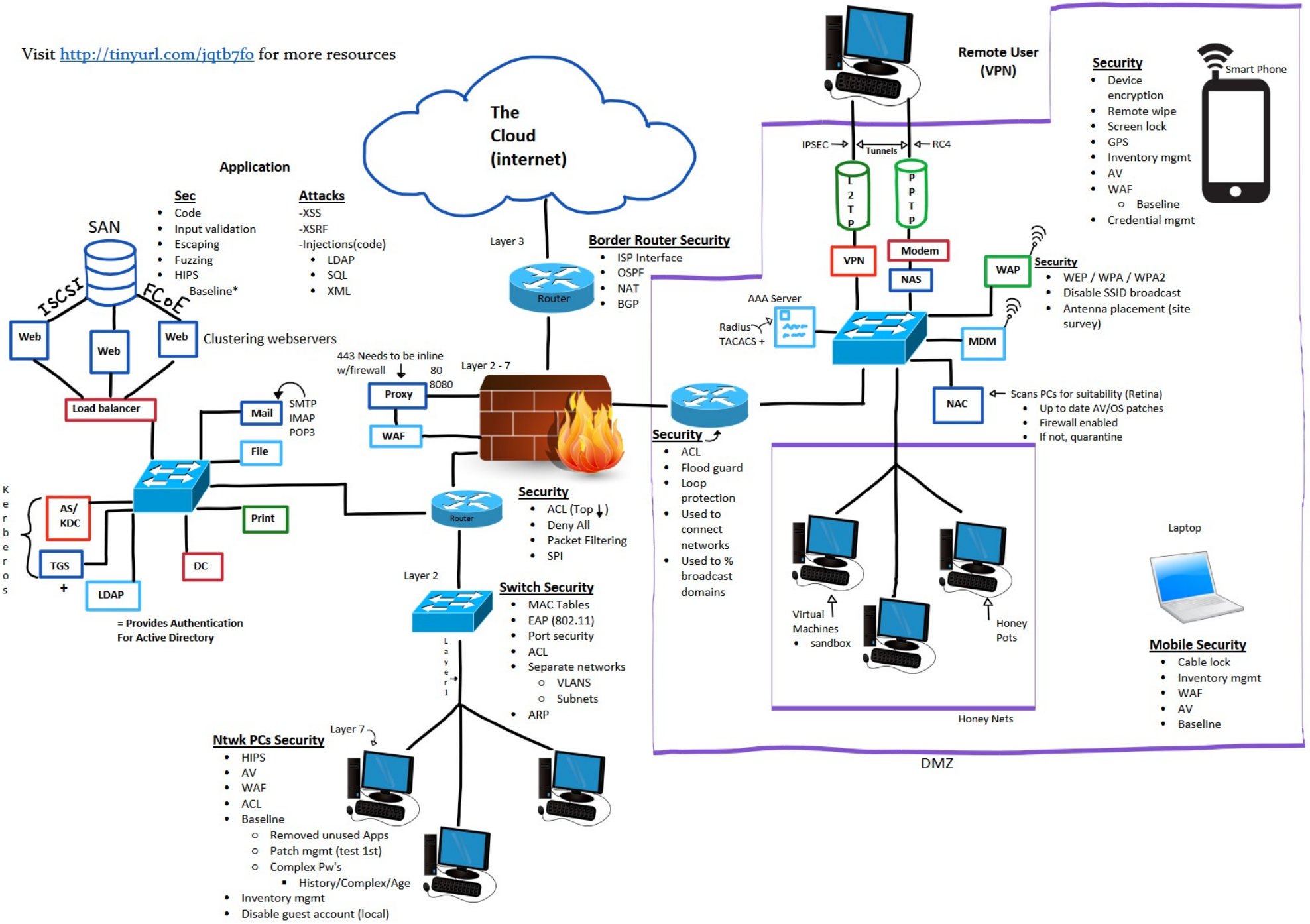
Screened Hosts

Smaller networks may not have the budget or technical expertise to implement a DMZ. In this case, Internet access can still be implemented using a dual-homed proxy/gateway server acting as a **screened host**.



Sometimes the term DMZ (or "DMZ host") is used by SOHO router vendors to mean a host on the local network that accepts connections from the Internet. This might be simpler to configure and solve some access problems, but it makes the whole network very vulnerable to intrusion and DoS. An enterprise DMZ is established by a separate network interface and subnet so that traffic between hosts in the DMZ and the LAN must be routed (and subject to firewall rules). Most SOHO routers do not have the necessary ports or routing functionality to create a true DMZ.

Visit http://tinyurl.com/jqtb7fo for more resources



Revised 5/9/2022

ALGORITHMS

TYPE	MN	NAME	KEY SIZE	NOTES
HASH	**M**	MD4/5	128	**M**essage **D**igest **5**; displayed as 32 hexadecimal characters; considered cracked-use is discouraged; MD4 prone to collisions
	R	RIPEMD	160	**R**ACE **I**ntegrity **P**rimitives **E**valuation **M**essage **D**igest; some versions create 128-, 256-, and 320-bit hashes; European standard
	S	SHA1/2/3	160/224/256,384,512	**S**ecure **H**ash **A**lgorithm; SHA0 is not used; SHA1 creates 160-bit hashes- it is prone to collisions; SHA2 has four versions: SHA-256, SHA-512, SHA-224, and SHA-384
NOTE: **HMAC** (**H**ash-based **M**essage **A**uthentication **C**ode) is a fixed-length string of bits similar to other hashing algorithms; however, HMAC also uses a shared secret key to add some randomness to the result. Only the sender and the receiver know the secret key. **IPsec** and **TLS** often use a version of HMAC such as HMAC-MD5 and HMAC-SHA1.				
SYMMETRIC ***(used with cipher modes)***	**B**	BLOWFISH/TWOFISH	64-bit key	(B) faster than AES in some circumstances
			128-bit key	128-bit block cipher. Blowfish is faster.
	R	RC4	40-bit key	(Ron) **R**ivest **C**ipher; the only stream cipher;
	A	AES	3 key strengths: 128/192/256; 128-bit blocks	**A**dvanced **E**ncryption **S**tandard Fast, efficient, strong; encrypts data in **128-bit blocks** (NIST)
	I	IDEA	160	**I**nternational **D**ata **E**ncryption **A**lgorithm
	D	DES/**3DES**	56-bit key 64-bit blocks	**D**ata **E**ncryption **S**tandard; 3DES Encrypts data in three separate passes using the DES algorithm; Fastest but weakest of the algorithms.
	s	(SYMMETRIC)		
ASYMMETRIC	**R**	RSA	1024, 2046, 3072, and 4096-bit keys	**R**ivest, **S**hamir, **A**dleman; Relies on the mathematical properties of prime numbers; used for both encryption and digital signatures
	E	ECC	160-224 bits	**E**lliptic **C**urve **C**ryptography; takes less processing power so is often used with small, wireless devices; small and fast (note: AES-256 is used in military mobile phones)
	D	DIFFIE HELMAN/DH EPHEMERAL		Modular (Clock) arithmetic; DH Groups from 768-3,072 bits in the key exchange. Not technically a key exchange algorithm, rather, it provides the ability of sender and recipient to compute the same key, DH creates the keys used in the Internet Key Exchange (IKE)
	D	DSA (Digital Signature Algorithm	2048	Digital signatures include a hash of the data for integrity
	h **a** **i** **r**	***NOTES***: 1. *Salting prevents brute force and rainbow table attacks. A pepper is like a salt but it must be kept secret and must NOT be stored with the hash.* 2. Key-Stretching Algorithms: (1) Bcrypt -adds additional random bits before encrypting multi[le times with Blowfish (2) PBKDF2 – Password-based key derivation function 2-Salts and hashes multiple times; used with WPA2 to protect pre-shared keys.		

NEW for the 601 Exam: Pretty Good Privacy (PGP) is used between two users to set up an asymmetric encryption and digital signatures. For PGP to operate, you need a private and public key pair. The first stage in using PGP is to exchange the keys. It uses RSA keys. PGP is used for encryption between two people. S/MIME is used for digital signatures between two people. GnuPG is a free version of OpenPGP. It is also known as PGP. It uses RSA keys.

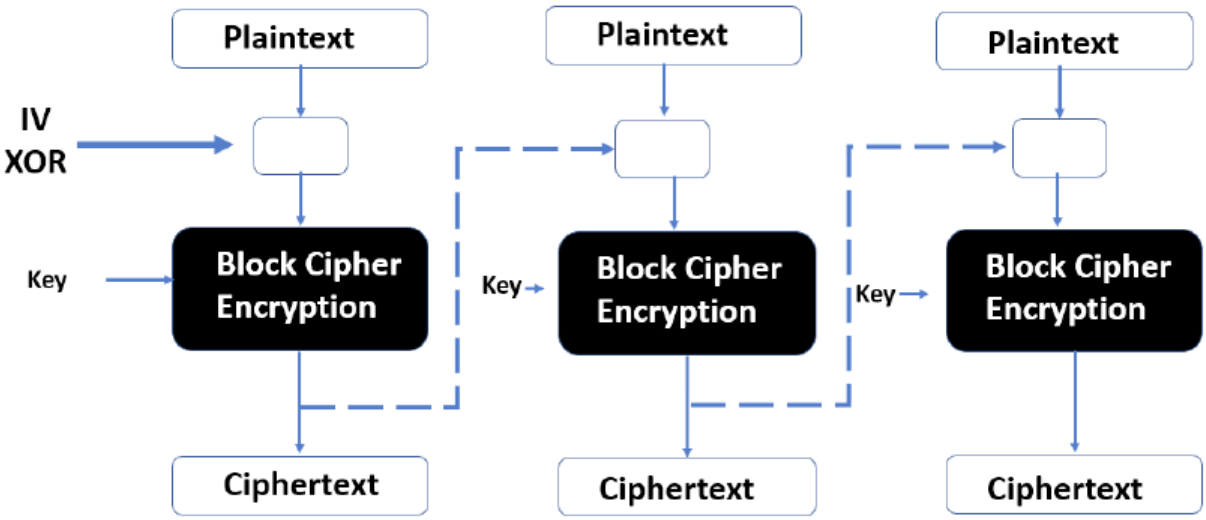
CIPHER MODES

Any given block cipher can be used in different modes of operation, which refers to the way a cryptographic product processes multiple blocks.

Mode	Name	Description	Notes
ECB	Electronic Code Book	An encryption Mode of operation where each plaintext block is encrypted with the same key	The simplest mode; identical plaintext blocks can output identical ciphertext making it vulnerable
CBC	Cipher Block Chaining *(See next page)*	An encryption mode of operation where an exclusive or (XOR) is applied to the first plaintext block	Applies an initialization vector (IV) to the first plaintext block to ensure that the key produce a unique ciphertext from any given plaintext. The output of the first operation is then combined with the next plaintext block using an XOR operation. This processes is repeated throughout the full "chain" of blocks.
CTM	Counter Mode	An encryption mode of operation where a numerical counter value is used to create a constantly changing IV. Also referred to as CM (Counter Mode) and CTM (Counter Mode)	Creates a stream from a cipher; each bock in combined with a nonce (number used once) counter value. Ensures unique ciphertext from identical plaintexts and allows each block to be processed individually, and consequently, in parallel. This improves performance.
GCM	Galois Counter Mode	An encryption mode of operation that adds authentication to the standard encryption services of a cypher mode	Used by most modern systems; Symmetric algorithms do not natively provide message integrity, and the GCM addresses this by combining the ciphertext message with a type of message authentication code (GMAC) similar to an HMAC. Where CBC is only considered secure when using a 256-bit key, GCM can be used with a 128-bit key to achieve the same level of security.

Cipher-Block Chaining Explained

- **Cipher Block Chaining (CBC)**: CBC adds XOR to each plaintext block from the ciphertext block that was previously produced. The first plaintext block has an IV that you XOR, and you then encrypt that block of plaintext. Refer to the following diagram:



The next block of plaintext is XOR'd against the last encrypted block before you encrypt this block. When decrypting a ciphertext block, you need the XOR from the previous ciphertext block. If you are missing any blocks, then decryption cannot be done.

XOR

Exclusive OR is a binary operand from Boolean Algebra. The operand will compare two bits and produce one bit in return.

- two bits that are the same = 0
- two bits that are different = 1

IV

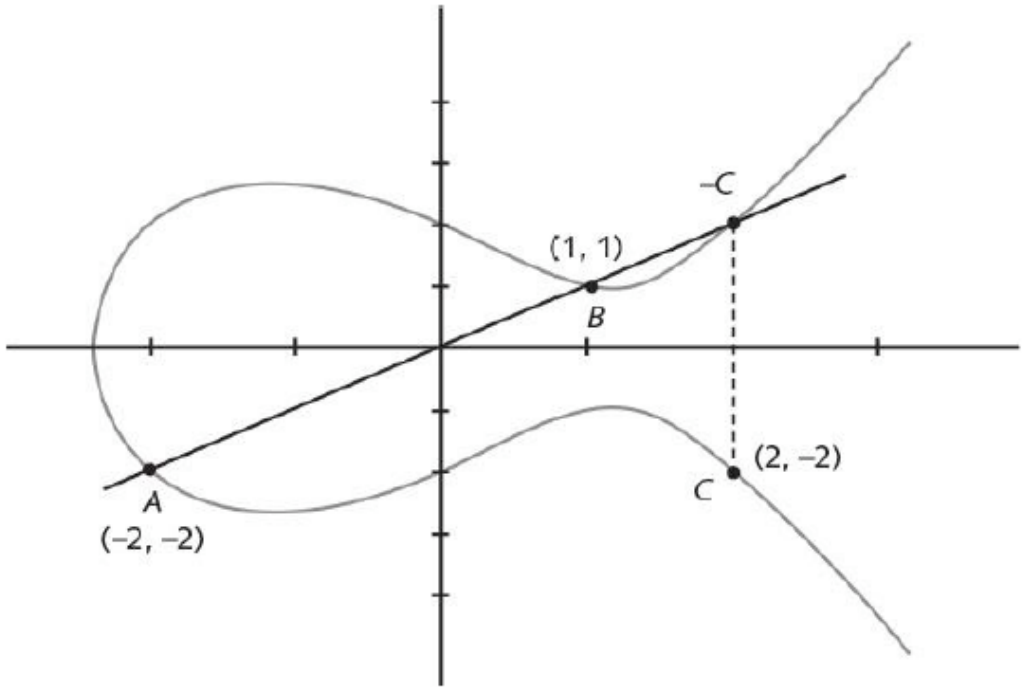
Initialization Vector- a data value used to seed a cryptographic algorithm providing for a measure of randomness.

ALGORITHMS

Elliptic Curve Cryptography (ECC)

- **Elliptic curve cryptography (ECC)** - Users share one elliptic curve and one point on curve
- Considered as an alternative for prime-number-based asymmetric cryptography for mobile and wireless devices
- Because mobile devices are limited in terms of computing power due to their smaller size, ECC offers security that is comparable to other asymmetric cryptography but with smaller key sizes
- Can result in faster computations and lower power consumption

Elliptic Curve Cryptography (ECC) (Figure 5-10)



Symmetric Encryption

- Session Key also known as secret key are both symmetric keys
- AES
  - 128 bit block cipher
  - 128, 192, or 256 bit keys
- Twofish
  - 128 bit block cipher
  - 128, 192, or 256 bit keys
- Blowfish
  - 64 bit block cipher
  - 32 to 448 bit key
  - Strong encryption that could be used instead of AES or Twofish because block sizes are 64 bits compared to 128 bits making it faster
- 3DES
  - 64 bit block cipher
  - Uses a "key bundle" that comprises three different DES keys, each of 56 bits
- RC4 – Stream cipher no longer used
  - 40 to 2048 bit key
- DES- Block encryption no longer used
  - 64 bit block cipher
  - 56 bit key

Asymmetric Encryption

- RSA (Rivest, Shamir, Adleman)
  - Keys commonly used with asymmetric encryption to privately share a symmetric key
- Diffie-Hellman (DH)
  - Secure method of sharing symmetric keys over a public network
  - Used for an IKE phase before the data is forwarded via symmetric encryption
  - Asymmetric technique that protects symmetric keys by setting up a secure channel
- ECC (Elliptic Curve Cryptography)
  - Commonly used with small wireless devices
  - Uses smaller key sizes requires less processing power
    - Note one exception is if military mobile devices are used AES-256 is used instead of ECC

Symmetric Encryption

Algorithm	Encryption Type	Method	Key Size
AES	Symmetric	128-bit block cipher	128-, 192-, or 256-bit key
Twofish	Symmetric	128-bit block cipher	128-, 192-, or 256-bit key
Blowfish	Symmetric	64-bit block cipher	32- to 448-bit key
3DES	Symmetric	64-bit block cipher	56-, 112-, or 168-bit key
RC4*	Symmetric	Stream cipher	40- to 2,048-bit key
DES*	Symmetric	64-bit block cipher	56-bit key

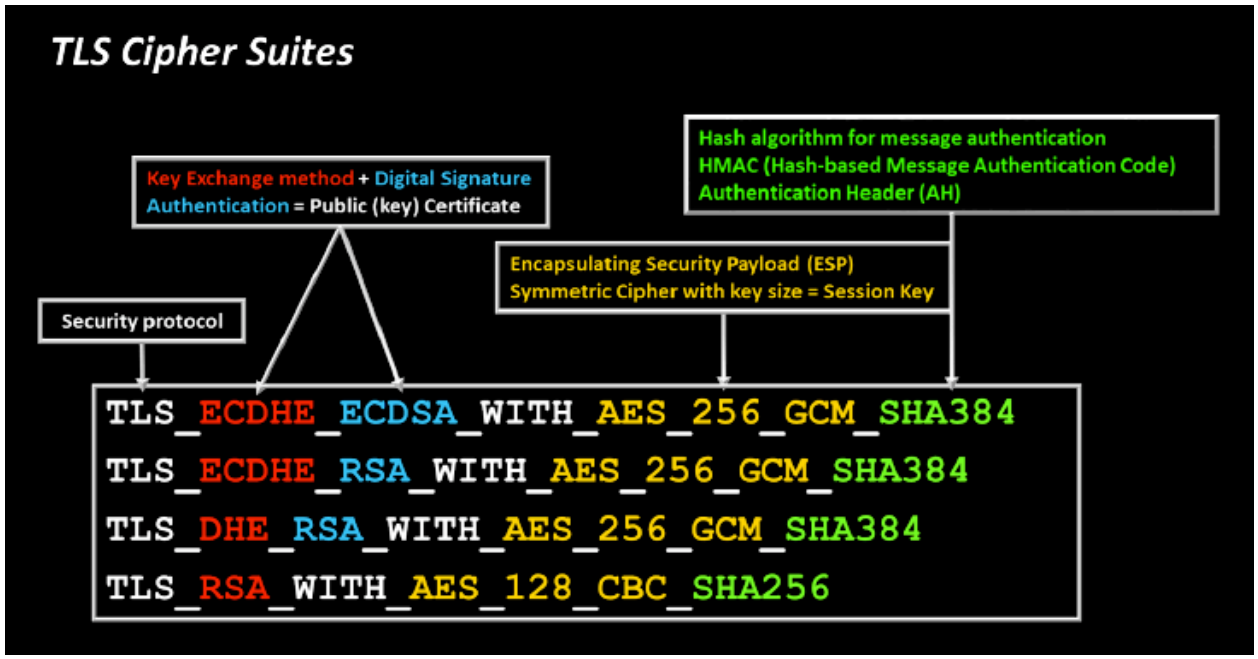
Hashing Algorithms

Algorithm	Type	Comments
MD5	Hashing - Integrity	Creates 128-bit hashes
SHA-1	Hashing - Integrity	Creates 160-bit hashes
SHA-2	Hashing - Integrity	Creates 224-, 256-, 384-, or 512-bit hashes
SHA-3	Hashing - Integrity	Creates 224-, 256-, 384-, or 512-bit hashes
HMAC-MD5	Integrity/Authenticity	Creates 128-bit hashes
HMAC-SHA1	Integrity/Authenticity	Creates 160-bit hashes

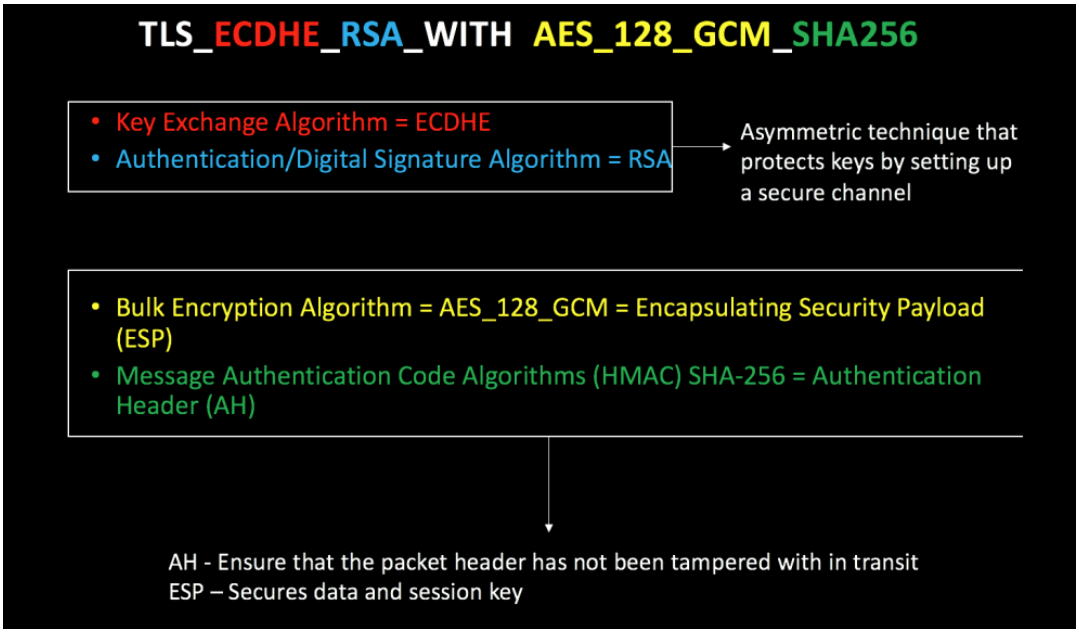
TLS Handshake

- The website's public key -> (ECDHE_RSA) encrypts
  - Asymmetric technique that protects symmetric keys by setting up a secure channel
- It encrypts a symmetric key (AES_256_GCM) ->Session Key
  - Once the secure tunnel has been created, then the symmetric encrypted data flows down the tunnel
- The website's private key decrypts the public key-> (ECDHE_RSA)
  It decrypts a symmetric key (AES_256_GCM) ->Session Key
- The symmetric key encrypts data in the website session (AES_256_GCM) ->Session Key

TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384



Algorithms for VPN



HMAC (Hash-based Message Authentication Code)

- **Authentication Header**
- Uses a shared secret to provide integrity/authenticity
- IPsec and TLS use:
  - HMAC-MD5
  - HMAC-SHA1
  - HMAC-SHA2

`TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384`

Digital Signatures Explained

When we send an email or document to someone, it could be intercepted in transit and altered. Your email address could be spoofed, and someone could send an email as if it was from you, but there is no guarantee of integrity. We sign the email or document with our private key and it is validated by our public key.

The first stage in digital signatures is to exchange public keys, the same principle as encryption. For example, George wants to send Mary an email and he wants to ensure that it has not been altered in transit:

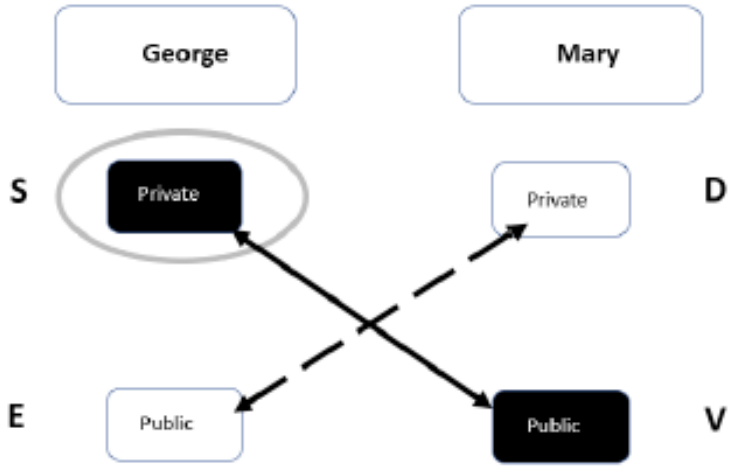


Figure 2.10 – Digital signature

In *Figure 2.10*, you can see that George is going to sign the email with his private key when he sends it to Mary, and she then validates it with the public key that George has already given to her. When the email has been validated, she knows that the email has not been tampered with. It could be read in transit, but not tampered with.

When people are asked to sign contracts, they sometimes use a third-party provider that asks them to digitally sign the contract. This then makes the contract valid as the digital signature proves the identity of the signatory.

Then there's non-repudiation. When I complete a digital signature, I am using my private key, which I should never give away to sign the email or document, proving that it has come from me. Non-repudiation means that I cannot deny that it was me who signed the document. I could not say it was done by someone else. In the early 6th century, King Arthur would send messages to his knights on a parchment scroll and then would put his wax seal on the scroll to prove it came from him. The digital signature in modern life is doing the same – it is proving who it came from. The digital signature creates a one-way hash of the entire document, so it also provides integrity similar to hashing.

> **Tip**
> Encryption uses the recipients' public key, where a digital signature used the sender's private key.

Digital Signatures vs Encrypted Email

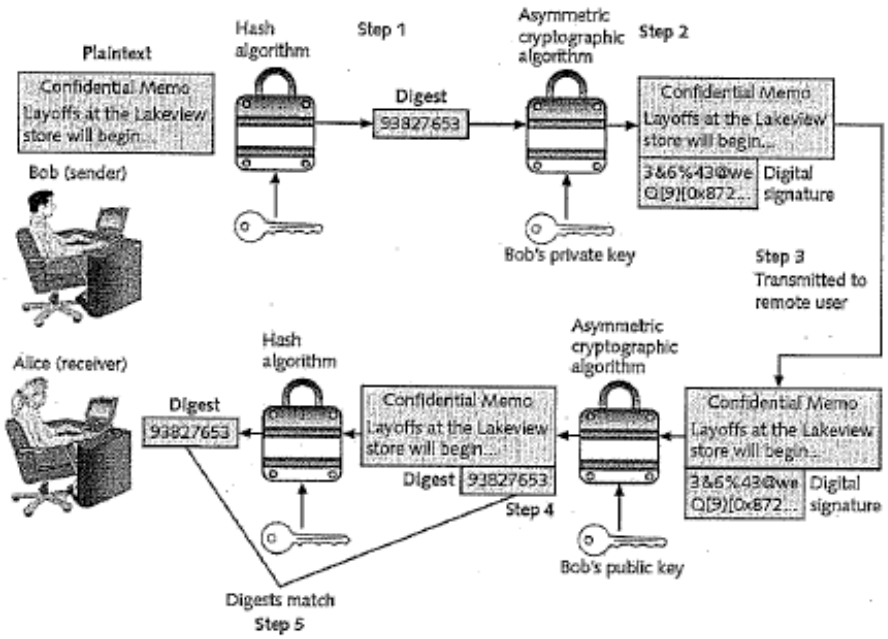
Digital Signatures

- Digital signatures provide integrity and non-repudiation
  - Uses private key to encrypt the hash of the digital signature
  - Recipient must use senders public key to decrypt

Encrypted Email

- Encrypting email provides confidentiality
  - Sender encrypts email with recipients public key
  - Recipient used private key to decrypt

Digital Signature



Digital Signatures

- **Provide integrity and non-repudiation**
- **Sender used private key to encrypt**
- **Recipient must use senders public key (RSA) to decrypt**

`TLS_RSA_WITH_AES_256_GCM_SHA384`

Encrypted Email

- **Encrypting email provides confidentiality**
- **Sender encrypts email with recipients public key (RSA)**
- **Recipient used private key to decrypt**

Registration Authorities and CSRs

Registration is the process by which end users create an account with the CA and become authorized to request certificates. The exact processes by which users are authorized and their identity proven are determined by the CA implementation. For example, in a Windows Active Directory network, users and devices can often auto-enroll with the CA just by authenticating to Active Directory. Commercial CAs might perform a range of tests to ensure that a subject is who he or she claims to be. It is in the CA's interest to ensure that it only issues certificates to legitimate users, or its reputation will suffer.

On a private network (such as a Windows domain), the right to issue certificates of different types must be carefully controlled. The Windows CA supports access permissions for each certificate type so that you can choose which accounts are able to issue them.

When a subject wants to obtain a certificate, it completes a **certificate signing request (CSR)** and submits it to the CA. The CSR is a Base64 ASCII file containing the information that the subject wants to use in the certificate, including its public key.

The CA reviews the certificate and checks that the information is valid. For a web server, this may simply mean verifying that the subject name and fully qualified domain name (FQDN) are identical, and verifying that the CSR was initiated by the person administratively responsible for the domain, as identified in the domain's WHOIS records. If the request is accepted, the CA signs the certificate and sends it to the subject.

The registration function may be delegated by the CA to one or more **registration authorities (RAs)**. These entities complete identity checking and submit CSRs on behalf of end users, but they do not actually sign or issue certificates.

This example is simplified. Using a root CA to issue leaf certificates directly is not robust. It is better to create one or more intermediate CAs.

Certificate Signing Requests

To configure a certificate on a host, create a certificate signing request (CSR) with a new key pair. This command is run on the web server:

```
openssl req -nodes -new -newkey rsa:2048 -out
www.csr -keyout www.key
```

Having run the command, you then complete the prompts to enter the subject information for the certificate, taking care to match the common name (CN) to the FQDN by which clients access the server. This key is created without a password, which would have to be input at any restart of the web server application. We can rely on general access control security measures to protect the key.

This CSR file must then be transmitted to the CA server. On the CA, run the following command to sign the CSR and output the X.509 certificate:

```
openssl ca -config openssl.cnf -extensions webserver
-infiles www.csr -out www.pem
```

The passphrase must be entered to confirm use of the `cakey.pem` private key. The `-extensions` argument selects an area of the configuration file for a particular certificate type. This sets the key usage attribute, plus any other extended attributes that are needed.

You can view the new certificate to check the details using the following two commands:

```
openssl x509 -noout -text -in www.pem
```

```
openssl verify -verbose -cafile cacert.pem www.pem
```

Transmit the `www.pem` file to the web server and update the server configuration to use it and the `www.key` private key.

Certificate signing request (CSR)

- Process of requesting a new certificate
- Two keys are generated and the public is sent to the CA
  - Create the RSA-based private key, which is used to create the public key
  - Include the public key in the CSR
  - The CA will embed the public key in the certificate

Leveraging the information supplied below, complete the CSR for the server to set up TLS (HTTPS)

- **Hostname: ws01**
- **Domain: comptia.org**
- **IPv4: 10.1.9.50**
- **IPV4: 10.2.10.50**
- **Root: home.aspx**
- **DNS CNAME: homesite.**

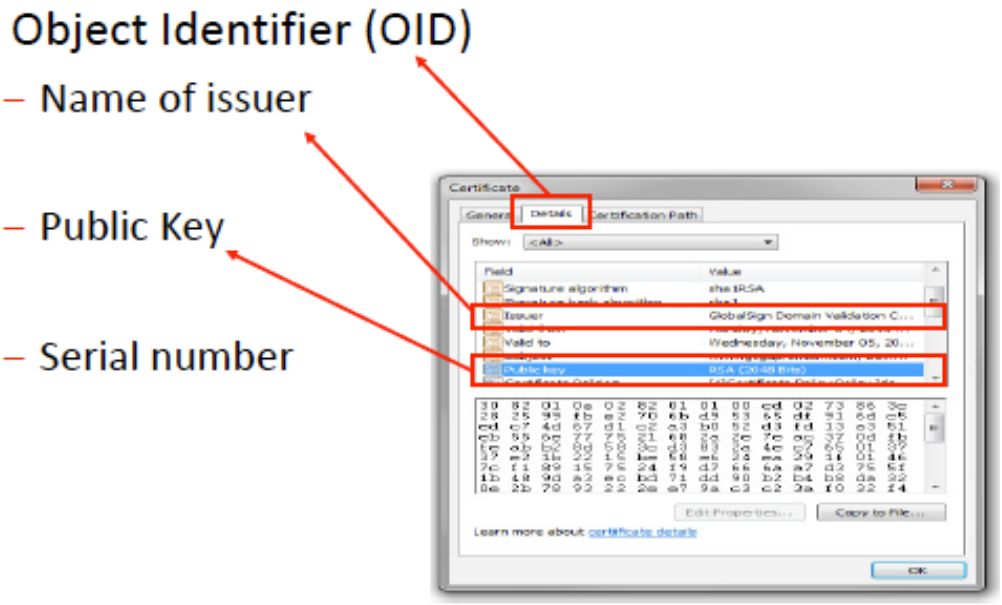
INSTRUCTIONS:

Drag the various data points to the correct locations within the CSR. Extension criteria belong in the let hand column and values belong in the corresponding row in the right hand column


Answer:
Server
Hostname: ws01
Domain: comptia.org
IPv4: 10.1.9.50
IPv4: 10.2.10.50
Root: home.aspx
DNS CNAME: homesite
Extensions
policyIdentifer
CommonName
SubjAltName
extendedKeyUsage
Values
serverAuth
OCSP;URI:http://ocsp.pki.comptia.org
URL=http://homesite.comptia.org/home.aspx
ws01.comptia.org
DNS comptia.orgName*=.comptia.org
clientAuth
DNS Name=homesite.comptia.org
Certificate Signing Request
Extensions
Values
CommonName
DNS Name=homesite.comptia.org
SubjAltName
DNS comptia.orgName*=.comptia.org
extendedKeyUsage
serverAuth
policyIdentifer
OCSP;URI:http://ocsp.pki.comptia.org

Digital Certificates

A **digital certificate** is essentially a wrapper for a subject's public key. As well as the public key, it contains information about the subject and the certificate's issuer or guarantor. The certificate is digitally signed to prove that it was issued to the subject by a particular CA. The subject could be a human user (for certificates allowing the signing of messages, for instance) or a computer server (for a web server hosting confidential transactions, for instance).



Digital certificates are based on the X.509 standard approved by the International Telecommunications Union and standardized by the Internet Engineering Taskforce (tools.ietf.org/html/rfc5280). The Public Key Infrastructure (PKIX) working group manages the development of these standards. RSA also created a set of standards, referred to as **Public Key Cryptography Standards (PKCS)**, to promote the use of public key infrastructure.

Certificate Attributes

The X.509 standard defines the fields or attributes that must be present in the certificate. Some of the main fields are listed in the following table.

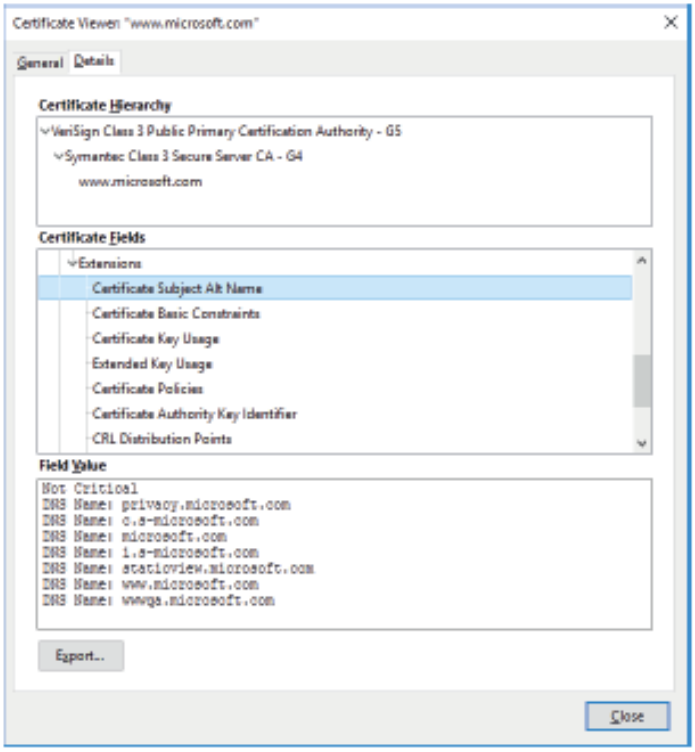
Field	Usage
Serial number	A number uniquely identifying the certificate within the domain of its CA.
Signature algorithm	The algorithm used by the CA to sign the certificate.
Issuer	The name of the CA.
Valid from/to	Date and time during which the certificate is valid.
Subject	The name of the certificate holder, expressed as a distinguished name (DN). Within this, the common name (CN) part should usually match either the fully qualified domain name (FQDN) of the server or a user email address.
Public key	Public key and algorithm used by the certificate holder.
Extensions	V3 certificates can be defined with extended attributes, such as friendly subject or issuer names, contact email addresses, and intended key usage.
Subject alternative name (SAN)	This extension field is the preferred mechanism to identify the DNS name or names by which a host is identified.

Subject Name Attributes

When certificates were first introduced, the **common name (CN)** attribute was used to identify the FQDN by which the server is accessed, such as `www.comptia.org`. This usage grew by custom rather than design, however. The CN attribute can contain different kinds of information, making it difficult for a browser to interpret it correctly. Consequently, the CN attribute is deprecated as a method of validating subject identity (tools.ietf.org/html/rfc2818#section-3.1).

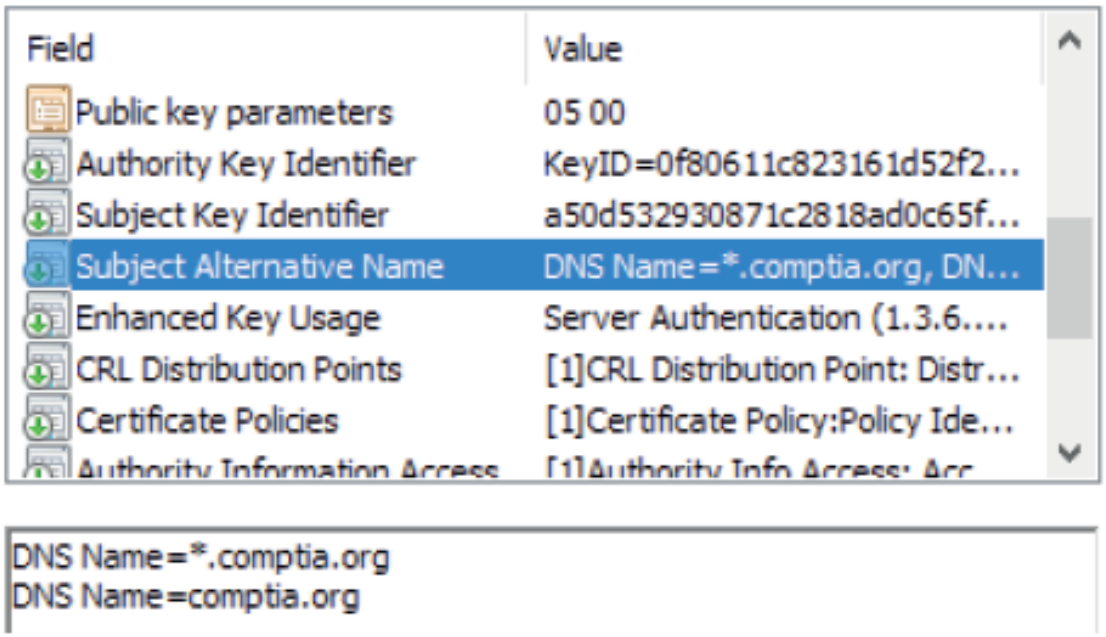
The **subject alternative name (SAN)** extension field is structured to represent different types of identifiers, including domain names. If a certificate is configured with a SAN, the browser should validate that, and ignore the CN value. It is still safer to put the FQDN is the CN as well, because not all browsers and implementations stay up-to-date with the standards.

The SAN field also allows a certificate to represent different subdomains, such as `www.comptia.org` and `members.comptia.org`.



Microsoft's website certificate configured with alternative subject names for different subdomains. (Screenshot used with permission from Microsoft.)

Listing the specific subdomains is more secure, but if a new subdomain is added, a new certificate must be issued. A wildcard domain, such as `*.comptia.org`, means that the certificate issued to the parent domain will be accepted as valid for all subdomains (to a single level).



CompTIA's website certificate configured with a wildcard domain, allowing access via either https://comptia.org or https://www.comptia.org. (Screenshot used with permission from Microsoft.)

Types of Certificates

As a security professional, you will be responsible for purchasing new certificates, and therefore, you must learn the certificate types thoroughly to ensure that you make the correct purchases. We will start with the self-signed certificate, which can roll out with applications such as Microsoft Exchange Server or Skype, and finish with extended validation where the certificate has a high level of trust:

- **Self-Signed Certificate**: A self-signed certificate is issued by the same entity that is using it. However, it does not have a CRL and cannot be validated or trusted.
- **Wildcard**: For a wildcard certificate for a domain called `securityplus.training`, the wildcard certification would be `*.securityplus.training` and could be used for the domain and a subdomain. For example, in the `securityplus.training` domain, there are two servers called `web` and `mail`. The wildcard certification is `*.securityplus.training` and, when installed, it would work for the **Fully Qualified Domain Names (FQDNs)** of both of these—`web.securityplus.training` and `mail.securityplus.training`. A wildcard can be used for multiple servers in the same domain.
- **Domain Validation**: A **Domain-Validated (DV)** certificate is an X.509 certificate that proves the ownership of a domain name.

Root Certificate

The **root certificate** is the one that identifies the CA itself. The root certificate is self-signed. A root certificate would normally use a key size of at least 2048 bits. Many providers are switching to 4096 bits. The CN for a root certificate is set to the organization/CA name, such as "CompTIA Root CA," rather than an FQDN.

Self-signed Certificates

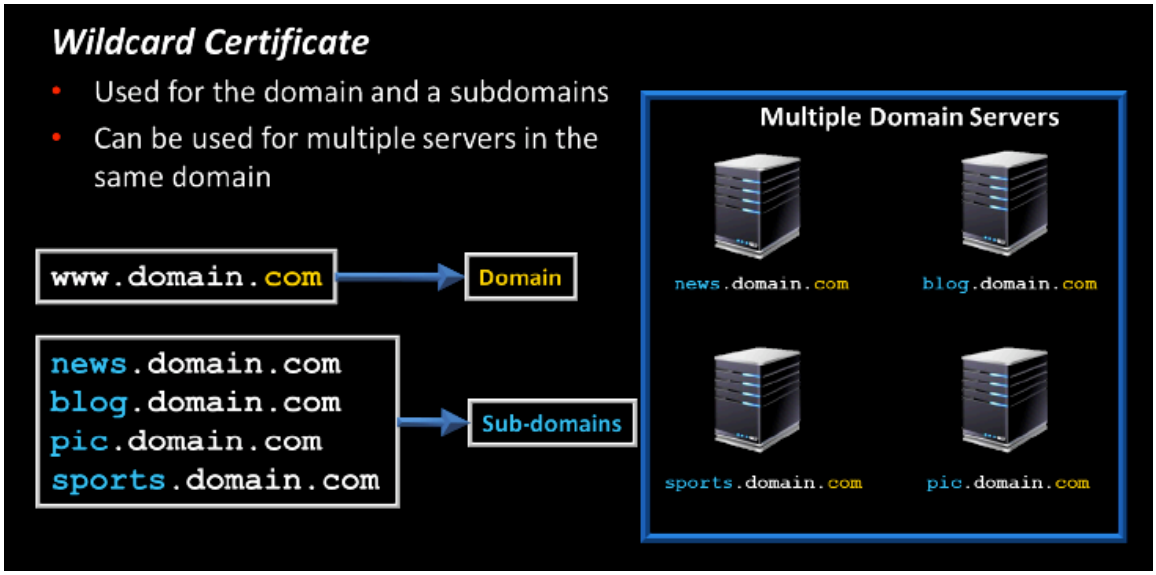
Any machine, web server, or program code can be deployed with a **self-signed certificate**. Self-signed certificates will be marked as untrusted by the operating system or browser, but an administrative user can choose to override this.

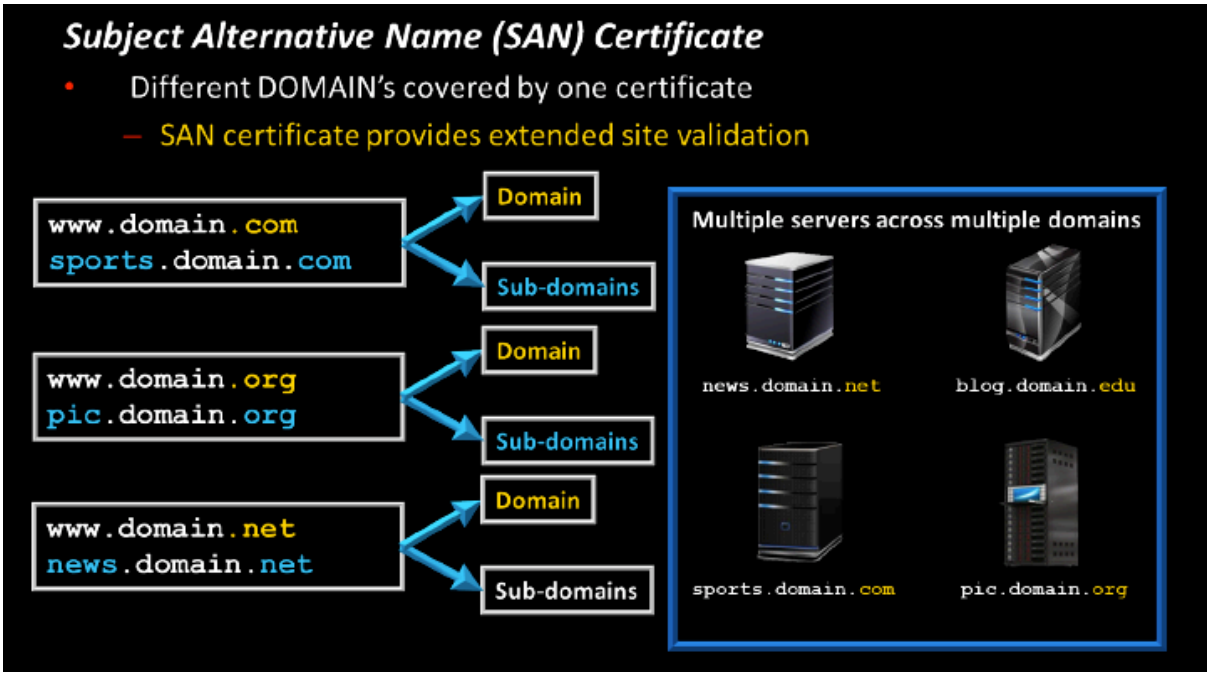
- **Subject Alternative Name (SAN)**: An SAN certificate can be used on multiple domain names, such as `abc.com` or `xyz.com`. You can also insert other information into an SAN certificate, such as an IP address.
- **Code Signing**: Code-signing certificates are used to digitally sign software so that its authenticity is guaranteed.
- **Computer/Machine**: A computer or machine certificate is used to identify a computer within a domain.
- **User**: A user certificate provides authenticity to a user for the applications that they use.

- **Extended Validation**: Extended validation certificates provide a higher level of trust in identifying the entity that is using the certificate. It would normally be used in the financial arena. You may have seen it in action where the background of the URL turns green, as shown in the following screenshot:

Companies applying for the extended validation certificate would have to provide more detailed information about the company.

> **Tip**
> A wildcard certificate can be installed on multiple public facing websites as a cheaper option. A self-signed certificate can be installed on internal facing websites as a cheaper option.



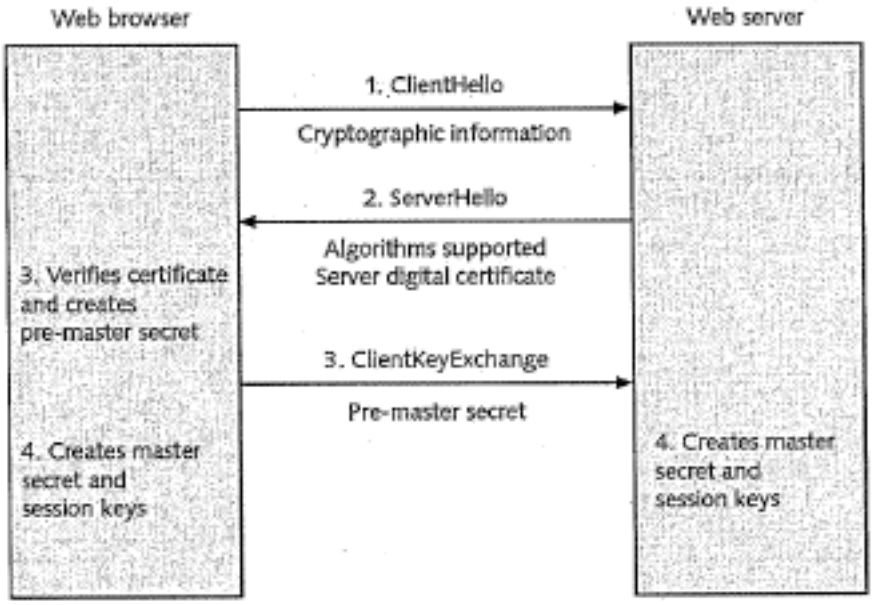


Self Signed Certificates

- Self-signed certificate is issued by the same entity that is using it. However, it does not have a CRL and cannot be validated or trusted.
- Used for internal sites without incurring additional costs or overhead
- No PKI infrastructure is required

1. A systems administrator wants to generate a self-signed certificate for an internal website
2. System Administrator needs to provide the private key to the internal CA prior to installing the certificate on the server
   - Private key is used to create public key

Server Digital Certificate Handshake



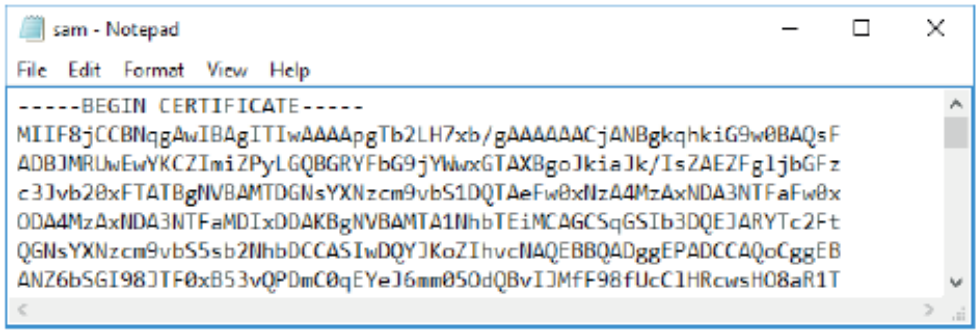
Certificate Formats

There are various formats for encoding a certificate as a digital file for exchange between different systems.

Encoding

Cryptographic data—both certificates and keys—are processed as binary using **Distinguished Encoding Rules (DER)**. Binary format files are not commonly used, however.

More typically, the binary data is represented as **ASCII** text characters using Base64 **Privacy-enhanced Electronic Mail (PEM)** encoding. ASCII-format data has descriptive headers, such as the "BEGIN CERTIFICATE" string.



Base64-encoded .CER file opened in Notepad. (Screenshot used with permission from Microsoft.)

File Extensions

A three character file extension is a *convention*, not a standard, and unfortunately file extensions do not always map cleanly to the type of encoding used within a certificate file, or even to the contents of a certificate file. The only certain way to check is to open it in a text editor.

- Both .DER and .PEM can be used as file extensions, although the latter is not recognized by Windows. .PEM is the the most widely used extension for ASCII format files in Linux.
- The .CRT and .CER extensions can also be used, but they they are not well-standardized. Most of the confusion arises from the way Windows handles certificates. In Linux, .CRT is most likely to represent an ASCII certificate. In Windows, the most common extension is .CER, but this does not tell you whether the file format is binary or ASCII.

Contents

A certificate file can also contain more than just a single certificate:

- The **PKCS #12 format** allows the export of the private key with the certificate. This would be used either to transfer a private key to a host that could not generate its own keys, or to back up/archive a private key. This type of file format is usually password-protected and always binary. On Windows, these usually have a **.PFX** extension, while MacOS and iOS use .P12. In Linux, the certificate and key are usually stored in separate files.
- The **P7B format** implements PKCS #7, which is a means of bundling multiple certificates in the same file. It is typically in ASCII format. This is most often used to deliver a chain of certificates that must be trusted by the processing host. It is associated with the use of S/MIME to encrypt email messages. P7B files do not contain the private key. In Linux, the .PEM extension is very widely used for certificate chains.

- **Certificate Formats**: There are different certificate formats, and these are as follows:

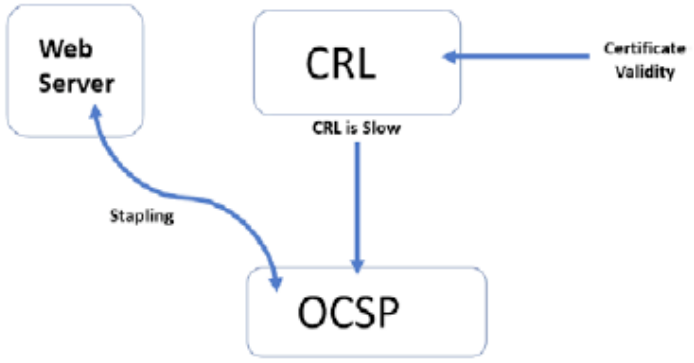
Certificate type	Format	File extension
Private	P12	.pfx
Public	P7B	.cer
PEM	Base64 format	.pem
DER	Extension for PEM	.der

Type	Common Extensions	Format	Common Purpose	Can Contain
CER	.cer	ASCII	Used for ASCII certificates	Varies
DER	.der	Binary	Used for binary certificates	Varies
PEM	.pem, .cer, .crt, .key	Binary (DER) or ASCII (CER)	Can be used for almost any certificate purpose	Server certificates, certificate chains, keys, CRL
P7B	.p7b, .p7c	ASCII (CER)	Used to share the public key	Certificates, certificate chains, CRL, but never the private key
P12 PFX	.p12, .pfx	Binary (DER)	Commonly used to store private keys with a certificate	Certificates, certificate chains, and private keys

> **Tip**
> Certificate chaining shows the trust from the vendor, the vendor CA, and the computer. Fewer than three layers results in trust errors

Certificate Validity

Each time a certificate is used, the first thing that must happen is that it must be checked for validity. The following diagram shows the certificate validity process:



There are three separate processes that you must know thoroughly, and these are as follows:

- **Certificate Revocation List (CRL)**: The first stage in checking whether a certificate is valid, no matter the scenario, is to check the CRL. If the X509 is in the CRL, it is no longer valid and will not be accepted. No matter how obscure the question posed in the exam, unless it is going slow or it is a web server looking for a faster lookup, it will be the CRL that provides certificate validity.
- **Online Certificate Status Protocol (OCSP)**: Only when the CRL is going slow will the OCSP come into play. It is much faster than the CRL and can take a load from the CRL in a very busy environment.
- **OCSP Stapling/Certificate Stapling**: Certificate stapling, also known as OCSP stapling, is used when a web server bypasses the CRL to use the OCSP for a faster confirmation, irrespective of whether or not a certificate is valid.

> **Tip**
> Certificate validity can only be done by the CRL or OCSP. OCSP is used only when the CRL is going slow or has been replaced by the OCSP

Certificate Expiration

Certificates are issued with a limited duration, as set by the CA policy for the certificate type. Root certificates might have long expiration dates (10+ years), whereas web server and user certificates might be issued for 1 year only. Typically, a certificate is renewed before it expires. Where a user is in possession of a valid certificate, less administration is required (in terms of checking identity) than with a request for a new certificate. When you are renewing a certificate, it is possible to use the existing key (referred to specifically as *key renewal*) or generate a new key (the certificate is *rekeyed*). A new key might be generated if the old one was no longer considered long enough or if any compromise of the key was feared.

When a certificate expires, there is the question of what to do with the key pair that it represents. A key can either be archived or destroyed. Destroying the key offers more security, but has the drawback that any data encrypted using the key will be unreadable. Whether a key is archived or destroyed will largely depend on how the key was used. In software terms, a key can be destroyed by overwriting the data (merely deleting the data is not secure). A key stored on hardware can be destroyed by a specified erase procedure or by destroying the device.

Certificate Revocation Lists

A certificate may be revoked or suspended:

- A revoked certificate is no longer valid and cannot be "un-revoked" or reinstated.
- A suspended certificate can be re-enabled.

A certificate may be revoked or suspended by the owner or by the CA for many reasons. For example, the certificate or its private key may have been compromised, the business could have closed, a user could have left the company, a domain name could have been changed, the certificate could have been misused in some way, and so on. These reasons are codified under choices such as Unspecified, Key Compromise, CA Compromise, Superseded, or Cessation of Operation. A suspended key is given the code Certificate Hold.

It follows that there must be some mechanism for informing users whether a certificate is valid, revoked, or suspended. CAs must maintain a **certificate revocation list (CRL)** of all revoked and suspended certificates, which can be distributed throughout the hierarchy.

With the CRL system, there is a risk that the certificate might be revoked but still accepted by clients because an up-to-date CRL has not been published. A further problem is that the browser (or other application) may not be configured to perform CRL checking, although this now tends to be the case only with legacy browser software.

Online Certificate Status Protocol Responders

Another means of providing up-to-date information is to check the certificate's status on an **Online Certificate Status Protocol (OCSP)** server, referred to as an *OCSP responder*. Rather than return a whole CRL, this just communicates the status of the requested certificate. Details of the OCSP responder service should be published in the certificate.

Most OCSP servers can query the certificate database directly and obtain the real-time status of a certificate. Other OCSP servers actually depend on the CRLs and are limited by the CRL publishing interval.

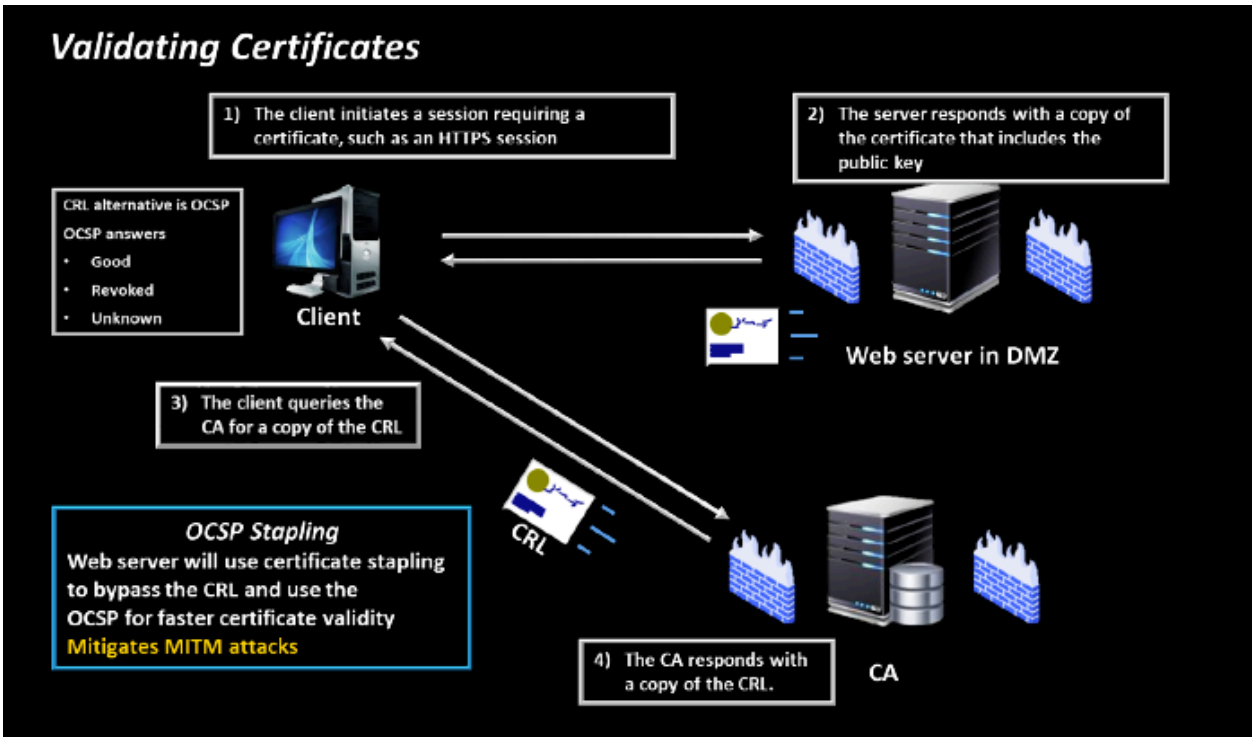
One of the problems with OCSP is that the job of responding to requests is resource intensive and can place high demands on the issuing CA running the OCSP responder. There is also a privacy issue, as the OCSP responder could be used to monitor and record client browser requests. OCSP **stapling** resolves these issues by having the SSL/TLS web server periodically obtain a time-stamped OCSP response from the CA. When a client submits an OCSP request, the web server returns the time-stamped response, rather than making the client contact the OCSP responder itself.

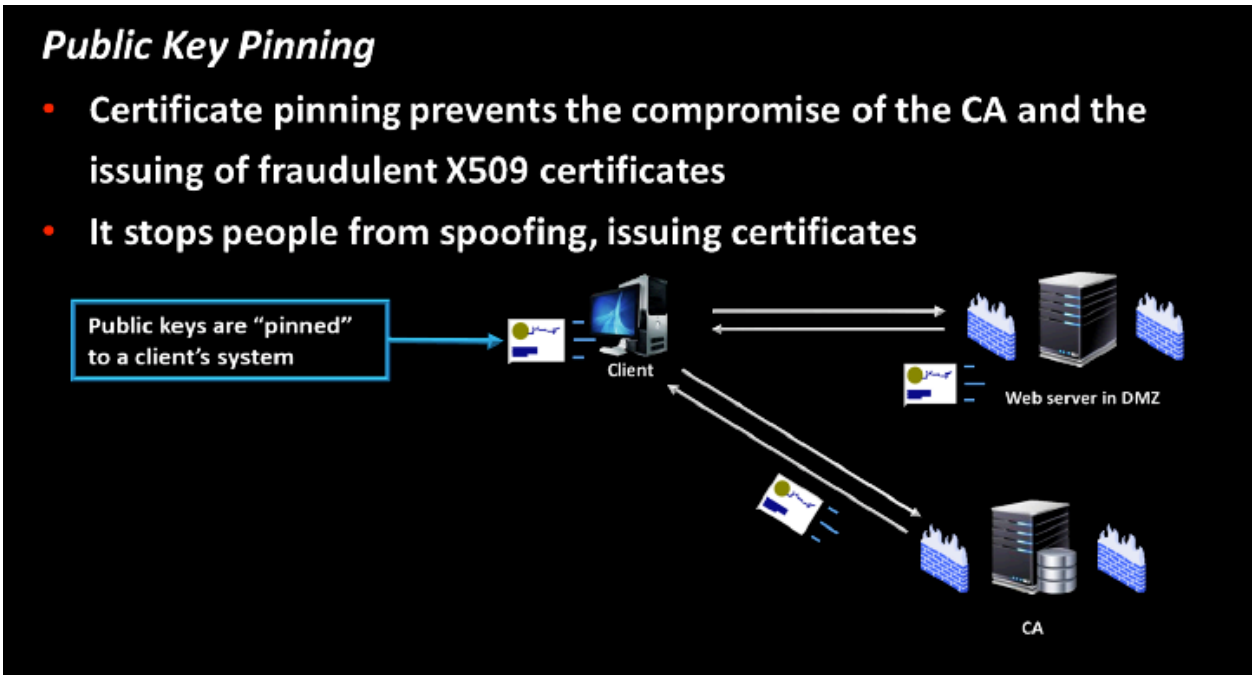
Certificate Pinning

When certificates are used by a transport protocol, such as SSL/TLS, there is a possibility that the chain of trust among the client, the server, and whatever intermediate and root CAs have provided certificates can be compromised. If an adversary can substitute a malicious but trusted certificate into the chain (using some sort of proxy or man-in-the-middle attack), they could be able to snoop on the supposedly secure connection.

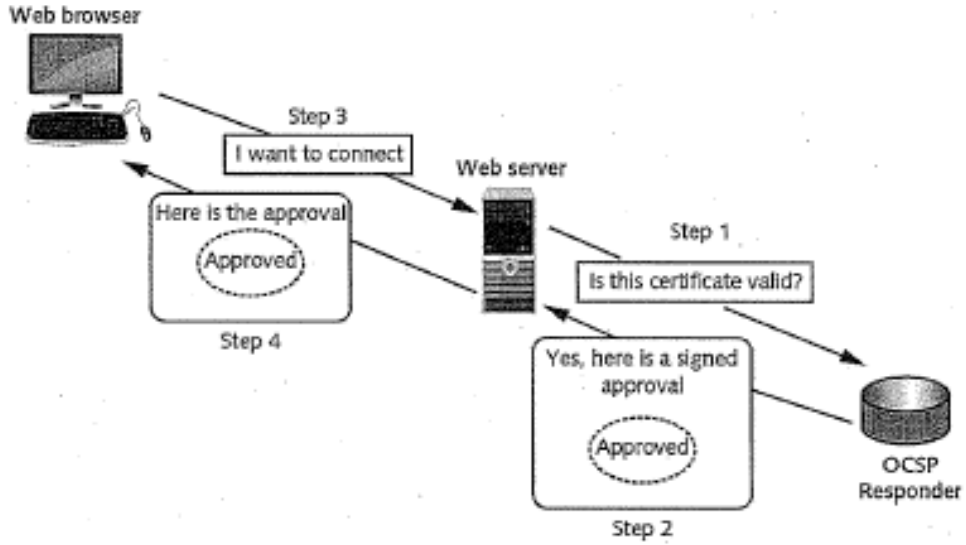
Pinning refers to several techniques to ensure that when a client inspects the certificate presented by a server or a code-signed application, it is inspecting the proper certificate. This might be achieved by embedding the certificate data in the application code, or by submitting one or more public keys to an HTTP browser via an HTTP header, which is referred to as *HTTP Public Key Pinning (HPKP)*.

HPKP has serious vulnerabilities and has been deprecated (developer.mozilla.org/en-US/docs/Web/HTTP/Public_Key_Pinning). The replacement mechanism is the Certificate Transparency Framework.


Validating Certificates
1) The client initiates a session requiring a certificate, such as an HTTPS session
2) The server responds with a copy of the certificate that includes the public key
CRL alternative is OCSP
OCSP answers
Good
Revoked
Unknown
Client
Web server in DMZ
3) The client queries the CA for a copy of the CRL
CRL
OCSP Stapling
Web server will use certificate stapling to bypass the CRL and use the OCSP for faster certificate validity
Mitigates MITM attacks
4) The CA responds with a copy of the CRL.
CA


Public Key Pinning
Certificate pinning prevents the compromise of the CA and the issuing of fraudulent X509 certificates
It stops people from spoofing, issuing certificates
Public keys are "pinned" to a client's system
Client
Web server in DMZ
CA

OCSP Stapling



Certificate Hierarchy

The **Certificate Authority** (**CA**) is the ultimate authority as it holds the master key, also known as the root key, for signing all of the certificates that it gives the **Intermediary**, which then, in turn, issues the certificate to the requester.

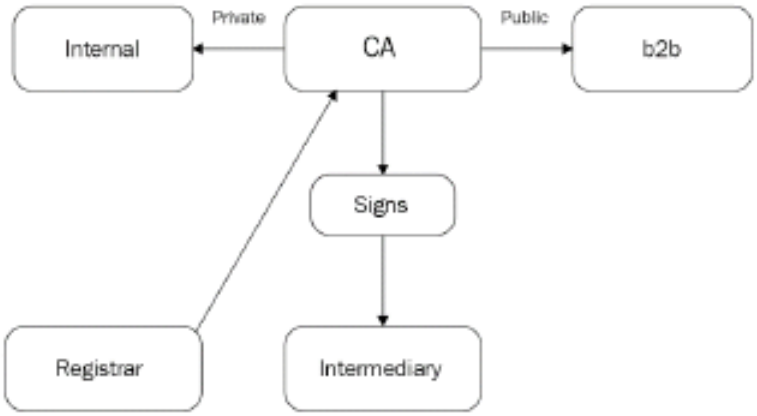


Figure 2.1 – CA Hierarchy

Let's look at the CA hierarchy shown in the preceding diagram in more depth:

- **Online CA**: An internal online CA is always up and running so that people in the company can request a certificate at any time of the day or night. This would not be the case in a government or top-security environment.
- **Offline CA**: An offline CA is for a military or secure environment where clearance and vetting must be completed before someone can be issued with a certificate. The CA is kept offline and locked up when it is not being used. It is switched off so that it cannot issue new certificates.

There are different types of CA:

- **Public CA**: A public CA is also known as a third-party CA and is commercially accepted as an authority for issuing public certificates. Examples include *Sectigo*, formerly known as *Comodo*, *Symantec*, *Go Daddy*, and more.

The benefit of using a third-party CA is that all of the management is carried out by them; once you purchase the certificate, all you have to do is install it. They keep an up-to-date **Certificate Revocation List** (**CRL**) where you can check whether your certificate is valid. A certificate that is not valid will not work if you are going to sell goods and services to other companies; this is known as a B2B transaction, which requires a public CA.

For example, I put gas in my car and go to pay for it. I give the attendant some monopoly money, but they refuse to take it; this would be the equivalent of a private CA. Businesses will not accept it as payment. I then go to the cash machine outside and withdraw $100 and I give this to the attendant; he smiles and accepts it and gives me some change. This is the equivalent of a public CA.

If you wish to trade and exchange certificates with other businesses, you need to get your certificate from a public CA. The certificate that follows has been issued to the Bank of Scotland from a public CA called DigiCert Global CA. You can see on the front of the certificate the purpose for use and also the dates that it is valid for. The X509 has an OID, which is basically the certificate's serial number – the same way that paper money has serial numbers:

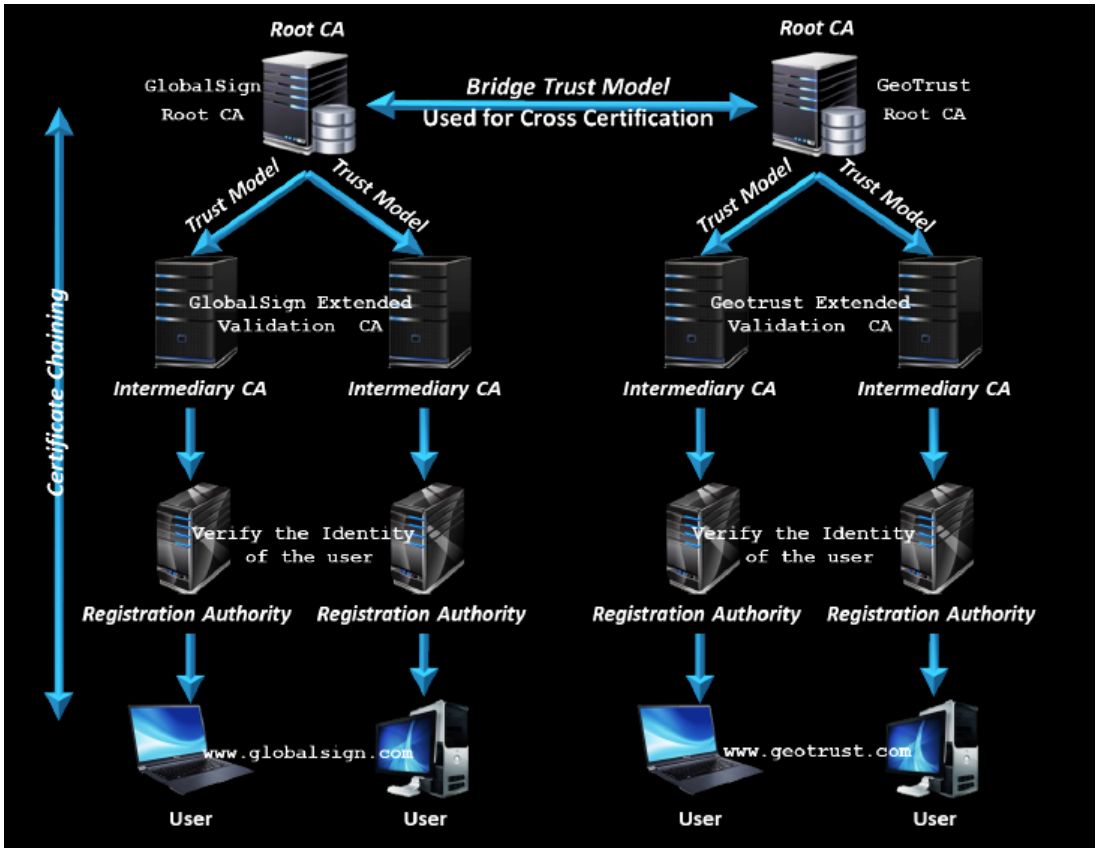
- **Private CA**: A private CA can only be used internally. However, although it is free, you must maintain the CA. Hopefully, your company has the skill set to do so.
- **Registration Authority** (**RA**): The RA validates and accepts the incoming requests for certificates from users on the network and notifies the CA to issue the certificates. The certificates that are issued are known as X509 certificates.
- **Subordinate CA**: It could be the RA that issues certificates to users. In the CompTIA exam, the subordinate CA could be called an intermediary.
- **Certificate Pinning**: Certificate pinning prevents the compromising of the CA and the issuing of fraudulent X509 certificates. It prevents SSL man-in-the-middle attacks.

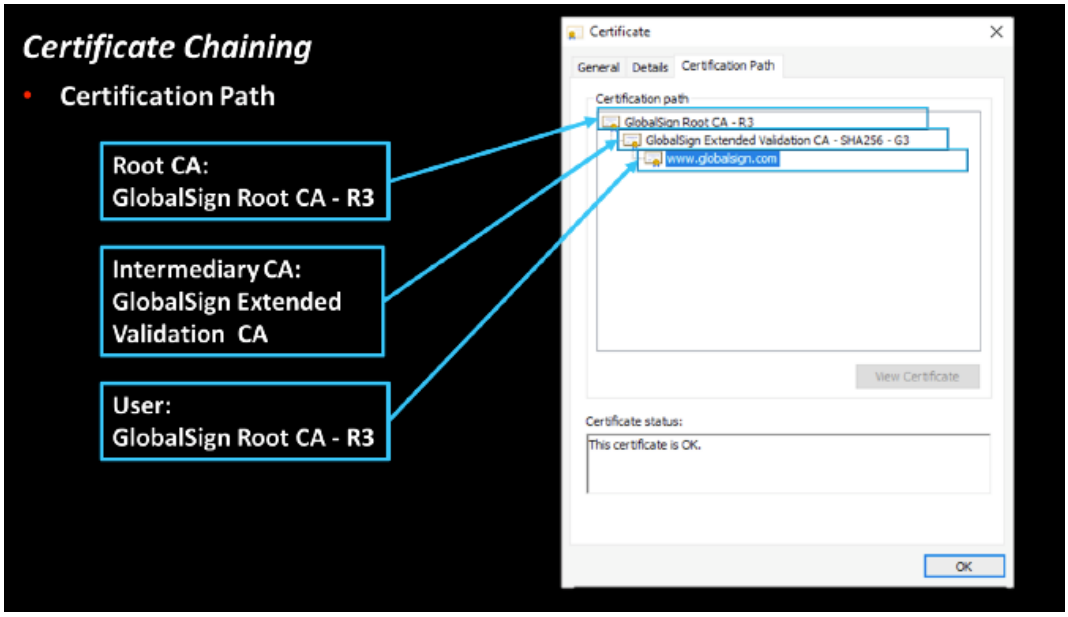
> **Tip**
> Certificate pinning prevents the compromising of the CA, certificate fraud, and SSL man-in-the-middle attacks.

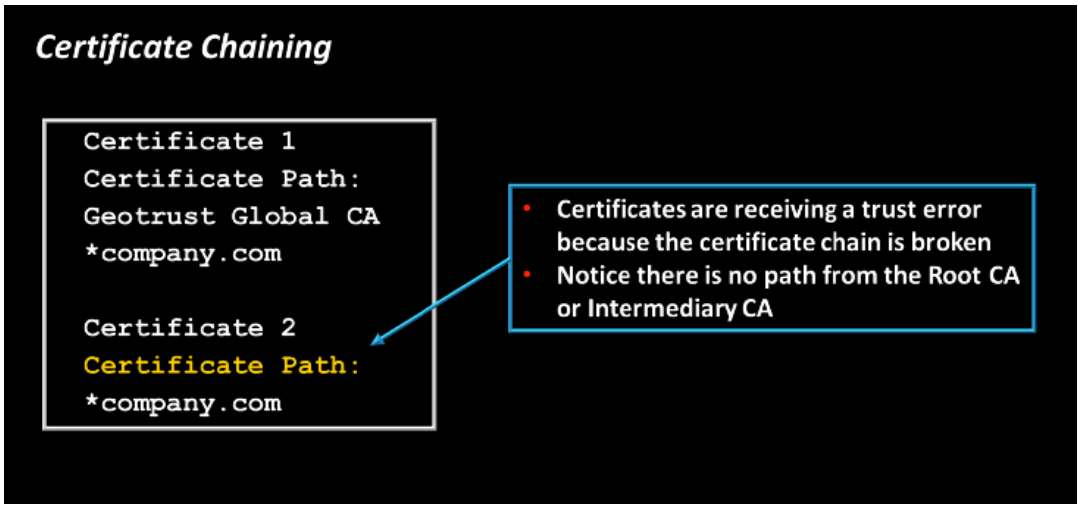
Certificate Trust

Certificates have some form of trust where the certificate can check whether or not it is valid. We are going to look at different trust models. You need to ensure that you know when each is used:

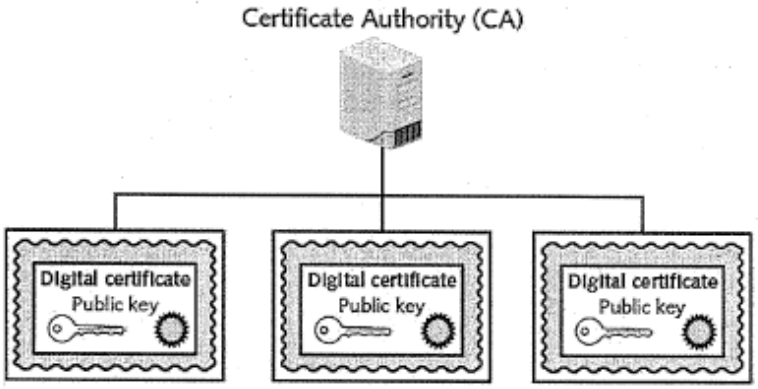
- **Trust Anchor**: A trust anchor in a PKI environment is the root certificate from which the whole chain of trust is derived; this is the root CA.
- **Trust Model**: A trust model proves the authenticity of a certificate; there are two trust models:
  a. **Hierarchical Trust Model**: This uses a hierarchy from the root CA down to the intermediary (also known as a subordinate); this is the normal PKI model. An example can be seen in the certificate hierarchy diagram earlier in this chapter.
  b. **Bridge Trust Model**: The bridge trust model is peer-to-peer, where two separate PKI environments trust each other. The certificate authorities communicate with each other, allowing for cross certification. Sometimes, this is referred to as the trust model.
- **Certificate Chaining**: This chain of trust is used to verify the validity of a certificate as it includes details of the CRL. The chain normally has three layers, the certificate vendor, the vendor's CA, and the computer where the certificate is installed.



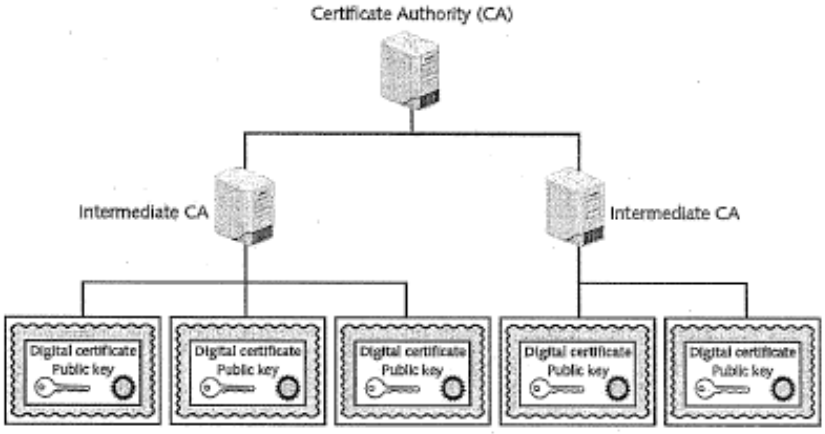

Certificate Chaining
Certification Path
Root CA:
GlobalSign Root CA - R3
Intermediary CA:
GlobalSign Extended Validation CA
User:
GlobalSign Root CA - R3
Certificate
General
Details
Certification Path
Certification path
GlobalSign Root CA - R3
GlobalSign Extended Validation CA - SHA256 - G3
www.globalsign.com
View Certificate
Certificate status:
This certificate is OK.
OK


Certificate Chaining
Certificate 1
Certificate Path:
Geotrust Global CA
*company.com
Certificate 2
Certificate Path:
*company.com
Certificates are receiving a trust error because the certificate chain is broken
Notice there is no path from the Root CA or Intermediary CA

Hierarchical Trust Model



Distributed Trust Model



Certificate and Key Management

Key management refers to operational considerations for the various stages in a key's life cycle. A key's life cycle may involve the following stages:

- Key generation—creating a secure key pair of the required strength, using the chosen cipher.
- Certificate generation—to identify the public part of a key pair as belonging to a subject (user or computer), the subject submits it for signing by the CA as a digital certificate with the appropriate key usage. At this point, it is critical to verify the identity of the subject requesting the certificate and only issue it if the subject passes identity checks.
- Storage—the user must take steps to store the private key securely, ensuring that unauthorized access and use is prevented. It is also important to ensure that the private key is not lost or damaged.
- Revocation—if a private key is compromised, the key pair can be revoked to prevent users from trusting the public key.
- Expiration and renewal—a key pair that has not been revoked expires after a certain period. Giving the key or certificate a "shelf-life" increases security. Certificates can be renewed with new key material.

Key management can be *centralized,* meaning that one administrator or authority controls the process, or *decentralized,* in which each user is responsible for his or her keys.

Certificate and key management can represent a critical vulnerability if not managed properly. If an attacker can obtain a private key, it puts both data confidentiality and identification/authentication systems at risk. If an attacker gains the ability to create signed certificates that appear to be valid, it will be easy to harvest huge amounts of information from the network as the user and computer accounts he or she sets up will be automatically trusted. Finally, if a key used for encryption is accidentally destroyed, the data encrypted using that key will be inaccessible, unless there is a backup or key recovery mechanism.

Key Recovery and Escrow

Keys such as the private key of a root CA must be subject to the highest possible technical and procedural access controls. If such a key were compromised, it would put the confidentiality and integrity of data processed by hundreds or thousands of systems at risk. Access to such critical encryption keys must be logged and audited and is typically subject to ***M*-of-*N* control**, meaning that of *N* number of administrators permitted to access the system, *M* must be present for access to be granted. *M* must be greater than 1, and *N* must be greater than *M*. For example, when *M* = 2 and *N* = 4, any two of four administrators must be present. Staff authorized to perform key management must be carefully vetted, and due care should be taken if these employees leave the business.

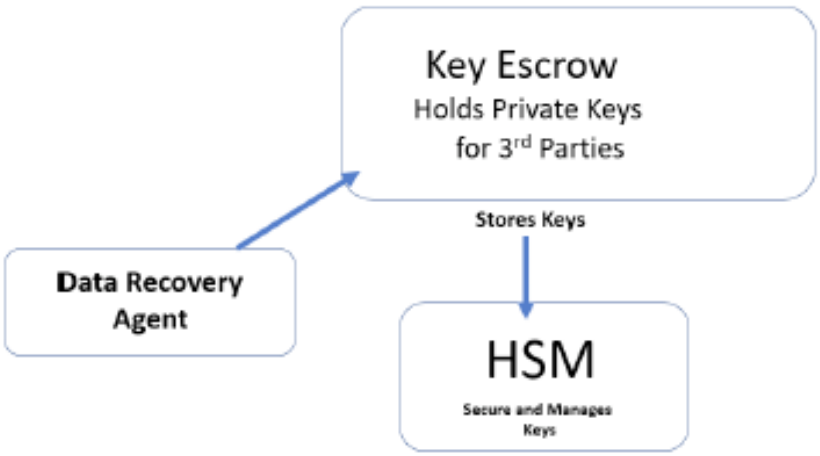
Another way to use M-of-N control is to split a key between several storage devices (such as three USB sticks, any two of which could be used to recreate the full key).

If the key used to decrypt data is lost or damaged, the encrypted data cannot be recovered unless a backup of the key has been made. A significant problem with key storage is that if you make multiple backups of a key, it is exponentially more difficult to ensure that the key is not compromised. However, if the key is not backed up, the storage system represents a single point of failure. Key recovery defines a secure process for backing up keys and/or recovering data encrypted with a lost key. This process might use *M*-of-*N* control to prevent unauthorized access to (and use of) the archived keys. **Escrow** means that something is held independently. In terms of key management, this refers to archiving a key (or keys) with a third party. This is a useful solution for organizations that don't have the capability to store keys securely themselves, but it invests a great deal of trust in the third party.

Certificate Management Concepts

We are now going to look at the different ways in which certificates are managed in a PKI environment, starting with the request for a new certificate and ending with different certificate formats. You must learn all of this information thoroughly as these aspects are heavily tested:

- **Certificate Signing Request** (**CSR**): This is the process of requesting a new certificate.
- **Key Escrow**: The key escrow holds the private keys for third parties and stores them in a **Hardware Security Module** (**HSM**):



- **Hardware Security Module (HSM)**: The HSM can be a piece of hardware attached to the server or a portable device that is attached to store the keys. See the preceding diagram for more on this. It stores and manages certificates.
- **Data Recovery Agent (DRA)**: If a user cannot access their data because their private key is corrupted, the DRA will recover the data. The DRA needs to get the private key from the key escrow.
- **Certificates**: There are two main certificate types: the *public key* and the *private key*. The public key is sent to third parties to encrypt the data, and the private key decrypts the data. If you think of the private key as your bank card, that's a thing you wouldn't give away. The public key is the deposit slip that is tied to your account. If you were in a room with 20 people who wanted to pay $20 into your account, you would definitely give them your deposit slip. You will always give your public key away because when you are encrypting data, you will always use the recipient's public key.

Key and Certificate Management

You might export a copy of the private key from this server to be held in escrow as a backup. For this usage, you must password-protect the key:

```
openssl rsa -aes256 -in www.key -out www.key.bak
```

You might need to convert the certificate format to make it compatible with an application server, such as Java. The following command takes a PEM-encoded certificate and outputs a DER binary-encoded certificate:

```
openssl x509 -outform der -in www.pem -out www.der
```

Another use case is to export a key and certificate for use in Windows:

```
openssl pkcs12 -export -inkey www.key -in www.pem
-out www.pfx
```

Asymmetric Cryptography Practices

Action	Whose key to use	Which key to use	Explanation
Bob wants to send Alice an encrypted message	Alice's key	Public key	When an encrypted message is to be sent, the recipient's, and not the sender's, key is used.
Alice wants to read an encrypted message sent by Bob	Alice's key	Private key	An encrypted message can be read only by using the recipient's private key.
Bob wants to send a copy to himself of the encrypted message that he sent to Alice	Bob's key	Public key to encrypt Private key to decrypt	An encrypted message can be read only by the recipient's private key. Bob would need to encrypt it with his public key and then use his private key to decrypt it.
Bob receives an encrypted reply message from Alice	Bob's key	Private key	The recipient's private key is used to decrypt received messages.
Bob wants Susan to read Alice's reply message that he received	Susan's key	Public key	The message should be encrypted with Susan's key for her to decrypt and read with her private key.
Bob wants to send Alice a message with a digital signature	Bob's key	Private key	Bob's private key is used to encrypt the hash.
Alice wants to see Bob's digital signature	Bob's key	Public key	Because Bob's public and private keys work in both directions, Alice can use his public key to decrypt the hash.

Protocol	Port	Name	Description
ARP	-	Address Resolution Protocol	Resolve IP address to MAC
TCP	-	Transmission Control Protocol	Connection-oriented network communication
UDP	-	User Datagram Protocol	Connectionless network communication
ICMP	-	Internet Control Message Protocol	Send management messages between clients (PING)
RTP	dynamic	Real-time Transport Protocol	Voice over IP media streaming
SRTP	dynamic	Secure Real-time Transport Protocol	Secure voice over IP media streaming
FTP	TCP 20-21	File transfer Protocol	Sends and receives files between systems
SSH	TCP 22	Secure Shell	Encrypted remote console access
SCP	TCP 22	Secure Copy	Simple file copy over SSH
SFTP	TCP 22	Secure File Transfer Protocol	SSH file transfer with file management
Telnet	TCP 23	Telecommunication Network	Insecure remote console access
SMTP	TCP 25	Simple Mail Transfer Protocol	Transfer email between mail servers (send mail)
TACACS+	TCP 49	Terminal Access Controller	AAA server used for device management

Protocol	Port	Name	Description
DNS	UDP 53 - TCP 53	Domain Name System	Convert domain names to IP address
DHCP	UDP 67-68	Dynamic Host Configuration Protocol	Update BOOTP
TFTP	UDP 69	Trivial File Transfer Protocol	A very simple file transfer application
HTTP	TCP 80	Hypertext Transfer Protocol	Web server communication
POP3	TCP 110	Post Office Protocol version 3	Receive email into a email client (does not store email)
NTP	UDP 123	Network Time Protocol	Used for network timing
IMAP4	TCP 143	Internet Message Access Protocol v4	Receive email into a email client (stores email)
SNMP	UDP 161-162	Simple Network Management Protocol	Gather metrics and manage network devices
LDAP	UDP 389 - TCP 389	Lightweight Directory Protocol	Directory Service Protocol
HTTPS	TCP 443	Hypertext Transfer Protocol Secure	Web server communication with encryption
TLS/SSL	TCP 443	Transport Layer Security and Socket Layer	Secure protocols for web browsing
NetBIOS	UPD 137	NetBIOS name service	Register, remove, and find services by name
NetBIOS	UDP 138	NetBIOS datagram Service	Connectionless data transfer
NetBIOS	TCP 139	NetBIOS session service	Connection-oriented data transfer

Protocol	Port	Name	Description
SMB	TCP 445	Server Message Block	Windows file transfer and printer sharing
SMTPS SSL/TLS	TCP 465	Simple Mail Transfer Protocol Secure	Secure transfer email between mail servers (send mail)
IPsec (IKE)	UDP 500	Internet Protocol Security	Authentication, integrity, confidentiality, encryption
LDAPS SSH	TCP 636	Lightweight Directory Protocol Secure	Directory Service Protocol over SSH
LDAPS SSL/TLS	TCP 636	Lightweight Directory Protocol Secure	Directory Service Protocol over SSL/TLS
IMAP4 SSL/TLS	TCP 993	Internet Message Access Protocol v4	Receive email into a email client (stores email) secure
POP3 SSL/TLS	TCP 995	Post Office Protocol version 3 Secure	Receive email into a email client secure
L2TP	UDP 1701	Layer 2 Tunneling Protocol	IPsec used to create L2TP to encrypt VPN traffic
PPTP	TCP 1723	Point-to-Point Tunneling Protocol	Used to encrypt VPN traffic
RADIUS	UDP 1812	Remote Authentication Dial-in User	User authentication
RADIUS	UDP 1813	Remote Authentication Dial-in User	Accounting
RDP	TCP/UDP 3389	Remote Desktop Protocol	Graphical display of remote devices
MS SQL	TCP 1433	Microsoft SQL Server	Used to encrypt databases

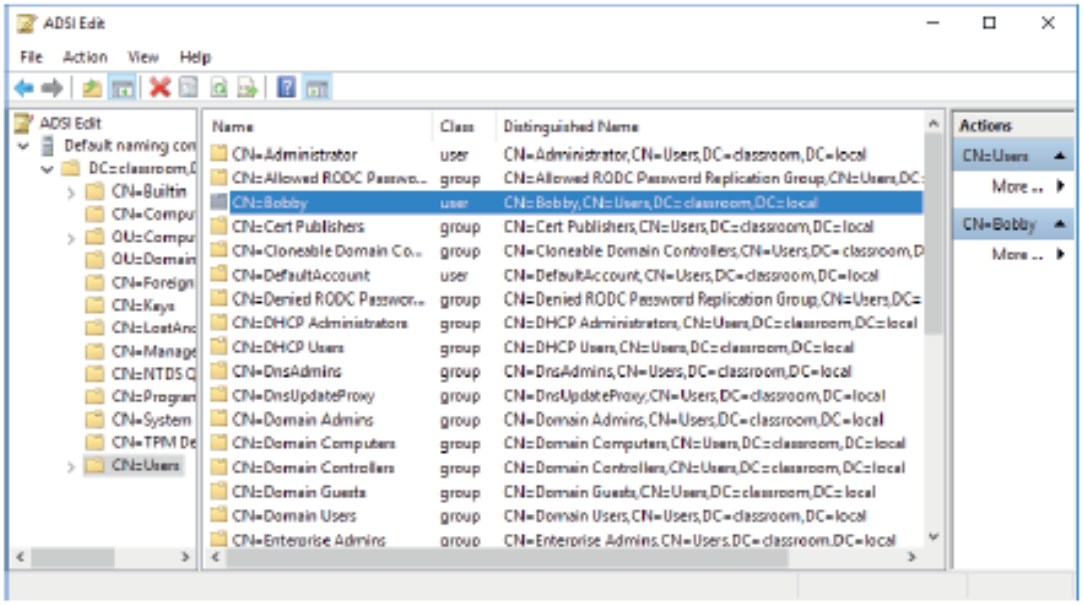
Kerberos UDP 88

LDAP & Active Directory

Directory Services

Directory services are the principal means of providing privilege management and authorization on an enterprise network, storing information about users, computers, security groups/roles, and services. A directory is like a database, where an object is like a record, and things that you know about the object (attributes) are like fields. In order for products from different vendors to be interoperable, most directories are based on the same standard. The Lightweight Directory Access Protocol (LDAP) is a protocol widely used to query and update X.500 format directories.

A distinguished name (DN) is a unique identifier for any given resource within an X.500-like directory. A distinguished name is made up of attribute=value pairs, separated by commas. The most specific attribute is listed first, and successive attributes become progressively broader. This most specific attribute is also referred to as the relative distinguished name, as it uniquely identifies the object within the context of successive (parent) attribute values.



Browsing objects in an Active Directory LDAP schema. (Screenshot used with permission from Microsoft.)

The types of attributes, what information they contain, and the way object types are defined through attributes (some of which may be required, and some optional) is described by the directory schema. Some of the attributes commonly used include common name (CN), organizational unit (OU), organization (O), country (C), and domain component (DC). For example, the distinguished name of a web server operated by Widget in the UK might be:

```
CN=WIDGETWEB, OU=Marketing, O=Widget, C=UK,
DC=widget, DC=foo
```

Secure Directory Services

A network directory lists the subjects (principally users, computers, and services) and objects (such as directories and files) available on the network plus the permissions that subjects have over objects. A directory facilitates authentication and authorization, and it is critical that it be maintained as a highly secure service. Most directory services are based on the **Lightweight Directory Access Protocol (LDAP)**, running over port 389. The basic protocol provides no security and all transmissions are in plaintext, making it vulnerable to sniffing and man-in-the-middle attacks. Authentication (referred to as binding to the server) can be implemented in the following ways:

- No authentication—anonymous access is granted to the directory.
- Simple bind—the client must supply its distinguished name (DN) and password, but these are passed as plaintext.
- Simple Authentication and Security Layer (SASL)—the client and server negotiate the use of a supported authentication mechanism, such as Kerberos. The STARTTLS command can be used to require encryption (sealing) and message integrity (signing). This is the preferred mechanism for Microsoft's Active Directory (AD) implementation of LDAP.
- **LDAP Secure (LDAPS)**—the server is installed with a digital certificate, which it uses to set up a secure tunnel for the user credential exchange. LDAPS uses port 636.

If secure access is required, anonymous and simple authentication access methods should be disabled on the server.

Generally two levels of access will need to be granted on the directory: read-only access (query) and read/write access (update). This is implemented using an access control policy, but the precise mechanism is vendor-specific and not specified by the LDAP standards documentation.

Unless hosting a public service, the LDAP directory server should also only be accessible from the private network. This means that the LDAP port should be blocked by a firewall from access over the public interface. If there is integration with other services over the Internet, ideally only authorized IPs should be permitted.

Directory Services

Identity management in a corporate environment will use a directory database. This is a centralized database that will authenticate all domain users. We are going to look at Microsoft's Active Directory, where a protocol called the **Lightweight Directory Access Protocol** (**LDAP**) manages the users in groups. Let's look at how it works.

LDAP

Most companies have identity and access services through a directory that stores objects such as users and computers as X500 objects. These were developed by the **International Telecommunication Union** (**ITU**). These objects form what is called a distinguished name and are organized and stored by the LDAP.

There are only three values in X500 objects; these are DC (domain), **Organization Unit** (**OU**), and CN (anything else).

In this example, we have a domain called *Domain A* and an OU called *Sales*; this is where all of the sales department users and computers reside. We can see inside the Sales OU a computer called **Computer 1**:

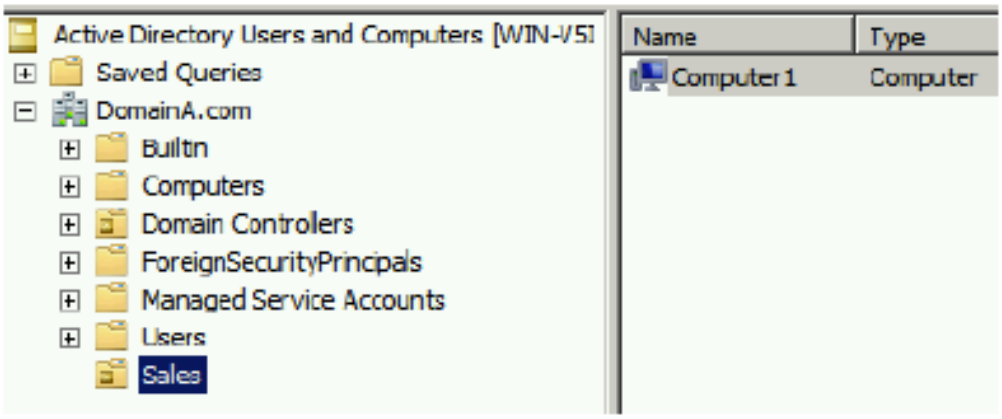


Figure 3.8 – Active Directory

When creating the X500 object, we start off with the object itself, **Computer 1**, and then continue up through the structure. As **Computer 1** is neither an OU nor a domain, we give it a value of CN. Then we move up the structure to **Sales**. As it is an OU, we give it that value. **Computer 1** is a CN, **Sales** is an OU, and the domain is divided into two portions, each having the value of DC. The distinguished name is here: **CN=Computer1**, **OU=Sales**, **DC=DomainA**, **DC=com**.

The way it is stored in the Active Directory can be viewed using a tool called **ADSI Edit**:

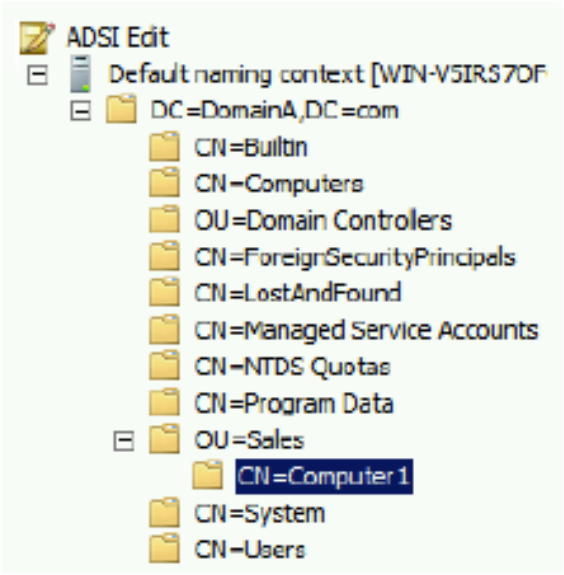


Figure 3.9 – ADSI Edit

LDAP is the active directory storeman responsible for storing the X500 objects. When the Active Directory is searched, then LDAP provides the information required. LDAPS is the secure version of LDAP.

Here are some examples.

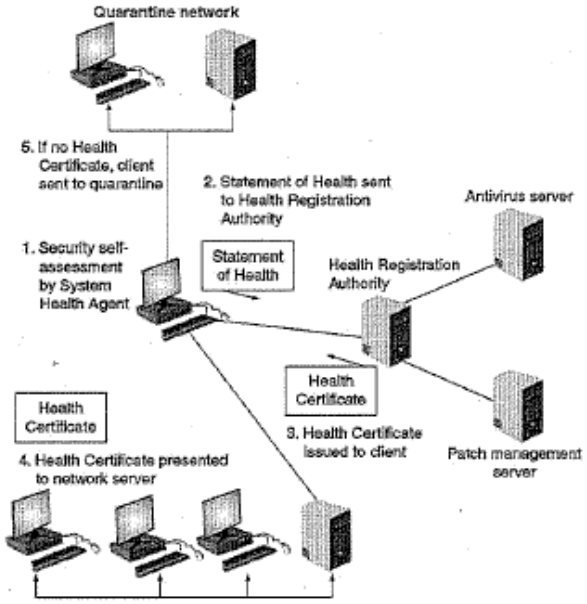
- **Example 1**: If I want to know how many people are in the IT OU, I can search the Active Directory. LDAP provides the search and returns a reply saying that the IT department has 10 members.
- **Example 2**: I am searching the Active Directory for a user called Fred. Once again, LDAP finds the user. If you have 10,000 people in your domain, you will have them in different OUs to make it easier to find and manage them. However, if you need to find someone, this will still be difficult. That is why we need LDAP to perform the search. It saves time.

Network Access Control (NAC)

C
Inst

- **Network access control (NAC)** - Examines current state of system or network device before allowing network connection
- Device must meet set of criteria
- If not met, NAC allows connection to quarantine network until deficiencies corrected

Network Access Control (NAC) Framework



Network Access Control (NAC).

- Host agent health checks
- Permanent NAC agent
- Dissolvable NAC agent
- Agentless NAC

Data Loss Prevention (DLP)

Securing Data

- Work today involves electronic collaboration, so data must flow freely but securely
- **Data loss prevention (DLP)** - System of security tools used to recognize and identify critical data and ensure it is protected
- Goal is protect data:
  - **Data in-use** - Data actions being performed by "endpoint devices"
  - **Data in-transit** - Actions that transmit the data across a network
  - **Data at-rest** - Stored on electronic media

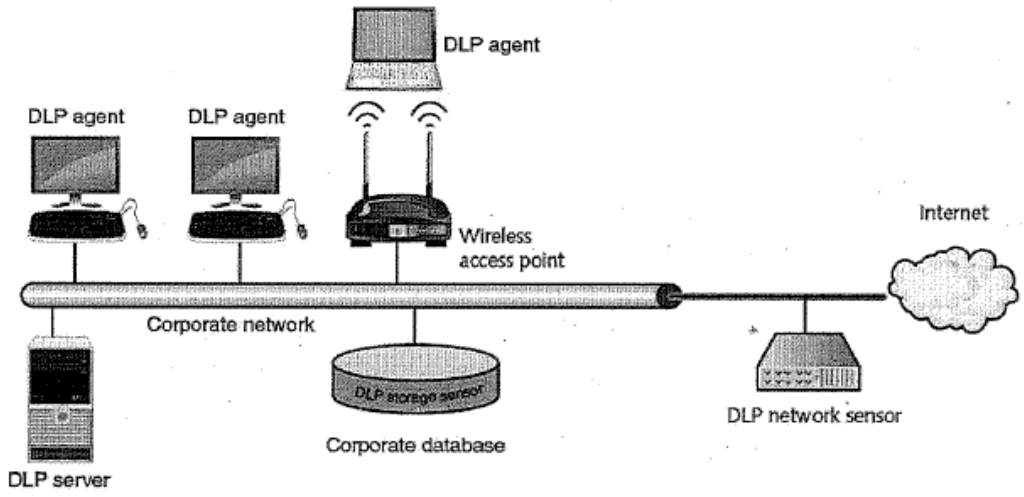
DLP Techniques

- *Content inspection* - Security analysis of transaction and takes context into account
- DLP systems also can use *index matching:*
  - Documents identified as needing protection, such as the program source code for a new software application, are analyzed by DLP system
  - Complex computations are conducted based on analysis

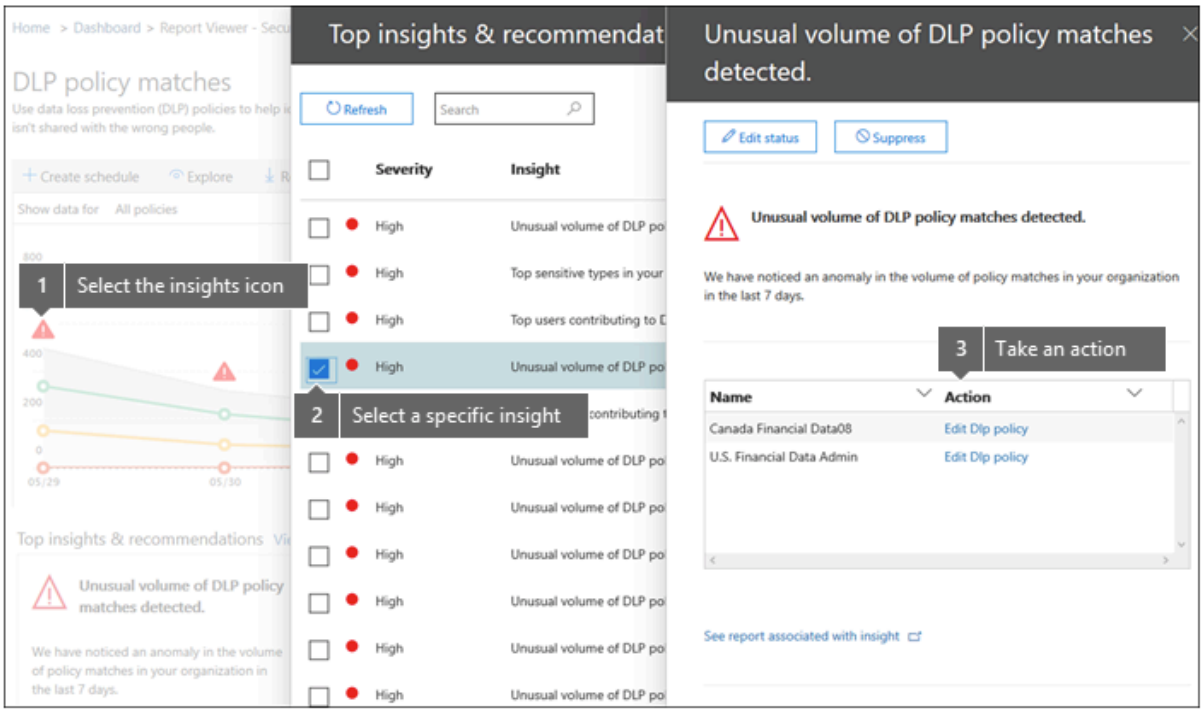
DLP Sensors

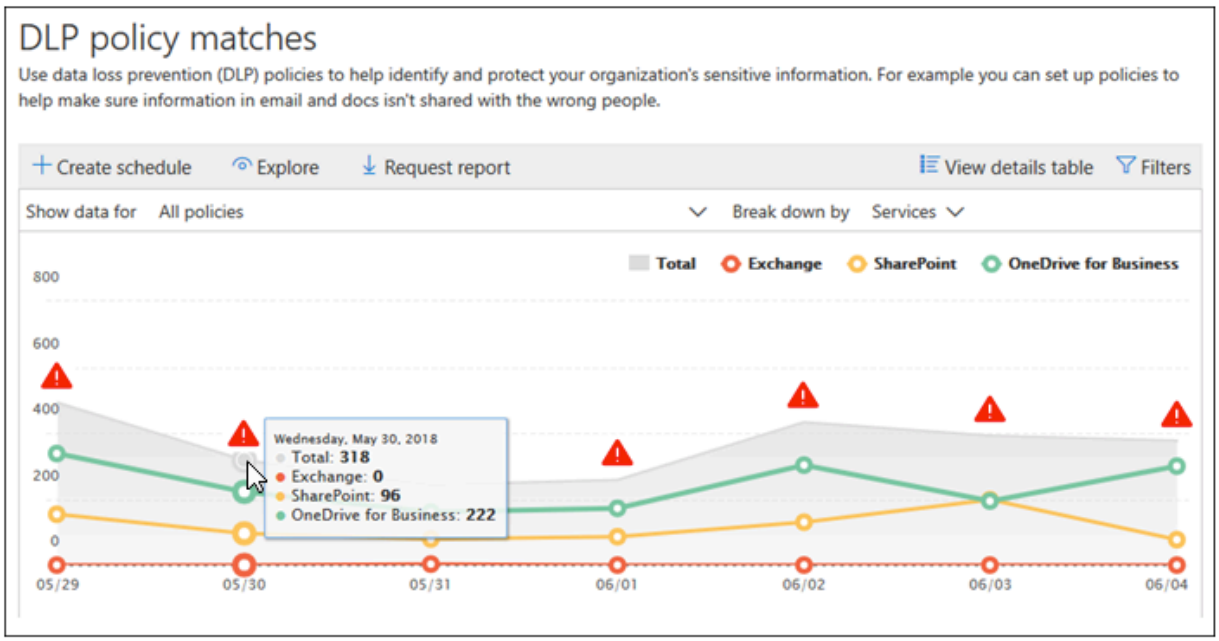
- DLP sensors:
  - *DLP network sensors* - Installed on perimeter of network to protect data in-transit by monitoring all network traffic
  - *DLP storage sensors* - Sensors on network storage devices are designed to protect data at-rest
  - *DLP agent sensors* - Sensors are installed on each host device (desktop, laptop, tablet, etc.) and protect data in-use

DLP Architecture



DLP Reports (Samples)




DLP policy matches
Use data loss prevention (DLP) policies to help identify and protect your organization's sensitive information. For example you can set up policies to help make sure information in email and docs isn't shared with the wrong people.
Create schedule
Explore
Request report
View details table
Filters
Show data for
All policies
Break down by
Services
Total
Exchange
SharePoint
OneDrive for Business
800
600
400
200
0
Wednesday, May 30, 2018
Total: 318
Exchange: 0
SharePoint: 96
OneDrive for Business: 222
05/29
05/30
05/31
06/01
06/02
06/03
06/04

Jump Servers

A jump server, also known as a jump host or jump box, is a hardened host that could be used as an intermediary device or as a gateway for administrators who would then connect to other servers for remote administration. It would only have secure remote access tools installed. It could be used to SSH into the screened subnet or an Azure public network.

Load Balancer

A network load balancer is a device that is used when there is a high volume of traffic coming into the company's network or web server. It can be used to control access to web servers, video conferencing, or email.

The web traffic, shown in *Figure 7.13*, comes into the load balancer from the **Virtual IP address** (**VIP**) on the frontend and is sent to one of the web servers in the server farm:

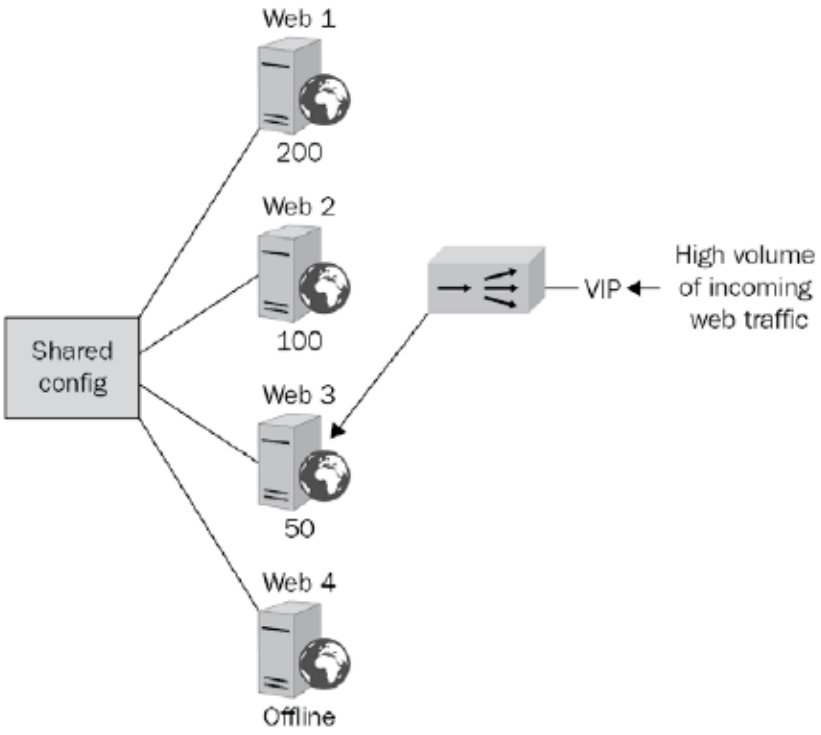


Figure 7.13 – Load balancer

Load Balancer Scheduling

Scheduling is how the load is distributed by the load balancer, let's look at these options in turn:

- **Least Utilized Host**: The benefits of a load balancer are that it knows the status of all of the web servers in the server farms and knows which web servers are the least utilized by using a scheduling algorithm.

  Example: The load balancer (see *Figure 7.13*) has selected to send the request to `Web 3`, which has the least number of requests (`50`), and `Web 4` will not be considered as it is currently offline. A user requesting three different pages may obtain them from different web servers but may not know this as the load balancer is optimizing the delivery of the web pages to the user.

- **Affinity**: When the load balancer is set to **Affinity**, the request is sent to the same web server based on the requester's IP address. This is also known as *persistence* or a *sticky session*, where the load balancer uses the same server for the session.
- **DNS Round Robin**: While using DNS round robin, when the request comes in, the load balancer contacts the DNS server and rotates the request based on the lowest IP address first. It rotates around `Web 1`, `Web 2`, and `Web 3`, and then keeps the sequence going by going back to `Web 1` on a rotational basis:

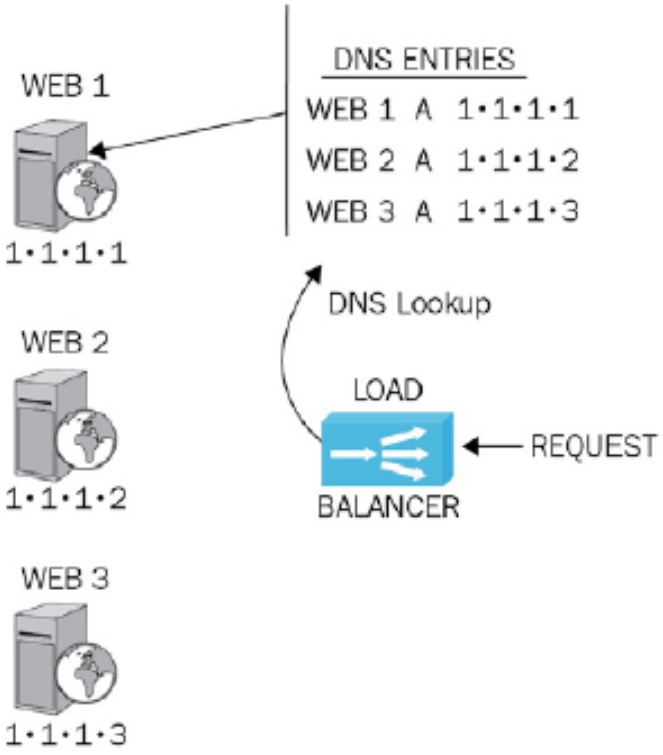


Figure 7.14 – DNS Round Robin

Load Balancer Configurations

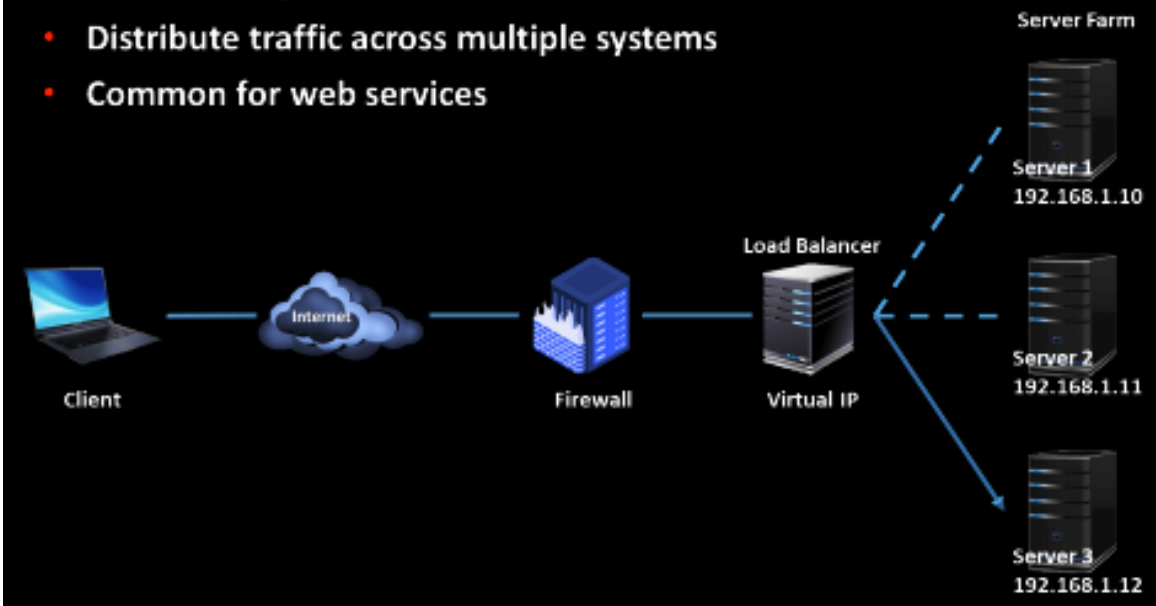
There are many ways to set up a load balancer and we are going to look at each of these in turn:

- **Active/Active**: With active/active load balancers, the load balancers act like an array, dealing with the traffic together as they are both active. They cache the requests. If someone comes back for a second time to look at a web page, they will get the same load balancer that dealt with their first request. The downside is that since they are working close to full capacity, should one fail, then it will look as if the load balancers are going much slower. This is due to one load balancer dealing with the workload of two.

- **Active/Passive**: With active/passive load balancers, you have a pair of load balancers. The active node is fulfilling load balancing duties and the passive node is listening and monitoring the active node. Should the active node fail, then the passive node will take over, giving you redundancy.

Load Balancing

- Distribute traffic across multiple systems
- Common for web services



Load Balancing

- Round-robin
  - The load balancer sends the first request to Server 1, the second request to Server 2, and so on
- Affinity
  - Sends requests to the same server based on the requestor's IP address
- Weighted round-robin
  - Prioritize the server use
- Dynamic round-robin
  - Monitor the server load and distributed to the server with the lowest use

Active/Active

In an *active/active* scheme, all the load balancers are active, sharing the load-balancing duties. Active/active load balancing can have performance efficiencies, but it is important to watch the overall load. If the overall load cannot be covered by $N-1$ load balancers (that is, one fails), then the failure of a load balancer will lead to session interruption and traffic loss. Without a standby passive system to recover the lost load, the system will trim the load based on capacity, dropping requests that the system lacks capacity to service.

Active/Passive

For high-availability solutions, having a single load balancer creates a single point of failure (SPOF). It is common to have multiple load balancers involved in the balancing work. In an *active/passive* scheme, the primary load balancer is actively doing the balancing while the secondary load balancer passively observes and is ready to step in any time the primary system fails.

Scheduling

When a load balancer moves loads across a set of resources, it decides which machine gets a request via a *scheduling* algorithm. There are a couple of commonly used scheduling algorithms: affinity-based scheduling and round-robin scheduling.

Affinity

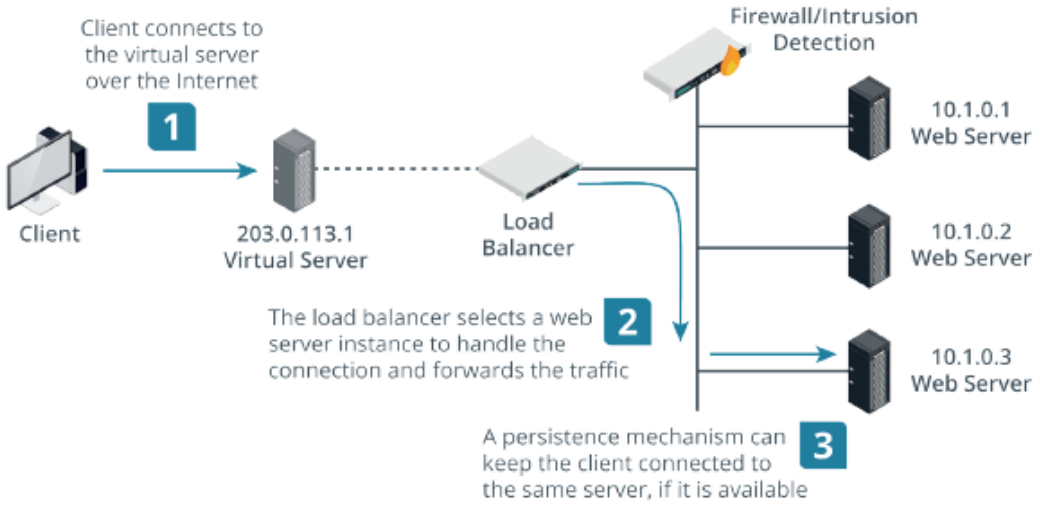
Affinity-based scheduling is designed to keep a host connected to the same server across a session. Some applications, such as web applications, can benefit from affinity-based scheduling. The method used by affinity-based scheduling is to have the load balancer keep track of where it last balanced a particular session and direct all continuing session traffic to the same server. If it is a new connection, the load balancer establishes a new affinity entry and assigns the session to the next server in the available rotation.

Round-Robin

Round-robin scheduling involves sending each new request to the next server in rotation. All requests are sent to servers in equal amounts, regardless of the server load. Round-robin schemes are frequently modified with a weighting factor, known as weighted round-robin, to take the server load or other criteria into account when assigning the next server.

There are two main types of load balancers:

- Layer 4 load balancer—basic load balancers make forwarding decisions on IP address and TCP/UDP port values, working at the transport layer of the OSI model.
- Layer 7 load balancer (content switch)—as web applications have become more complex, modern load balancers need to be able to make forwarding decisions based on application-level data, such as a request for a particular URL or data types like video or audio streaming. This requires more complex logic, but the processing power of modern appliances is sufficient to deal with this.



Topology of basic load balancing architecture. (Images © 123RF.com.)

Virtual Firewalls

Virtual firewalls are usually deployed within data centers and cloud services. A virtual firewall can be implemented in three different ways:

- Hypervisor-based—this means that filtering functionality is built into the hypervisor or cloud provisioning tool. You can use the cloud's web app or application programming interface (API) to write access control lists (ACLs) for traffic arriving or leaving a virtual host or virtual network.
- Virtual appliance—this refers to deploying a vendor firewall appliance instance using virtualization, in the same way you might deploy a Windows or Linux guest OS.
- Multiple context—this refers to multiple virtual firewall instances running on a hardware firewall appliance. Each context has a separate interface and can perform a distinct filtering role.

While they can be deployed like "regular" firewalls for zone-based routing and filtering, virtual firewalls most significant role is to support the east-west security and zero-trust microsegmentation design paradigms. They are able to inspect traffic as it passes from host-to-host or between virtual networks, rather than requiring that traffic be routed up to a firewall appliance and back.

Open-Source versus Proprietary Firewalls

The ability to inspect source code will be a requirement for high-security environments that cannot rely on implicit trust when selecting vendors. The code underpinning appliance-based, software, and virtual firewalls can be developed as open-source or proprietary or somewhere in between:

- Wholly proprietary—implemented as a proprietary OS, such as Cisco ASA, Juniper JunOS, PaloAlto PANOS, or Barracuda's Windows-based appliance.
- Mostly proprietary—developed from a Linux kernel, but with proprietary features added. Examples include Check Point IPSO, FortiGate FortiOS, and Sonicwall. Any code developed from a GPL source should be available, but in general terms these products cannot be used independently of a commercial contract with the vendor.
- Wholly open-souce—these can be used independently of the vendor, but the vendors typically have commercial appliances and support contracts too. Examples include pfSense and Smoothwall.

In determining whether to follow a self-installed versus supported deployment, as well as the core appliance code, you need to consider access to support, update availability, and access to subscription-based features, such as signatures and threat feeds.

Next-Generation Firewalls and Content Filters

While intrusion detection was originally produced as standalone software or appliances, its functionality very quickly became incorporated into a new generation of firewalls. The original **next-generation firewall (NGFW)** was released as far back as 2010 by Palo Alto. This product combined application-aware filtering with user account-based filtering and the ability to act as an intrusion prevention system (IPS). This approach was quickly adopted by competitor products. Subsequent firewall generations have added capabilities such as cloud inspection and combined features of different security technologies.

Content/URL Filter

A firewall has to sustain high loads, and overloads can increase latency or even cause outages. The high complexity of application-aware NGFW and UTM solutions can reduce their suitability as an edge device, because while they might provide high confidentiality and integrity, lower throughput reduces availability. One solution to this is to treat security solutions for server traffic differently from that for user traffic. User traffic refers to web browsing, social networking, email, and video/VoIP connections initiated by local network clients.

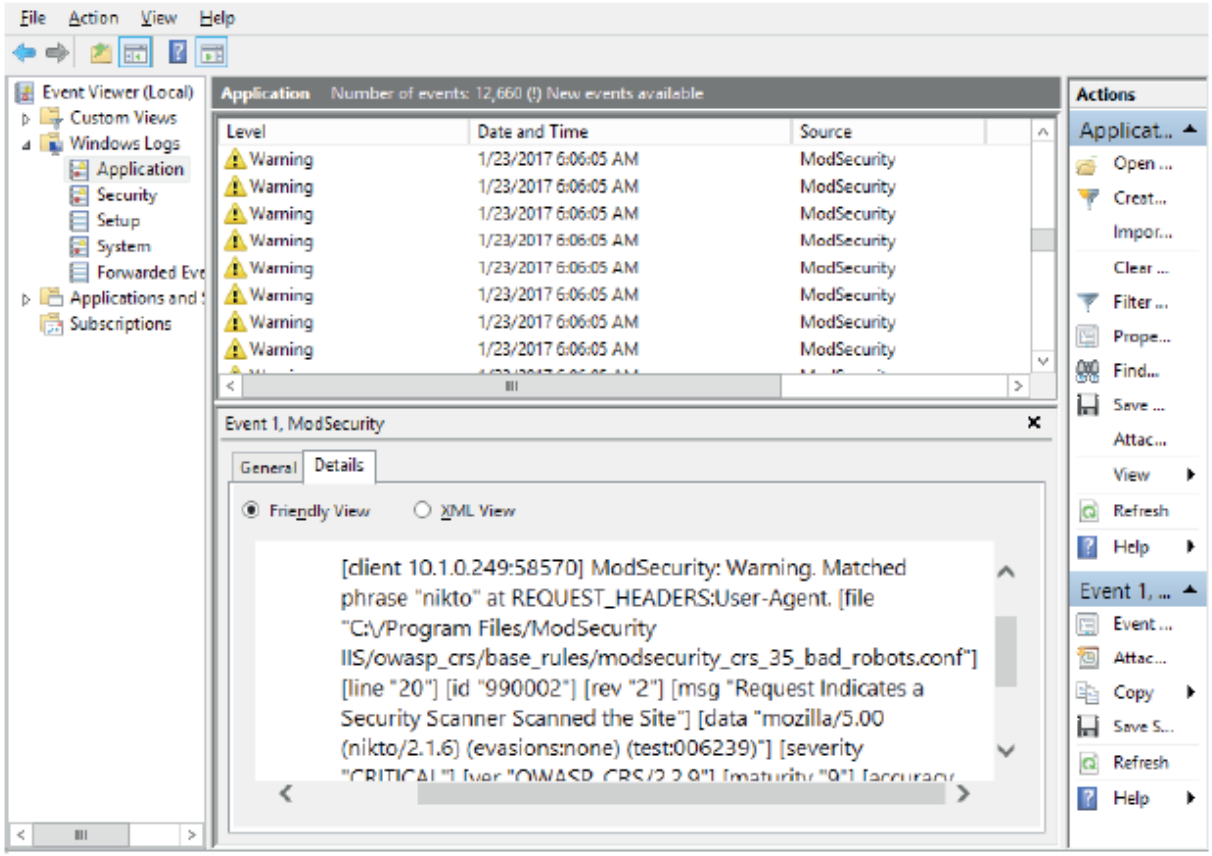
Consequently, where a stateful or NGFW firewall may be deployed for application server traffic, the job of filtering user traffic is often performed by a separate appliance or proxy host. A **content filter** is designed to apply a number of user-focused filtering rules, such as blocking uniform resource locators (URLs) that appear on content blacklists or applying time-based restrictions to browsing. Content filters are now usually implemented as a class of product called a **secure web gateway (SWG)**. As well as filtering, a SWG performs threat analysis and often integrates the functionality of data loss prevention (DLP) and cloud access security brokers (CASB) to protect against the full range of unauthorized egress threats, including malware command and control and data exfiltration.

Next-Generation Firewall Integration

An analytics-driven next-gen antivirus product is likely to combine with the perimeter and zonal security offered by next-gen firewalls. For example, detecting a threat on an endpoint could automate a firewall policy to block the covert channel at the perimeter, isolate the endpoint, and mitigate risks of the malware using lateral movement between hosts. This type of functionality is set out in more detail in Sophos's white paper on synchronized security (sophos.com/en-us/lp/synchronized-security.aspx).

Web Application Firewalls

A **web application firewall (WAF)** is designed specifically to protect software running on web servers and their backend databases from code injection and DoS attacks. WAFs use application-aware processing rules to filter traffic and perform application-specific intrusion detection. The WAF can be programmed with signatures of known attacks and use pattern matching to block requests containing suspect code. The output from a WAF will be written to a log, which you can inspect to determine what threats the web application might be subject to.



A WAF may be deployed as an appliance or as plug-in software for a web server platform. Some examples of WAF products include:

- ModSecurity (modsecurity.org) is an open source (sponsored by Trustwave) WAF for Apache, nginx, and IIS.
- NAXSI (github.com/nbs-system/naxsi) is an open source module for the nginx web server software.
- Imperva (imperva.com) is a commercial web security offering with a particular focus on data centers. Imperva markets WAF, DDoS, and database security through its SecureSphere appliance.

Cloud Firewall Security

As in an on-premises network, a firewall determines whether to accept or deny/discard incoming and outgoing traffic. Firewalls work with multiple accounts, VPCs, subnets within VPCs, and instances within subnets to enforce the segmentation required by the architectural design. Segmentation may be needed for many different reasons, including separating workloads for performance and load balancing, keeping data processing within an isolated segment for compliance with laws and regulations, and compartmentalizing data access and processing for different departments or functional requirements.

Filtering decisions can be made based on packet headers and payload contents at various layers, identified in terms of the OSI model:

- Network layer (layer 3)—the firewall accepts or denies connections on the basis of IP addresses or address ranges and TCP/UDP port numbers (the latter are actually contained in layer 4 headers, but this functionality is still always described as basic layer 3 packet filtering).
- Transport layer (layer 4)—the firewall can store connection states and use rules to allow established or related traffic. Because the firewall must maintain a state table of existing connections, this requires more processing power (CPU and memory).
- Application layer (layer 7)—the firewall can parse application protocol headers and payloads (such as HTTP packets) and make filtering decisions based on their contents. This requires even greater processing capacity (or load balancing), or the firewall will become a bottleneck and increase network latency.

While you can use cloud-based firewalls to implement on-premises network security, here we are primarily concerned with the use of firewalls to filter traffic within and to and from the cloud itself. Such firewalls can be implemented in several ways to suit different purposes:

- As software running on an instance. This sort of host-based firewall is identical to ones that you would configure for an on-premises host. It could be a stateful packet filtering firewall or a web application firewall (WAF) with a ruleset tuned to preventing malicious attacks. The drawback is that the software consumes instance resources and so is not very efficient. Also, managing the rulesets across many instances can be challenging.
- As a service at the virtualization layer to filter traffic between VPC subnets and instances. This equates to the concept of an on-premises network firewall.

Native cloud application-aware firewalls incur transaction costs, typically calculated on time deployed and traffic volume. These costs might be a reason to choose a third-party solution instead of the native control.

Wireless Bandwidth/Band Selection

There are different wireless standards, and we need to know the limitations of each. The band selection is also known as the frequency:

IEEE Standard	Frequency Band	Maximum Allowable Streams
802.11a	5 GHz	1
802.11b	2.4 GHz	1
802.11g	2.4 GHz	1
802.11n	2.4 GHz and 5 GHz	4
802.11ac	5 GHz	8
802.11ax	2.4 GHz and 5 GHz	16

Wireless Channels

In the Security+ exam, the wireless channels go from *channel 1* up to *channel 11*, and the device placement should be as follows:

- **Channel 1**: Your first wireless device
- **Channel 11**: Your second wireless device
- **Channel 6**: Your third wireless device

We place the device's channels as far apart as possible to prevent the overlap of adjacent channels and interference. Wireless devices can suffer interference from elevators, baby monitors, cordless phones, metal racking, and load-bearing walls, to name but a few things.

Wireless Antenna Types

WAP uses antennas to operate. There are three main antenna types:

- **Omnidirectional**: Omnidirectional antennas provide the most coverage as they transmit over 360 degrees.
- **Directional**: Directional antennas transmit only in one direction. Therefore, if the antenna is pointing in the wrong direction, there will be no connection to the wireless network.
- **Yagi**: A Yagi fin is an antenna that can transmit in two directions. Therefore, it is suitable for placement between two buildings.

Wireless Coverage

One of the security implications of having a wireless network is to ensure that wireless networks will have coverage. This will give access to resources in a timely fashion without the coverage being extended outside of the companies' boundaries where it could be hacked.

Let's look at options that need to be considered before setting up a wireless network:

- **Site Survey**: Before we install a wireless network, we need to complete a site survey so that we identify what could cause interference with the wireless network. In certain areas, we may need an extra WAP because of potential interference with the network. If we install a wireless network and it does not function properly or runs at a slow speed, then we have not carried out a thorough site survey.

- **WAP Placement**: There are many things that could interfere with or prevent your wireless network from working, and these include load-bearing walls, cordless phones, microwaves, elevators, metal frames, metal doors, and radio waves.
- **Heat Map**: A heat map shows your wireless coverage. The red and orange areas indicate good coverage, but the blue areas indicate poor coverage:

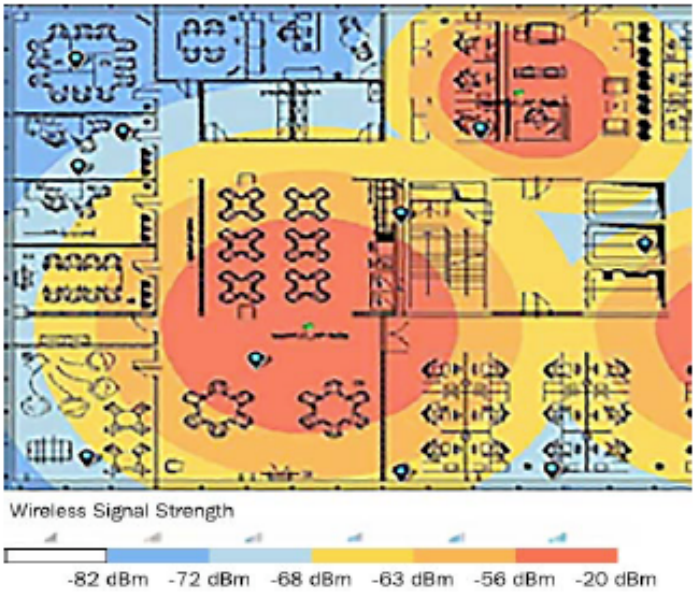


Figure 8.3 – Heat map

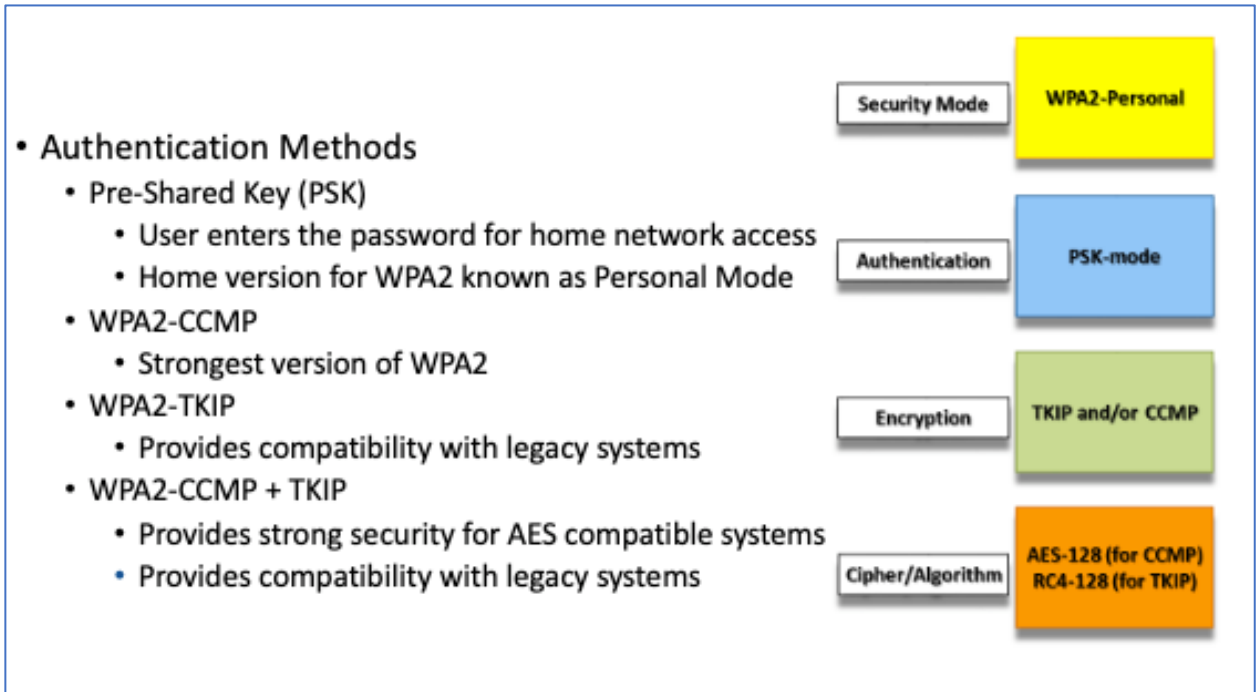
The heat map could also show green as poor coverage and it can help you identify where you have channel overlap.

- **Low-Power Directional Antennas**: If the wireless network goes outside of the boundary of a company's network, it may be hacked or attacked. To prevent these attacks, we turn down the power of the WAP and this reduces the distance of the wireless network coverage.
- **Wireless Speed Slow**: If the speed of the wireless network is very slow, we may be too far away from the WAP or the connection may be saturated as a result of downloading large files.

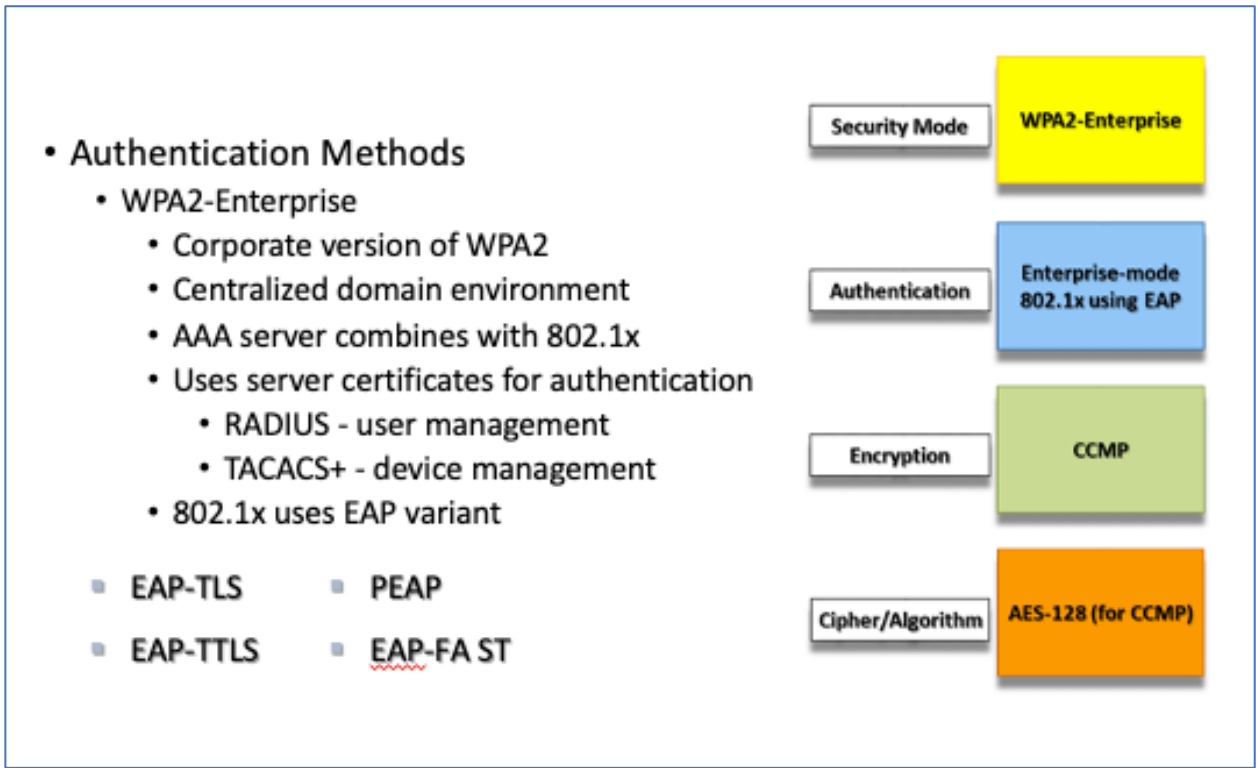
Exam tip

If my newly installed WLAN is not fully functional, we may not have carried out the site survey properly or placed it incorrectly.

WPA2 personal



WPA2 Enterprise



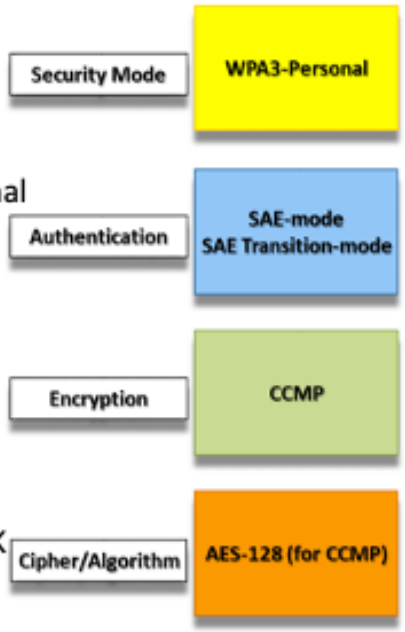
WPA3

- Authentication Methods
  - Wi-Fi Protected Access Version 3 (WPA 3)
    - Released in 2018 to address the weaknesses in WPA2
    - Uses a much stronger 256-bit Galois/Counter Mode Protocol (GCMP-256) for encryption also known as:
      - AES256_GCM
    - Features of WPA3 include:
      - WPA3-SAE (Simultaneous Authentication of Equals)

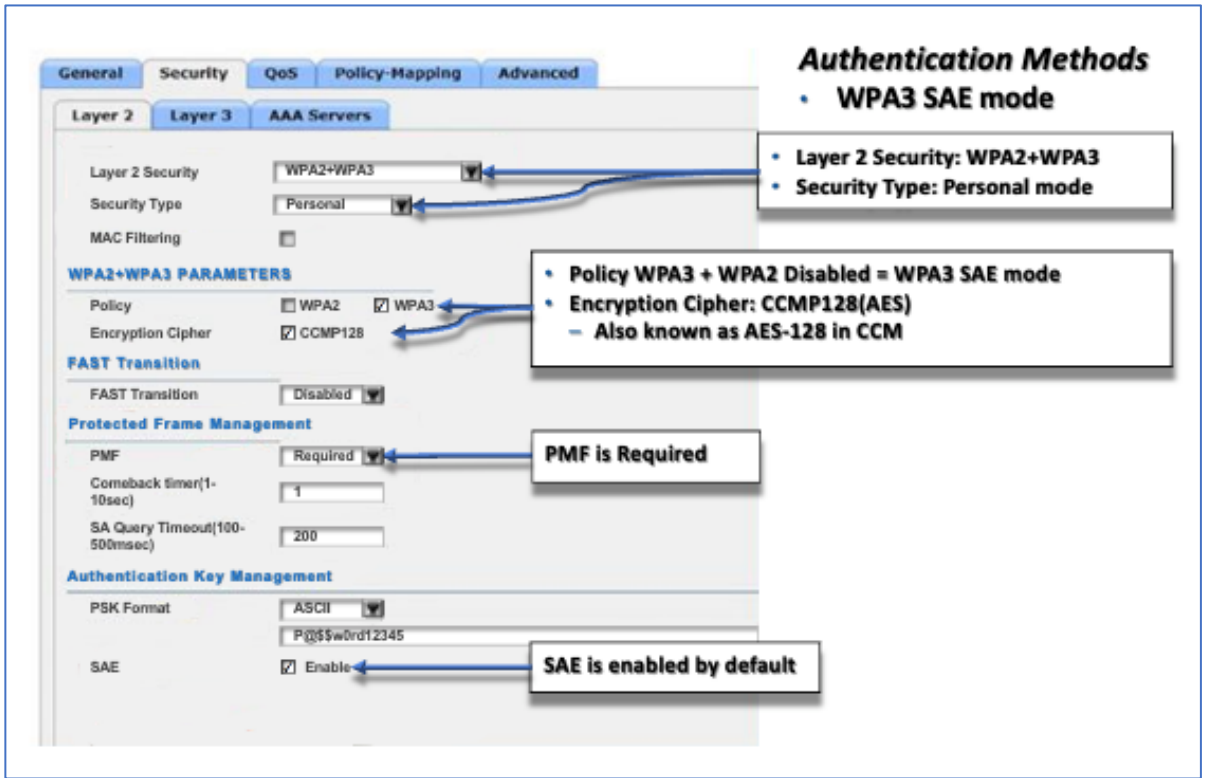
- **WPA3-SAE Transition**
- **WPA3-Enterprise**
- **PMF (Protected Management Frames)**
- **Wi-Fi Easy Connect**
- **Wi-Fi Enhanced Open**

WPA3-SAE

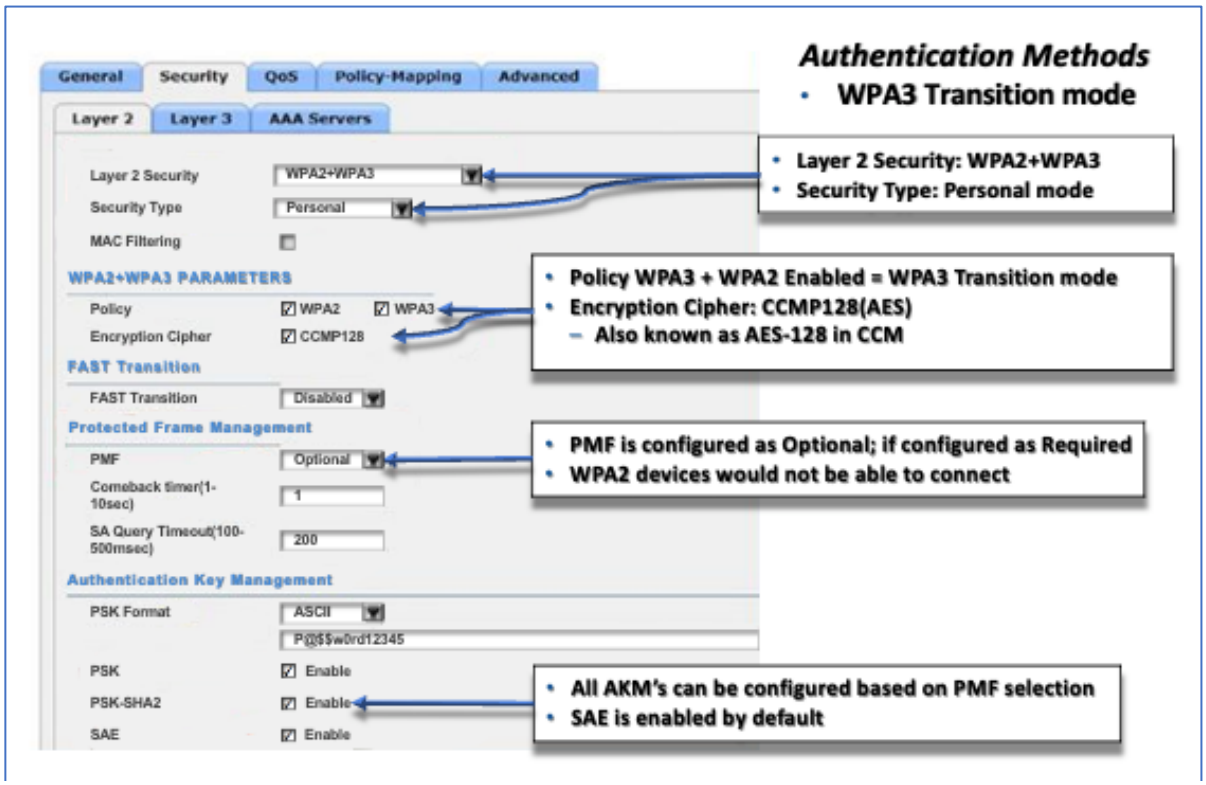
- Authentication Methods
  - WPA3-SAE
    - Home version of WPA3 known as WPA3-Personal
    - Replaces the WPA2-PSK
    - Protects against brute-force attacks
    - Provides Perfect Forward Secrecy (PFS)
    - Ensures session keys cannot be compromised
    - Simpler passwords for home network access
    - Immune to offline attacks
  - WPA3-SAE Transition
    - Allows backwards compatibility with WPA2-PSK



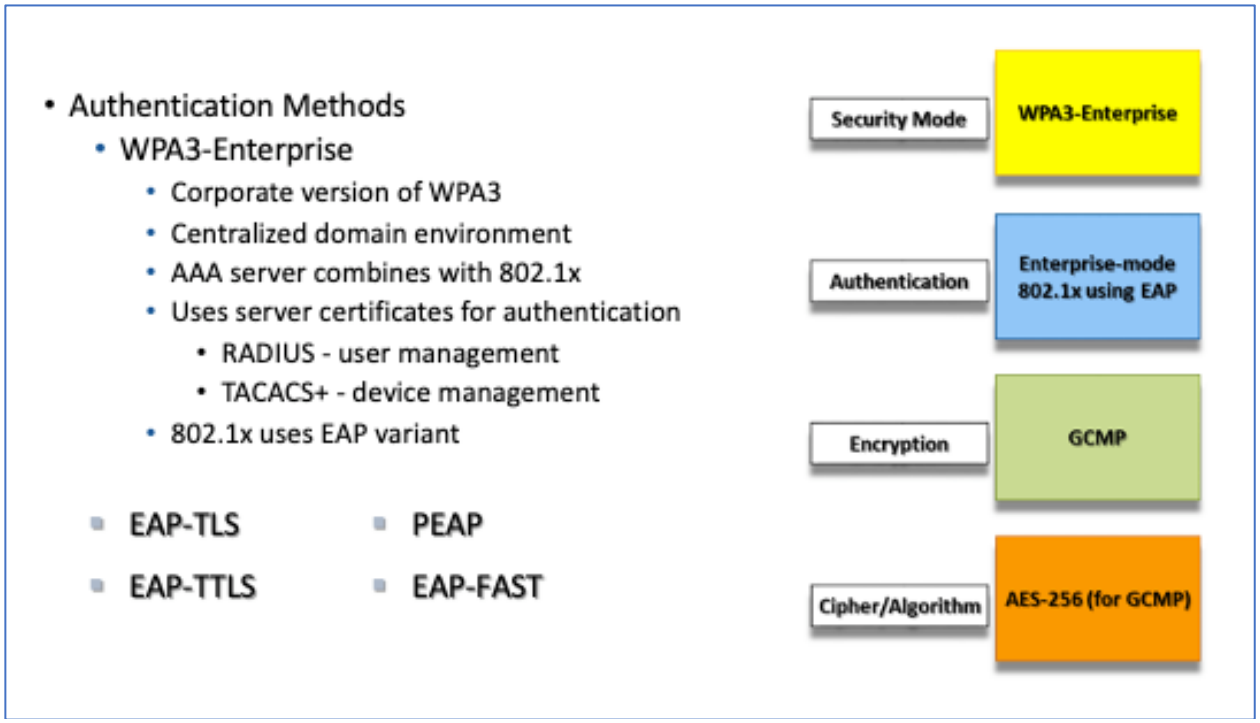
WPA3-SAE Mode



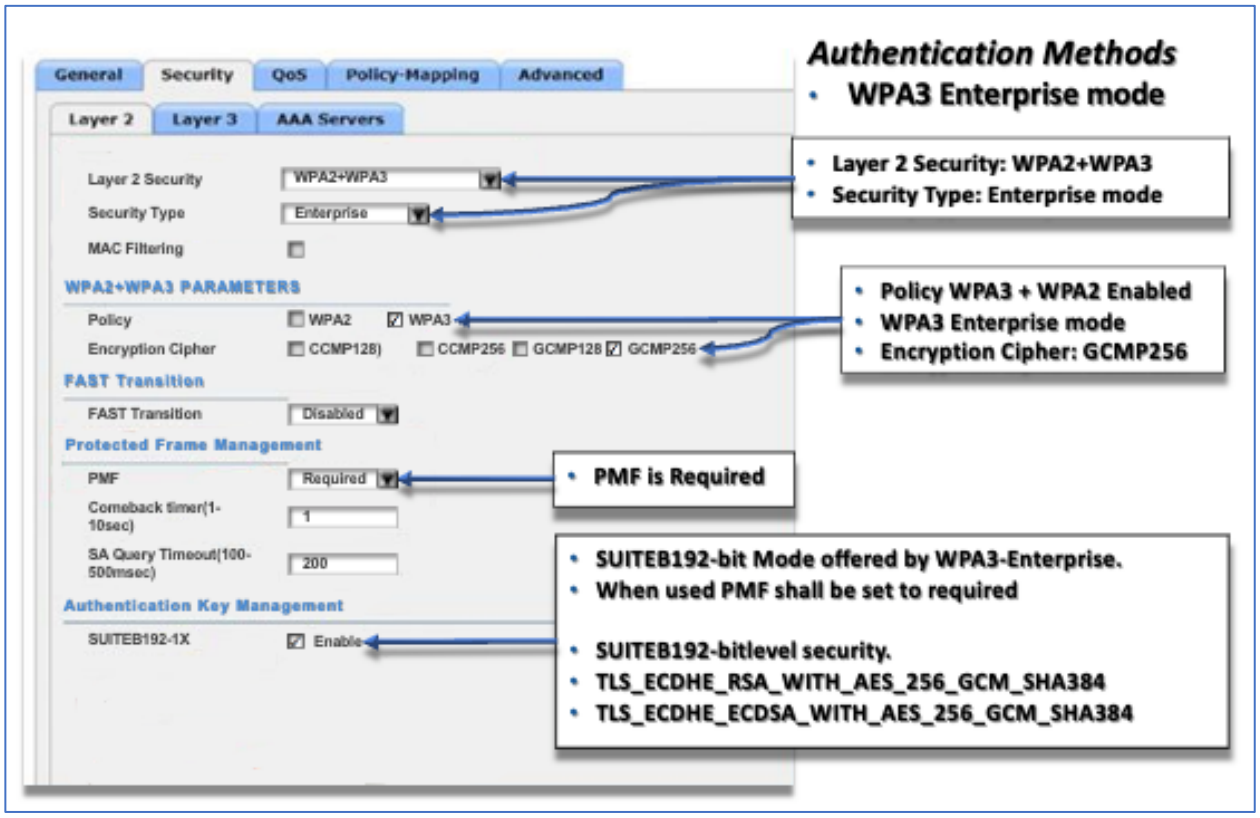
WPA3 Transition Mode



WPA3-Enterprise



WPA3-Enterprise Mode



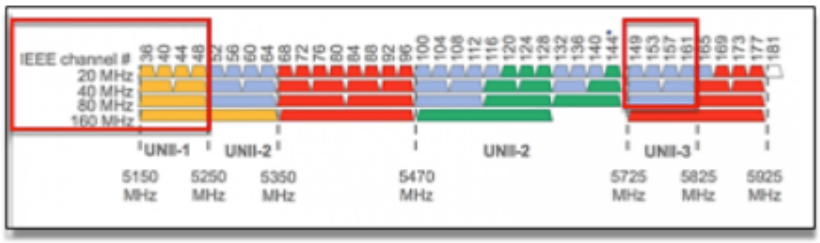
PMF/Easy Connect/OWE

- Authentication Methods
  - Protected Management Frames (PMF)
    - WPA3 uses Protected Management Frames (PMF)
      - PMF Prevents attackers from capturing the encryption keys (IV Attack)
  - Wi-Fi Easy Connect:
    - Easy to connect IoT devices, such as a smartphone, by simply using a QR code
  - Wi-Fi Enhanced Open (OWE)
    - Enhancement of WPA2 open authentication
    - Uses encryption for open authentication
    - No password is required
    - Prevents eavesdropping as it uses PMF

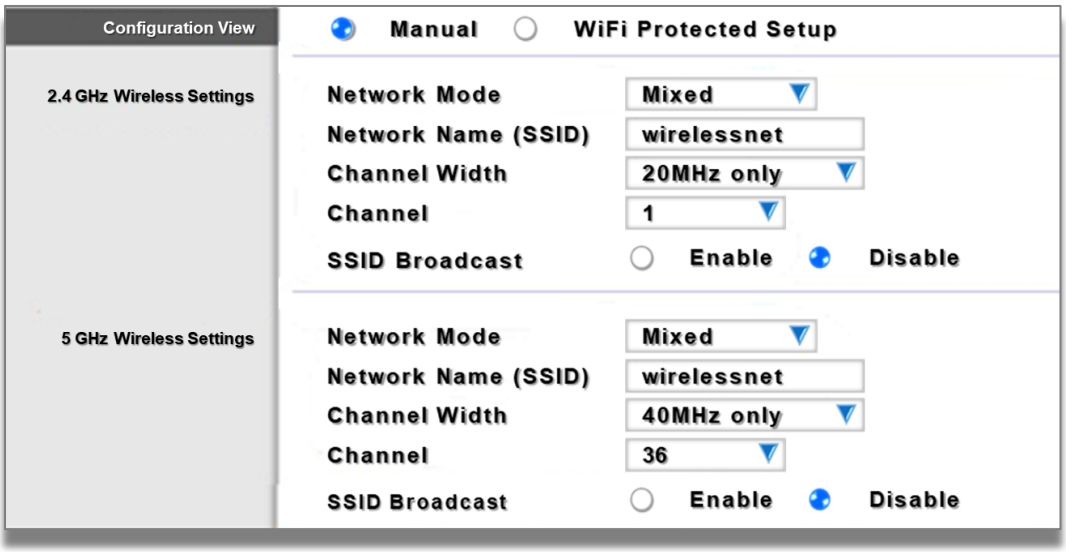
Installation considerations

- 5 GHz, it is recommended to use at least 40 MHz channel width.

20 MHz channel width	40 MHz channel width	80 MHz channel width
36	36 - 40	36 - 48
40	44 - 48	149 - 161
44	149 - 153	
48	157 - 161	
149		
153		
157		
161		
165*		



*Channel 165 only supports 20MHz channel width



*Mixed mode allows backwards compatibility with standards and devices

802.1x Wireless Infrastructure

- Enterprise
  - (WPA2 Enterprise CCMP AES)
  - Credentialed authentication
- AAA server
  - RADIUS
    - Uses shared secret, secret, Key or secret key (all are the same thing)
  - TACACS+
- EAP-TLS
  - Client-side certificated
  - Server-side certificate
  - Most secure method of authentication
    - No user intervention
- PEAP
  - Client-side Username & password
  - Server-side certificate
  - Encrypts credentials over wireless network
  - Used with MSCHAP & GTC
- EAP-TTLS
  - Client-side Username & password
  - Server-side certificate
  - Encrypts credentials over wireless network
  - Used with PAP
- EAP-FAST
  - Server generates a key and provides it to trusted clients
  - Does not use certificates but can support the use of them
  - Replaced LEAP both CISCO proprietary

WPA2 PSK (Pre-shared Key)

- WPA2 PSK CCMP AES
  - Credentialed authentication
  - Also known as personal mode
  - Uses Password or Passphrase (same thing)
- Used for small networks
  - Home networks
  - Small business (coffee shop)

WPS Wi-Fi Protected Setup

- Non-credentialed Authentication
- Uses PIN or push button to Authenticate

OPEN mode

- Non-credentialed non-authentication
- NO SECURITY!

Extensible Authentication Protocol

The Extensible Authentication Protocol (EAP) defines a framework for negotiating authentication mechanisms rather than the details of the mechanisms themselves. Vendors can write extensions to the protocol to support third-party security devices. EAP implementations can include smart cards, one-time passwords, biometric identifiers, or simpler username and password combinations.

EAP-TLS is one of the strongest types of authentication and is very widely supported. An encrypted Transport Layer Security (TLS) tunnel is established between the supplicant and authentication server using public key certificates on the authentication server and supplicant. As both supplicant and server are configured with certificates, this provides mutual authentication. The supplicant will typically provide a certificate using a smart card or a certificate could be installed on the client device, possibly in a Trusted Platform Module (TPM).

PEAP, EAP-TTLS, and EAP-FAST

Provisioning certificates to each wireless device is a considerable management challenge. Other types of EAP are designed to provide secure tunneling with server-side certificates only.

Protected Extensible Authentication Protocol (PEAP)

In **Protected Extensible Authentication Protocol (PEAP)**, as with EAP-TLS, an encrypted tunnel is established between the supplicant and authentication server, but PEAP only requires a server-side public key certificate. The supplicant does not require a certificate. With the server authenticated to the supplicant, user authentication can then take place through the secure tunnel with protection against sniffing, password-guessing/dictionary, and on-path attacks. The user authentication method (also referred to as the "inner" method) can use either MS-CHAPv2 or EAP-GTC. The Generic Token Card (GTC) method transfers a token for authentication against a network directory or using a one-time password mechanism.

EAP with Tunneled TLS (EAP-TTLS)

EAP-Tunneled TLS (EAP-TTLS) is similar to PEAP. It uses a server-side certificate to establish a protected tunnel through which the user's authentication credentials can be transmitted to the authentication server. The main distinction from PEAP is that EAP-TTLS can use any inner authentication protocol (PAP or CHAP, for instance), while PEAP must use EAP-MSCHAP or EAP-GTC.

EAP with Flexible Authentication via Secure Tunneling (EAP-FAST)

EAP with Flexible Authentication via Secure Tunneling (EAP-FAST) is similar to PEAP, but instead of using a certificate to set up the tunnel, it uses a Protected Access Credential (PAC), which is generated for each user from the authentication server's master key. The problem with EAP-FAST is in distributing (provisioning) the PAC securely to each user requiring access. The PAC can either be distributed via an out-of-band method or via a server with a digital certificate (but in the latter case, EAP-FAST does not offer much advantage over using PEAP). Alternatively, the PAC can be delivered via anonymous Diffie-Hellman key exchange. The problem here is that there is nothing to authenticate the access point to the user. A rogue access point could obtain enough of the user credential to perform an ASLEAP password cracking attack (techrepublic.com/article/ultimate-wireless-security-guide-a-primer-on-cisco-eap-fast-authentication).

EAP-TLS	Most Secure	Certificates required on Server and Client
PEAP	Large Networks	Certificates required on Server; Client logs in with user name and password
EAP-TTLS	Backwards Compatibility	Certificates required on Server; Client logs in with user name and password
EAP-FAST	Fastest/Certificates Optional	Certificates required on Server; Client logs in with user name and password

Authentication, Authorization, and Accounting (AAA) Servers

The two main AAA servers are Microsoft's **Remote Authentication Dial-In User Service** (**RADIUS**) and CISCO's **Terminal Access Controller Access-Control System Plus** (**TACACS+**). Both of these servers provide authentication, authorization, and accounting. Let's look at each of these in turn:

- **RADIUS Server**: The RADIUS server is UDP-based, and it authenticates servers such as **Virtual Private Network** (**VPN**) servers, **Remote Access Services** (**RAS**) servers, and the 802.1x authenticating switch. Each of these are known as RADIUS clients, even though they are servers themselves. If I had a small company, I could outsource my remote access server and put in a RADIUS server that would check any remote-access policies and verify that authentication was allowed by contacting a domain controller.
- **RADIUS Clients**: RADIUS clients are VPN servers, RAS servers, and the 802.1x authentication switch. Every RADIUS client needs the secret key that is sometimes known as the session key or shared secret to join the RADIUS environment. RADIUS authentication communicates over the UDP port `1812`. RADIUS accounting uses UDP Port `1813`.
- **TACACS+**: This is the CISCO AAA server that used TCP, and uses TCP port `49` for authentication.

Enterprise/IEEE 802.1X Authentication

The main problems with personal modes of authentication are that distribution of the key or passphrase cannot be secured properly, and users may choose unsecure phrases. Personal authentication also fails to provide accounting, as all users share the same key.

As an alternative to personal authentication, the enterprise authentication method implements IEEE 802.1X to use an Extensible Authentication Protocol (EAP) mechanism. 802.1X defines the use of EAP over Wireless (EAPoW) to allow an access point to forward authentication data without allowing any other type of network access. It is configured by selecting WPA2-Enterprise or WPA3-Enterprise as the security method on the access point.

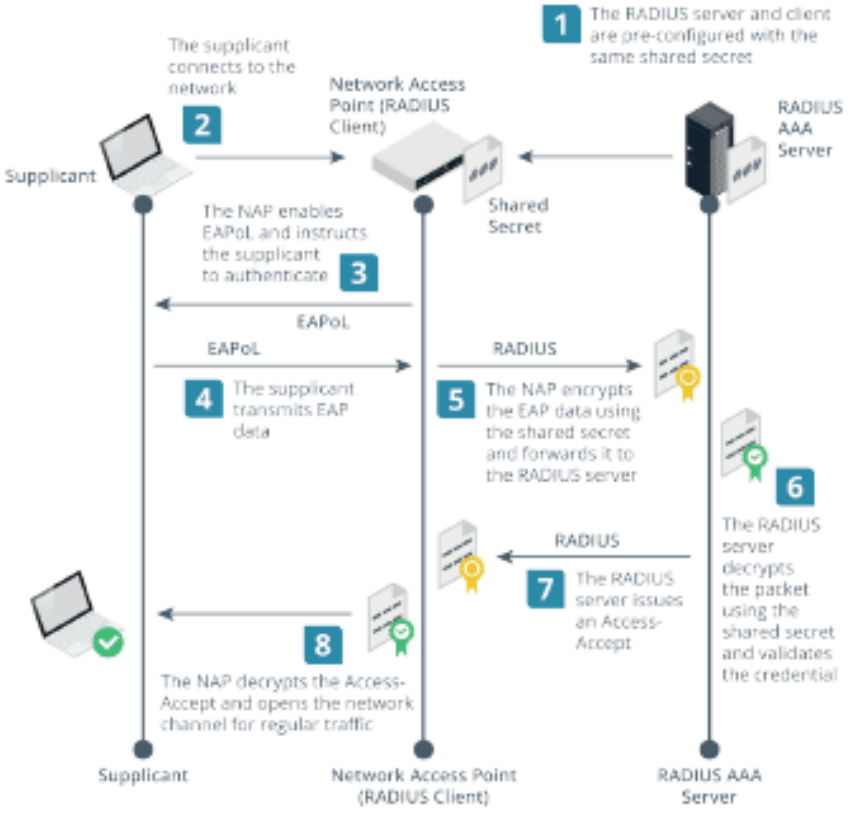
With enterprise authentication, when a wireless station requests an association, the WAP enables the channel for EAPoW traffic only. It passes the credentials of the supplicant to an AAA (RADIUS or TACACS+) server on the wired network for validation. When the supplicant has been authenticated, the AAA server transmits a master key (MK) to the supplicant. The supplicant and authentication server then derive the same pairwise master key (PMK) from the MK. The AAA server transmits the PMK to the the access point. The wireless station and access point use the PMK to derive session keys, using either the WPA2 4-way handshake or WPA3 SAE methods.

Remote Authentication Dial-in User Service

The **Remote Authentication Dial-In User Service (RADIUS)** standard is published as an Internet standard. There are several RADIUS server and client products.

The NAS device (RADIUS client) is configured with the IP address of the RADIUS server and with a shared secret. This allows the client to authenticate to the server. Remember that the client is the access device (switch, access point, or VPN gateway), not the user's PC or laptop. A generic RADIUS authentication workflow proceed as follows:

1. The user's device (the supplicant) makes a connection to the NAS appliance, such as an access point, switch, or remote access server.



RADIUS authentication with EAP overview. (Images © 123RF.com.)

2. The NAS prompts the user for their authentication credentials. RADIUS supports PAP, CHAP, and EAP. Most implementations now use EAP, as PAP and CHAP are not secure. If EAP credentials are required, the NAS enables the supplicant

to transmit **EAP over LAN (EAPoL)** data, but does not allow any other type of network traffic.

3. The supplicant submits the credentials as EAPoL data. The RADIUS client uses this information to create an Access-Request RADIUS packet, encrypted using the shared secret. It sends the Access-Request to the AAA server using UDP on port 1812 (by default).

4. The AAA server decrypts the Access-Request using the shared secret. If the Access-Request cannot be decrypted (because the shared secret is not correctly configured, for instance), the server does not respond.

5. With EAP, there will be an exchange of Access-Challenge and Access-Request packets as the authentication method is set up and the credentials verified. The NAS acts as a pass-thru, taking RADIUS messages from the server, and encapsulating them as EAPoL to transmit to the supplicant.

6. At the end of this exchange, if the supplicant is authenticated, the AAA server responds with an Access-Accept packet; otherwise, an Access-Reject packet is returned.

Optionally, the NAS can use RADIUS for accounting (logging). Accounting uses port 1813. The accounting server can be different from the authentication server.

RADIUS Federation

Most implementations of EAP use a RADIUS server to validate the authentication credentials for each user (supplicant). RADIUS federation means that multiple organizations allow access to one another's users by joining their RADIUS servers into a RADIUS hierarchy or mesh. For example, when Bob from widget.foo needs to log on to grommet.foo's network, the RADIUS server at grommet.foo recognizes that Bob is not a local user but has been granted access rights and routes the request to widget.foo's RADIUS server.

One example of RADIUS federation is the eduroam network (eduroam.org), which allows students of universities from several different countries to log on to the networks of any of the participating institutions using the credentials stored by their "home" university.

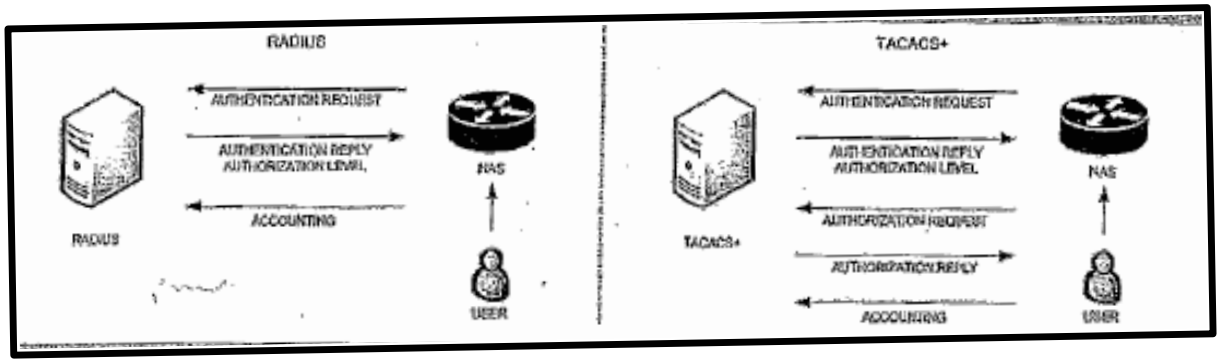
Terminal Access Controller Access-Control System

RADIUS is used primarily for network access control. AAA services are also used for the purpose of centralizing logins for the administrative accounts for network appliances. This allows network administrators to be allocated specific privileges on each switch, router, access point, and firewall. Whereas RADIUS can be used for this network appliance administration role, the Cisco-developed **Terminal Access Controller Access-Control System Plus (TACACS+)** is specifically designed for this purpose (https://www.cisco.com/c/en/us/support/docs/security-vpn/remote-authentication-dial-user-service-radius/13838-10.html):

- TACACS+ uses TCP communications (over port 49), and this reliable, connection-oriented delivery makes it easier to detect when a server is down.
- All the data in TACACS+ packets is encrypted (except for the header identifying the packet as TACACS+ data), rather than just the authentication data. This ensures confidentiality and integrity when transferring critical network infrastructure data.
- Authentication, authorization, and accounting functions are discrete. Many device management tasks require reauthentication (similar to having to re-enter a password for sudo or UAC) and per-command authorizations and privileges for users, groups, and roles. TACACS+ supports this workflow better than RADIUS.

COMPARE: RADIUS VS TACACS+

RADIUS - Open Standard	TACACS+ - CISCO Proprietary (Not backwards compatible)
• Combines AUTHENTICATION and AUTHORIZATION	• Separates all 3 elements of AAA: Flexible design structure
• Detailed ACCOUNTING	• Basic ACCOUNTING support (Not enabled by default)
• Minimal vender support for AUTHORIZATION	• Supported by many leading vendors
• Not easily Scaled	• Easily Scaled
• Flexible, supports PPP, CHAP, EAP, PAP, etc.	• More secure, supports EAP
• Only encrypts PASSWORD (secret key sent as HASH)	• Encrypts both USERNAME and PASSWORD (Payload)
• UDP: Connectionless Ports 1645/1646 1812/1813 (default)	• TCP: Connection oriented Port 49
• Designed for Subscriber (user) Dial-in AAA: Usually used for Network Access, but can also perform Device Admin	• Designed for Administrator AAA: Usually Device Administration but can also perform Network Access
• Accounting Info may be lost due to UDP	• Accepts variety of credentials, such as Kerberos, tokens, etc.
• No command logging takes place	• All entered commands are logged
• Commonly used to connect routers, switches, firewalls, etc. in conjunction with *802.1x*	• Commonly used to connect via VTY, AUX, Console, and TTY Commands include *login*, *enable*, and *exec*
• Requires each network device to contain authorization configuration	• Central management for authorization configuration



Network Access: Secure network by requiring authentication prior to accessing access to the network by extending EAP.

RADIUS typically used for Network Access. Initially used for Layer 2 PPP connections between client and Network Access Server (NAS), and from NAS to AAA Server responsible for Authentication. RADIUS uses EAP and IEEE 802.1x to cross Layer 3 boundaries.

Network Access utilizes controls such as VLANS, Security Groups, and Access Control Lists (ACL's).

Focus: Who connected, how did they authenticate, where did they go, how long where they there, etc.

Device Administration: Controlling login access to network devices using methods such as Telnet, secure shell (SSH), Remote Desktop Protocol (RDP), etc.

TACACS+ was designed for device administration AAA. It was used to authenticate and authorize users logging into UNIX and mainframe terminals, consoles, etc.

Device Administration deals with Rights and Permissions, such as Administrator vs. User.

Focus: Who is allowed to access specific resources, what they can do, and what commands they issue.

XML-based authentication	Non-XML-based authentication
TOTP (time based one time password) SAML (Security assertion markup language) • Allows an application to securely authenticate a user by receiving credentials from a web domain • Used with federated services and federated identity management across multiple organizations – Single sign-on (SSO) ▪ Employ the use of SAML ▪ Use one set of credentials multiple internal resources – Federation Services ▪ Employ the use of SAML ▪ Allow two or more companies one another's resources multiple external resources) ▪ Shibboleth (open source federation)	Kerberos • Mutual Authentication and delegation • Ticket granting • Time stamps to prevent replay attacks • Uses KDC (Key distribution center) • Supports Single sign-on and smart card logons • SSO (Single sign-on) • Used with Active Directory LDAP • Protocol that communicates with database directories or acts as a identity management server. • Single sign-on communication with the access control database OAuth – OpenID Connect • Tokenized authentication • Single sign-on - HOTP

Single sign-on employs the technology of SAML which is based off of XML making Single sign-on XML based the others like Kerberos support Single sign-on meaning Kerberos or LDAP support this as a way to receive credentials Kerberos and LDAP are on the back end providing Authorization it just means it's compatible like Active Directory Federation the third party needs to support Kerberos and SAML

Security Assertions Markup Language

A federated network or cloud needs specific protocols and technologies to implement user identity assertions and transmit attestations between the principal, the relying party, and the identity provider. **Security Assertions Markup Language (SAML)** is one such solution. SAML attestations (or authorizations) are written in eXtensible Markup Language (XML). Communications are established using HTTP/HTTPS and the **Simple Object Access Protocol (SOAP)**. These secure tokens are signed using the XML signature specification. The use of a digital signature allows the relying party to trust the identity provider.

As an example of a SAML implementation, Amazon Web Services (AWS) can function as a SAML service provider. This allows companies using AWS to develop cloud applications to manage their customers' user identities and provide them with permissions on AWS without having to create accounts for them on AWS directly.

```
<samlp:Response
xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
ID="200" Version="2.0"

 IssueInstant="2020-01-01T20:00:10Z "
 Destination="https://sp.foo/saml/acs"
 InResponseTo="100".
  <saml:Issuer>https://idp.foo/sso</saml:Issuer>
  <ds:Signature>...</ds:Signature>
  <samlp:Status>...(success)...</samlp:Status.
 <saml:Assertion xmlns:xsi="http://www.w3.org/2001/
 XMLSchema-instance"
 xmlns:xs="http://www.w3.org/2001/XMLSchema" ID="2000"
 Version="2.0"
 IssueInstant="2020-01-01T20:00:09Z">
 <saml:Issuer>https://idp.foo/sso</saml:Issuer>
 <ds:Signature>...</ds:Signature>
   <saml:Subject>...
   <saml:Conditions>...
    <saml:AudienceRestriction>...
    <saml:AuthnStatement>...
    <saml:AttributeStatement>
       <saml:Attribute>...
       <saml:Attribute>...
    </saml:AttributeStatement>
   </saml:Assertion>
</samlp:Response>
```

Single Sign-On (SSO)

SSO is used in a domain environment. This is where someone logs in to the domain and then can access several resources, such as the file or email server, without needing to input their credentials again. Think of it as an all-inclusive holiday, where you book into your hotel and the receptionist gives you a wristband that you produce when you want to consume food and drink. Federation services and Kerberos (Microsoft authentication protocol) are both good examples of SSO. You log in once and access all of your resources without needing to insert your credentials again.

Internet-Based Open Source Authentication

More and more people are accessing web-based applications and need an account to log in. However, applications hosting companies do not want to be responsible for the creation and management of the account accessing the application. They use OAuth to help them facilitate this:

- **OAuth 2.0**: OAuth 2.0 provides authorization to enable third-party applications to obtain limited access to a web service.
- **Open ID Connect**: Open ID Connect uses OAuth to allow users to log in to a web application without needing to manage the user's account. It allows users to authenticate by using their Google, Facebook, or Twitter account. For example, the Airbnb website that finds users accommodation allows you to sign up using your Google or Facebook account.

OAuth and OpenID Connect

Many public clouds use application programming interfaces (APIs) based on Representational State Transfer (REST) rather than SOAP. These are often called RESTful APIs. Where SOAP is a tightly specified protocol, REST is a looser architectural framework. This allows the service provider more choice over implementation elements. Compared to SOAP and SAML, there is better support for mobile apps.

OAuth

Authentication and authorization for a RESTful API is often implemented using the **Open Authorization (OAuth)** protocol. OAuth is designed to facilitate sharing of information (resources) within a user profile between sites. The user creates a password-protected account at an identity provider (IdP). The user can use that account to log on to an OAuth consumer site without giving the password to the consumer site. A user (resource owner) can grant a client an authorization to access some part of their account. A client in this context is an app or consumer site.

The user account is hosted by one or more resource servers. A resource server is also called an API server because it hosts the functions that allow clients (consumer sites and mobile apps) to access user attributes. Authorization requests are processed by an authorization server. A single authorization server can manage multiple resource servers; equally the resource and authorization server could be the same server instance.

The client app or service must be registered with the authorization server. As part of this process, the client registers a redirect URL, which is the endpoint that will process authorization tokens. Registration also provides the client with an ID and a secret. The ID can be publicly exposed, but the secret must be kept confidential between the client and the authorization server. When the client application requests authorization, the user approves the authorization server to grant the request using an appropriate method. OAuth supports several grant types—or flows—for use in different contexts, such as server to server or mobile app to server. Depending on the flow type, the client will end up with an access token validated by the authorization server. The client presents the access token to the resource server, which then accepts the request for the resource if the token is valid.

OAuth uses the JavaScript object notation (JSON) web token (JWT) format for claims data. JWTs can easily be passed as Base64-encoded strings in URLs and HTTP headers and can be digitally signed for authentication and integrity.

OpenID Connect (OIDC)

OAuth is explicitly designed to authorize claims and not to authenticate users. The implementation details for fields and attributes within tokens are not defined. There is no mechanism to validate that a user who initiated an authorization request is still logged on and present. The access token once granted has no authenticating information. **Open ID Connect (OIDC)** is an authentication protocol that can be implemented as special types of OAuth flows with precisely defined token fields.

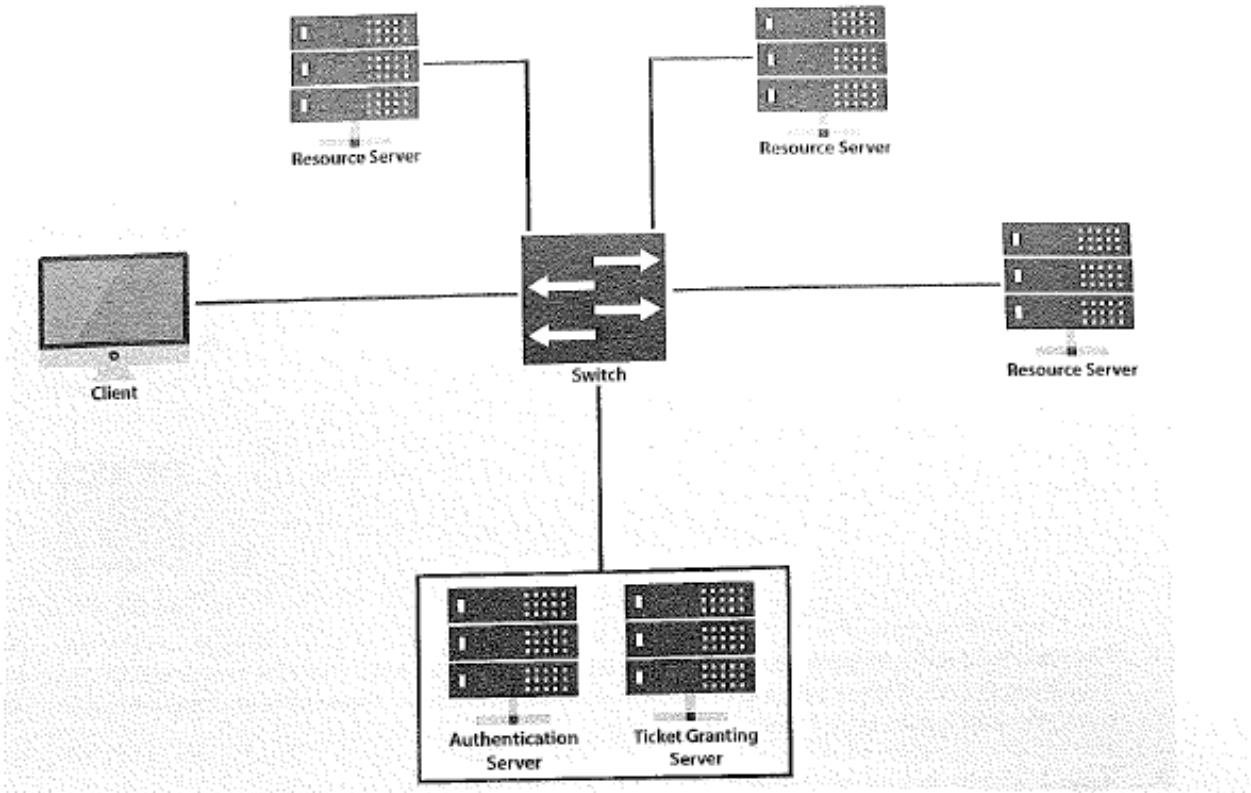
Note that OpenID can also refer to an earlier protocol developed between 2005 and 2007. This implemented a similar framework and underpinned early "sign on with" functionality, but is now regarded as obsolete. OpenID uses XML-format messaging and supports only web applications and not mobile apps.

Kerberos

Kerberos was developed at MIT as part of *Project Athena*, a project to develop a distributed computing[128] system. It was named for the three-headed guard dog owned by Hades in Greek mythology. It was designed to provide mutual authentication and encryption for secure communication between clients and servers on a non-secure network. Currently at version 5, Kerberos has been widely adopted: it's the default authentication protocol for Windows domains, and is also used by many Unix-like operating systems, web applications, embedded devices, and other products. The original implementation is also available under a free license from MIT at `http://web.mit.edu/kerberos/`, though since it uses strong encryption it's still subject to some US cryptographic export laws.

Kerberos is basically network security via a single sign-in method: nodes negotiate with each other on the word of a *trusted third-party*, the Kerberos server. Users go through the authentication process when they first connect, and after that they can communicate securely with any other node; the Kerberos protocol presents their credentials and sets up security without further user input.

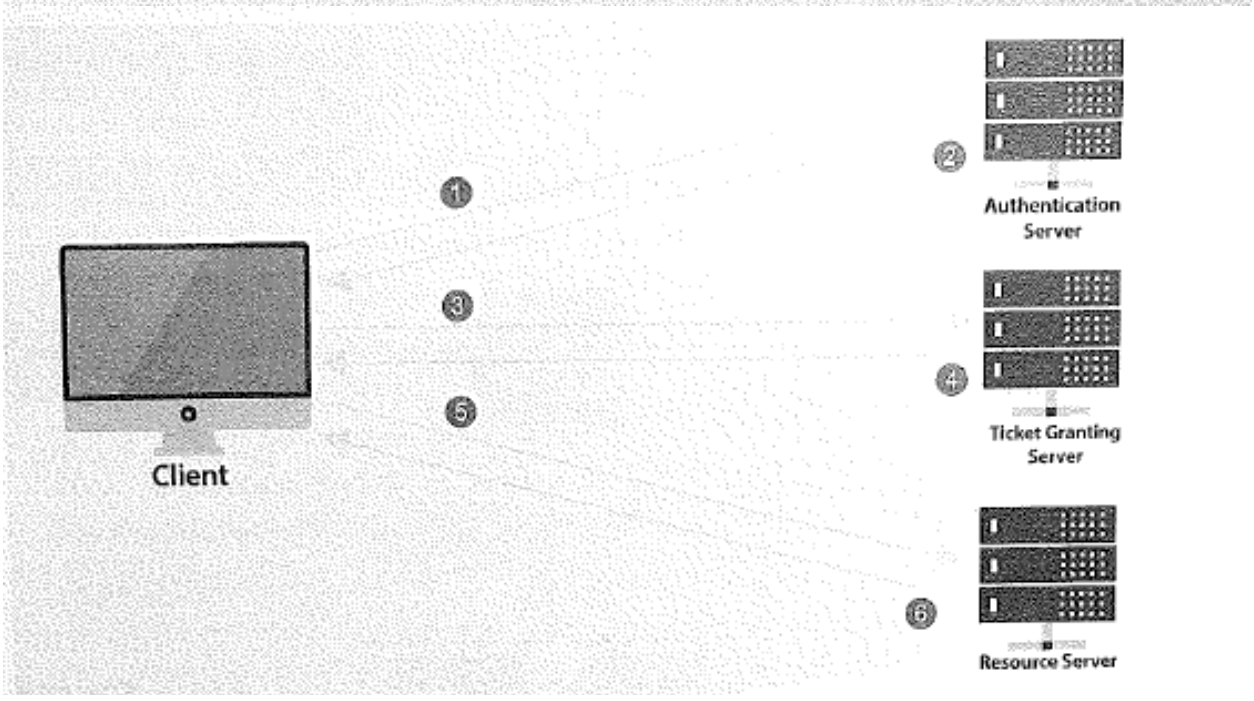
A Kerberos system has several components.



- The basic unit of a Kerberos network is a *realm*. Large organizations can have multiple realms, but each realm has a unique name within the organization. Every *principal*, or node, is a member of a realm.
- Each realm is controlled by a *key distribution center* (*KDC*) which distributes the cryptographic tokens, or *tickets*, used to manage network access. The KDC has two components, which usually are on the same physical host.
  - The *authentication server* (*AS*) authenticates users and gives them a special *ticket-granting ticket* (*TGT*). It contains the full list of users and servers in the realm, and the secret key of each.
  - The *ticket-granting service* (*TGS*) validates TGT holders, and issues them temporary credential tickets and cryptographic *session keys* to access specific resource servers. A remote TGS (in another realm) is called an *RTGS*.

Kerberos authentication

Some implementations of Kerberos use public key cryptography for authentication, but the core protocols are designed to work solely with symmetric cryptography. Instead of the private half of a key pair, the secret key for a user or service is a salted, hashed password created during setup. Additionally, there's no direct communication between the KDC and resource servers: all communication is between clients and servers. Messages are also time-stamped to prevent replay attacks, so all nodes need to be time-synchronized. Guaranteeing secure communication under those restraints takes a fairly exacting process, but it allows mutual authentication without any sensitive information being shared as plaintext.



1. The user logs into a client workstation with a user name and password. The client immediately requests a TGT from the AS. This is a plaintext message, so it doesn't include a password: only the user name, AS name, network address, and requested ticket lifetime.

2. If the AS finds the user in its database, it sends back two messages.

   - A TGT, encrypted with the TGS secret key. It contains the information from the initial request, plus the TGS ID, a timestamp and a newly created TGS session key to be shared between the client and TGS.
   - A matching message encrypted with the user's secret key. It contains the TGS idea, timestamp, lifetime and the same session key. This is how authentication actually happens: only the valid user password can decrypt this message and learn the session key.

3. After learning the session key, the client sends three messages to the TGS to request access to a specific server.

   - The TGT
   - An *authenticator* encrypted with the session key. It contains the client name and a timestamp.
   - A plaintext message containing the name of a resource server and requested ticket lifetime.

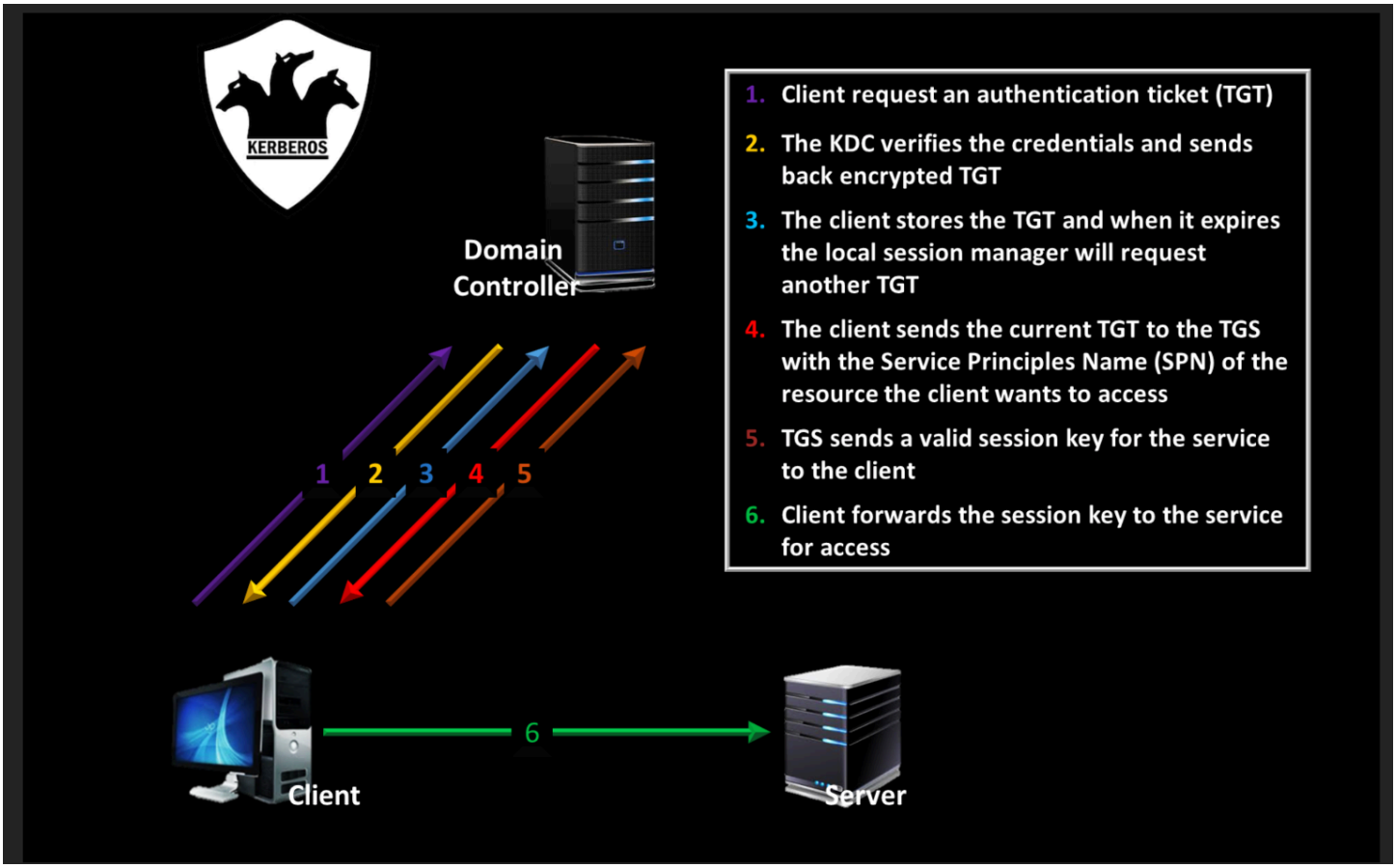
4. The TGS decrypts the TGT with its secret key to learn the session key, then uses the session key to decrypt the authenticator. If the service, timestamp, and user all seem valid, it sends the following messages in response.

   - A credential ticket for the service, encrypted with the resource server's secret key. Much like a TGT it has the user's information, a timestamp, lifetime, and a service session key created by the TGS.
   - A matching message encrypted by the TGS session key, and containing the service session key.

5. The client decrypts the second message to learn the service session key, and uses it to create a new authenticator. It sends the authenticator and service ticket to the resource server.

6. The resource server decrypts the service ticket to learn the session key, and uses the session key to decrypt the authenticator. If all information checks out, the client is authenticated on the resource server, and they can communicate securely. This step can also use mutual authentication.

Whenever the client wants to log into a new service, or when a service ticket expires, it can present its TGT to the TGS again for a new request. What happens when the TGT itself expires depends on network settings and implementation: the user might need to log in again, or the client might be able to transparently request a new TGT.


KERBEROS
Domain Controller
1 2 3 4 5
6
Client
Server
1. Client request an authentication ticket (TGT)
2. The KDC verifies the credentials and sends back encrypted TGT
3. The client stores the TGT and when it expires the local session manager will request another TGT
4. The client sends the current TGT to the TGS with the Service Principles Name (SPN) of the resource the client wants to access
5. TGS sends a valid session key for the service to the client
6. Client forwards the session key to the service for access

Mandatory Access Control

Mandatory Access Control (**MAC**) is based on the classification level of the data. MAC looks at how much damage could be inflicted to the interests of the nation. These are as follows:

- **Top secret**: Highest level, exceptionally grave damage
- **Secret**: Causes serious damage
- **Confidential**: Causes damage
- **Restricted**: Undesirable effects

Examples of MAC based on the classification level of data are as follows:

- **Top secret**: Nuclear energy project
- **Secret**: Research and development
- **Confidential**: Ongoing legal issues

MAC Roles

Once classified data has been written, it is owned by the company. For example, if a Colonel writes a classified document, it belongs to the Army. Let's look at three roles:

- **Owner**: This is the person who writes data, and they are the only person that can determine the classification. For example, if they are writing a secret document, they will pitch it at that level, no higher.
- **Steward**: This is the person responsible for labeling the data.
- **Custodian**: The custodian is the person who stores and manages classified data.
- **Security Administrator**: The security administrator is the person who gives access to classified data once clearance has been approved.

Role-Based Access Control

Role-based access control is a subset of the department carrying out a subset of duties within a department. An example would be two people within the finance department who only handle petty cash. In IT terms, it could be that only two people of the IT team administer the email server.

Rule-Based Access Control

In **Rule-Based Access Control** (**RBAC**), a rule is applied to all of the people within a department, for example, contractors will only have access between 8 a.m. and 5 p.m., and the help desk people will only be able to access building 1, where their place of work is. It can be time-based or have some sort of restriction, but it applies to the whole department.

Attribute-Based Access Control

In **Attribute-Based Access Control** (**ABAC**), access is restricted based on an attribute in the account. John could be an executive and some data could be restricted to only those with the executive attribute. This is a user attribute from the directory services, such as a department or a location. You may wish to give different levels of control to different departments.

Group-Based Access Control

To control access to data, people may be put into groups to simplify access. An example would be if there were two people who worked in IT who needed access to IT data. For example, let's call them *Bill* and *Ben*. We first of all place them into the IT group, and then that group is given access to the data:

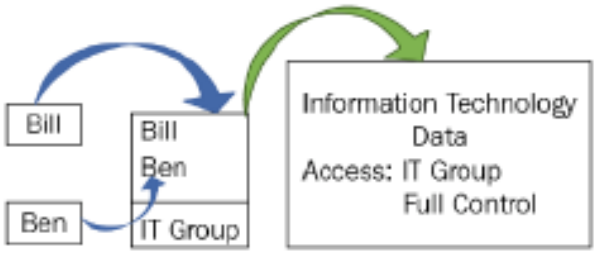


Figure 1.4 – Group-based access

Another example is where members of a sales team may have full control of the sales data by using group-based access, but you may need two new starters to have only read access. In this case, you would create a group called new starters and give those people inside that group only read permission to the data.

WINDOWS

Discretionary access control involves New Technology File System (NTFS)

- File permissions, which are used in Microsoft operating systems

Windows New Technology File System (NTFS)
• Full control: Full access
• Modify: Change data, read, and read and execute
• Read and execute: Read the file and run a program if one is inside it
• List folder contents: Expand a folder to see the subfolders inside it
• Read: Read the contents
• Special permissions: Allows granular access; for example, it breaks each of the previous permissions down to a more granular level

LINUX

Linux permissions

Number	Permission Type	Symbol
0	no permission	`---`
1	execute	`--x`
2	write	`-w-`
3	execute + write	`-wx`
4	read	`r--`
5	read + execute	`r-x`
6	read + write	`rw-`
7	read +write + execute	`rwx`

Linux permissions

- If I have 764 access to a file, this could be broken down as:

  `rwxrw-r--`

  - Owner: Read, write, and execute
  - Group: Read, write
  - All other users (world): Read

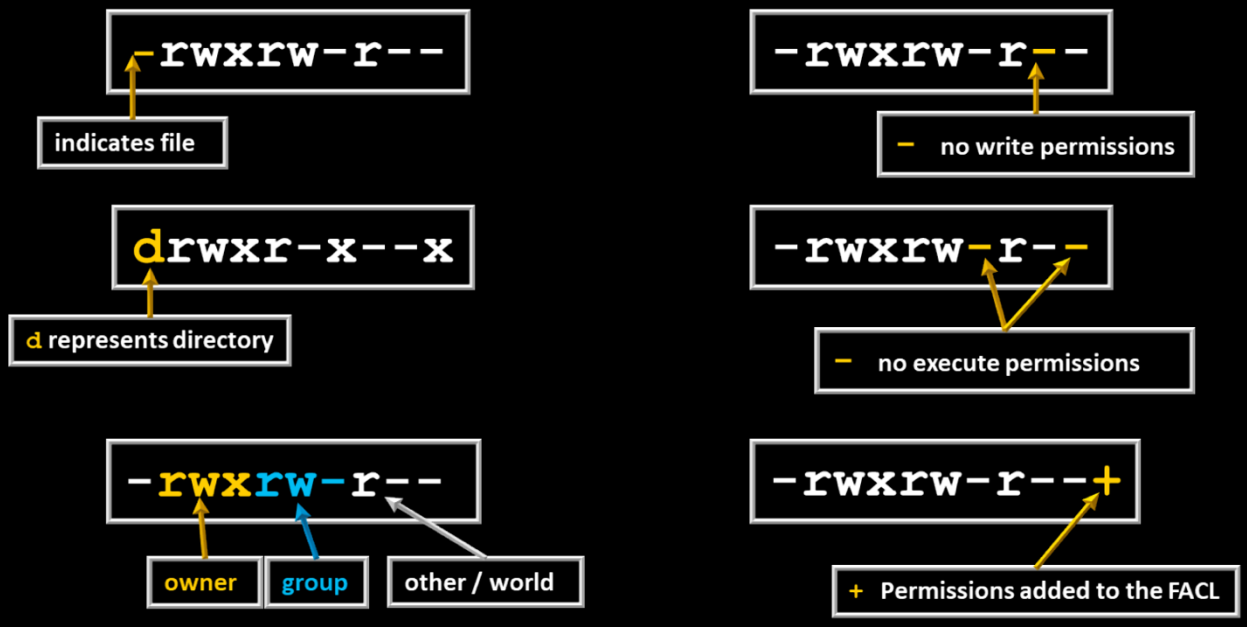
Permissions:

- Owner: First number
- Group: Second number
- All other users (world): Third number

Numerical values:

- 4: Read (r)
- 2: Write (w)
- 1: Execute (x)

Linux permissions



COMMON COMMAND-LINE COMMANDS

Tool	Notes
tracert (Windows)/ traceroute (Linux)	The tracert command is a Windows command for tracing the route that packets take over the network. The tracert command provides a list of the hosts, switches, and routers in the order in which a packet passes through them, providing a trace of the network route from source to target. As tracert uses Internet Control Message Protocol (ICMP), if ICMP is blocked, tracert will fail to provide information. On Linux and macOS systems, the command with similar functionality is traceroute.
route	The route command works in Linux and Windows systems to provide information on current routing parameters and to manipulate these parameters. In addition to listing the current routing table, it has the ability to modify the table.
nslookup (Windows) / dig (Linux)	The Domain Name System (DNS) is used to convert a human-readable domain name into an IP address. This is not a single system, but rather a hierarchy of DNS servers, from root servers on the backbone of the Internet, to copies at your Internet service provider (ISP), your home router, and your local machine, each in the form of a DNS cache. To examine a DNS query for a specific address, you can use the nslookup command. At times, nslookup will return a nonauthoritative answer. This typically means the result is from a cache as opposed to a server that has an authoritative (that is, known to be current) answer, such as from a DNS server. While nslookup works on Windows systems, the command dig, which stands for Domain Information Groper, works on Linux systems. One difference is that dig is designed to return answers in a format that is easy to parse and include in scripts, which is a common trait of Linux command-line utilities.
ipconfig (Windows) / ifconfig (Linux/Mac) / IP (Linux)	Both ipconfig (for Windows) and ifconfig (for Linux) are command-line tools to manipulate the network interfaces on a system. They have the ability to list the interfaces and connection parameters, alter parameters, and release/renew connections. If you are having network connection issues, this is one of the first tools you should use, to verify the network setup of the operating system and its interfaces. The ip command in Linux is used to show and manipulate routing, devices, policy routing, and tunnels. The ipconfig command is an important command for troubleshooting because it displays current TCP/IP configurations on a local system. The command displays adapter information such as MAC address, current IP addresses (both IPv4 and IPv6), subnet mask, default gateway, as well as DNS servers and whether DHCP is enabled. This is an important troubleshooting tool because when you can't connect to something, it is the first place to start exploring network connections, as it gives you all of your settings.
nmap	Nmap is a is free, open source port scanning tool developed by Gordon Lyon and has been the standard network mapping utility for Windows and Linux since 1999. The nmap command is the command to launch and run the nmap utility. Nmap is used to discover what systems are on a network and the open ports and services on those systems. This tool has many other additional functions, such as OS fingerprinting, finding rogue devices, and discovering services and even application versions. It operates via the command line, so it's very scriptable. It also has a GUI interface called Zenmap. Nmap works on a wide range of operating systems, including Microsoft Windows, Linux, and macOS. This is one of the top ten tools used by system administrators on a regular basis. Nmap includes a scripting engine using the Lua programming language to write, save, and share scripts that can automate different types of scans. All sorts of tasks can be automated, including regular checks for well-known network infrastructure vulnerabilities.

ping	The ping command sends echo requests to a designated machine to determine if communication is possible. The syntax is ping [options] targetname/address. The options include items such as name resolution, how many pings, data size, TTL counts, and more.
netstat	The netstat command is used to monitor network connections to and from a system. The following are some examples of how you can use netstat: • netstat –a Lists all active connections and listening ports • netstat –at Lists all active TCP connections • netstat –an Lists all active UDP connections Many more options are available and useful. The netstat command is available on Windows and Linux, but availability of certain netstat command switches and other netstat command syntax may differ from operating system to operating system.
netcat	Netcat is the network utility designed for Linux environments. It has been ported to Windows but is not regularly used in Windows environments. The actual command to invoke netcat is nc –options –address. The netcat utility is the tool of choice in Linux for reading from and writing to network connections using TCP or UDP. Like all Linux command-line utilities, it is designed for scripts and automation. Netcat has a wide range of functions. It acts as a connection to the network and can act as a transmitter or a receiver, and with redirection it can turn virtually any running process into a server. It can listen on a port and pipe the input it receives to the process identified.
arp	The arp command is designed to interface with the operating system's Address Resolution Protocol (ARP) caches on a system. In moving packets between machines, a device sometimes needs to know where to send a packet using the MAC or layer 2 address. ARP handles this problem through four basic message types: • ARP request "Who has this IP address?" • ARP reply "I have that IP address; my MAC address is…" • Reverse ARP (RARP) request "Who has this MAC address?" • RARP reply "I have that MAC address; my IP address is…" These messages are used in conjunction with a device's ARP table, where a form of short-term memory associated with these data elements resides. The commands are used as a simple form of lookup. When a machine sends an ARP request to the network, the reply is received and entered into all devices that hear the reply. This facilitates efficient address lookups, but also makes the system subject to attack. The arp command allows a system administrator the ability to see and manipulate the ARP cache on a system. This way they can see if entries have been spoofed or if other problems, such as errors, occur.

Linux Command-line Commands

Admin accounts: Root top level

sudo: Admin

su: Lower admin

kill : Stops applications

ls : List

grep: Search

pwd : Parent Working Directory

chown: Changes ownership

chmod : Changes permissions

mkdir: Make directory

rm -rf: Remove directory and files

SetFACL: Used to set permissions on a given file

Ifconfig: Equivalent of ipconfig

Ip**Tables**: Firewall rules

chroot: Change root directory

Root directories: /bin, /boot, /dev, /etc, /home, /mnt, /sbin, and /usr

COMMON SWITCHES

Ipconfig	iwconfig - Linux used exclusively for wireless
ipconfig	– Windows TCP/IP configuration
ipconfig /all	– Show all TCP/IP details
ipconfig /release	– Release the DHCP lease
ipconfig /renew	– Renew the DHCP lease
ipconfig /flushdns	– Flush the DNS resolver cache
ifconfig or ip	– Linux - Unix TCP/IP configuration
ifconfig <interface>	– Linux - Unix Show TCP/IP details

Ping	
ping	– Test reachability of a host
ping -t	– Ping the specified host until stopped <Ctrl-c>
ping -a	– Resolve addresses to hostname
ping -n count	– Number of echo request to send
ping -c count(Linux)	– Specifies number of ping packets sent
ping -l size	– Send buffer size
ping -f	– Send with Don't Fragment flag set
ping -i TTL	– Time to live
ping -i interval(Linux)	– Sets the time interval in seconds
ping -v TOS	– Type of service

Netstat	-o shows which processes are using which protocols
netstat -a ✓	– Displays a listing of all TCP and User Datagram Protocol (UDP) ports that a system is listening on, in addition to all open connections
netstat -r ✓	– Displays the routing table
netstat -e	– Displays details on network statistics. – Including how many bytes the system sent and received
netstat -s	– Displays statistics of packets sent or received for specific protocols Such as: – IP, ICMP, TCP, and UDP
netstat -n	– Displays addresses and port numbers in numerical order
netstat -p protocol	– Shows statistics on a specific protocol, such as TCP or UDP
netstat -anp tcp	– Netstat displays the state of a connection, such as ESTABLISHED to indicate an active connection

Netstat ✓	
ESTABLISHED	– Normal state for the data transfer phase of a connection
LISTEN	– Indicates the system is waiting for a connection request
CLOSE_WAIT	– This indicates the system is waiting for a connection termination request
TIME_WAIT	– Indicates the system is waiting for enough time to receive a TCP-based acknowledgment of the connection
SYN_SENT	– This indicates the system sent a TCP SYN (synchronize) packet – First part of the SYN, SYN-ACK (synchronize-acknowledge) – ACK (acknowledge) handshake process and it is waiting for the SYN-ACK response

SOFTWARE TOOLS YOU SHOULD BE FAMILIAR WITH

Tool	Notes
Tcpreplay	Tcpreplay is the name for both a tool and a suite of tools. As a suite, tcpreplay is a group of free, open source utilities for editing and replaying previously captured network traffic. As a tool, it specifically replays a PCAP file on a network. Originally designed as an incident response tool, tcpreplay has utility in a wide range of circumstances where network packets are used. It can be used to test all manner of security systems through the use of crafted PCAP files to trip certain controls. It is also used to test online services such as web servers. If you have a need to send network packets to another machine, tcpreplay suite has your answer.
tcpdump	The tcpdump utility is designed to analyze network packets either from a network connection or a recorded file. You also can use tcpdump to create files of packet captures, called PCAP files, and perform filtering between input and output, making it a valuable tool to lessen data loads on other tools. For example, if you have a complete packet capture file that has hundreds of millions of records, but you are only interested in one server's connections, you can make a copy of the PCAP file containing only the packets associated with the server of interest. This file will be smaller and easier to analyze with other tools.
Wireshark	Wireshark is the gold standard for graphical analysis of network protocols. With dissectors that allow the analysis of virtually any network protocol, this tool can allow you to examine individual packets, monitor conversations, carve out files, and more. When it comes to examining packets, Wireshark is the tool. When it comes to using this functionality in a scripting environment, TShark provides the same processing in a scriptable form, producing a wide range of outputs, depending on the options set. Wireshark has the ability to capture live traffic, or it can use recorded packets from other sources.
Zenmap	GUI version of Nmap; another example is AngryIP
Pacu	scanning and exploit tools for reconnaissance and exploitation of Amazon Web Service (AWS) accounts (rhinosecuritylabs.com/aws/pacu-open-source-aws-exploitation-framework).
Zed Attack Proxy (ZAP)	scanning tools and scripts for web application and mobile app security testing (owasp.org/www-project-zap).
fireELF	fireELF—injecting fileless exploit payloads into a Linux host (github.com/rek7/fireELF).
BeEF	Browser Exploitation Framework (BeEF)—recovering web session information and exploiting client-side scripting (beefproject.com).
RouterSploit	RouterSploit—vulnerability scanning and exploit modules targeting embedded systems (github.com/threat9/routersploit).
Metasploit	The best-known exploit framework. The platform is open-source software, now maintained by Rapid7. There is a free framework (command-line) community edition with installation packages for Linux and Windows. Rapid7 produces pro and express commercial editions of the framework and it can be closely integrated with the Nexpose vulnerability scanner.
Netflow/sFlow	NetFlow and sFlow are protocols designed to capture information about packet flows (that is, a sequence of related packets) as they traverse a network. NetFlow is a proprietary standard from Cisco. Flow data is generated by the network devices themselves, including routers and switches. The data that is collected and shipped off to data collectors is a simple set of metadata—source and destination IP addresses, source and destination ports, if any (ICMP, for example, doesn't use ports), and the protocol. NetFlow does this for all packets, while sFlow (sampled flow) does a statistical sampling. On high-throughput

	networks, NetFlow can generate large quantities of data—data that requires de-duplication. However, having all that data will catch the rare security event packets. sFlow is more suited for statistical traffic monitoring. Cisco added statistical monitoring to NetFlow on its high-end infrastructure routers to deal with the traffic volumes.
IPFIX	Internet Protocol Flow Information Export (IPFIX) is an IETF protocol that's the answer to the proprietary Cisco NetFlow standard. IPFIX is based on NetFlow version 9 and is highly configurable using a series of templates. The primary purpose of IPFIX is to provide a central monitoring station with information about the state of the network. IPFIX is a push-based protocol, where the sender sends the reports and receives no response from the receiver.
pathping	a TCP/IP-based utility that provides additional data beyond that of a ping command. Pathping will first display the path as if you were using traceroute. Pathping then calculates loss information.
hping	a TCP/IP packet creation tool that allows a user to craft raw TCP, IP, UDP, and ICMP packets from scratch. This tool provides a means of performing a wide range of network operations; anything that you can do with these protocols can be crafted into a packet. This includes port scanning, crafting ICMP packets, host discovery and more. The current version is hping3, and it is available on most operating systems, including Windows and Linux.
curl	Curl is a tool designed to transfer data to or from a server, without user interaction. It support a long list of protocols (DICT, FILE, FTP, FTPS, Gopher, HTTP, HTTPS, IMAP, IMAPS, LDAP, LDAPS, MQTT, POP3, POP3S, RTMP, RTMPS, RTSP, SCP, SFTP, SMB, SMBS, SMTP, SMTPS, Telnet, and TFTP) and acts like a Swiss army knife for interacting with a server. Originally designed to interact with URLs, curl has expanded into a jack-of-all-trades supporting numerous protocols. It works on both Linux and Windows systems, although the command options are slightly different.
theHarvester	theHarvester is a Python-based program designed to assist penetration testers in the gathering of information during the reconnaissance portion of a penetration test. This is a useful tool for exploring what is publicly available about your organization on the Web, and it can provide information on employees, e-mails, and subdomains using different public sources such as search engines, PGP key servers, and Shodan databases. Designed for Linux and included as part of Kali and other penetration testing distributions.
sn1per	Sn1per is a Linux-based tool used by penetration testers. Sn1per is an automated scanner designed to collect a large amount of information while scanning for vulnerabilities. It runs a series of automated scripts to enumerate servers, open ports, and vulnerabilities, and it's designed to integrate with the penetration testing tool Metasploit. Sn1per goes further than just scanning; it can also brute force open ports, brute force subdomains and DNS systems, scan web applications for common vulnerabilities, and run targeted nmap scripts against open ports as well as targeted Metasploit scans and exploit modules. This tool suite comes as a free community edition, with limited scope, as well as an unlimited professional version for corporations and penetration testers.
scanless	Scanless is a command-line utility to interface with websites that can perform port scans as part of a penetration test. When you use this tool, the source IP address for the scan is the website, not your testing machine. Written in Python, with a simple interface, scanless anonymizes your port scans.
dnsenum	Dnsenum is a Perl script designed to enumerate DNS information. Dnsenum will enumerate DNS entries, including subdomains, MX records, and IP addresses. It can interface with Whois, a public record that identifies domain owners, to gather additional information. Dnsenum works on Linux distros that support Perl.
Nessus	Nessus is one of the leading vulnerability scanners in the marketplace. It comes in a free version, with limited IP address capability, and fully functional commercial versions. Nessus is designed to perform a wide range of testing on a system, including the use of user credentials, patch level testing, common misconfigurations, password attacks, and more. Designed as a full suite of vulnerability and configuration testing tools, Nessus is commonly used to audit systems for compliance to various security standards such as PCI DSS, SOX, and other compliance schemes. Nessus free version was the original source of

	the OpenVAS fork, which is a popular free vulnerability scanner.
Cuckoo	Cuckoo is a sandbox used for malware analysis. Cuckoo is designed to allow a means of testing a suspicious file and determining what it does. It is open source, free software that can run on Linux and Windows. Cuckoo is a common security tool used to investigate suspicious files, as it can provide reports on system calls, API calls, network analysis, and memory analysis.
dd	Data dump (dd) is a Linux command-line utility used to convert and copy files. On Linux systems, virtually everything is represented in storage as a file, and dd can read and/or write from/to these files, provided that function is implemented in the respective drivers. As a result, dd can be used for tasks such as backing up the boot sector of a hard drive, obtaining a fixed amount of random data, or copying (backing up) entire disks. The dd program can also perform conversions on the data as it is copied, including byte order swapping and conversion to and from the ASCII and EBCDIC text encodings. dd has the ability to copy everything, back up/restore a partition, and create/restore an image of an entire disk.
memdump	Linux has a utility program called memory dumper, or memdump. This program dumps system memory to the standard output stream, skipping over any holes in memory maps. By default, the program dumps the contents of physical memory (/dev/mem). The output from memdump is in the form of a raw dump. Because running memdump uses memory, it is important to send the output to a location that is off the host machine being copied, using a tool such as netcat.
WinHex	WinHex is a hexadecimal file editor. This tool is very useful in forensically investigating files, and it provides a whole host of forensic functions such as the ability to read almost any file, display contents of the file, convert between character sets and encoding, perform hash verification functions, and compare files. As a native file reader/hex editor, it can examine specific application files without invoking the application and changing the data. WinHex is a commercial program that is part of the X-Ways forensic suite, which is a comprehensive set of digital forensic tools.
FTK Imager	FTK Imager is the company AccessData's answer to dd. FTK Imager is a commercial program, free for use, and is designed to capture an image of a hard drive (or other device) in a forensic fashion. Forensic duplications are bit-by-bit copies, supported by hashes to demonstrate that the copy and the original are exact duplicates in all ways. As with all forensically sound collection tools, FTK Imager retains the file system metadata (and the file path) and creates a log of the files copied. This process does not change file access attributes. FTK Imager is part of the larger, and commercial, FTK suite of forensic tools.
Autopsy	Autopsy is the open source answer for digital forensic tool suites. This suite, developed by Brian Carrier, has evolved over the past couple of decades into a community-supported open source project that can perform virtually all digital forensic functions. It runs on Windows and offers a comprehensive set of tools that can enable network-based collaboration and automated, intuitive workflows. It has tools to support hard drives, removable devices, and smartphones. It supports MD5 hash creation and lookup, deleted file carving, EXIF data extraction from JPEG images, indexed keyword searches, extension mismatch detections, e-mail message extractions, and artifact extractions from web browsers. It has case management tools to support the functions of case analysis and reporting, including managing timelines.
NXLog	NXLog is a multiplatform log management tool designed to assist in the use of log data during investigations. This tool suite is capable of handling syslog-type data as well as other log formats, including Microsoft Windows. It has advanced capabilities to enrich log files through context-based lookups, correlations, and rule-based enrichments. NXLog has connectors to most major applications and can act as a log collector, forwarder, aggregator, and investigative tool for searching through log data. As logs are one of the most used data sources in investigations, tools such as NXLog can enable investigators to identify security issues, policy violations, and operational problems in systems.

Command Line Security Tools

Professor Messer (Video transcript)

The seasoned security administrator needs to be comfortable at the command line. In this video, you'll learn about some of the most important command line security tools for the security professional.

If you've ever needed a test to see if another device was available on your network, then you've probably used the ***ping*** command. The ***ping*** command can also be used to determine a round trip time between your device and another device on the internet using a protocol called the Internet Control Message Protocol, or *ICMP*. If you're doing any type of network troubleshooting or security analysis, you will be using ***ping*** quite a bit. It's usually the first thing that you use.

You want to know if a device is available, the first thing you try to do is to ***ping*** that device. This is a utility that was written by Mike Moose in 1983. So it's been around for a while. He named it ping because that's the sound used by sonar. Some people have stated that the term ***ping*** is an acronym for Packet Internet Groper. But, in fact, this was an acronym that was created after the fact, something we call a backroym. And Mike says, no. This is simply the sound that you make whenever you're using sonar to try to find another device that might be out there.

Ping is a relatively easy utility to use. You simply type ***ping***, and then the IP address of the device that you'd like to see is available. I'm going to use Google's DNS server since that is not only easy to remember– the IP addresses 8.8.8.8– But it's also an IP address that you could use if you'd like to try to ***ping*** this yourself. And now we get the results back from the ping. It tries on my machine until I choose to stop it with ***Control C***.

You can see that it's sending 64 bytes of data from 8.8.8.8. That's what we're receiving. It shows the sequence number of what we're pinging. The time to live is 57. On my system, it probably starts with 64 as the time to live. And every time the packet goes through a router, it decreases by one. So you can guess by looking at the time to live just how many hops this had to go through to get back to my workstation. And the time it takes to be able to get round trip to that device is anywhere between 17, 14, 28 milliseconds. And you can see exactly how quickly we can get data between point A and point B.

And we know that 8.8.8.8 must be available because we're receiving an answer back from that device. We can also use names. In ***ping***, we could use the DNS capability. So I could ***ping*** www.professormesser.com, and we'll get a response back from the Professor Messer server, or at least the device that sits in front of all the Professor Messer servers. And now you can see exactly how long it takes to get between this device and that web server. Another utility that's available on many different operating systems is ***netstat***. You can find ***netstat*** on Linux, on Unix, on Windows, and many others.

Netstat stands for network statistics, and it provides you with many different views of what the statistics are for network communications on that particular device. So, for example, if you wanted to see all of the active connections that were on an individual machine, you can run ***netstat dash a***. On a Windows machine, you could combine the ***netstat dash a with the a netstat dash b***, which would not only show you the number of active connections, it would tell you exactly the Windows binary that was used to create that connection across the network. And if you wanted to get rid of the resolved names and just get IP addresses with netstat, you can run ***netstat dash n.***

Let's run a ***netstat*** on my computer. I'm just going to run netstat with no other parameters, and it's going to show me all of the network communications that I have made. You can see a number of them out to professormesser.com and to other web services. And you can see as it's working through all of the active connections, it begins updating those on the screen. When we ran the ping command earlier, we could tell from the time to live settings that there were a number of hops between my computer and the Google DNS server. With the ***traceroute*** command, we could map out every single router along the way, and effectively build a map that shows the route between my system and another device.

If you're running this utility on ***Linux or Mac OS, it's the entire word trace route***. If you are on ***Windows***, it's ***trace RT***. Trace route uses a function of ICMP to be able to map out this network communication. If you ever send traffic that exceeds the time to live, that device will send a response back to you saying that it had to drop the packet because your time to live was exceeded. What we'll do is send some messages out across the network with intentionally small time to lives that will automatically expire. So we should expect to receive a message back from a number of the devices out across the network.

Not every device will send you back a time to live exceeded message. So occasionally in your trace route, you'll see asterisks appear instead of getting information about a route. And occasionally, once it gets past that particular router, it will pick up after that point and continue with its mapping function but you can always try running a trace route and see exactly what route happens to appear between you and the other device. Behind the scenes, here's what's really happening with traceroute. In this example, I'm going to be Sam, and I'm going to trace route to Jack's IP address, which is 10.10.30.10.

So when I run the ***traceroute***, it will tell me that it's going to trace the route to that IP address. And in this particular version of trace route that I'm using here, it will do it to a maximum of 30 hops. But we can see already from this map it's nowhere close to that number of hops to get to Jack's device. The first packet that is sent to jack is one that has a ***traceroute*** set with a time to live of one, which means as soon as it hits the first router, that time to live of one will suddenly be decreased by one, making the time to live zero, and the router will identify that the time to live has been exceeded, and it drops that particular frame.

This router then sends back to Sam a time to live exceeded message with the source IP address as the router that sent the message. So we know that it only took, in this case, two milliseconds because Sam was counting this response. And this response came from 10.10.10.1. That is our first hop along the way. Now Sam will send another message, but this one's going to set the time to live exceeded to two, which means it will make it through the first router and it will decrease by one. But as soon as it gets to the next router along the way and the time to live is decreased to zero, it again is exceeded, and a message is sent back to Sam, and Sam makes a note of the IP address of the router where that particular time to live was exceeded.

So we know the first hop and we know the second hop of the IP addresses of those routers. We perform the process again by taking the time to live and making it three, which means it's going to get through the first two routers. But when it gets to the third router, and the time to live is decreased that last time, a time to live is again exceeded, and that message is sent back to Sam, and the IP address of that router is put into the traceroute. And finally, we'll do a time to live equal to 4, which means it will make it through the first router, the second router, and the third router. But finally, when it gets to Jack's machine, it will decrease one last time to zero, and Jack's device will send back a time to live exceeded message completing the ***traceroute***.

And now we have our complete map of all of the IP addresses that it takes to get between Sam's device and Jack's device. Let's run a ***traceroute*** on my operating system. I'm going to run it with the ***parameter dash q1***. That means it will send a single frame each time it performs that time to live change. And I'm going to send that to 8.8.8.8, which is Google's DNS server. And if I hit Enter, we can see that I've got a ***traceroute*** to that IP address. This is 64 hops maximum. Obviously, we only went 11 hops between my IP address and that Google DNS server. And you can see every hop along the way, every router that was used.

Now, a lot of these were also changed from IP addresses to names. So you could run ***traceroute*** again ***with the dash q1***. I'm going to ***specify dash n*** to tell traceroute not to resolve the names. And when it performs the ***traceroute***, this looks very similar to the ***traceroute*** we were just looking at in our example where it went through the connections again and finally got 11 hops. You can see in this particular ***traceroute***, whichever router ended up as being the ninth route on this particular route, did not respond back with an ICMP time exceeded message.

This isn't completely unusual when you're communicating to a device across the internet. There's multiple paths and redundancy built in. You may take one path on one instance to that device, and a completely different path on another instance. Whenever we're putting the name of a web server in our browser, we don't even think about the IP addresses associated with that web server. We leave it to the domain name services to be able to convert from a fully qualified domain name down to the IP address so that things can communicate across the network. But, of course, from a security perspective, you may want to perform that same look up yourself.

And there are a number of tools that you can use to be able to do that on your computer. One that's very common is ***nslookup***. This is something available on Windows and your POSIX-based systems. So you'll find this already built in on Linux, Mac OS, and other types of systems. This is a utility where you can put in an IP address or the name of a device and it will either perform a lookup or a reverse lookup up using the ***nslookup*** utility. Although ***nslookup*** is still available on most of these operating systems, it is a utility that has been deprecated, and they highly recommend you use the more updated utility ***dig***.

Dig stands for domain information groper. And this is going to provide you with a lot more detail about what it finds on that DNS server. And there are many more options available in ***dig*** that you didn't have available in ***nslookup***. Let's run an ***nslookup*** on Professor Messer. So I'll type in ***nslookup*** and I'll type In www.professormesser.com. And you

can see that it went out to the Google domain server because that is the default for this particular computer. And it got an answer saying that professormesser.com is 104.20.215.49.

Let's run the same lookup, but let's use the ***dig*** utility instead. So I'll use ***dig***, and we'll choose again www.professormesser.com. And you can see we get a lot more output from the ***dig*** command than we did the **nslookup** command. You can see all of the different queries that were done for this particular default ***dig*** command, but you have a lot of other options that you could add here. The question section is going to show you what the actual query was to the DNS server. In this case, it was professormesser.com,. and I'm looking for the address record.

And here's the answers back. The two records from that server is www.professormesser.com. 299 is the cache that is available. I have a very small cache configured so that I could change the IP addresses for my servers very quickly. And you can see we did get the same IP address, 104.20.215.41. And we have a lot of other options in ***dig***. And if you're planning to do a lot of queries, and a lot of lookups, you may want to make sure that you have both of these utilities available on your system.

ARP is the Address Resolution Protocol. This is the protocol that's used across our local network to be able to associate a local IP address with the MAC address of these local devices. There is a cache of these that is stored on our computer in the ARP cache. And we're able to view that ARP cache by using the ***ARP*** command. On most operating systems, we would view that ARP cache by using ***ARP dash a***. It's a very simple command to use, and that allows us to verify that the IP address and MAC address associations are correct for the devices that we're connecting to.

Let's see what's in my ARP cache. I'm going to type ***ARP dash a***. And we're going to see that there are a number of IP addresses, 10.1.10.1 through 10.1.10.249. And for each of those, we can see the MAC addresses that are associated with those IP addresses. If we were ever trying to determine if there might be a man in the middle, or if any of these are not resolving properly, we'll be able to see all of that by looking at our ARP cache.

When you're doing any type of network analysis or network troubleshooting, it's useful to know what the local IP address configuration is of the device that you happen to be working on. The way that you would view this is by either using the ***ipconfig*** command. Or the ***ifconfig*** command. Both of these commands are designed to give you information about the hardware address and the IP addressing on your device. If you have more than one ethernet adapter in your device, then you'll have hardware and IP address information available for each one of those ethernet adapters.

If you're using a ***Windows*** operating system, the command is ***ipconfig***. If using ***Linux, Mac OS***, or almost any other operating system, it's ***ifconfig***. I'm running Mac OS. So let's run the ***ifconfig*** command. And I'm going to ***specify en0***, which is the ethernet adapter on my computer. And you can see it does show me hardware information about this device along with the IPv6 and the IPv4 addresses that are defined on this computer. If we were running Windows, and we ran ***ipconfig***, then we'd be able to see similar information. Here's the Mac address information on this Windows device.

This device was not able to get a DHCP address. So we can see an APIPA address on this device. And you can see if there are any IPv6 or other DNS server configurations all from the ***ipconfig*** command. In a previous video, we talked about using a protocol analyzer to capture packets, but there are utilities you can use to capture packets at the command line. One of these utilities is ***tcpdump***. This allows you to capture packets in most ***Linux and Unix*** flavors. It's also available on Mac OS. And you can also download it for Windows in the utility called ***WinDump***.

You can not only capture the packets, you can apply filters, you can view these packets going by in real time, and if you're just trying to view some very basic information, this might be a very easy way to do it. You can save these packets, and then you can load them up in a protocol analyzer later on if you wanted to because ***tcpdump*** will write them into a standard format called the ***pcap*** format. Whenever you're capturing data on any device, it can be a lot of information. So it may take a little bit of time to figure out how to filter and save the information you want. But this can be a very valuable tool if all you need to do is capture a little bit of data.

Here's some data I captured on my system using ***tcpdump***. You can see it's a combination of IPv6 and IPv4 data. There's a lot of different kinds of data going back and forth. And this can be a little difficult to read if you're not familiar with exactly the format. This becomes a little bit easier over time, but you can see why it's useful sometimes to capture large amounts of data and then import that data later on into a protocol analyzer.

One security utility you do not want to be without is ***nmap***. ***Nmap*** is the network mapper and allows you to gather information from all of the different devices across the network. Using ***nmap***, you can perform a port scan to identify

what services might be available on a device. ***Nmap*** can perform an operating system scan, which means it can determine what operating system is running on a machine without authenticating into that device. It can also perform a service scan. So not only does it know what services are available on that device, it can tell you the name, the version, and other details about that service without actually using that service.

And there are other scripts built into and ***nmap*** that extend the capabilities and allow you to perform vulnerability scans and much more. Let's run a very simple ***nmap*** scan. I'm going to choose one of the devices on my network, and it tells me very quickly that there were 991 ports that were closed, but it did show that port 22 was open, which is normally the SSH service. Port 80 was open for HTTP. Port 443 was open using HTTPS, and there's others here as well.

We can also have ***nmap*** perform an operating system scan. In this particular case, ***nmap*** requires that I run it as roots. So I'm going to perform a ***[? Sudu, ?]*** and I'm going to choose to ***run nmap with a dash O option*** to that same IP address to see if we can figure out what operating system is running on that device. It asks for my password since I asked to run as root. And as it goes out and performs the scan, it will show me information about those open ports.

But notice, it also shows me the Mac address of this device, which is a sinology device. And it is indeed running Linux, and it was able to tell me that all based from this in ***nmap*** scan. When we're accessing services on these devices across the network, we're usually using a front end application. So to access a web server, we're using a browser. To access an FTP server, we're using an FTP client. But you are able to access those services without using those specific clients. And one of the utilities that allows you to do that is ***netcat***.

Netcat as a way to read or write information to or from the network, and you can listen to a particular port number, transfer data to a particular port number, or scan a number of ports and send data to all of them to see what kind of responses you get. We sometimes will see people enable ***netcat*** to listen on a particular port number so that it will act as a shell or a backdoor to a remote device. There's lots of other utilities that do similar functions, such as ***Ncat***. And you can usually find one of these utilities available for the operating system that you're using.

On my machine, I have ***Ncat***. So we're going to use ***Ncat*** to query that device we were scanning earlier. I'm going to run ***Ncat***, and I'm going to specify the IP address of that device and tell it that I want to communicate to that vise on port 80. I'm going to then tell it to perform a get command and specify that I'm using HTTP version 1 to do that. And here's the output that I get from that particular device. It tells me that it's a server running engine x. It is a web server.

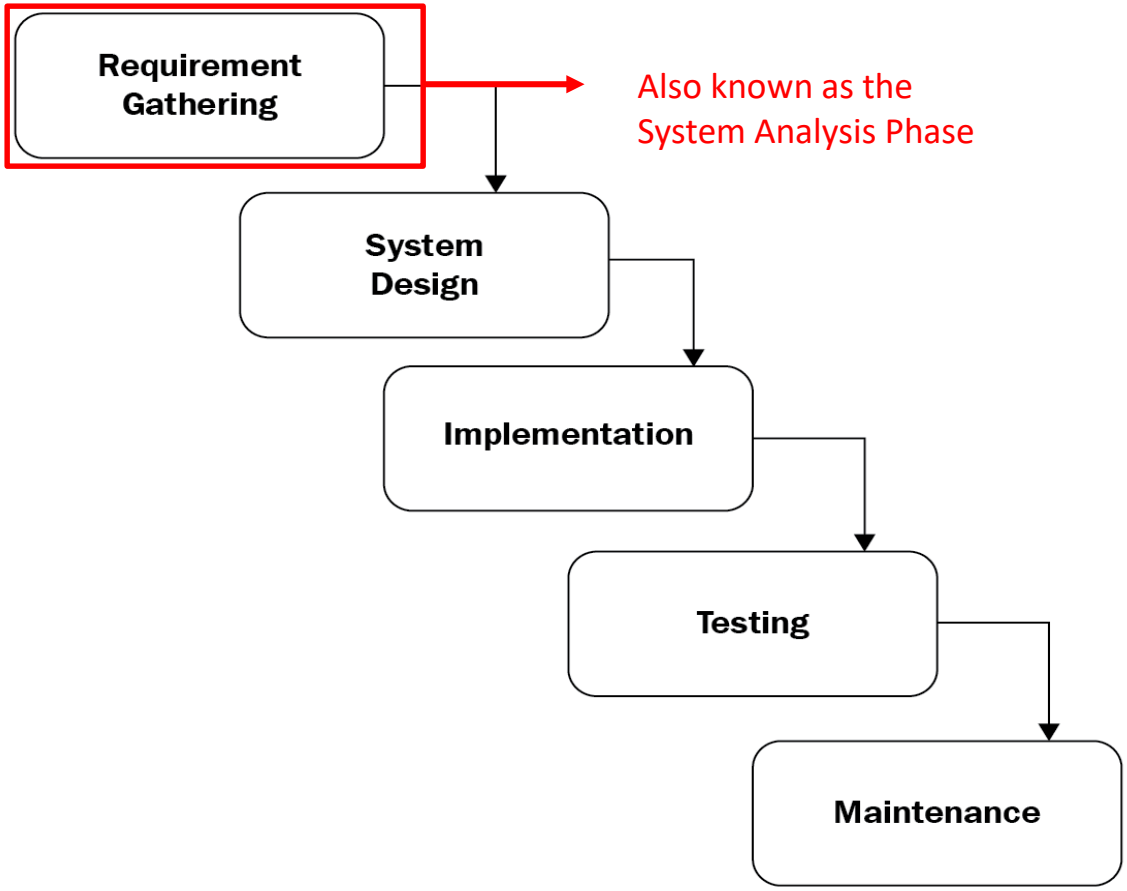
Notice that it says that this particular root of this web server gives me an error 302, which means it has been moved temporarily. The new location is the same thing except using a port number of 5,000. So now I'm found that this server, although it has a port 80 available, it's really telling me that this server is running on port 5,000. So let's run the same command with ***Ncat*** 10.1.10.222. But in this case, let's choose port 5,000.

And I'm going to run the same command using HTTP of 1.0. And you can see I get a much larger web server page back– this is the entire page for this particular web server– all by using ***Ncat*** at the command line rather than querying any of this with a browser.

Waterfall Software Development Life Cycle (SDLC)

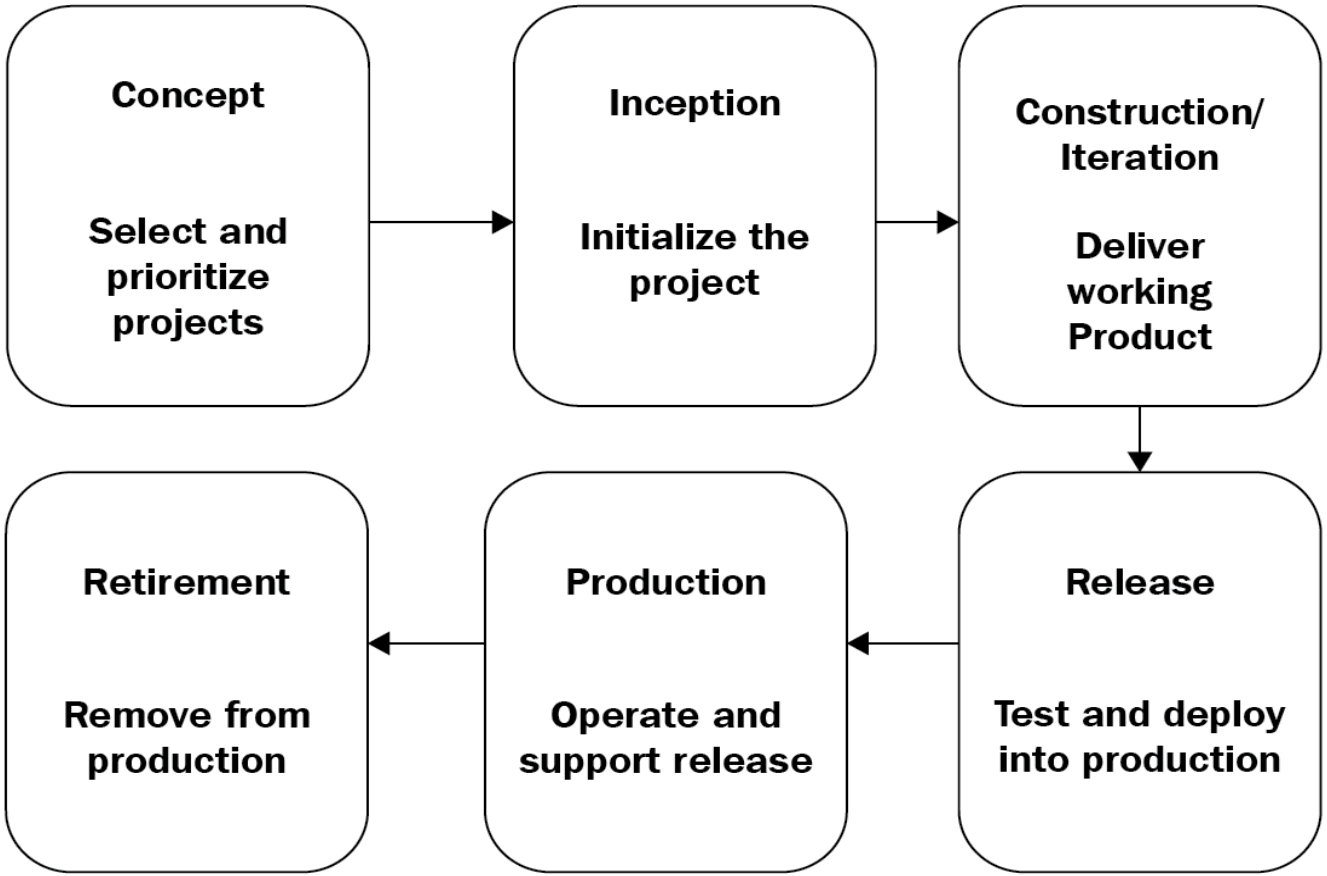
The waterfall model is the traditional method used in the SDLC as it has a linear and sequential pattern to it. The development of the software moves from the top to the bottom, with each phase needing to be completed before the next phase can begin

1. Starts with gathering information about the requirement and analysis
2. Put into the design phase
3. Then it is implemented.
4. The testing phase is carried out before it goes into production
   - Any testing carried out will be rolled back prior to deployment.
5. The maintenance phase is for patching and fixing any bugs.



Agile Development life cycle models

The Agile method anticipates change and breaks down each project into prioritized requirements, delivering each individually within an iterative cycle. Adaptability and customer satisfaction by rapid delivery are the key concepts of this model:



Secure Staging

- Development Phase
  - Requirements are gathered
- Test Phase
  - Code review and dynamic analysis
- Staging Phase
  - Personnel learn how to support software and beta is sent out for customer feedback
- Production Stage
  - Software is fully supported and deployed to the market for sale

Input Validation

- Verifies that an application is properly handling user error exceptions

Regression Testing

- Tests software for security flaws to ensure the vulnerabilities have been remediated and the application is still functioning properly

Analysis: **Dynamic** (code is being executed)

Technique: **Fuzzing** - allows an auditor to test proprietary-software compiled code for security flaws.

Enviroment: **Sandboxing** - best used to isolate and test an identified unknown vulnerability.

Tool: **Fuzzer**

DevSecOps

Network operations and use of cloud computing make ever-increasing use of automation through software code. Traditionally, software code would be the responsibility of a programming or development team. Separate development and operations departments or teams can lead to silos, where each team does not work effectively with the other.

Development and operations (DevOps) is a cultural shift within an organization to encourage much more collaboration between developers and system administrators. By creating a highly orchestrated environment, IT personnel and developers can build, test, and release software faster and more reliably. Many consider a DevOps approach to administration as the only way organizations can take full advantage of the potential benefits offered by cloud service providers.

DevSecOps extends the boundary to security specialists and personnel, reflecting the principle that security is a primary consideration at every stage of software development and deployment. This is also known as shift left, meaning that security considerations need to be made during requirements and planning phases, not grafted on at the end. The principle of DevSecOps recognizes this and shows that security expertise must be embedded into any development project. Ancillary to this is the recognition that security operations can be conceived of as software development projects. Security tools can be automated through code. Consequently, security operations need to take on developer expertise to improve detection and monitoring.

RAID

Disk

There is a need for the disk setup on servers to provide redundancy, sometimes known as *fault tolerance*. In simple terms, this means that if one or more disks fail, the data is still available. There are different **Redundant Array of Independent Disks** (**RAID**) levels, so let's look at each of these in turn, starting with RAID 0:

- **RAID 0**: RAID 0 uses a minimum of two disks with a maximum of 32 disks; see *Figure 12.4*:

Figure 12.4 – RAID 0

This is known as a *stripe set*, as the data is written across *Disks 1-3* in 64 KB stripes. Should one disk fail, then all of the data will be lost, so RAID 0 does not provide fault tolerance or redundancy. The benefit of RAID 0 is its faster read access, so it may be used for the proxy server's cache.

- **RAID 1**: RAID 1 is two disks, known as a *mirror set* where you have an original disk that is live with a copy on the second disk. See *Figure 12.5*:

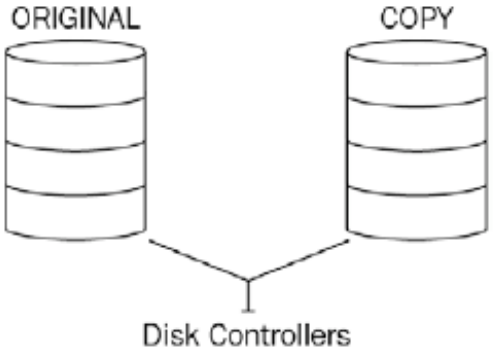


Figure 12.5 – RAID 1

RAID 1 is fault tolerant, and so should *Disk 1* fail, you would *break the mirror* and then activate *Disk 2*. At a later stage, we will add another disk and then re-establish the mirror set.

- **RAID 5**: RAID 5 has a minimum of three disks and is known as a *stripe set with parity*. It is written across the disks in 64 KB stripes just like RAID 0 but, when each stripe is written, one of the disks has a single parity block for each line of data. The parity is shown as shaded in *Figure 12.6*:

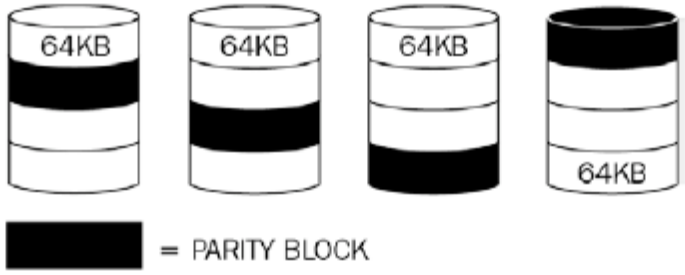


Figure 12.6 – RAID 5

Example: The following diagram (*Figure 12.7*) represents a RAID 5 set where we are using a mathematical equation to represent the disk set so that you can see the impact of losing one disk and then losing two disks:

Figure 12.7 – RAID 5 as a mathematical equation

Each of the disks has a numerical value. For example, if *Disk 3* fails, the equation would be (*7 +? = 10*) and the answer would be *3*. If we lose a second disk, *Disk 1*, the equation would then be (*? + ? = 10*) and you could not work it out. The same happens if you lose two disks; parity cannot recreate the missing data.

- **RAID 6**: RAID 6 has a minimum of four disks and the same configuration as RAID 5, but it has an additional disk that holds another copy of the parity:

Figure 12.8 – RAID 6

A RAID 5 disk set can afford to lose one disk and the data will still be available as it has single parity. The good thing about a RAID 6 set is that it can lose two disks and still be redundant as it has double parity.

- **RAID 10**: RAID 10 is also known as *RAID 1+0*. This is a RAID configuration that combines both mirroring and striping to protect data. It has a mirrored set that is then striped. As long as one disk in each mirrored pair is functional, data can be retrieved:

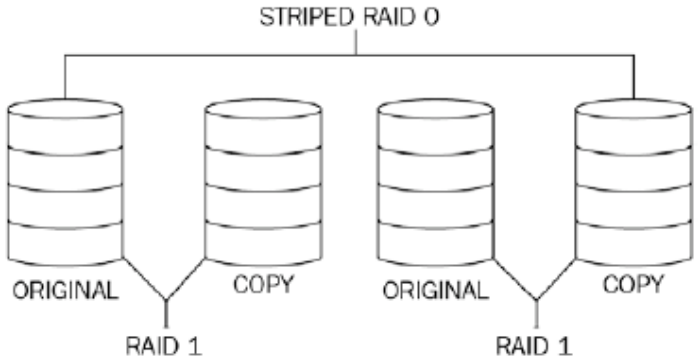


Figure 12.9 – RAID 10

From this diagram, you can see a *RAID 1* on the left and then it is striped, meaning you could lose an entire dataset.

- **Multipath**: This is normally used by a SAN storage solution where there is more than one network path between the SAN storage and the target server. This prevents a single point of failure and provides redundancy as well as a load-balancing capability.

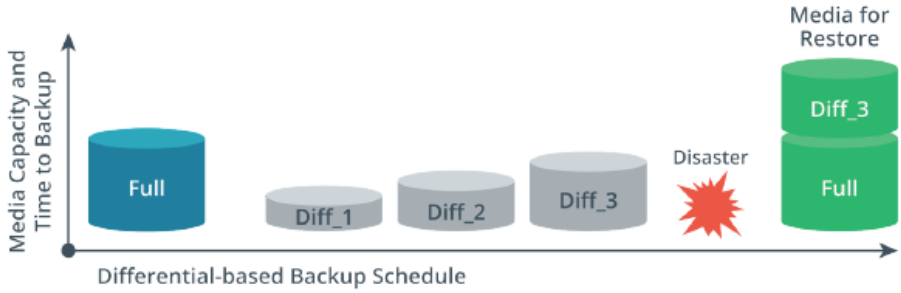
Implementation		Use	Parity	Min # of Disks
RAID 0	Striping	Video/Audio Streaming	No	2
RAID 1	Mirroring	Authentication Servers	No	2
RAID 5	Striping	Email Archives	Single	3
RAID 6	Striping	Identity Management Servers	Double	4
RAID 1+0	Striping + Mirroring	Fast Databases & Application Servers	No	4

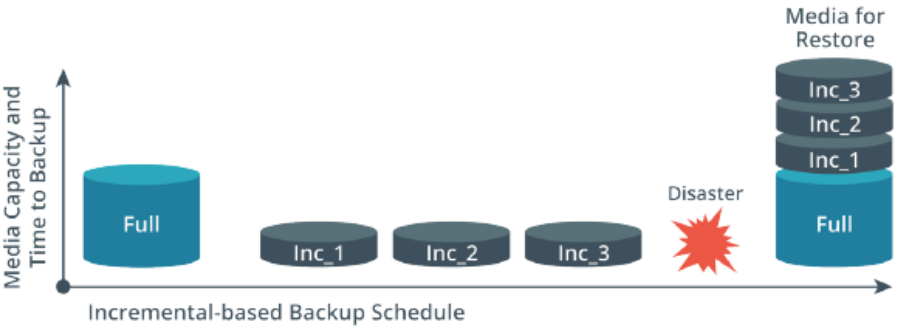
BACKUPS

Full, Incremental, and Differential Backup Types

The following table summarizes the three different backup types.

Type	Data Selection	Backup/Restore Time	Archive Attribute
Full	All selected data regardless of when it was previously backed up	High/low (one tape set)	Cleared
Incremental	New files, as well as files modified since the last backup	Low/high (multiple tape sets)	Cleared
Differential	All new and modified files since the last full backup	Moderate/moderate (no more than two sets)	Not Cleared





Snapshot

A *snapshot is* a copy of a virtual machine at a specific point in time. A snapshot is created by copying the files that store the virtual machine. One of the advantages of a virtual machine over a physical machine is the ease with which the virtual machine can be backed up and restored—the ability to revert to an earlier snapshot is as easy as clicking a button and waiting for the machine to be restored via a change of the files.

Backdoor	user/etc/passwd if!grep--quiet joeuser/etc/passwd then rm –rf/ fi
Buffer Overflow	strcpy x(any#) strcat char *code = “AAAABBBBCCCCDDD”; {char buf [8]; Echo “vrfy ’perl –e print ’hi” x 500’’”
Command Injection	Cd%20../../../etc;cat$20shadow *[note: this is a password attack, but command injection is occurring.]*
JavaScript data insertion	var data=”<test test test>” ++ <../../../../../../etc/passwd>
keylogger	http://www.badsite.com<enter>StanUsr<BackSPACE>erPASSWORD<enter>
LDAP injection	Comptia)(&) (&(username=BOB)(&))
Logic Bomb:	~$ crontab -1 5**** If ($members -notcontains “Kyle”) {Remove-Item -path c:\database -recurse -force} fi
Netcat (nc,ncat)	"questionable socket"
Ping sweep of the class c network	>for i in seq 255; ping –c 1 192.168.0. $i; done
PKI transfer vulnerability	QID 42366 – SSLv3.0 / TLSv1.0 Protocol weak CBC mode Server side vulnerability
RAT	"erratic behavior" open ports: 137(NETBIOS), 443(CHROME EXT) also ".exe" file extension *.*
Rootkit	/kernel32 (Windows) /usr/bin/directory (Linux)
SQL Injection	1=1, 20=-20 union all select OWASP_TOP_10/A1
Trojan	inserts .dll into SysWOW64 or /system32 file
Worm	open ports: 135, 445, 1900, 4444, 5000
XMAS	FIN, PSH, URG
Xml injection	*Look for the* <?xml version=1.0” *and also* <!DOCTYPE foo [<!ELEMENT foo ANY ><!ENTITY bar SYSTEM file:///etc/config>]> <bar><&bar;</bar>
XSS	<script>document.cookie<\script>

This is called a (null) pointer dereference. If this code is executed, the vulnerability that would occur is called a missing null check and the error that would occur is called a null pointer exception.	public class donuts {
	public static void main (String [] args) {
	object stuffed = null;
	stuffed.heat ();
	…
	}
	}

Syslog (System Logging Protocol)
Used to send system log or event messages to a specific server, called a syslog server. Primarily used to collect various device logs from several different machines in a central location for monitoring and review

Router Log	Monitors and records every Internet Protocol (IP) address that every computer on the network visits

Firewall Log	Logs connections that are permitted and traffic that is dropped

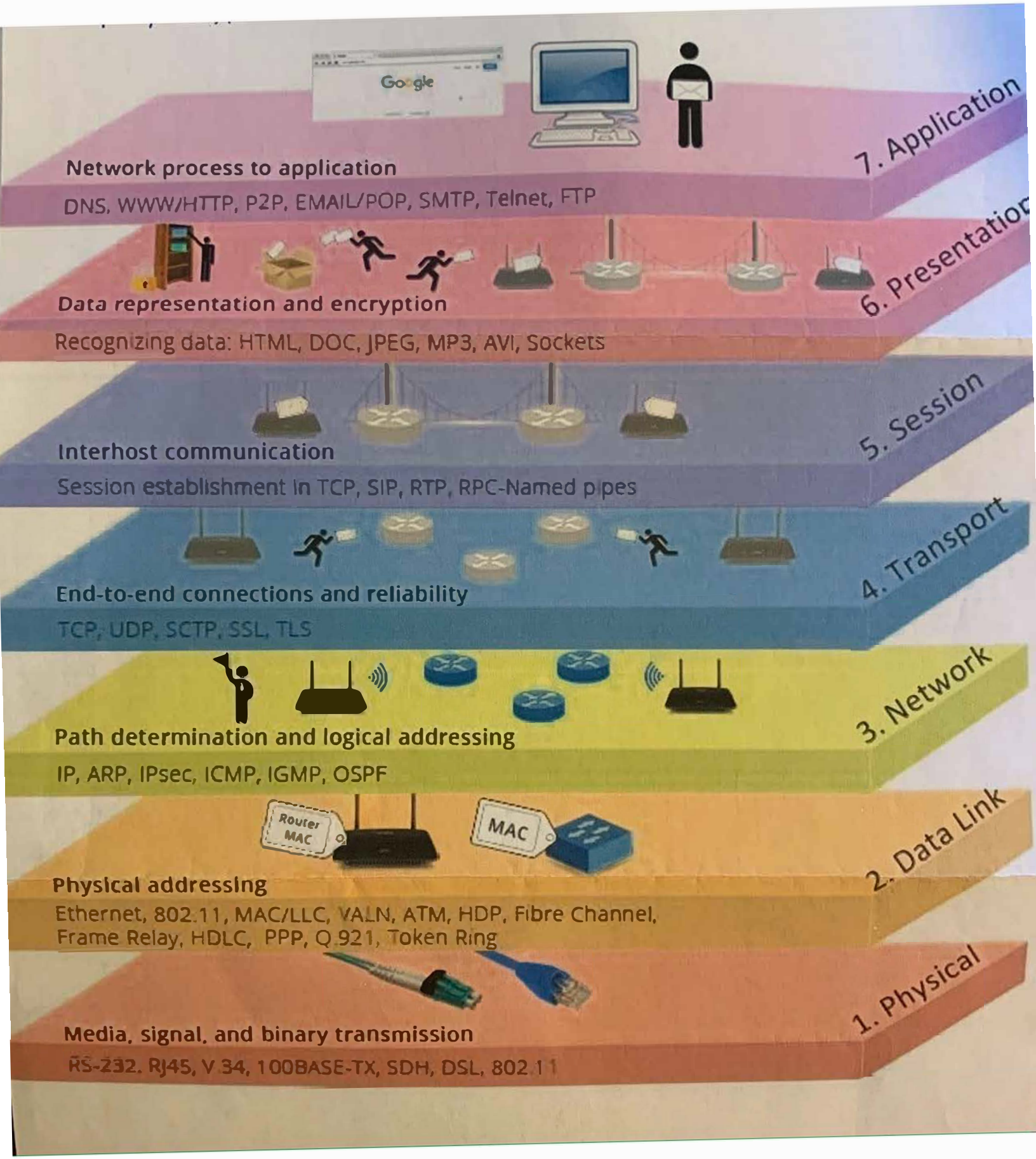
Monitoring Logs

- Continuously record information that can be useful in troubleshooting and gaining information **(performing trend analysis)**

Windows Event Viewer	
• Application Log	– File of events that are logged by a software application.
• System log	– Contains events logged by the operating system components
• Security log	– Contains records of login/logout activity or security-related events

Linux "var/log/" directory	
• var/log/auth.log	– Contains information on successful/unsuccessful logins
• var/log/faillog	– Contains information on failed logins
• var/log/httpd/	– Directory unique to Apache web servers

This page intentionally left blank.


Google
7. Application
Network process to application
DNS, WWW/HTTP, P2P, EMAIL/POP, SMTP, Telnet, FTP
6. Presentation
Data representation and encryption
Recognizing data: HTML, DOC, JPEG, MP3, AVI, Sockets
5. Session
Interhost communication
Session establishment in TCP, SIP, RTP, RPC-Named pipes
4. Transport
End-to-end connections and reliability
TCP, UDP, SCTP, SSL, TLS
3. Network
Path determination and logical addressing
IP, ARP, IPsec, ICMP, IGMP, OSPF
Router MAC
MAC
2. Data Link
Physical addressing
Ethernet, 802.11, MAC/LLC, VALN, ATM, HDP, Fibre Channel,
Frame Relay, HDLC, PPP, Q.921, Token Ring
1. Physical
Media, signal, and binary transmission
RS-232, RJ45, V.34, 100BASE-TX, SDH, DSL, 802.11

Subnetting Basics

In Chapter 7, "IP Addressing," you learned how to define and find the valid host ranges used in a Class A, Class B, or Class C network address by turning the host bits all off and then all on. This is very good, but here's the catch: You were defining only one network. What would happen if you wanted to take one network address range and create six networks from it? You would have to do something called *subnetting*, because that's what allows you to take one larger network and break it into a bunch of smaller networks.

There are loads of reasons in favor of subnetting, including the following benefits:

Reduced Network Traffic We all appreciate less traffic of any kind. With networks, it's no different. Without trusty routers, packet traffic could grind the entire network down to a near standstill. With routers, most traffic will stay on the local network; only packets destined for other networks will pass through the router. Routers create broadcast domains. The more broadcast domains you create, the smaller the broadcast domains and the less network traffic on each network segment.

Optimized Network Performance This is the very cool reward you get when you reduce network traffic!

Simplified Management It's easier to identify and isolate network problems in a group of smaller connected networks than within one gigantic network.

Facilitated Spanning of Large Geographical Distances Because WAN links are considerably slower and more expensive than LAN links, a single large network that spans long distances can create problems in every area previously listed. Connecting multiple smaller networks makes the system more efficient.

Next, we're going to move on to subnetting a network address. This is the good part—ready?

How to Create Subnets

To create subnetworks, you take bits from the host portion of the IP address and reserve them to define the subnet address. This means fewer bits for hosts, so the more subnets, the fewer bits are left available for defining hosts.

Soon, I'll show you how to create subnets, starting with Class C addresses. But before you actually implement subnetting, you really need to determine your current requirements as well as plan for future conditions.

Follow these steps—they're your recipe for solid design:

1. Determine the number of required network IDs:
   - One for each subnet
   - One for each wide area network (WAN) connection
2. Determine the number of required host IDs per subnet:
   - One for each TCP/IP host
   - One for each router interface
3. Based on the previous requirements, create the following:
   - One subnet mask for your entire network
   - A unique subnet ID for each physical segment
   - A range of host IDs for each subnet

Understanding the Powers of 2

By the way, powers of 2 are really important to memorize for use with IP subnetting. To review powers of 2, remember that when you see a number with another number to its upper right (an exponent), this means you should multiply the number by itself as many times as the upper number specifies. For example, 2^3 is $2 \times 2 \times 2$, which equals 8. Here's a list of powers of 2 that you should commit to memory:

$2^1 = 2$

$2^2 = 4$

$2^3 = 8$

$2^4 = 16$

$2^5 = 32$

$2^6 = 64$

$2^7 = 128$

$2^8 = 256$

$2^9 = 512$

$2^{10} = 1{,}024$

$2^{11} = 2{,}048$

$2^{12} = 4{,}096$

$2^{13} = 8{,}192$

$2^{14} = 16{,}384$

If you hate math, don't get stressed out about knowing all these exponents—it's helpful to know them, but it's not absolutely necessary. Here's a little trick, because you're working with 2s: Each successive power of 2 is double the previous one.

For example, all you have to do to remember the value of 2^9 is to first know that $2^8 = 256$. Why? Because when you double 2 to the eighth power (256), you get 2^9 (or 512). To determine the value of 2^{10}, simply start at $2^8 = 256$, and then double it twice.

You can go the other way as well. If you needed to know what 2^6 is, for example, you just cut 256 in half two times: once to reach 2^7 and then one more time to reach 2^6. Not bad, right?

Subnet Masks

For the subnet address scheme to work, every machine on the network must know which part of the host address will be used as the subnet address. This is accomplished by assigning a *subnet mask* to each machine. A subnet mask is a 32-bit value that allows the recipient of IP packets to distinguish the network ID portion of the IP address from the host ID portion of the IP address.

The network administrator creates a 32-bit subnet mask composed of 1s and 0s. The 1s in the subnet mask represent the positions that refer to the network, or subnet, addresses.

Not all networks need subnets, meaning they use the default subnet mask. This is basically the same as saying that a network doesn't have a subnet address. Table 8.1 shows the default subnet masks for Classes A, B, and C. These default masks cannot and do not change. In other words, you can't make a Class B subnet mask read 255.0.0.0. If you try, the host will read that address as invalid and usually won't even let you type it in. For a Class A network, you can't change the first byte in a subnet mask; it must read 255.0.0.0 at a minimum. Similarly, you cannot assign 255.255.255.255, because this is all 1s—a broadcast address. A Class B address must start with 255.255.0.0, and a Class C has to start with 255.255.255.0. Check out Table 8.1.

Table 8.1 Default subnet masks

Class	Format	Default subnet mask
A	*network.host.host.host*	255.0.0.0
B	*network.network.host.host*	255.255.0.0
C	*network.network.network.host*	255.255.255.0

Classless Inter-Domain Routing (CIDR)

Another term you need to know is *Classless Inter-Domain Routing (CIDR)*. It's basically the method that Internet service providers (ISPs) use to allocate a number of addresses to a company or a home connection. They provide addresses in a certain block size; I'll be going into that in greater detail later in this chapter. Another term for the use of different length subnet masks in the network is *variable length subnet masking (VLSM)*.

When you receive a block of addresses from an ISP, what you get will look something like this: 192.168.10.32/28. This is telling you what your subnet mask is. The slash notation (/) means how many bits are turned on (1s). Obviously, the maximum could only be /32 because a byte is 8 bits and there are 4 bytes in an IP address: 4 × 8 = 32. But keep in mind that the largest subnet mask available (regardless of the class of address) can only be a /30 because you have to keep at least 2 bits for host bits.

Take, for example, a Class A default subnet mask, which is 255.0.0.0. This means that the first byte of the subnet mask is all ones (1s), or 11111111. When referring to a slash notation, you need to count all the 1s bits to figure out your mask. The 255.0.0.0 is considered a /8 because it has 8 bits that are 1s—that is, 8 bits that are turned on.

A Class B default mask would be 255.255.0.0, which is a /16 because 16 bits are (1s): 11111111.11111111.00000000.00000000.

Table 8.2 offers a listing of every available subnet mask and its equivalent CIDR slash notation.

Table 8.2 CIDR values

Subnet Mask	CIDR Value
255.0.0.0	/8
255.128.0.0	/9
255.192.0.0	/10
255.224.0.0	/11
255.240.0.0	/12
255.248.0.0	/13
255.252.0.0	/14
255.254.0.0	/15
255.255.0.0	/16
255.255.128.0	/17
255.255.192.0	/18
255.255.224.0	/19
255.255.240.0	/20
255.255.248.0	/21
255.255.252.0	/22
255.255.254.0	/23
255.255.255.0	/24
255.255.255.128	/25
255.255.255.192	/26
255.255.255.224	/27
255.255.255.240	/28
255.255.255.248	/29
255.255.255.252	/30

Although according to RFC 1518, any device or software that claims to be CIDR compliant will allow supernetting, meaning a traditionally Class C address can be used with a /23 subnet mask, in almost all cases. The /8 through /15 can be used only with Class A network addresses; /16 through /23 can be used by Class A and B network addresses; /24 through /30 can be used by Class A, B, and C network addresses. This is a big reason most companies use Class A network addresses. By being allowed the use of all subnet masks, they gain the valuable benefit of maximum flexibility for their network design.

Subnetting Class C Addresses

There are many different ways to subnet a network. The right way is the way that works best for you. In a Class C address, only 8 bits are available for defining the hosts. Remember that subnet bits start at the left and go to the right, without skipping bits. This means that the only Class C subnet masks can be those listed here:

Binary	Decimal	CIDR
00000000	0	/24
10000000	128	/25
11000000	192	/26
11100000	224	/27
11110000	240	/28
11111000	248	/29
11111100	252	/30

We can't use a /31 or /32 because, remember, we have to leave at least 2 host bits for assigning IP addresses to hosts.

Get ready for something special. I'm going to teach you an alternate method of subnetting that makes it a whole lot easier to subnet larger numbers in no time. And trust me, you really do need to be able to subnet fast!

Subnetting a Class C Address: The Fast Way!

When you've chosen a possible subnet mask for your network and need to determine the number of subnets, valid hosts, and broadcast addresses of a subnet that the mask provides, all you need to do is answer five simple questions:

- How many subnets does the chosen subnet mask produce?
- How many valid hosts per subnet are available?
- What are the valid subnets?
- What's the broadcast address of each subnet?
- What are the valid hosts in each subnet?

At this point, it's important that you both understand and have memorized your powers of 2. Please refer to the sidebar "Understanding the Powers of 2" earlier in this chapter if you need some help. Here's how you get the answers to those five big questions:

- *How many subnets?* 2^x = number of subnets. x is the number of masked bits, or the 1s. For example, in 11000000, the number of 1s gives us 2^2 subnets. In this example, there are 4 subnets.
- *How many hosts per subnet?* $2^y - 2$ = number of hosts per subnet. y is the number of unmasked bits, or the 0s. For example, in 11000000, the number of 0s gives us $2^6 - 2$ hosts. In this example, there are 62 hosts per subnet. You need to subtract 2 for the subnet address and the broadcast address, which are not valid hosts.
- *What are the valid subnets?* 256 – *subnet mask* = *block size*, or increment number. An example would be 256 – 192 = 64. The block size of a 192 mask is always 64. Start counting at zero in blocks of 64 until you reach the subnet mask value, and these are your subnets. 0, 64, 128, 192. Easy, huh?
- *What's the broadcast address for each subnet?* Now here's the really easy part. Because we counted our subnets in the last section as 0, 64, 128, and 192, the broadcast address is always the number right before the next subnet. For example, the 0 subnet has a broadcast address of 63 because the next subnet is 64. The 64 subnet has a broadcast address of 127 because the next subnet is 128. And so on. And remember, the broadcast address of the last subnet is always 255.
- *What are the valid hosts?* Valid hosts are the numbers between the subnets, omitting all the 0s and all the 1s. For example, if 64 is the subnet number and 127 is the broadcast address, then 65–126 is the valid host range—it's *always* the numbers between the subnet address and the broadcast address.

I know this can truly seem confusing. But it really isn't as hard as it seems to be at first—just hang in there! Why not try a few and see for yourself?

IPv6 Addressing

186

fe80::5d18:652:cffd:8f52

fe80:0000:0000:0000:5d18:0652:cffd:8f52

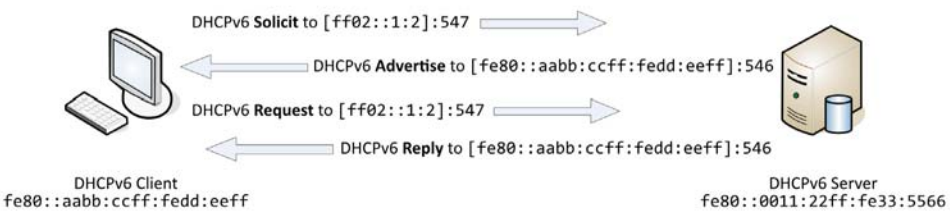
fe80	:	0000	:	0000	:	0000	:	5d18	:	0652	:	cffd	:	8f52
1111111010000000:		0000000000000000:		0000000000000000:		0000000000000000:		0101110100011000:		0000011001010010:		1100111111111101:		1000111101010010

16 bits = 2 bytes = 2 octets

128 bits = 16 bytes

DHCPv6

Very similar process to DHCPv4 - udp/546 (client) and udp/547 (server)



Subnet Classes

Class	Leading Bits	Network Bits	Remaining Bits	Number of Networks	Hosts per Network	Default Subnet Mask
Class A	0xxx (1-126)	8	24	128	16,777,214	255.0.0.0
Class B	10xx (128-191)	16	16	16,384	65,534	255.255.0.0
Class C	110x (192-223)	24	8	2,097,152	254	255.255.255.0
Class D (multicast)	1110 (224-239)	Not defined	Not defined	Not defined	Not defined	Not defined
Class E (reserved)	1111 (240-254)	Not defined	Not defined	Not defined	Not defined	Not defined

RFC 1918 Private Addresses

IP address range	Number of addresses	Classful description	Largest CIDR block (subnet mask)	Host ID size
10.0.0.0 – 10.255.255.255	16,777,216	single class A	10.0.0.0/8 (255.0.0.0)	24 bits
172.16.0.0 – 172.31.255.255	1,048,576	16 contiguous class Bs	172.16.0.0/12 (255.240.0.0)	20 bits
192.168.0.0 – 192.168.255.255	65,536	256 contiguous class Cs	192.168.0.0/16 (255.255.0.0)	16 bits

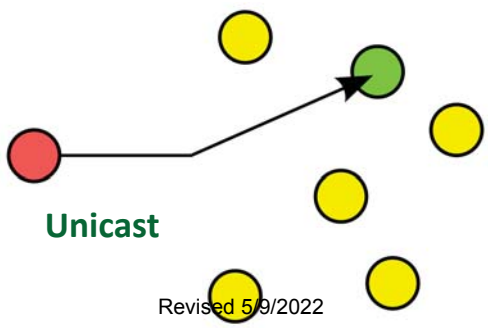
CIDR (Classless Inter-Domain Routing

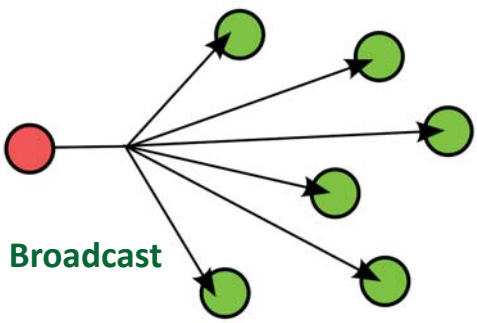
CIDR	Mask	Classful		IPv4 Addresses	
/0	0.0.0.0			2^32	4,294,967,296
/1	128.0.0.0			2^31	2,147,483,648
/2	192.0.0.0			2^30	1,073,741,824
/3	224.0.0.0			2^29	536,870,912
/4	240.0.0.0			2^28	268,435,456
/5	248.0.0.0			2^27	134,217,728
/6	252.0.0.0			2^26	67,108,864
/7	254.0.0.0			2^25	33,554,432
/8	255.0.0.0	A	16,777,216	2^24	16,777,216
/9	255.128.0.0			2^23	8,388,608
/10	255.192.0.0			2^22	4,194,304
/11	255.224.0.0			2^21	2,097,152
/12	255.240.0.0			2^20	1,048,576
/13	255.248.0.0			2^19	524,288
/14	255.252.0.0			2^18	262,144
/15	255.254.0.0			2^17	131,072
/16	255.255.0.0	B	65,536	2^16	65,536
/17	255.255.128.0			2^15	32,768
/18	255.255.192.0			2^14	16,384
/19	255.255.224.0			2^13	8,192
/20	255.255.240.0			2^12	4,096
/21	255.255.248.0			2^11	2,048
/22	255.255.252.0			2^10	1,024
/23	255.255.254.0			2^9	512
/24	255.255.255.0	C	256	2^8	256
/25	255.255.255.128			2^7	128
/26	255.255.255.192			2^6	64
/27	255.255.255.224			2^5	32
/28	255.255.255.240			2^4	16
/29	255.255.255.248			2^3	8
/30	255.255.255.252			2^2	4
/31	255.255.255.254			2^1	2
/32	255.255.255.255			2^0	1

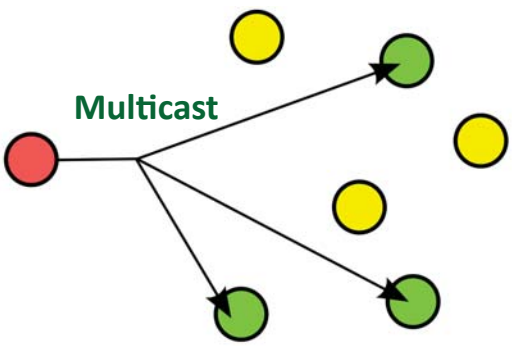
APIPA (Automatic Private IP Addressing)

- 169.254.0.1 through 169.254.255.254
- First and last 256 addresses are reserved, making the functional block 169.254.1.0 through 169.254.254.255

Network Communication







Revised 5/9/2022

POINT-TO-POINT -PPP – EXPLAINED

WHAT IS PPP?

A point-to-point connection is one of the most common types of WAN connection. PPP connections are used to connect LANs to service provider WANs, and to connect LAN segments within an organization network. A LAN-to-WAN point-to-point connection is also referred to as a serial connection or leased-line connection because the lines are leased from a carrier (usually a telephone company) and are dedicated for use by the company leasing the lines.

Simply, when you establish a connection to your ISP (Internet Service Provider) through a modem. The connection between the ISP and you make up two points on the network. Therefore, the protocol that is used for establishing this connectivity between the two of you is the Point-to-Point Protocol or the PPP.

Note: The default serial encapsulation method when you connect two Cisco routers is HDLC. This means Cisco HDLC can only work with other Cisco devices. However, when you need to connect to a non-Cisco router, you should use PPP encapsulation.

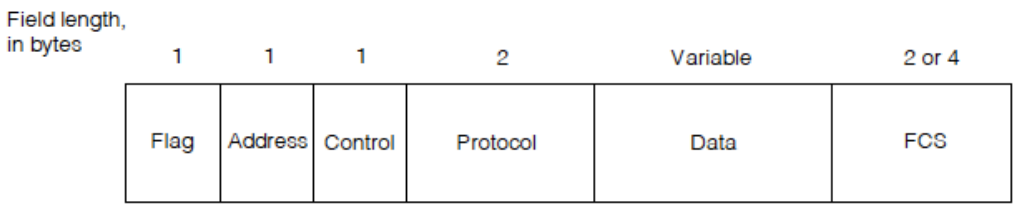
The basic purpose of PPP at this point is to transport layer-3 packets across a Data Link layer point-to-point link. This is one of many advantages to using PPP, it is not proprietary.

PPP can be used over twisted pair, fiber-optic lines, and satellite transmission. PPP provides transport over ATM, Frame Relay, ISDN and optical links. For security, PPP allows you to authenticate or secure connections using either Password Authentication Protocol (PAP) or the more effective Challenge Handshake Authentication Protocol (CHAP).

PPP contains four main components:
EIA/TIA-232-C, V.24, V.35, and ISDN A Physical layer international standard for serial communication.

Data Encapsulations: this is a method used to encapsulate multi-protocol datagrams. Different network-layer protocols are simultaneously transported and encapsulated over the same link, the flexibility of the PPP design enables it to be compatible with most supporting network devices.

The PPP control procedures use the definitions and control field encodings standardized in ISO 4335-1979 and ISO 4335-1979/Addendum 1-1979. The PPP frame format appears in the figure below.



Six Fields Make Up the PPP Frame

• Flag—A single byte that indicates the beginning or end of a frame. The flag field consists of the binary sequence 01111110.

• Address—A single byte that contains the binary sequence 11111111, the standard broadcast address. PPP does not assign individual station addresses.

• Control—A single byte that contains the binary sequence 00000011, which calls for transmission of user data in an unsequenced frame. A connectionless link service similar to that of Logical Link Control (LLC) Type 1 is provided.

• Protocol—Two bytes that identify the protocol encapsulated in the information field of the frame. The most up-to-date values of the protocol field are specified in the most recent Assigned Numbers Request For Comments (RFC).

• Data—Zero or more bytes that contain the datagram for the protocol specified in the protocol field. The end of the information field is found by locating the closing flag sequence and allowing 2 bytes for the FCS field. The default maximum length

of the information field is 1,500 bytes. By prior agreement, consenting PPP implementations can use other values for the maximum information field length.

• Frame check sequence (FCS)—Normally 16 bits (2 bytes). By prior agreement, consenting PPP implementations can use a 32-bit (4-byte) FCS for improved error detection.

The LCP can negotiate modifications to the standard PPP frame structure. Modified frames, however, always will be clearly distinguishable from standard frames.

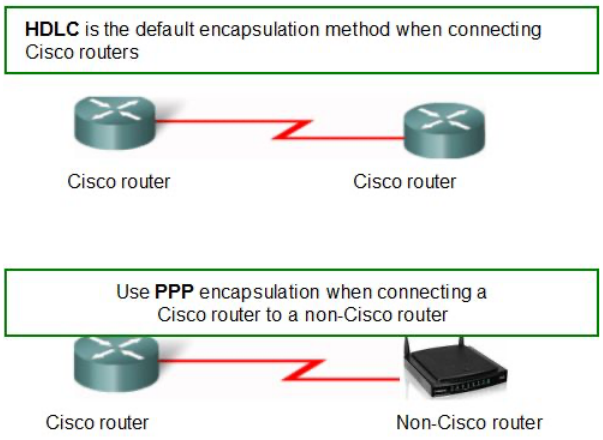
HDLC: A method for encapsulating datagrams over serial links.

Link Control Protocol: The LCP is used to establish, configure, and test the data link connection. It's flexible in handling different sizes of packets, detect a looped-back link, configuration errors, and terminate the link.

Network Control Protocol: NCP is used for establishing and configuring different Network layer protocols. PPP enables the simultaneous use of multiple Network layer protocols. Some of the more familiar NPCs are:

- Internet Protocol Control Protocol
- AppleTalk Control Protocol
- Novell IPX Control Protocol
- Cisco Systems Control Protocol
- SNA Control Protocol
- Compression Control Protocol.

Layer	PPP Encapsulation Protocols
3	Upper-Layer Protocols (IP, IPX, Apple Talk)
2	1. Network Control Protocol (NCP) (Specific to each Network-layer Protocol) 2. Link Control Protocol (LCP) 3. High-Level Data Link Control Protocol (HDLC)
1	Physical Layer (EIA/TIA-232, V24, V35, ISDN)



Summary of Establishing a Point-to-Point WAN Connection with PPP

■ PPP is a common Layer 2 protocol for the WAN. Two components of PPP exist: LCP negotiates the connection and NCP encapsulates traffic.
■ You can configure PPP to use PAP or CHAP. PAP sends everything in plain text. CHAP uses an MD5 hash.
■ Common PPP verification commands include show interface to verify PPP encapsulation and debug ppp negotiation to verify the LCP handshake.

PPP – CHAP EXPLANATION WITH EXAMPLES

CHALLENGE HANDSHAKE AUTHENTICATION PROTOCOL (CHAP).

Challenge Handshake Authentication Protocol (CHAP) is more secure than PAP. It involves a three-way exchange of a shared secret. During link establishment, CHAP conducts periodic challenges to make sure that the remote host still has a valid password value. While PAP basically stops working once authentication is established, this leaves the network vulnerable to attack.

HOW CHAP WORKS

After the PPP link encapsulation phase is complete, the local router sends a challenge message to the remote host.
The remote host sends a response with a value calculated using a one-way hash function, which is normally Message Digest 5 (MD5) based on the password and challenge message.

The local router checks the response from the remote host against its own calculation of the expected hash value. If there is a match, the initiating host acknowledges the authentication. If the values don't match, it immediately terminates the connection.

ADVANTAGES OF CHAP

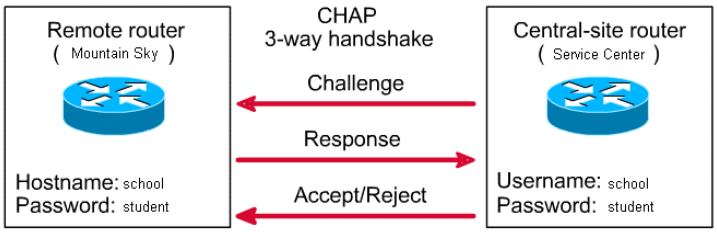
CHAP provides protection against playback attack by using different challenge values that are unique and random. Because the challenge is unique and unpredictable, the resulting hash value is also unique and random. Which makes it difficult for 'guessing'.

The use of repeated and different challenges limits the time of exposure to any single attack. The local router or a third-party authentication server is in control of the frequency and timing of the challenges.

NOTE:

You can enable either PAP or CHAP or both on a network. If both authentications are enabled, PAP is requested during link negotiation. If the network device suggests using CHAP or simply refuses the first method, then the second method is tried.

Some remote devices support CHAP only and some PAP only. It is highly recommended to use both on Cisco routers for maximum data security.
PAP usernames and passwords are sent as clear-text strings and can be intercepted and reused.
CHAP has eliminated most of the known security holes.



The Central-site router initiates the 3-way handshake and sends a challenge message to the Mountain Sky router, who responds to the Central-site's CHAP challenge by sending its username and password. The Central-site router checks to see if Mountain Sky's username and password is in its local database for a possible match, and if there is a match, it accepts the connection. If not, it rejects.

PAP EXPLAINED WITH EXAMPLES

WHAT IS PASSWORD AUTHENTICATION PROTOCOL?

Password Authentication Protocol (PAP) is a very basic two-way process. The username and password are sent in plain text, there is no encryption or protection. If it is accepted, the connection is allowed. PAP is not interactive in anyway, PAP is not considered a strong authentication protocol.

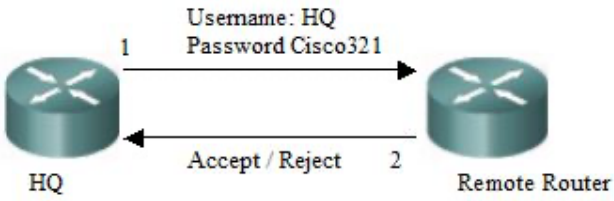
When the ppp authentication pap command is used, the username and password are sent as one LCP data package, rather than the server sending a login prompt and waiting for a response.

Although, PAP cannot be said to be a very strong authentication protocol but, there are times when using PAP can be justified. PAP may be used in the following situations:

1. When an installed network applications does not support CHAP
2. Incompatibilities between different vendor implementations of CHAP
3. Circumstances where a plain text password must be available to simulate a login at the remote host

Password Authentication Protocol (PAP)

PAP 2-way handshake



The example above is of a two-way PAP authentication configuration. Both routers authenticate and are authenticated, so the PAP authentication commands emulate each other. The PAP username and password that each router sends must match those specified with the username name password command of the other router.

SUMMARY:

PAP provides a simple method for a remote host to establish its identity using a two-way handshake. This is done only on initial link establishment. The hostname on one router must match the username the other router has configured. The passwords do not have to match.

Make sure you know…

ISO 27001 – Security techniques for *Information Security Management Systems*:

ISO 27002 – *Code of Practice for Information Security Controls*. The aim of this standard is to improve the management of information

ISO/IEC 27005:2018 -- Information technology — Security techniques — Information security risk management

ISO/IEC 27017:2015 - Information technology — Security techniques — Code of practice for information security controls based on ISO/IEC 27002 for cloud services

ISO 27701 – An extension to 27001/27002 for *Privacy Information Management* – Requirements and Guidelines

ISO 31000 – About managing risk for company organizations and management in general; information can be found on its website

Statements on Standards Attestation Engagements (SSAE): SSAE 18 is an audit standard to enhance the quality and usefulness of Service Organization Control (SOC) reports.

SOC Type 2 Reports: These are reports on the internal controls of the security, processing, and handling of users' data to ensure that it is kept confidential and that privacy is maintained. There are two types: type 1 is to do with the suitability of the design of controls, and type 2 is to do with the effectiveness of the controls.

General Data Protection Regulation (GDPR): The European Union's (EU's) deals with the handing of data while maintaining the privacy and rights of an individual.

Payment Card Industry Data Security Standard (PCI DSS): PCI DSS deals with the handling and storage of data used for card payments.

Vulnerability Databases: The National Institute of Standards and Technology (NIST) is a US Government body that provides a National Vulnerability Database, which comprises:

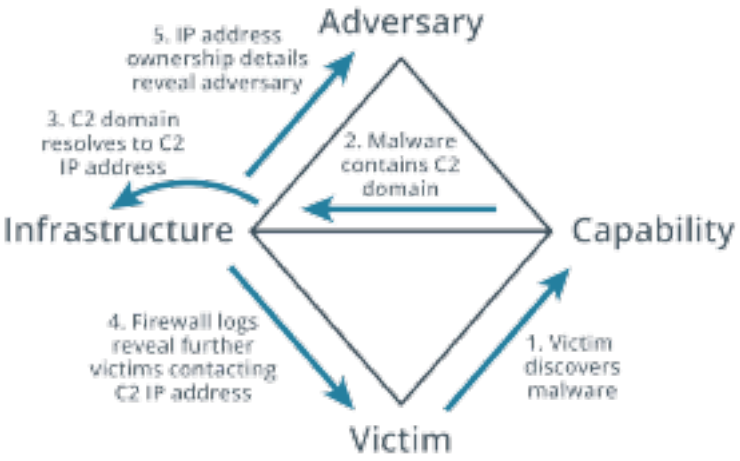
Common Vulnerabilities and Exposures (CVE) and uses the:

Common Vulnerability Scoring System (CVSS) to show the level of severity of each of the vulnerabilities.

The MITRE ATT&CK framework is a database of threat actors, their techniques, and the vulnerabilities that they exploit. This helps security teams protect themselves against such attacks.

The Diamond Model of Intrusion Analysis

The Diamond Model of Intrusion Analysis suggests a framework to analyze an intrusion event (E) by exploring the relationships between four core features: adversary, capability, infrastructure, and victim. These four features are represented by the four vertices of a diamond shape. Each event may also be described by meta-features, such as date/time, kill chain phase, result, and so on. Each feature is also assigned a confidence level (C), indicating data accuracy or the reliability of a conclusion or assumption assigned to the value by analysis.

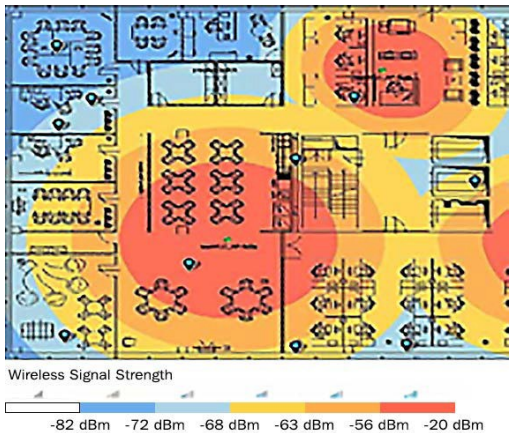


Stages of the Cyber Kill Chain	
Reconnaissance	Calling employees, sending emails, social engineering, dumpster diving
Weaponization	Create malware payload
Delivery	Delivery medium such as USB, email, web page
Exploitation	Executing code via a vulnerability
Installation	Installing malware on the asset
Command and Control	Infected system sends back information to the attacker
Action on Objectives	'Hands-on keyboard' – attack complete

FTK imager: This can be used to collect copies of data without making changes to the original evidence

Autopsy: This can be used to analyze hard drives, smartphones, and media cards. It has a built-in translator to translate foreign languages into English.

Heat Map: A heat map shows your wireless coverage. The red and orange areas indicate good coverage, but the blue areas indicate poor coverage:



Site Survey: Before we install a wireless network, we need to complete a site survey so that we identify what could cause interference with the wireless network.

Managed Security Service Provider (MSSP): An MSSP will maintain the security environment for companies that will include enterprise firewalls, intrusion prevention and detection systems, and SIEM systems.

NMAP flags that are in "The Folder of Knowledge"
Netstat flags that are in "The Folder of Knowledge"

WPA3 SAE --- Simultaneous Authentication of Equals replaces PSK

Next Generation Secure Web Gateway (SWG): An SWG acts like a reverse proxy, content filter, and an inline NIPS.

Next-Generation Firewall (NGFW): An NGFW is more than a traditional firewall. It has the ability to act as a stateful firewall by carrying out deep packet filtering

CGI –Common Gateway Interface - know languages used to communicate from the web server to the database (this is some current scenarios found in 501)

LAMP –Server stack - **LAMP** is an acronym for a stack typically consisting of the Linux operating system, the Apache HTTP **Server**, the MySQL relational database management system, and the PHP programming language. It is a platform mainly used to develop dynamic websites.

MEAN –Server stack - **MEAN** (MongoDB, Express.js, AngularJS (or Angular), and Node.js) is a free and open-source JavaScript software **stack** for building dynamic web sites and web applications.

Reflect XSS - Reflected cross-site scripting (or **XSS**) arises when an application receives data in an HTTP request and includes that data within the immediate response in an unsafe way.

Stored XSS - **Stored** cross-site scripting (also known as second-order or persistent **XSS**) arises when an application receives data from an untrusted source and includes that data within its later HTTP responses in an unsafe way.

DOM Based XSS - DOM Based XSS (or as it is called in some texts, "type-0 XSS") is an XSS attack wherein the attack payload is executed as a result of modifying the DOM "environment" in the victim's browser used by the original client side script, so that the client side code runs in an "unexpected" manner.

SSRF (Server Side Request Forgery) – See "Threats, Attacks, and Vulnerabilities."

Shadow IT: New threat actor. This is where people plug their own computers and devices into your network without consent.

Spraying Attack: The hacker first of all searches the internet for people who work within an organization and sees whether they can work out the standard naming convention.

AGILE

- Daily standups
- Sprints
- Iterations

Scrum --- Scaled down AGILE model

- Scrum Master
- Team
- Product

Netflow: This is a CISCO product that monitors network traffic, so that they can identify the load on the network. This helps you utilize your network traffic efficiently. During an investigation, it can help identify patterns in network traffic.

Sflow: This is a multi-vendor product that gives you clear visibility of network traffic patterns. This can help identify malicious traffic so that we can keep the network secure and safe.

Curl: This is a command-line tool used to transfer data. It can also be used in banner grabbing; this is fetching remote banner information from web servers. -s is silent and -I fetches the HTTP headers.

SOAP (Simple object access protocol) https://www.soapui.org/learn/api/soap-vs-rest-api/

Right to Audit Clause - Audits are the mechanism used to verify that systems are performing to their designed levels of purpose, security, and efficiency. The ability to audit involves access to a system and the data. When the information is stored or processed in the cloud, users need the ability to audit the cloud provider. The level and scope of the audit can vary given the dynamic natures of both the cloud and the regulatory environment, but one thing does not vary. The only rights the customer has are detailed in the service level agreements/contracts with the cloud provider. This makes the Right to Audit clause a critical requirement of any service level agreement, and its specificity needs to match the operational and regulatory scope of the cloud engagement.

HTTP Strict Transport Security

HSTS ensures that the browser will ignore all HTTP connections and accept only secure connections. Prevents XSS.

DNS Sink hole - https://resources.infosecinstitute.com/topic/dns-sinkhole/

A primary source of threat intelligence is the **dark web**. Here is a summary of what you learned today:

Terminology:

Please note: The Internet and the World Wide Web are NOT synonymous. They are two different animals!

- The **Internet** is an electronic communications network that connects computer networks and organizational computer facilities around the world.

- The **World Wide Web** is actually a layer that sits on top of the Internet and uses Internet technology. The WWW is a global collection of documents and other resources, linked by hyperlinks and URIs. **Web** resources are accessed using HTTP or HTTPS, which are application-level, by means of a software application called a web browser.

- The Clearnet is another term for the World Wide Web

- The Deep Web— any part of the World Wide Web that is not indexed by a search engine. This includes pages that require registration, pages that block search indexing, unlinked pages, pages using nonstandard DNS, and content encoded in a nonstandard manner. It is still part of the Clearnet.

- The Dark Net—a network established as an overlay to the Internet infrastructure by software, such as The Onion Router (TOR), Freenet, or I2P, that acts to anonymize usage and prevent a third party from knowing about the existence of the network or analyzing any activity taking place over the network. Onion routing, for instance, uses multiple layers of encryption and relays between nodes to achieve this anonymity.

- The Dark Web— sites, content, and services accessible ***only*** over a dark net. While there are dark web search engines, many sites are hidden from them. Access to a dark web site via its URL is often only available via "word of mouth" bulletin boards.

Safety Tips

The Dark Net is a very interesting place, but it can also be very dangerous. There are unscrupulous people of the very WORST kind trolling the Dark Net. You can browse the Dark Net with confidence if you take the necessary precautions to stay SAFE!

1. Turn on your VPN! I use PIA (Private Internet Access), but there are other reliable products. ProtonVPN is very good. There are others, just make sure you investigate them first.

2. LEARN before you browse! I use the TOR browser (The Onion Router), and I'm very happy with it. Once you launch the browser, you will want to click the onion in the upper left-hand corner and READ EVERYTHING!. There is also some useful information that you can find on the OSINT Framework (https://osintframework.com). If you use the TOR browser, there are a lot of resources there to learn just about anything.

3. Make sure you are setting your browsers security setting to "Safest." In some situations, such as if you want to access the OSINT framework from the Dark Net, you may need to use "Safer" instead of "Safest," but when you are just browsing, ALWAYS use "Safest." You can access the security settings by clicking the little shield to the right of the address bar.

4. Do not EVER provide your true contact details and name to ANYONE on the Dark Net. It is very dangerous to do that! You can establish a Dark Net email account and use that for Dark Net communications. I realize some of you will want to check out Facebook's Dark Net Onion site- do it safely!

5. You will know if you are on an onion site because there will be an onion where you will see a lock on a Clearnet site. You can click that little onion for additional information.

6. If you want to find The Hidden Wiki, you can access it here: http://zqktlwiuavvvqqt4ybvgvi7tyo4hjl5xgfuvpdf6otjiycgwqbym2qad.onion/wiki/index.php/Main_Page

7. DO NOT DO NOT DO NOT DO NOT DO ANYTHING ILLEGAL! Do not do anything that you wouldn't want your mother or grandmother to know about! You are UNITED STATES AIRMAN! Keep us proud of you!

8. If you have questions, please ASK!

donna.schwartz.1@us.af.mil
228-256-9111