# CISCO CCNA SECURITY 210-260

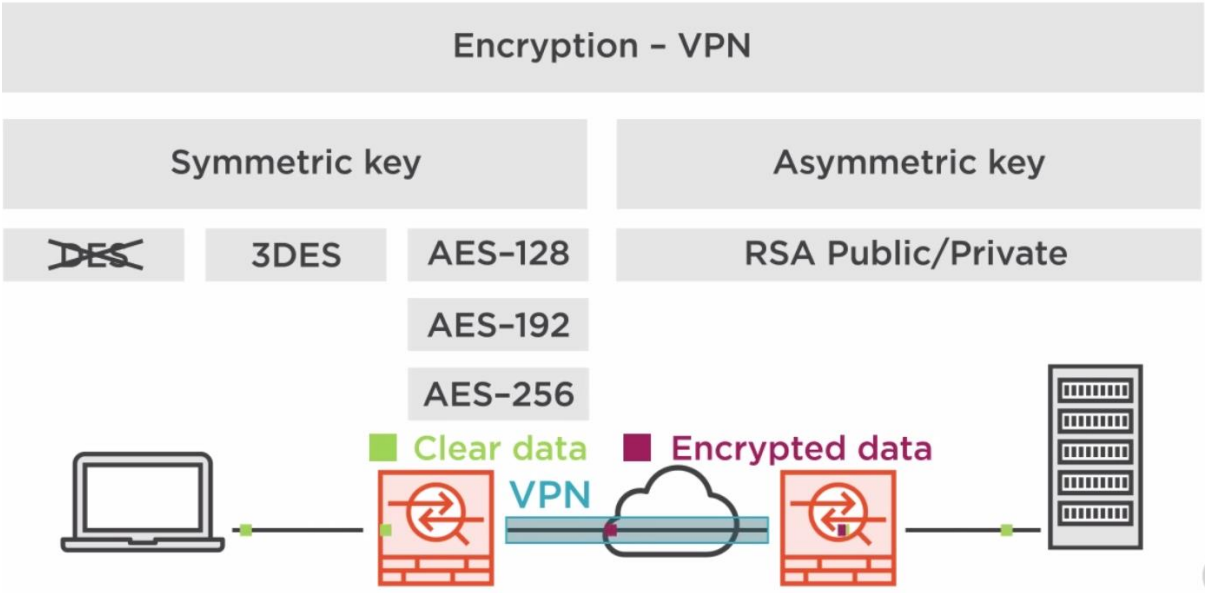
Clive Micallef

# Table of Contents

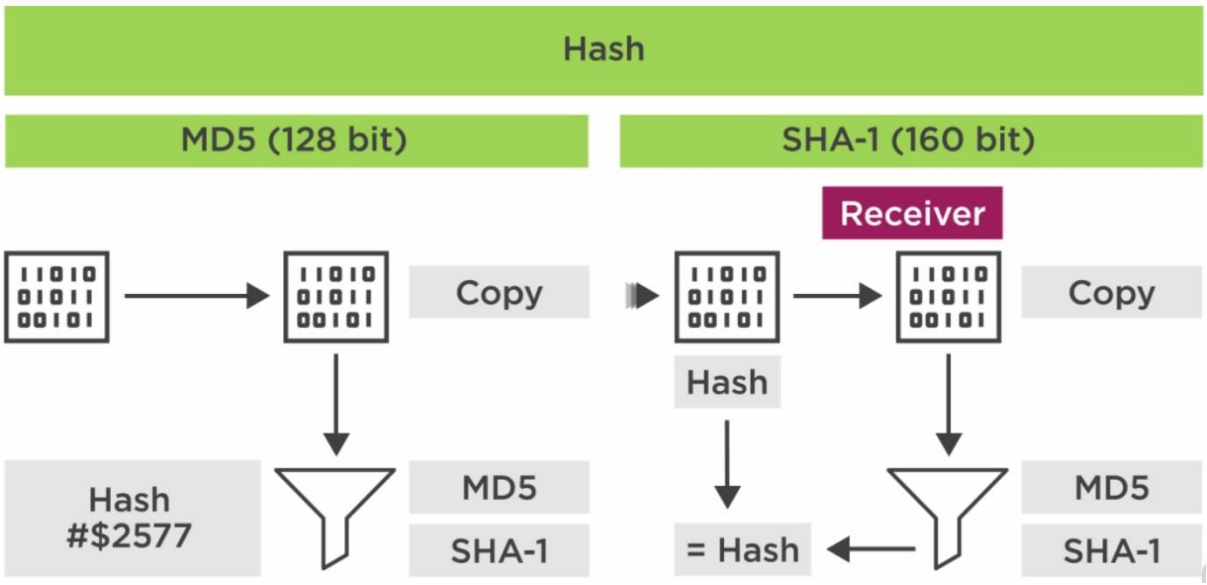
# CHAPTER 1: Networking Security Concepts

**Confidentiality, Integrity and Availability**

- **Confidentiality:** Only the **authorized individuals/systems** can view sensitive or classified information. This also implies that unauthorized individuals should not have any type of access to the data. Regarding data in motion, the primary way to protect that data is to **encrypt** is **before** sending it over the network. Another option you can use with encryption is to use **separate networks** for the transmission of confidential data.




- **Integrity:** Changes made to the data are done only by **authorized individual/systems**. Corruption of data is a failure to maintain data integrity.



- **Availability:** This applies to **systems** and to **data**. If the network or data is **not available** to authorized users, perhaps because of a **denial of service** attack or maybe because of a general **network failure**, the impact may be significant to companies and users who rely on that network as a business tool. The failure of a system, to include data, applications, devices and networks generally equates to loss of revenue.

    - Maintain hardware
    - Make upgrades
    - Have a failover plan
    - Prevent bottleneck

**SEIM Technology (Security Incident Events Monitoring)**

**Security information and event management** (SIEM) is an approach to security management that seeks to provide a holistic view of an organization's information technology (IT) security.
In Cisco terms this applies to the collection, correlation and acting on security information.

**SEIM is supposed to integrate into the overall architecture of the network.** What this means is the security and monitoring of a network should be aspects of the initial design and not afterthoughts as they so often are.

**Logging**

- Along with monitoring is a critical aspect of network logging.
- Needed for troubleshooting and policy-compliance auditing.
- Enables you to recognize the start of an attack or an attack in progress.
- Supported by all Cisco security devices.

**Internal Logging locations**

- Console
- Monitor
- Memory Buffer
- Flash Memory

**External Logging locations**

- Syslog server
- SNMP Trap

**SEIM Capabilities**

- Data aggregation
- Correlation
- Alerting
- Dashboards
- Compliance
- Retention
- Forensic Analysis

**Cost-Benefit Analysis of Security**

**Asset**: An item that is to be protected and can include **property**, people and **information**/**data** that gave **value** to the company. This includes intangible items such as proprietary information or trade secrets and the reputation of the company. The data could include company records, client information, and proprietary software and so on.

**Vulnerability**: An **exploitable weakness** of some type. That exploitation might result from a **malicious attack** or it might be accidentally triggered because of a failure or weakness in the policy implementation or software running on the network. Example: Physical environment, software bugs, Protocol/ system flaws, weak passwords.

**Threat**: This is what you are protecting against. A threat is anything that **attempts** to **gain unauthorized access** to compromise, destroy or damage an asset. Threats are often realized via an attack or exploit that takes advantages of an existing vulnerability. Example: Physical, malicious codes, phishing, social engineering.

**Risk**: The potential for **unauthorized access** to **compromise**, **destroy** or **damage** an **asset**. If a threat exists but proper countermeasures and protections are in place, the potential for the threat to be successful is reduced.

**Countermeasure**: A **device** or **process** that is implemented to counteract a potential threat, which thus reduces risk.

**Identify Common Network Security Zones**

Zones are important concept in security. In fact the definition of a firewall is that is controls access between zones. In terms of firewalls (ASA, IOS) these are 3 zones to be aware of:

- **Inside** – The local LAN side of the firewall. Typically high security, #100 and trusted.
- **Outside** – The wild untamed internet, lowest security, #0 and not trusted.
- **DMZ** – This is a zone that has one foot in each camp – lower security for internet facing servers, proxies, #1-99.

Modern firewalls perform inspection of traffic and use that inspection to form tables of what traffic is allowed to flow from untrusted to trusted network. By default, no traffic is allowed to flow from untrusted to trusted so:

- Inside to Outside traffic is ok
- Inside to DMZ traffic is ok
- Outside to Inside in not allowed
- DMZ to Inside is not allowed
- DMZ to Outside is allowed
- Outside to DMZ is not allowed

**Classifying Assets**

One reason to classify an asset is so that you can take **specific action**, based on policy with regard to assets in a given class. Consider, for example a VPN. We classify the traffic that should be sent over a VPN tunnel. By classifying data and labelling it, we can then focus that **appropriate amount** of **protection** or **security** on that data. More security for top secret data.

| | |
|---|---|
| Governmental Classifications: | • Unclassified<br>• Sensitive but unclassified<br>• Confidential<br>• Secret<br>• Top secret |
| Private sector classifications: | • Public<br>• Sensitive<br>• Private<br>• Confidential |
| Classification criteria: | • Value<br>• Age<br>• Replacement cost<br>• Useful lifetime |
| Classification roles: | • Owner – The group responsible for the data<br>• Custodian – The group responsible for implementing the policy as decided by the owner.<br>• User – Those who access the data and abide by the rules of acceptable use. |

**Traffic Light Protocol (TLP)**

TLP is a set of **designations** developed by the **US-CERT** division to ensure that sensitive information is shared with the **correct audience**.

**TLP Classification Levels**

| Colour | When should it be used? | How may is be shared? |
|---|---|---|
| RED | When information cannot be effectively acted upon by additional parties and could lead to impacts on party's privacy, reputation or operations if misused. | Recipients **may not share** TLP:RED information with any parties outside of the specific exchange, meeting or conversation in which it is originally disclosed. |
| AMBER | Sources may use TLP:AMBER when information requires support to be effectively acted upon, but carries risks to privacy, reputation or operations if shared outside of the organizations involved. | Recipients may only **share** TLP:AMBER information with **members** of their **own organization** who need to know and only as widely as necessary to act on that information. |
| GREEN | Sources may use TLP:GREEN when information is useful for the awareness of all participating organizations as well as with peers within he broader community or sector. | Recipients may **share** TLP:GREEN information with **peers** and **partner organisations** within their sector of community by not via publicly accessible channels. |
| WHITE | Sources may use TLP:WHITE when information carries minimal or no foreseeable risk of misuse in accordance with applicable rules and procedures for public release. | TLP:WHITE information may be **distributed without restriction**, subject to copyright controls. |

**Classifying Vulnerabilities**

Understanding the **weakness** and **vulnerabilities** in a system or network is a huge step toward correcting the vulnerability or putting in **appropriate countermeasures** to mitigate threats against those vulnerabilities. Potential network vulnerabilities abound with many resulting from one of the following:

- Policy flaws
- Design errors
- Protocol weaknesses
- Misconfiguration
- Software vulnerabilities
- Human factors
- Malicious software
- Hardware vulnerabilities
- Physical access to network resources

**Classifying Countermeasures**

A company may implement **various countermeasures** in order to **reduce** the **risk** of a successful **attack**. Common control methods used to implement countermeasures include the following.

**Administrative**: These consist of written policies, procedures, guidelines and standards.

**Physical**: Physical controls are exactly what they sound like, physical security for the network servers, equipment and infrastructure.

**Logical**: Logical controls include passwords, firewalls, intrusion prevention systems, access lists, and VPN tunnels and so on. These are often referred to as technical controls.

**Potential Attackers:**

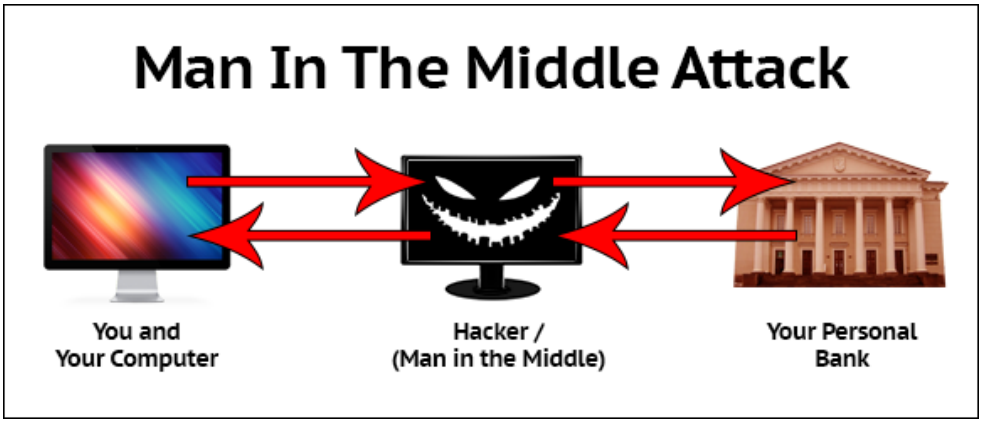
- Terrorists
- Criminals
- Government agencies
- Nation states
- Hackers
- Disgruntled employees
- Competitors
- Anyone with access to a computing device.

**Attacks Methods:**

| Action | Description |
|---|---|
| **Reconnaissance** | This is the **discovery process** used to find **information** about the network. It could include **scans** of the network to find out which **IP addresses** respond and further scans to see which **ports** on the devices at these IP address are **open**. Usually the first step taken to discover what is on the network and to determine **potential vulnerabilities**. |
| **Social engineering** | If an attacker can get to the user to reveal information, it is much easier for the attacker than using some other method of **reconnaissance**.<br>Can be done through **e-mail** or **misdirection** of web pages and also **in person** or **over the phone**.<br><br>• **Phishing**: Presents a link that looks like a **valid trusted** resource to a user but when the user clicks on it he us prompted to disclose confidential information.<br>• **Pharming**: Used to **direct** a customer's **URL** from a valid resource to a malicious one that could be made to applier as the valid site to the user. |
| **Privilege escalation** | The process of taking some level of **access** and achieving an even greater level of access. Example an attacker who gains user mode access to a router and then uses a brute force attack against the router, determining what the enable secret is for privilege level 15 access. |
| **Back doors** | When an attacker gain access to a system, then usually want future access, as well and they want it to be easy. A backdoor application can be installed to either **allow future access** or to **collect information** to use in further attacks. Many backdoors are installed by users clicking something without realizing the **link** they click or the **file** they open is a threat. |
| **Code execution** | When attackers can gain access to a device, they might be able to take several actions. The type of action depends on the level of access the attacker has or can achieve based on **permissions** granted to the account compromised by the attacker. |

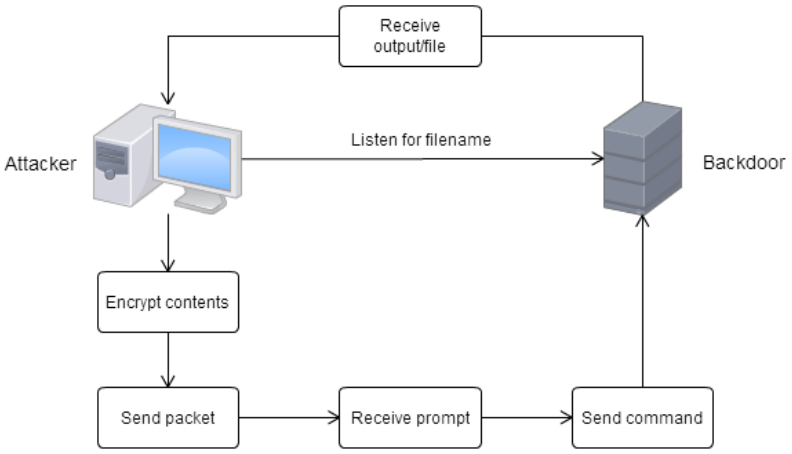
**Man in the Middle Attacks**

A **man in the middle** attack results when attackers place themselves in line **between two devices** that are communicating with the intent to perform **reconnaissance** or to **manipulate** the **data** as it moves between them. This can happen at **Layer 2 or Layer 3**. The main purpose is **eavesdropping** so the attacker can see all the **traffic**.



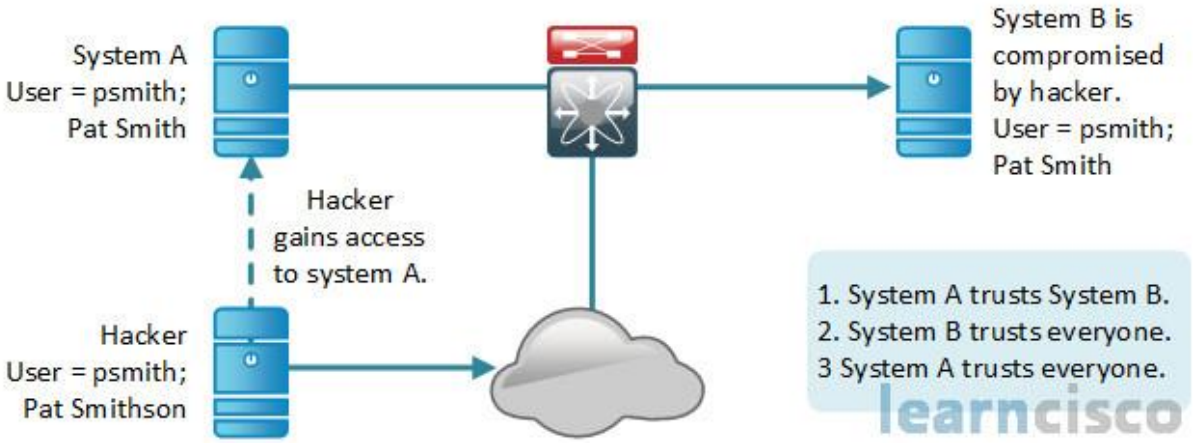

**Covert channel**

This method uses **programs** or **communications** in unintended ways. For example, if the security policy says that web traffic is allowed but peer to peer messaging is not, users can attempt to tunnel their peer to peer traffic **inside** of HTTP traffic. An attacker may use a similar technique to hide traffic by tunnelling it inside of some other allowed protocol to avoid detection. An example of the is a backdoor application collecting keystroke information from the workstation and then slowly sending is out disguised as ICMP, this is a covert channel.



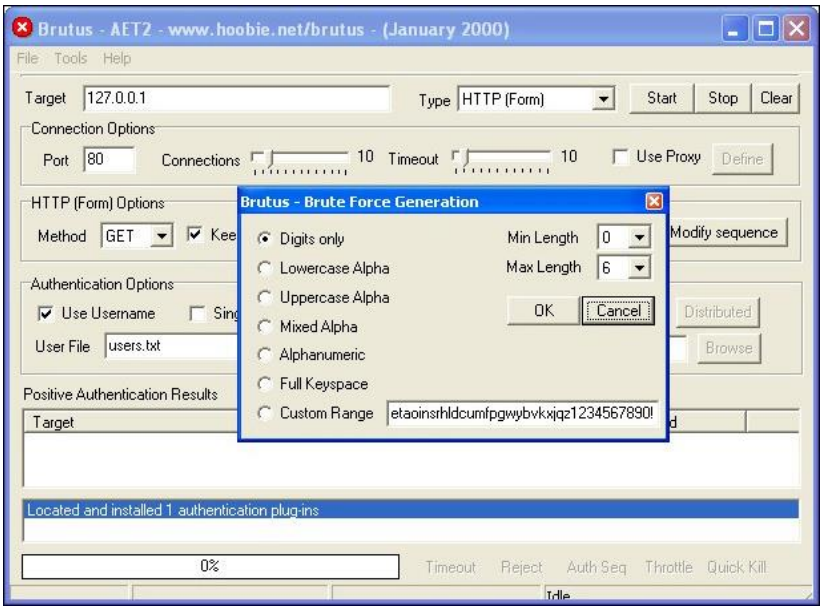
**Trust exploitation**

If the firewall has three interfaces and the outside interface allows all traffic to the demilitarized zone (DMZ) but not to the inside network and the DMZ allows access to the inside network from the DMZ, an attacker could leverage that by gaining access to the DMZ and using that location to launch his attacks from there to the inside network. Others trust models, if incorrectly configured may allow unintentional access to an attacker including active directory and NFS.



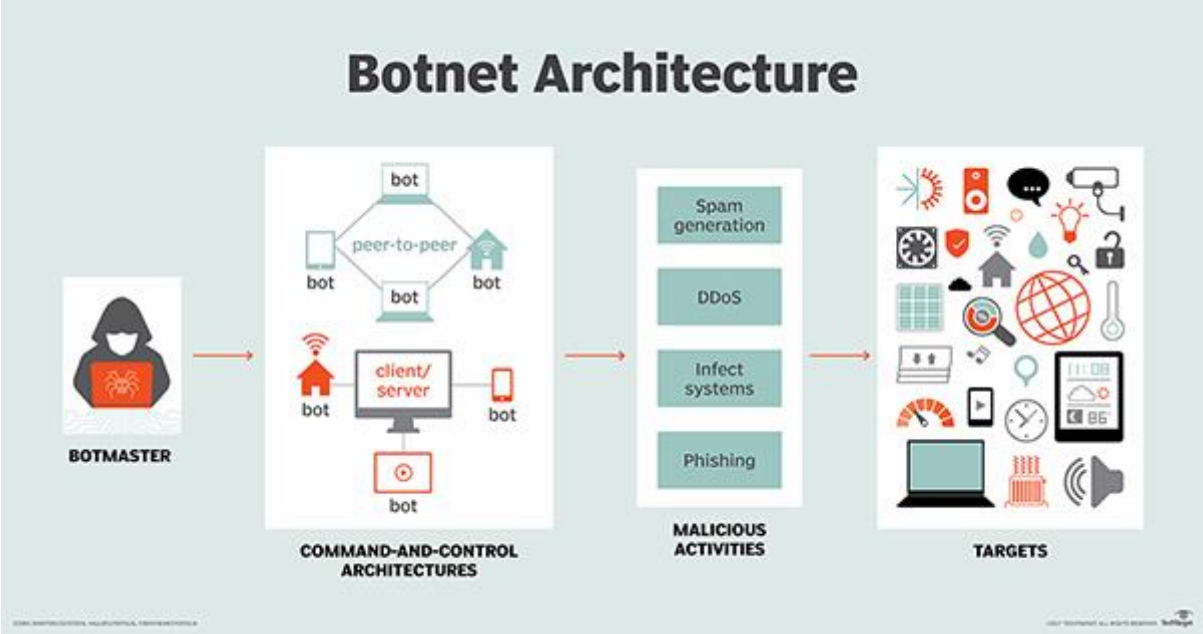

**Brute force attacks**

**Brute force** types of attacks are performed when an attacker's system attempts **thousands** of **possible passwords** looking for the **right match**. This is best protected against by specifying limits on how many unsuccessful authentication attempts can occur within specified time frame. They can be done through malware, man in the middle attacks by using sniffers or key loggers.



**Botnet**

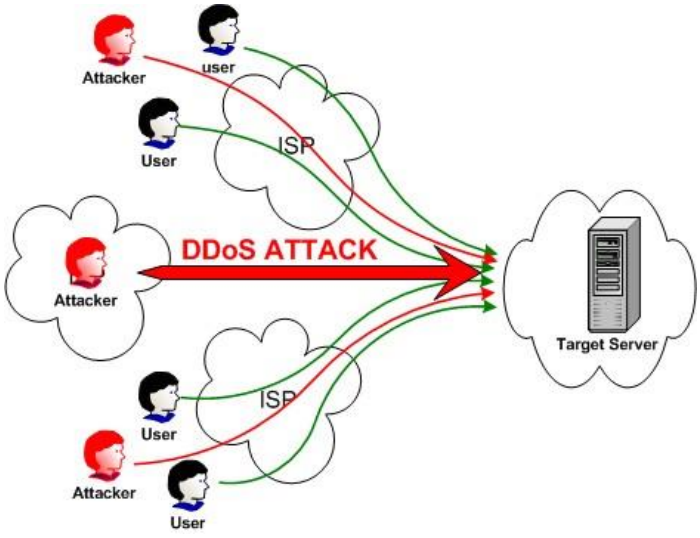
A **bot net** is a **collection** of **infected computers** that are ready to take instructions from the attacker. For example, if the attacker has the malicious backdoor software installed on 10000 computers from his central location, he could instruct those computers to all send **TCP SYN** requests or **ICMP echo** requests repeatedly to the **same destination**. To add insult to injury, he could also spoof the source IP address of the request so that reply traffic is sent to yet another victim. The attacker generally uses a covert channel to manage the individual device that make up the botnet.




**Denial of service attack and distributed denial of service attacks (DoS and DDOS attacks)**

An example is using a botnet to attack a target system. If an attack is launched from a single device with the intent to cause damage to an asset, the attack could be considered a DoS attempt rather than a DDoS attack. Both attacks want the same result and whether it is called DoS or DDoS attack just depends on how many source machines are used in the attack.

A more advanced and increasingly popular type of DDoS attack is called a reflected DDoS attack (RDDoS). An RDDoS takes place when the source of the initial packets is actually spoofed by the attacker. The response packets are then reflected back from the unknowing participant to the victim of the attack that is the original (spoofed) source of the initial packets.



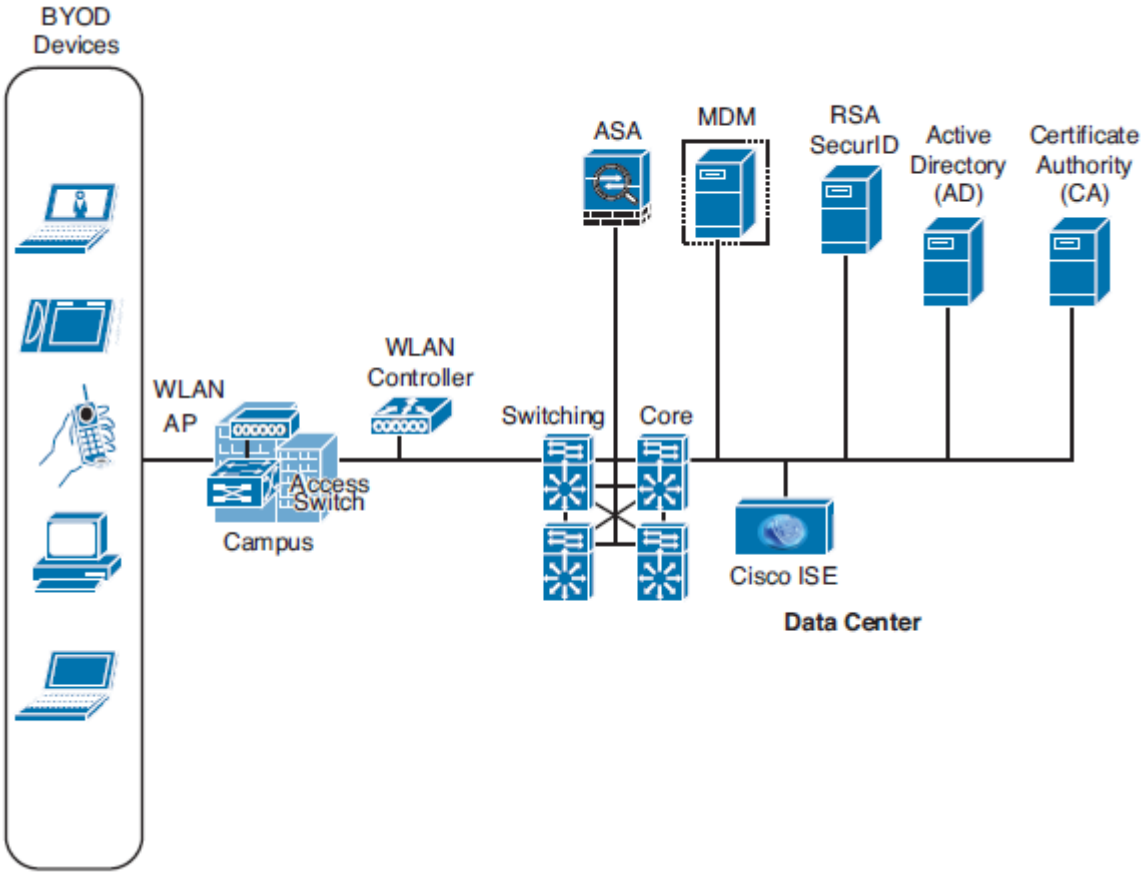
**Guidelines for Secure Network Architecture**

| Guideline | Explanation |
|---|---|
| **Rule of least privilege** | The rule states that **minimal access** is **only provided** to the **requited network resources**, and **not more** than that. An Example of this is an access list applied to an interface for filtering that says deny all. Before this, specific entries could be added allowing only the bare minimum of required protocols and only then between the correct source and destination address. |
| **Defence in depth** | This concept suggests that you have **security implemented** on **nearly every point** of your **network**. An example is filtering at a perimeter router, filtering again at a firewall using IPSs to analyse traffic before it reaches your servers as well. Additional methods that can be used to implement a defence in depth approach include using authentication and authorization mechanisms, web and e-mail security, content security, application inspection monitoring traffic monitoring and malware protection.<br><br>**The concept** behind defence in depth is that if a **single security technology fails**. **Additional levels** or mechanisms or security are **still in place** to **protect** the data applications and devices on the network. |
| **Separation of duties** | When you place specific individuals into specific roles, there can be checks and balances in place regarding the implementation of the security policy. Rotating individuals into different roles periodically will also assist in verifying that vulnerabilities are being addressed, because a person who moves into a new role will be required to review the polices in place. |
| **Auditing** | This refers to **accounting** and keeping **records** about what is occulting on the network. Most of this can be automated through the features of **authentication, authorization and accounting (AAA).** |

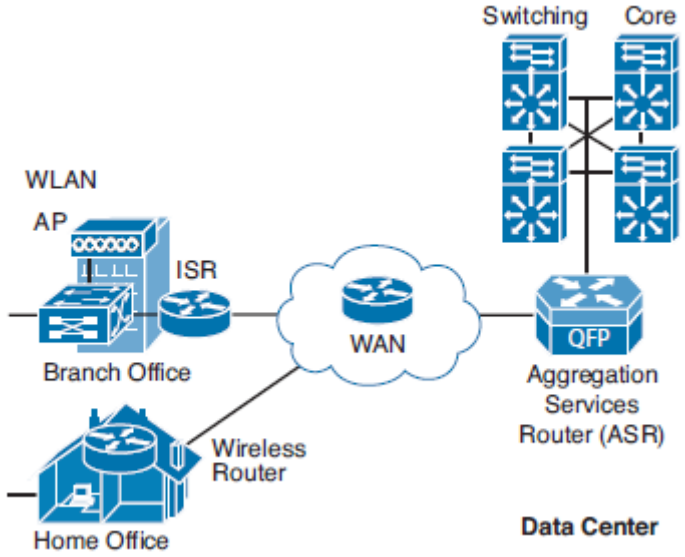
**Network Topologies**

There exist a number of network topologies that depend on the size and type of each organization. Some organizations will have a presence of each of the following topologies while others may only utilize a subset of this list

- **Campus-Area Network (CAN):** A campus-area network is the network topology used to provide connectivity, data, applications, and services to users of an organization that are physically located at the corporate office (headquarters). The CAN includes a module for each building in the campus, for the data center, for WAN Aggregation, and for the Internet Edge. Security with the Campus Area Network.

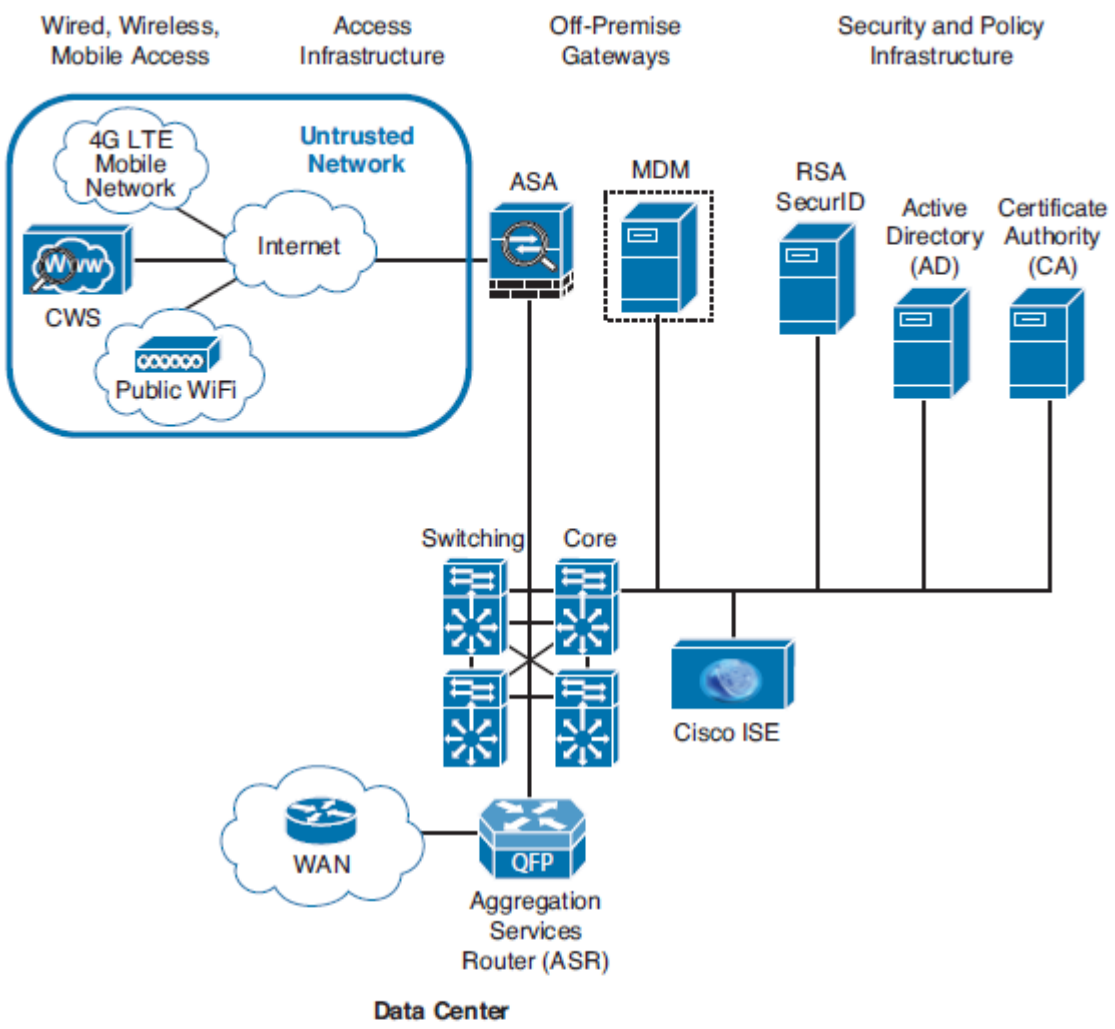


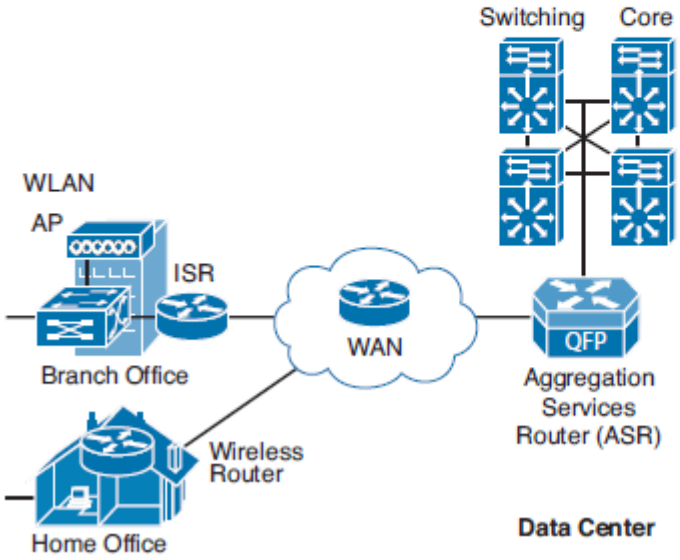
**Cloud, Wide-Area Network (WAN):** The cloud and WAN provide a logical and physical location for data and applications that an organization prefers to have moved off-site. This alleviates an organization from having to expend resources to operate, maintain, and manage the services that have been previously located within the organization's purview.




- **Data Center:** The Data Center network contains the Unified Computing System (UCS) servers, voice gateways, and CUCM servers supporting the VoIP environment, all of which is provided network connectivity by a series of Nexus switches. The entire Data Center network is protected by a set of firewalls at the edge that filters all traffic ingressing and egressing the Data Center.

- **Small office/Home office (SOHO):** The remote SOHO site will provide connectivity to the SOHO users through the use of WAN routers that find their way back to the WAN Aggregation module in the CAN via MPLS WANs. Within the SOHO, users are provided network connectivity through the presence of access switches.

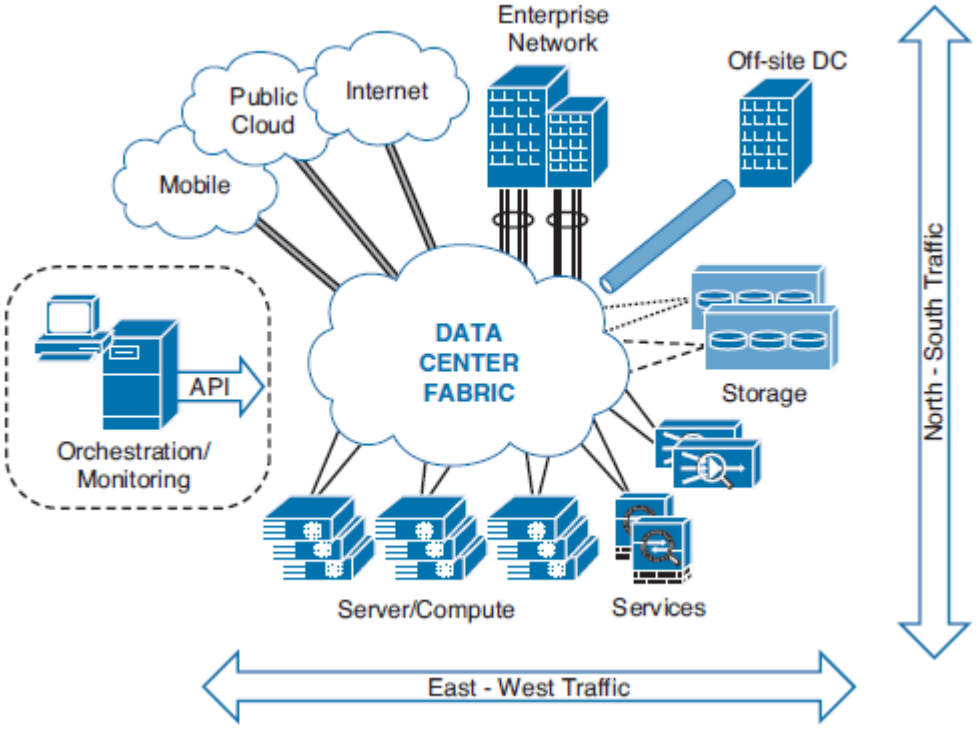

Wired, Wireless, Mobile Access
Access Infrastructure
Off-Premise Gateways
Security and Policy Infrastructure
4G LTE Mobile Network
Untrusted Network
Internet
CWS
Public WiFi
ASA
MDM
RSA SecurID
Active Directory (AD)
Certificate Authority (CA)
Switching
Core
Cisco ISE
WAN
QFP
Aggregation Services Router (ASR)
Data Center


Switching
Core
WLAN AP
ISR
Branch Office
WAN
QFP
Aggregation Services Router (ASR)
Wireless Router
Home Office
Data Center

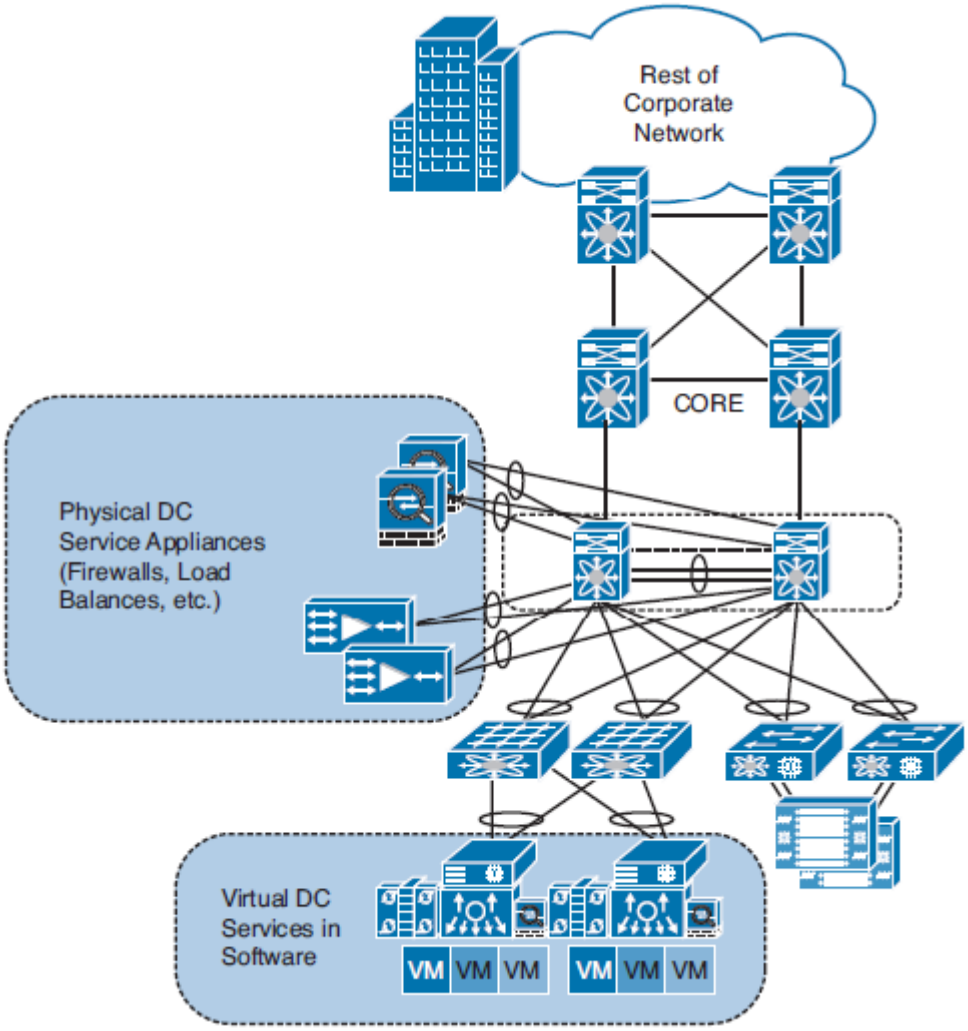
**Network Security for a Virtual Environment**

Today's data center environments must be designed to significantly reduce administrative overhead and improve flexibility and operational efficiency. Critical security functions must be able to dynamically scale to protect assets as business demands change. Cisco has created technologies and products such as the Application Centric Infrastructure (ACI) ecosystem and the Cisco ASAv (virtual ASA) to provide security solutions for today's data center demands.

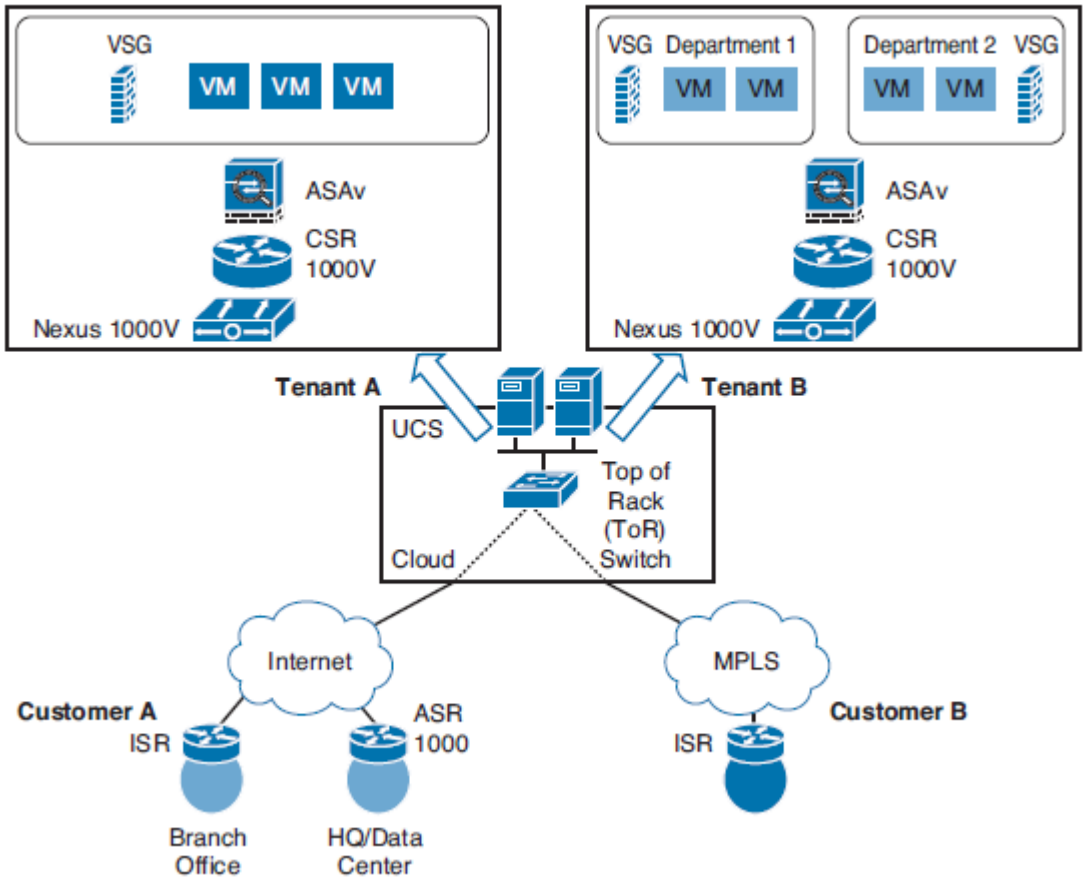
For example, ACI provides a centralized application-level policy engine for physical, virtual, and cloud infrastructures. The Cisco ASAv provides detailed visibility and control of application and services within the virtual environment. The picture below illustrates a high-level data center environment with multiple network connections, and it defines the concept of east-west versus north-south traffic.



The picture below shows a virtualized data center where multiple software applications (such as VMWare, KVM, Xen) are used to divide one physical server into multiple isolated virtual environments. In this example physical firewalls are deployed to provide protection and segmentation to the data center from the rest of the corporate network.



The challenge of using physical firewalls and other security appliances in a virtualized environment is that sometimes the traffic does not leave the physical server (often referred to as bare metal). Subsequently, a virtual security solution is needed.The picture below demonstrates how a security administrator can provide detailed visibility and control of application and services within the virtual environment by deploying the Cisco ASAv.



# CHAPTER 2: Common Security Threats

**Network Security Threat Landscape**

**Financial**: They can compromise a **point of sale (PoS)** system at a retail organisation and siphon off millions of **credit/debit cards** which can subsequently be sold on the **online black market**. Threat actors can also penetrate financial organizations for the sole purples of compromising **user accounts** and transferring **money** to accounts of their choosing.

**Disruption**: Unfortunately, many individuals and groups exist solely to cause disruption to the core business of many organizations and institutions. This disruption is created for several reasons:

- To protest the actions, decisions or behaviours of an enterprise.
- To serve as a distraction while the malicious actors plant something within the network to be leveraged at a future point in time.
- To gain media attention for the action of the malicious group or individual.

**Geopolitical**: Groups that engage in **cyber warfare** and launch **attacks** against countries who they believe do not have their best interests.

**Denial of service (DoS) and Distributed DoS (DDoS) attacks**

**Direct**: Direct DDoS attacks occur when the source of the attack generates the packets regardless of protocol, application and so on, that are sent directly to the victim of the attack.

**Reflected**: Reflected DDoS attacks occur when the sources of the attack are send spoofed packets that appear to be from the victim and then the sources become unwitting participants in the DDoS attack by sending response traffic back to the intended victim. **UDP** is often used as the transport mechanism because it is more easily spoofed due to the lack of a three way handshake.

For example, if the attacker (A) decides he wants to attack a victim (V), he will send packets (for example, *Network Time Protocol [NTP]* requests) to a source (S) who thinks these packets are legitimate. The source (S) then responds to the NTP requests by sending the responses to the victim (V), who was never expecting these NTP packets from source (S).

**Amplification**: Amplification attacks are a form of reflected attacks in which the response traffic (sent by the unwitting participants) is made up of packets that are much larger than those that were initially send by the attacker (spoofing the victim). An example of this is when DNS queries are send and the DNS responses are much larger in packet size than the initial query packets. The end result is that the victim gets flooded by larger packets for which it never actually issued queries.

**Social Engineering Tactics**

**Phishing**: Phishing clients secure information through an e-mail message that appears to come from a legitimate source such as a service provider or financial institution. The e-mail message may ask the user to reply with the sensitive data or to access a website to update information such as a bank account number.

**Malvertising**: This is the act of incorporating malicious ads on trusted websites, which results in users' browsers being inadvertently redirected to sites hosting malware.

**Phone scams**: It is not uncommon for someone to call up an employee and attempts to convince employees to divulge information about themselves or others within the organisation. An example is a miscreant posing as a recruiter asking for names, e-mail addresses and so on for members of the organization and then using that information to start building a database to leverage for a future attack, reconnaissance mission and so forth.

**Defences against Social Engineering**

**Password management:** Guidelines such as the number and type of characters that each password must include how often a password must be changed and even a simple declaration that employees should not disclose passwords to anyone will help secure information assets.

**Two-factor Authentication:** Authentication for high-risk network services such as modem pools and VPNs should use two-factor authentication rather than fixed passwords.

**Antivirus/antiphishing defences:** Multiple layers of antivirus defences such as at mail gateways and end user desktops can minimise the threat of phishing and other social engineering attacks.

**Change management:** A documented change-management process is more secure than an ad hoc process, which is more easily exploited by an attacker who claims to be in a crisis.

**Information classification:** A classification policy should clearly describe what information is considered sensitive and how to label and handle it.

**Document handling and destruction:** Sensitive documents and media must be securely disposed of and not simply thrown out with the regular office trash.

**Physical security:** The organization should have effective physical security controls such as visitor logs, escort requirements and background checks.

**Malware Identification Tool**

Several factors make this identification particularly difficult:

The sheer **amount** of **malware** that exists and is created on a daily basis is almost incomprehensible. The creation of new malware often results in the rendering useless of signature-based detection tools.

Malware is often **embedded** in otherwise-**trusted applications** and sent over **protocols** that are traditionally allowed through firewalls and access lists.

Organizations have **limited resources** to keep up with the massive amounts of traffic that traverse the network. The volume of network traffic, both good and bad has become so large that it is almost too much for any one organization to keep up.

The increasing use of encryption has not surprisingly added another layer of complexity for organizations trying to gain visibility into malicious traffic residing on the network.

**Methods Available for Malware Identification**

**Packet captures:** Collecting, storing and analysing the raw packets that are traversing the network is certainly one way of inspecting traffic for the presence of malware. Although packer captures provide the most granular look into that traffic that is on the network, one primary hurdle in the use of packet capture for malware identification is that face that you are looking for the proverbial “needle in a haystack” due to the volume of data generated by packet captures.

**Snort:** Packet capture is often referred to as **micro**-**analytical** in terms of the granularity of data being analysed. NetFlow data is considered more of a **macro**-**analytical** approach. The use of **NetFlow data collection** consist of the creation of **bucker** or **flows** of data that are based on a set of **predefined parameters** such as source IP address, source port, destination IP address, Destination port, IP protocol, ingress interface and type of service (ToS). **Each time** one of these **parameters differs**, a **new flow** is **created**. Flows are stored **locally** on the **device** for a configured time interval, after which time the flows are exported to **external collectors**.

**IPS events:** When using **IPS devices** on your network, it is possible to leverage the **alarms triggered** on the IPS device as an **emergency** flare that network traffic should be further analysed for the presence of malware. Often IPS devices have signatures for specific strains of malware, which when triggered can be an indication that malicious traffic exists on the network.

**Advanced Malware Protection:** Cisco Advanced Malware Protection (AMP) is designed for Cisco Firepower network security appliances. It provides **visibility** and **control** to **protect** against **highly sophisticated**, targeted, zero-day and persistent advanced malware threats. AMP helps to identify inconspicuous attacks by continuously analysing and monitoring files after they’ve entered the network, utilizing retrospective security alerts to help administrators take action during and after an attack and provides multi-source indications of compromise to aid in the correlation of discrete events for better detection.

**NGIPS:** The Cisco Firepower next-generation intrusion prevention system (**NGIPS**) solution provides **multiple layers** of **advanced threat protection** at high inspection throughput rate. The NGIPS threat protection solution is **centrally managed** through the **Cisco FireSIGHT Management Centre** and can be expanded to include additional features such as AMP, application visibility and control and URL filtering.

**Several types of data are particularly attractive to the miscreants of the cyber (under) world:**

**Intellectual property (IP):** This consists of any type of data or documentation that is the property of an organization and has been created or produced by employees of the organization. IP often refers to the designs, drawings and documents that support the development, sale and support of an organization product.

**Personally identifiable information (PII):** This is the type of information that has, unfortunately been talked about in the press all too often lately when we hear about data breaches. This information includes names, dates of birth, addresses and Social Security numbers (SSN).

**Credit/debit cards:** In addition to PII, which is often stolen or compromised during data breaches, credit and debit card information is extremely desired by malicious actors.

# Chapter 3: Implementing AAA in Cisco IOS

**Four Ways to Implement AAA**

- Self-contained
- Cisco Secure ACS for Windows
- Cisco Secure ACS Appliance
- Cisco ISE

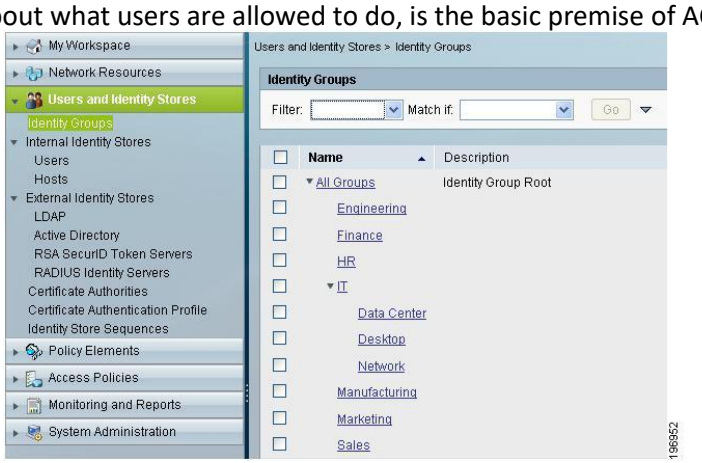
**Why use Cisco ACS?**

By configuring **users** locally on the **ACS server**, and then having the dozens or hundreds of routers and switches act as clients to the ACS server, you can use the Cisco ACS server as a central clearinghouse for the authentication of users. This way, you can create a **user account** one time on the ACS server, and configure the routers and switches to use the ACS server for any type of user, whether an administrator trying to access the router for configuration or an end user who just needs access through a router for some network application or service such as browsing the web. If all your network devices use the ACS server, you can avoid having to create that same user account on each of the individual routers' and switches' local database (in their **running** config).

1. A user connects to a router, and the router prompts the user for **authentication**. In this example, assume it is an administrator who wants CLI access to the router.
2. The router being configured to use the ACS server prompts the user for his **username** and **password**. After getting the username and password, the router sends those **credentials** to the **AAA server** (in this case, the ACS server) and waits for a reply.
3. At the ACS server, if it is configured to use an **external database** such as Microsoft Active Directory, the ACS server makes an inquiry out to Active Directory to validate whether the username and password that the user provided are accurate. If they are, Active Directory can indicate that to the ACS server, and the ACS server in turn can indicate that the credentials are correct back to the router, and then the router can provide the access to the user.
4. If there were no Active Directory, the ACS server would consult its own **local configuration** to verify the username and password instead of handing it off to Active Directory.

- Extends access security by combining authentication, user access, and administrator access with policy control
- Allows greater flexibility and mobility, increased security, and user-productivity gains
- Enforces a uniform security policy for all users
- Reduces the administrative and management efforts

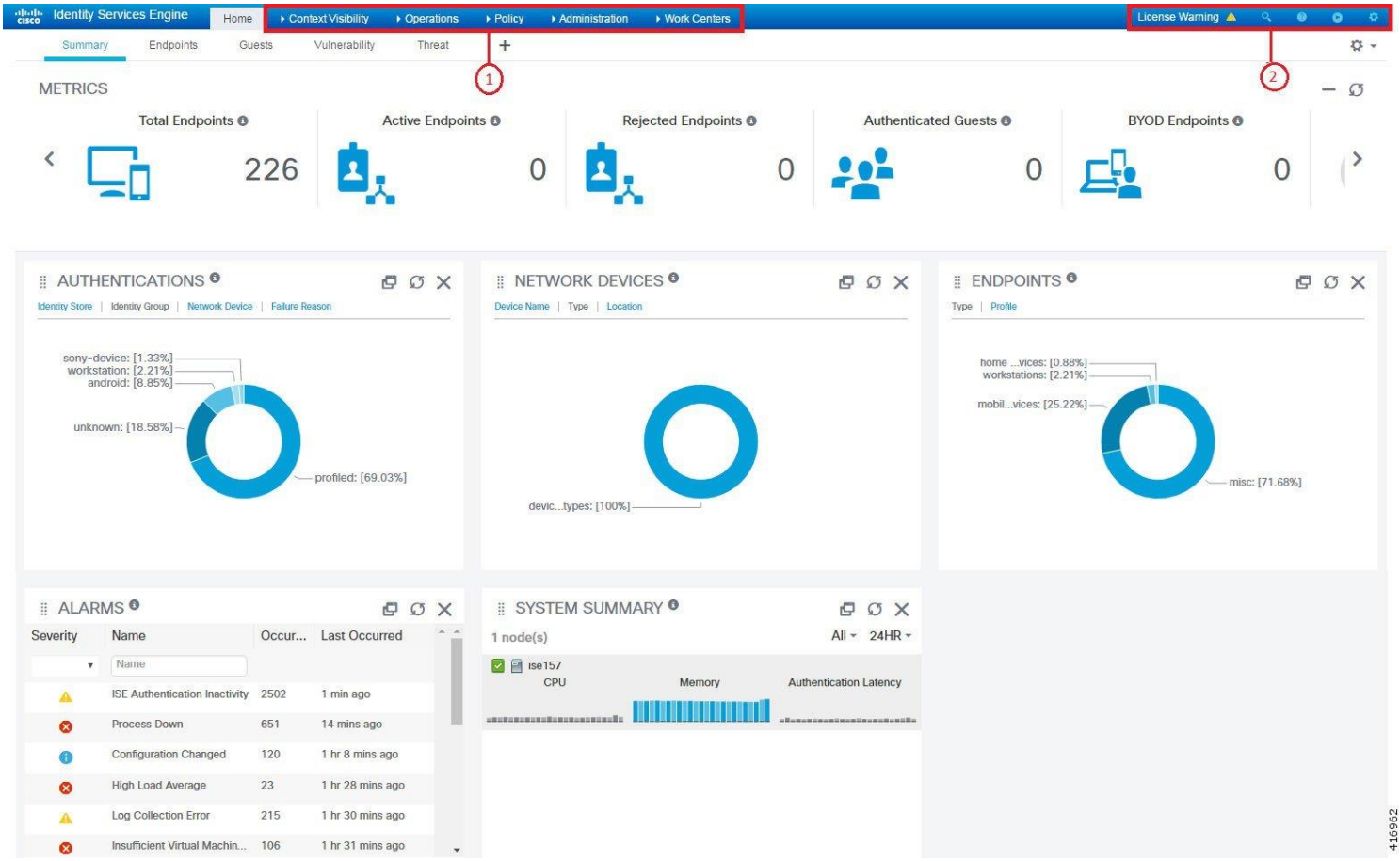
The core functionality of having a **centralized database** of **users**, along with **authorization rules** about what users are allowed to do, is the basic premise of ACS.



**What is ISE?**

**Identity Services Engine (ISE)** is an identity and access control policy platform that can validate that computer meets the requirements of a company's policy related to virus definition files, service pack levels and so on before allowing the device on the network.

This solution leverages many AAA-like features but does not replace ACS.



**Protocols Used Between the ACS and the Router**

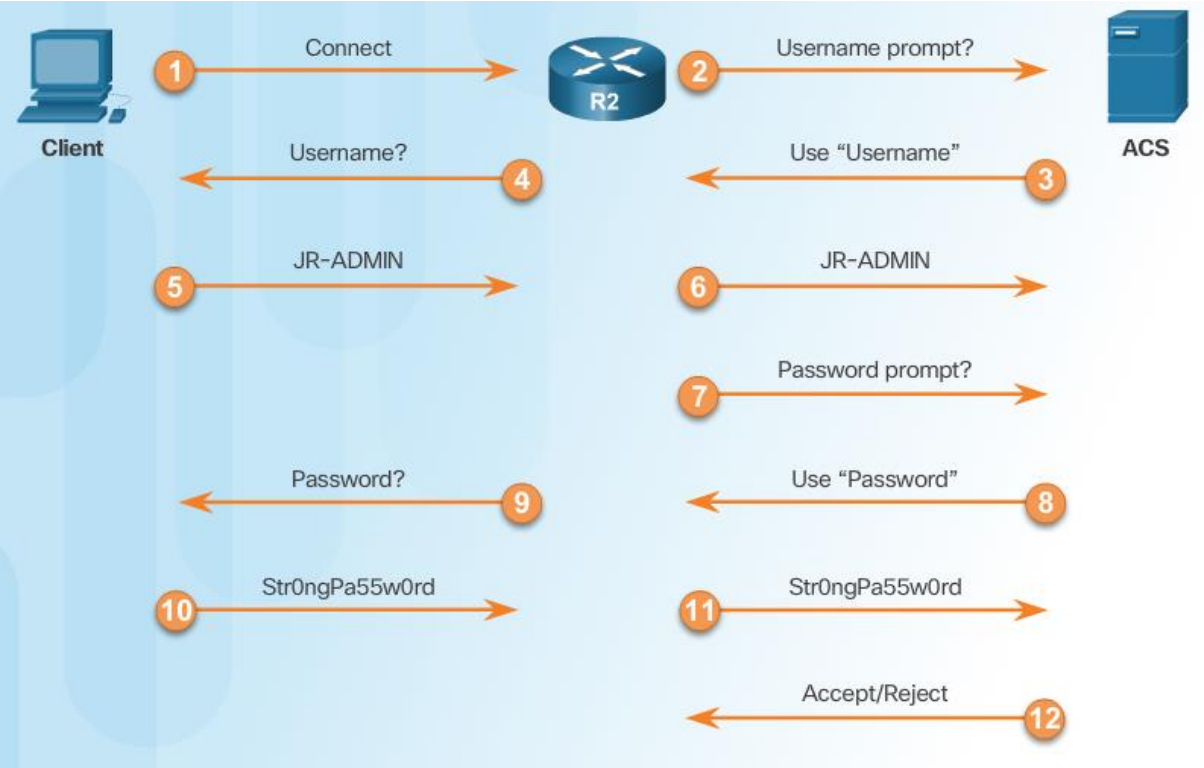
Two main protocols may be used between the **ACS server** and its **clients**:

- **TACACS+**
- **RADIUS**

**TACACS+**

**Terminal Access Control Access Control Servers** is Cisco proprietary. If you configure the router and the ACS server to use TACACS+, all the AAA packets that are sent between the router and the ACS server use the TACACS+ protocol, which encrypts each packet before it is sent on the network.

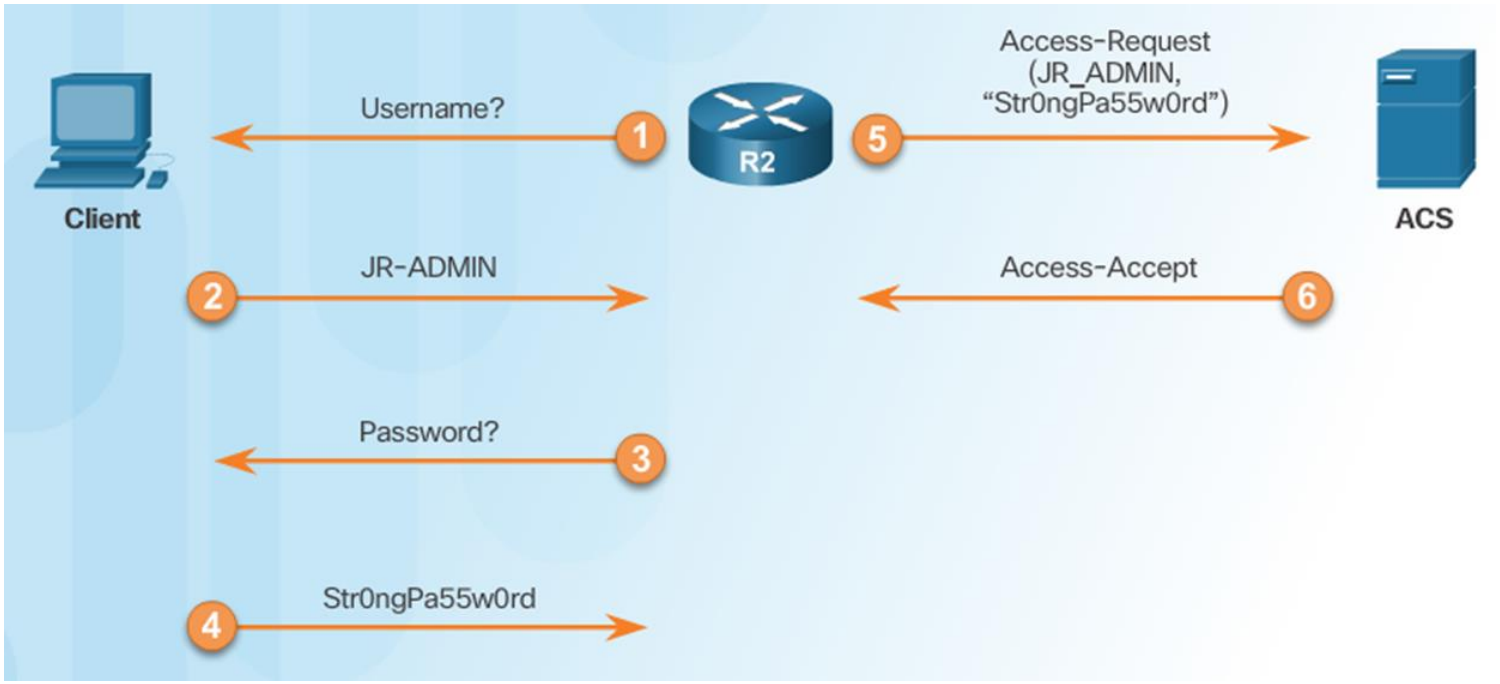
Traditionally , and in common practice, if you are authenticating and authorizing administrators for command-line access, it is likely that you will configure TACACS+ on both the ACS server and the router for their communication with each other. A large reason for this is because TACACS+ has clearly defined and separate techniques and configurations for each aspect of AAA. For example, if you want to tell the router to check authorization for each individual command before allowing an administrator to put that command in, and only give the administrator a subset or portion of commands, TACACS+ and its authorization component allows extremely granular control in communicating which commands would be allowed. RADIUS, however, does not have the same level of granular control as TACACS+ command-by-command authorization.



**RADIUS**

**Remote Authentication Dial-In User Service** is an open standard protocol. It encrypts only the passwords.

If you are authenticating and authorizing end users who just want their packets to go through a network device (when authentication and authorization are required), it is likely that you are using RADIUS as the communications method between the ACS server on the router. You may configure the router and ACS server to use both TACACS+ and RADIUS simultaneously between the ACS server and its client, the router.




| | TACACS+ | RADIUS |
|---|---|---|
| **Functionality** | **Separates** AAA functions into distinct elements, Authentication is separate from authorization and both of those are separate from accounting | **Combines** many of the functions of authentication and authorization together. Has detailed accounting capability when accounting is configured for use. |
| **Standard** | Cisco Proprietary | Open standard |
| **L4 Protocol** | TCP | UDP |
| **Confidentiality** | **All packets** are **encrypted** between the ACS server and the router(which is the client) | **Only password** is **encrypted** with regard to packets sent back and forth between the ACS server and the router |
| **Granular command by command authorization** | This is supported and the rules are defined on the ACS server about which commands are allowed or disallowed | No explicit command authorization checking rules can be implemented |
| **Accounting** | Provides accounting support | Provide accounting support and generally acknowledged as providing more detailed or extensive accounting capability than TACACS+ |

**Configuring Routers to Interoperate with an ACS Server**

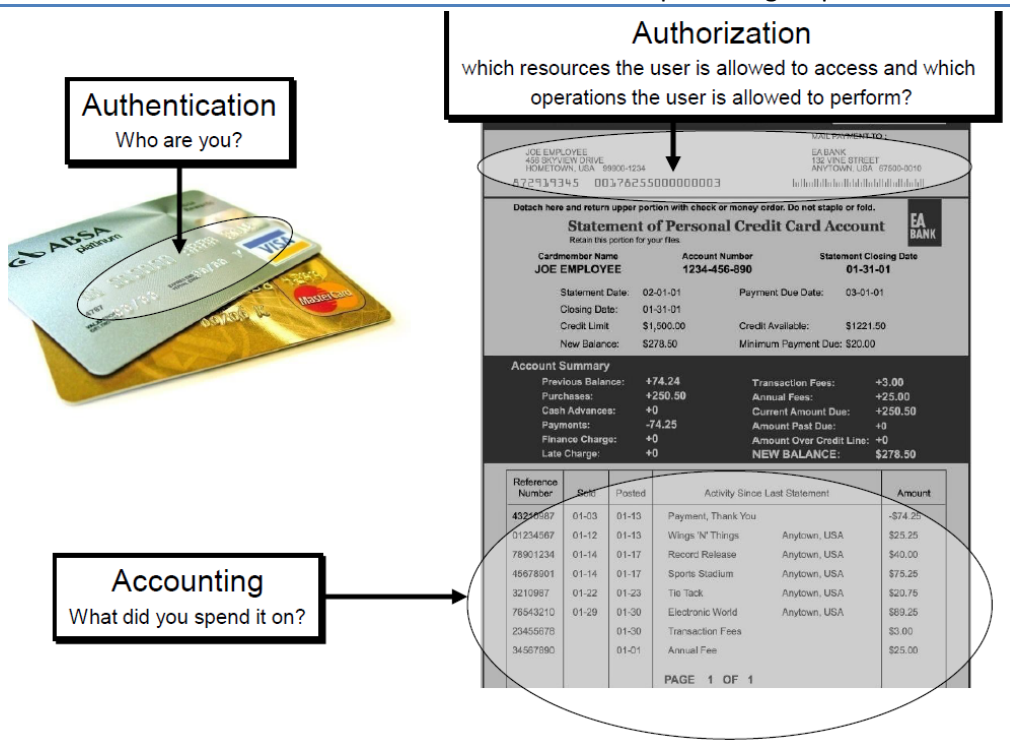
We want the router to implement the following:
■ For administrators/users who are accessing the router via the vty lines, regardless of whether they are using Telnet or *Secure Shell (SSH)*, the router should check with a TACACS+ server (the ACS server using TACACS+ to communicate with this router) for the authentication check (username/password).

■ Authenticated users need to be authorized to have access to a *command-line interface (CLI)* (EXEC) session, including the privilege level they should be placed into. The authorization check should be done by the router referring to the ACS server, using TACACS+.

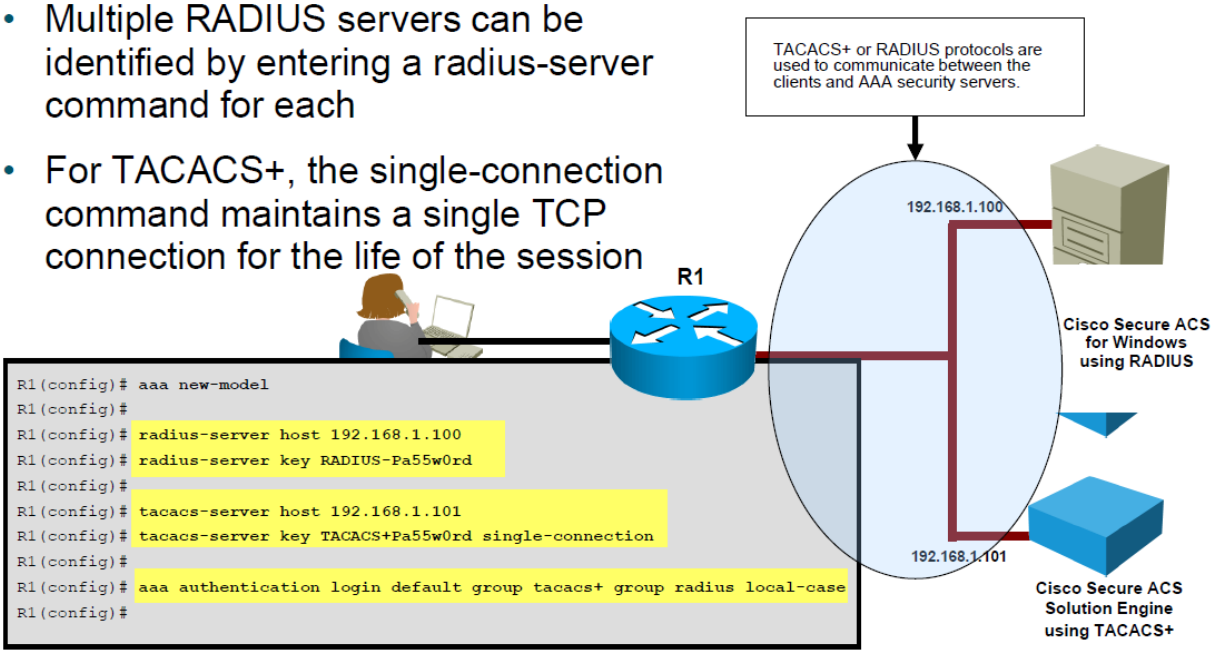
```
aaa new –model
aaa authentication login AUTHEN_via_TACACS group tacacs+ local
aaa authorization exec Author-Exec_via_TACACS group tacacs+ local
username admin privilege 15 secret cisco
tacacs-server host 192.168.1.252 key cisco123

line vty 0 15
authorization exec Author-Exec_via_TACACS
login authentication AUTHEN_via_TACACS
```

**Configuring the Router to use ACS via TACACS+**

| Task | How to Do It |
|---|---|
| Decide what the **policy** should be.<br>Ex. Which vty lines should require authentication/authorization and which methods should be used | This step is done before you configure the router and is based on your security policy for the network. |
| Enable the ability to **configure** AAA | **aaa new-model** is not enabled by default. |
| Specify the address of an ACS server to use | **Tacacs-server host** including the IP address of the ACS server and the password. |
| Create a named method list for authentication and another for authorization based of your policy | Each method list is created in global configuration mode, specifying which methods this list uses, in order from left to right. |
| Apply the method lists to the location that should use those methods. | In the vty line configuration mode, specify the authentication authorization method lists that you created in the preceding step. |



- Multiple RADIUS servers can be identified by entering a radius-server command for each
- For TACACS+, the single-connection command maintains a single TCP connection for the life of the session




```
R1# debug aaa authentication
AAA Authentication debugging is on
R1#
14:01:17: AAA/AUTHEN (567936829): Method=TACACS+
14:01:17: TAC+: send AUTHEN/CONT packet
14:01:17: TAC+ (567936829): received authen response status = PASS
14:01:17: AAA/AUTHEN (567936829): status = PASS
```

- The debug aaa authentication command provides a view of login activity
- For successful TACACS+ login attempts, a status message of PASS results

| Command | Description |
| --- | --- |
| **aaa new-model** | Enable the configuration of method lists and other AAA-related elements, including the use of ACS. |
| **test aaa group tacacs+** *admin cisco123* **legacy** | Allow verification of the authentication function working between the AAA client (the router) and the ACS server (the AAA server). |
| **aaa authentication login** *MYLIST1* **group tacacs+ local** | Create an authentication method list that, when applied elsewhere in the configuration, requests the services of an ACS server via TACACS+, and if no server responds, the next method local (which is the local router configuration) is checked to verify the credentials of the user. |
| **aaa authorization exec** *MYLIST2* **group tacacs+ none** | Create an authorization method list that, when applied to a vty line, requests the services of an ACS server (via TACACS+). If no server responds, the second method "none" is used. This results in no username prompt being provided to the user, and authentication is not required. |
| **tacacs-server host** *192.168.1.252* **key** *cisco123* | Places a server into the group of ACS servers the router can use for TACACS+ requests. It includes the IP address and the secret used to encrypt packets between this router (the client) and the ACS server. |

**Configuring the ACS Server to Interoperate with a Router**

| Component of ACS | How It Is Used |
|---|---|
| Network device groups | Groups of network devices, normally based on routers or switches with similar functions/devices managed by the same admins. |
| Network devices (ACS clients/routers/switches) | The individual network devices that go into the device groups |
| Identity groups (user/admin groups) | Groups of administrators, normally based on users who will need similar rights and access to specific groups of network devices. |
| User accounts | Individual administrator/user accounts that are placed in identity groups. |
| Authorization profiles | These profiles control what rights are permitted. The profile is associated with a network device groups and a user/administrator identify groups |

*Testing AAA Between the Router and the ACS*

```
R1# test aaa group tacacs+ admin cisco123 legacy
Attempting authentication test to server-group tacacs+ using tacacs+
User was successfully authenticated.
```

# CHAPTER 4: Bring Your Own Device

**BYOD** brings some **challenges** such as providing **seamless connectivity** for users bringing their own network-connected devices while also maintaining an appropriate **security** posture. The organization must provide a level of security that meets the organization's security policies and ensures that network devices, systems and data do **not get compromised** through the proliferation of vulnerable devices starting with the devices brought in by employees.

**Reasons to Implement BYOD**

- Wide variety of consumer devices
- Blurred lines between work and play
- Connect me anytime, anywhere

**BYOD Architecture Framework**

The **Cisco BYOD** solution architecture leverages the **Cisco Borderless Network Architecture** and is based on the assumption that best common practices (BCP) are followed in network designs for campus, branch offices, Internet edge and home office implementations.

**High-Level BYOD Solution Architecture**

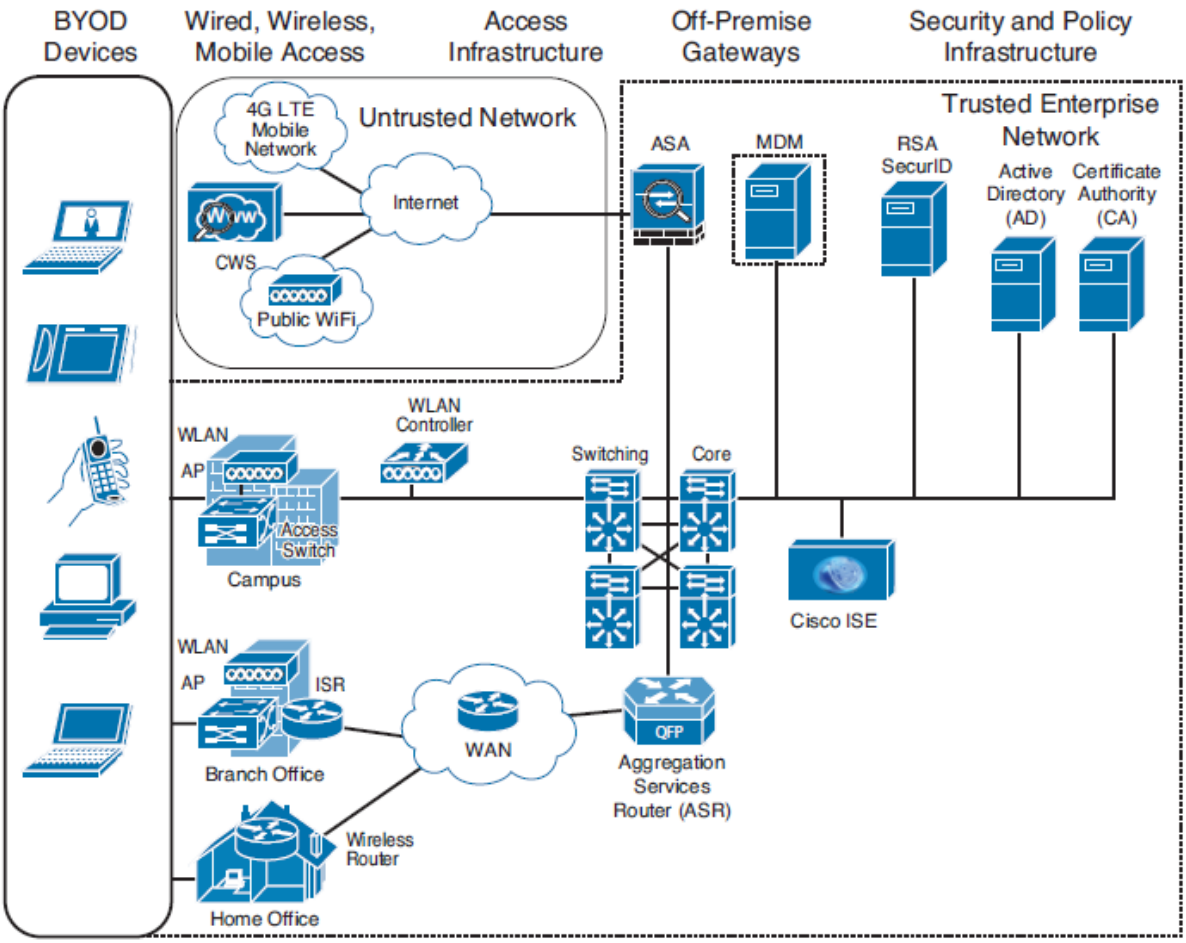


*Figure 3-1* *High-Level BYOD Solution Architecture—Campus View*

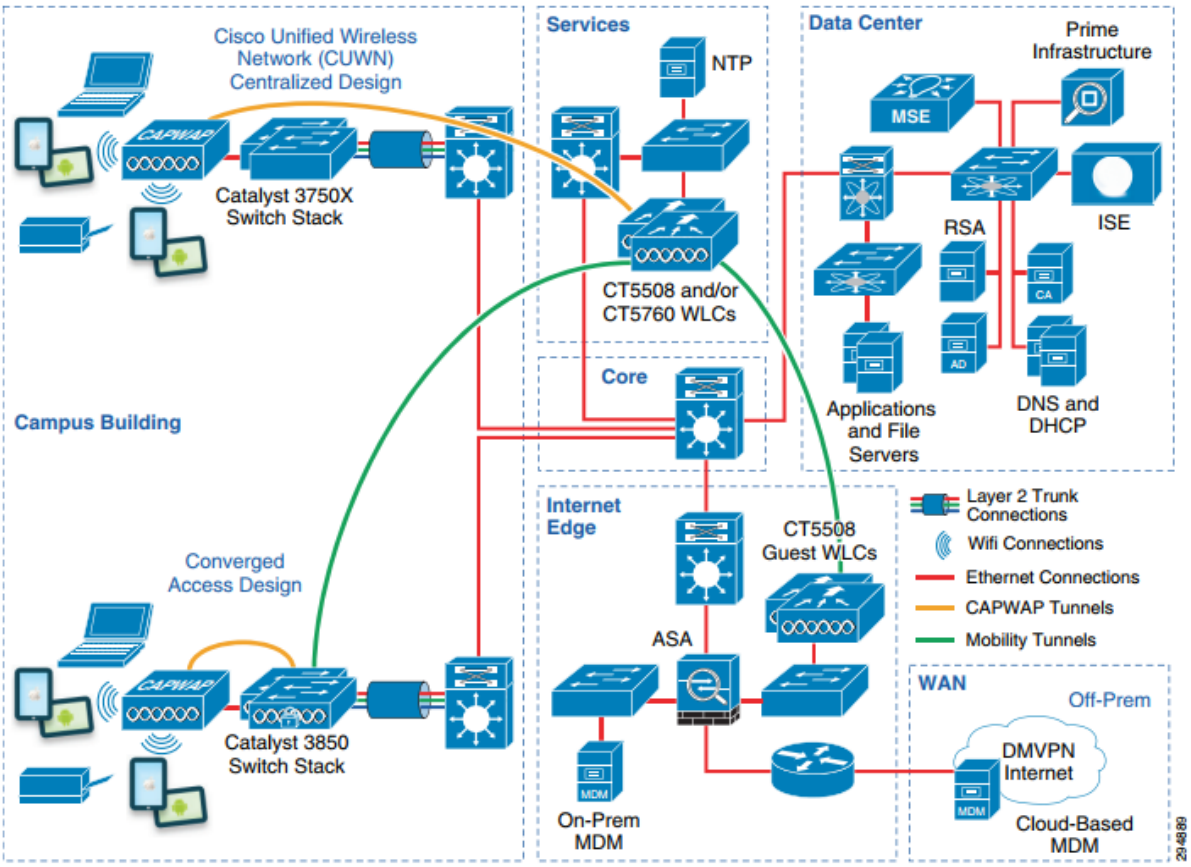



*Figure 3-2* *High-Level BYOD Branch Solution Architecture—Branch View*

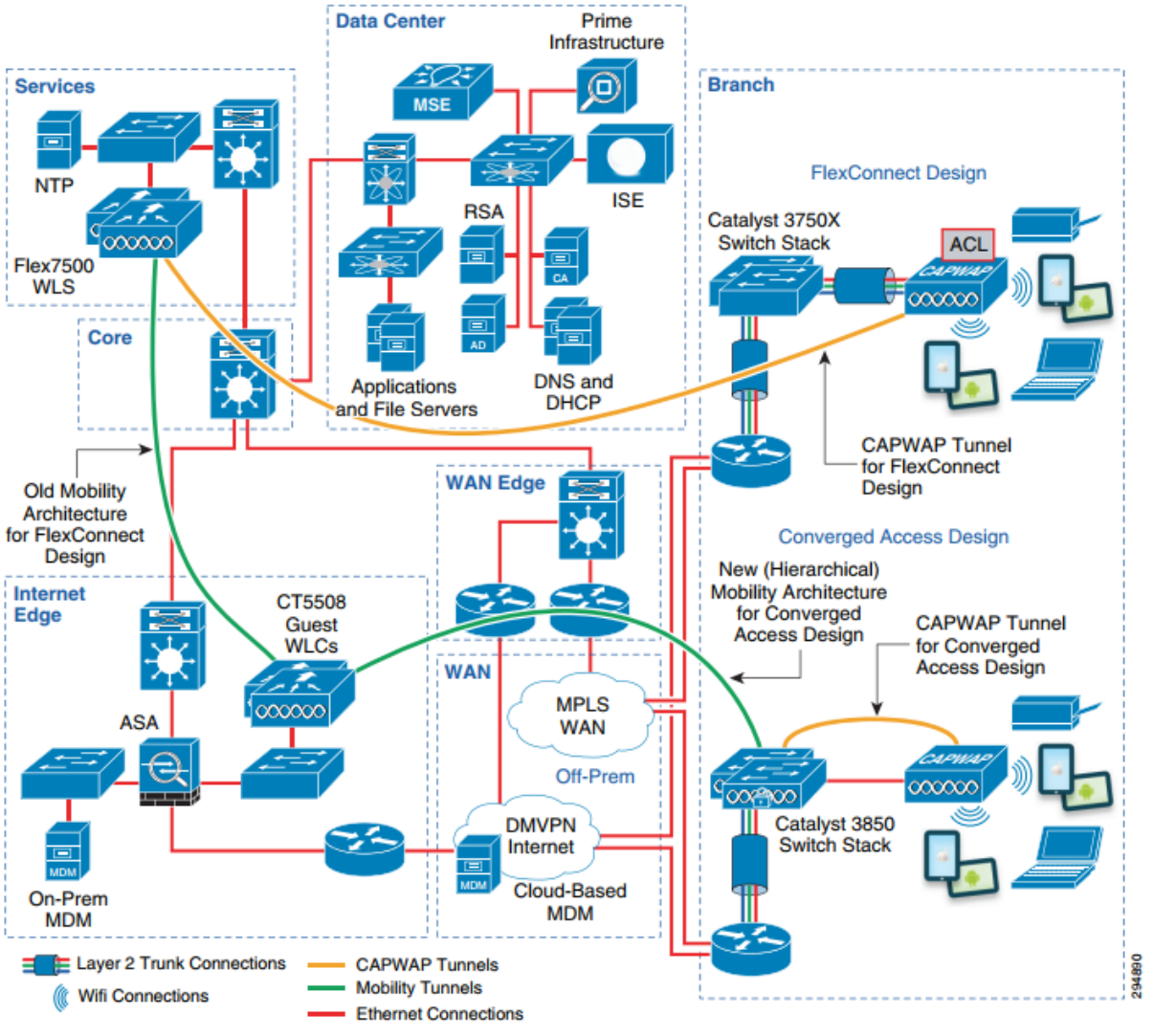
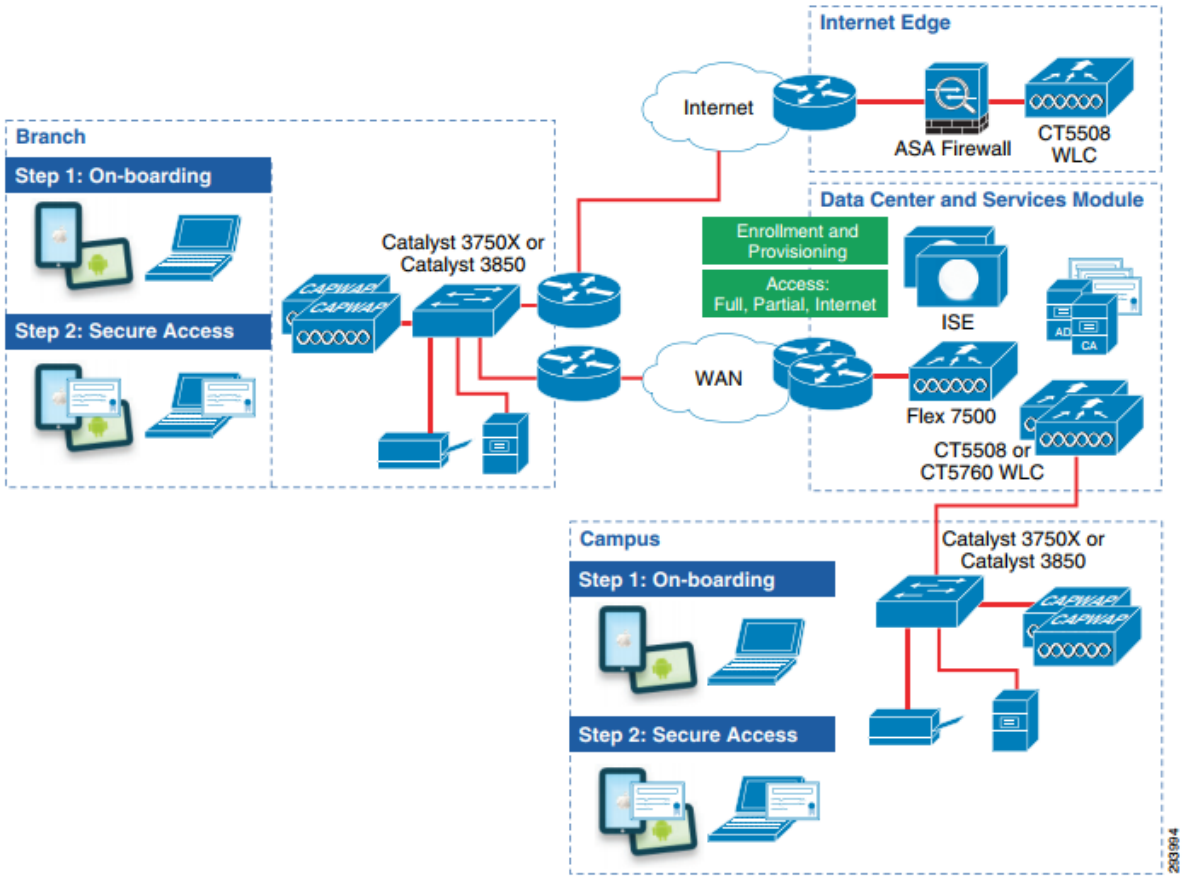


*Figure 3-3 Enrollment and Provisioning for Mobile Devices*



**BYOD Solution Components**

**BYOD devices:** These are the **corporate**-**owned** and **personally owned** endpoints that require **access** to the **corporate network** regardless of their physical **location**. This physical location can be within the corporate campus or from a public location. BYOD devices include laptops, smartphones, tablets, e-readers and notebooks.

**Wireless Access Points (AP):** Cisco wireless APs provide **wireless network connectivity** to the **corporate network** for both corporate-owned and personally owned **BYOD** devices. These APs can be physically located in the corporate campus, the branch office environment or in home offices.

**Wireless LAN Controllers (WLAN):** Cisco WLAN controllers serve as a **centralized point** for the **configuration**, **management** and **monitoring** of the Cisco WLAN solution. **WLC** works with **ISE** to enforce both **authentication** and **authorization** polices on each of the BYOD endpoints that require connectivity to the corporate network, both direct and remotely.

**Identity Services Engine (ISE):** Cisco ISE is a critical piece to the Cisco BYOD solution. It is the cornerstone of the **authentication**, **authorization** and **accounting** (AAA) requirements for endpoint access, which are governed by the security policies, put forth by the organization.

**Cisco AnyConnect Secure Mobility Client:** Provides **connectivity** for end users who need **access** to the **corporate network**. For users within the corporate campus, Branch and home offices the AnyConnect Client leverages 802.1X to provide secure access to the corporate network. For users who are using public Internet access, the AnyConnect Client provides secure VPN connectivity, including posture checking for the user's BYOD device.

**Integrated Services Routers (ISR):** Cisco ISRs will be used in the Cisco BYOD solution to provide **WAN** and **Internet access** for the branch offices and Internet access for home office environments. In addition, the ISR will provide both **wired** and **WLAN** connectivity in the branch office environments. Finally the ISR can be leveraged to provide **VPN** connectivity for mobile devices that are part of the BYOD solution.

**Aggregation Services Routers (ASR):** Cisco Aggregation Services Routers provide **WAN** and **Internet** access at the corporate campus and serve as aggregation points for all the branch and home office networks connecting back to the corporate campus for the Cisco BYOD solution.

**Cloud Web Security (CWS):** Cisco Cloud Web Security provides **enhanced security** for all the **BYOD** solution endpoints while that access Internet website using publicly available wireless hotspots and 3G, 4G and 4G LTE mobile networks.

**Adaptive Security Appliance (ASA):** The Cisco ASA provides all the **standard security** function for the **BYOD** solution at the Internet edge. In addition to traditional firewall and intrusion prevention systems (IPS) functions, the ASA also serves as a **VPN termination point** for mobile devices connecting over the **Internet** from home offices, branch offices, public network and mobile networks.

**RSA SecurID:** The RSA SecurID server provides **one**-**time password** (OTP) generation and **logging** for users that **access network** devices and other applications which require **OTP authentication**.

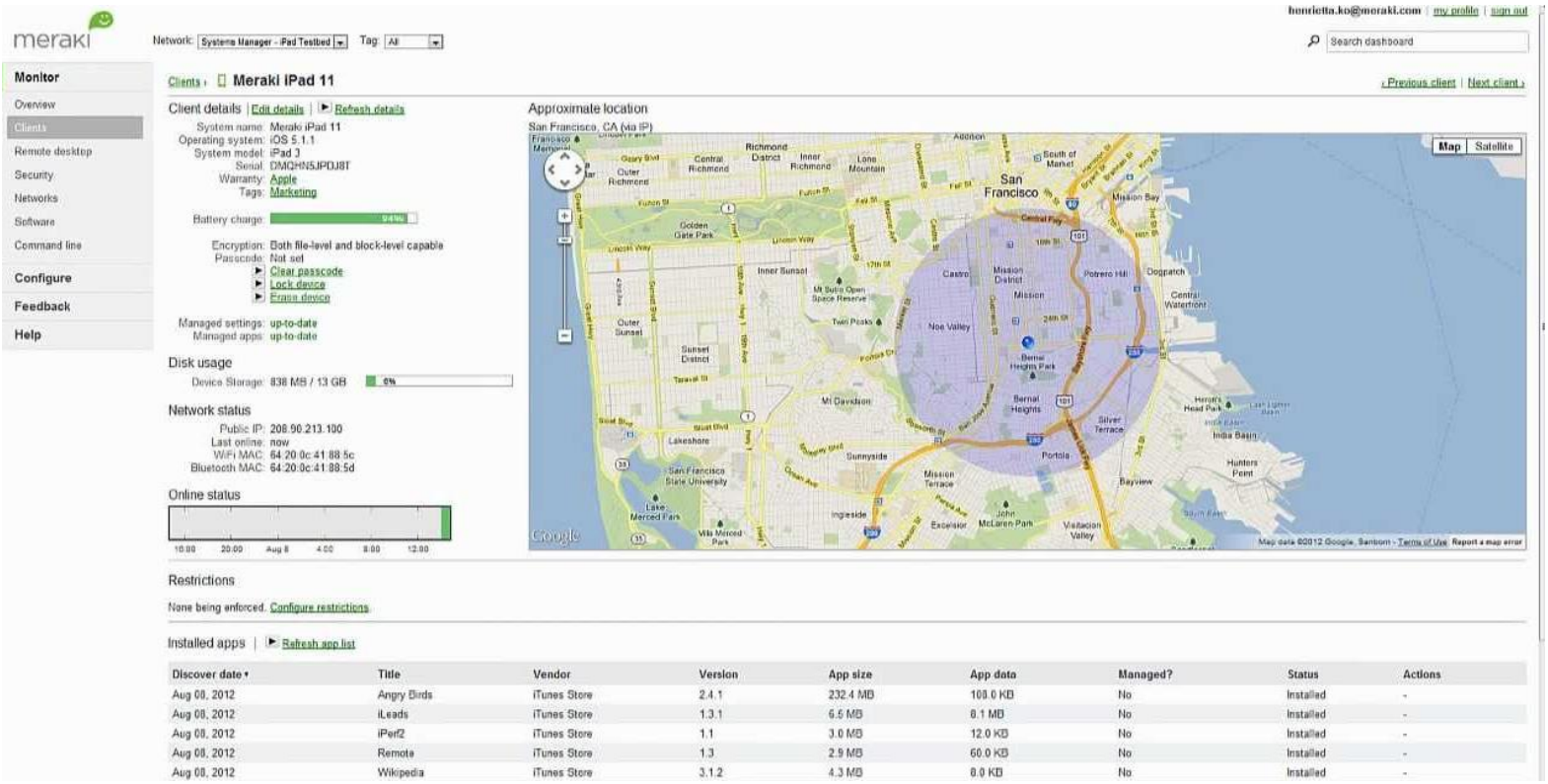
**Active Directory:** The Active directory (AD) server **enforces access control** to the network, to servers and to applications. It **restricts** access to those users with **valid authentication** credentials.

**Certificate Authority:** The Certificate authority (CA) server provides for, among other things, the on boarding of endpoints that meet **certificate requirements** for access to the corporate network. The CA server ensures that **only** devices with corporate certificates can access the corporate network.

**Mobile Device Management (MDM)**

Used to **deploy**, **manage** and **monitor** the **mobile devices** that make up the Cisco BYOD solution. These devices consist not only of mobile phones, smartphones and tablets but also notebooks, laptops and any other user device that connect back to the corporate network and that can physically be move from the office to the home, hotels and other remote locations offering public Internet connectivity. Specific functions provided by **MDM** include the following:

- Enforcement of a **PIN locks** (locking a device after a set threshold of failed login attempts has been reached).
- Enforcement of **strong passwords** for all BYOD devices. Strong password policies can also be enforced by a MDM, reducing the likelihood of brute-force-attacks.
- Detection of attempts to **jailbreak** or **root** BYOD devices, specifically smartphones and then attempting to use these compromised devices on the corporate network. MDM can be used to detect these types of actions and immediately restrict a device's access to the network or other corporate assets.
- Enforcement of **data encryption** requirements based on an organization's security policies and regulatory requirements. MDM can ensure that only devices that support data encryption and have it enabled can access the network and corporate content.
- Provide the ability to **remotely wipe** a stolen or lost BYOD device so that all data is completely removed.
- Administration and execution of **data loss prevention** (DLP) for BYOD devices. DLP prevent authorized users from doing careless or malicious things with critical data.



**MDM Deployment Options**

**On-Premise MDM Deployment:**

MDM application software is installed on **servers** that are located within the **corporate data centre** and are completely supported and maintained by the **network staff** of the corporation

The **benefits** of having an on premise MDM solution include **greater control** over management of the BYOD solution, a potentially **higher** degree of **security**, particularly with respect to intellectual property.

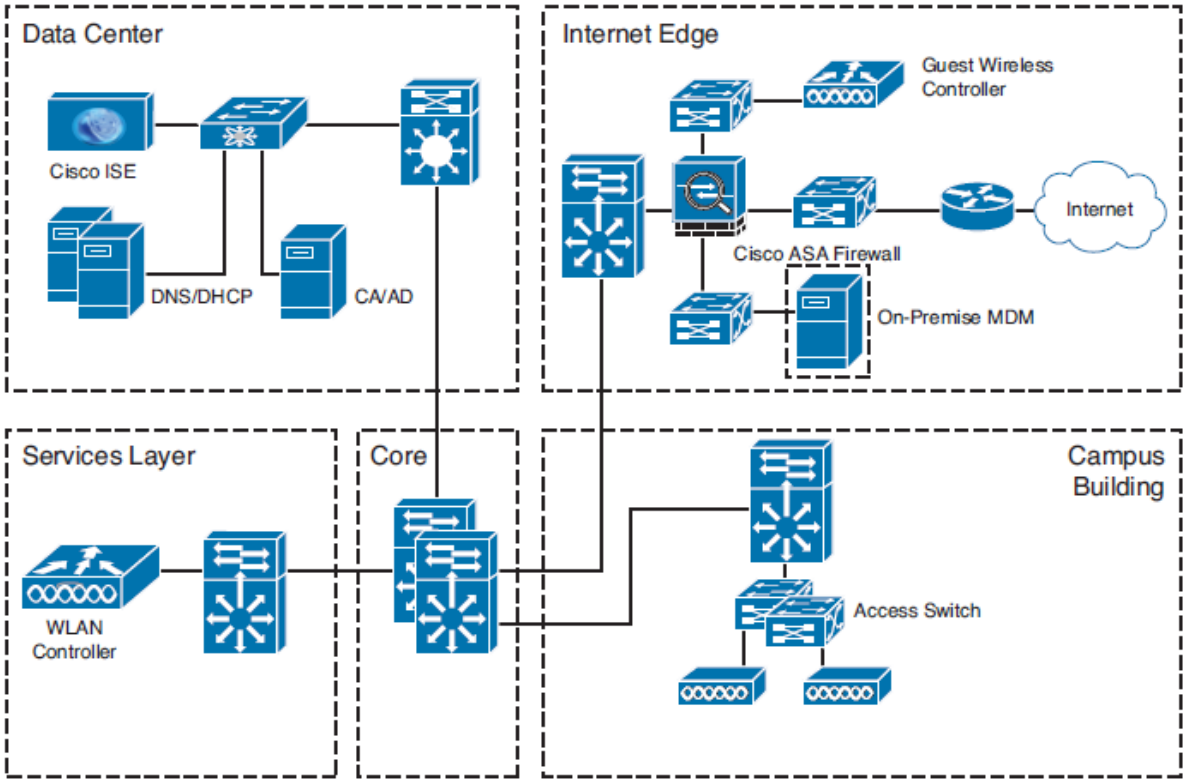
**Data Centre:** In addition to the core and distribution layer switches, the data centre consists of the Cisco ISE to enforce posture assessment and access control as well as DNS/DHCP servers to provide those services for network connectivity. A CA server to enable on boarding of endpoints that meet certificate requirements for access to the corporate network and an AD server that restricts access to only those users with valid authentication credentials.

**Internet Edge:** In addition to providing connectivity to the public internet, includes an ASA firewall to enforce security controls for all traffic going to and coming from the internet. Also located in the Internet edge layer is a WLC, which is dedicated to any of the Aps in the network to which guest users can connect. The last key component in the Internet edge layer is the on premise MDM, which provides all policies and profiles, digital certificate, applications, data and configuration settings for all the BYOD devices that require connectivity to the corporate network,

**Services:** Contains the WLC for all Aps to which the corporate users connect. However, any other network-based services required for the corporate network can be found within the Services module.

**Core:** There are no other functions served by the Core module for the BYOD solution beyond what it normally provides. The core serves as the man distribution and routing point for al network traffic traversing the corporate network environment.

**Campus Building:** A distribution switch provides the main ingress/egress point for all network traffic entering and exiting from the campus environment. All users requiring network connectivity within the campus building do so through either hardwired connections to the access switches or via WLAN access to the corporate APs.


Data Center
Cisco ISE
DNS/DHCP
CA/AD
Internet Edge
Guest Wireless Controller
Cisco ASA Firewall
Internet
On-Premise MDM
Services Layer
WLAN Controller
Core
Campus Building
Access Switch

**Cloud-Based MDM Deployment:**

**MDM** application software is hosted by a **managed service provider** who is solely responsible for the **deployment**, **management** and **maintenance** of the **BYOD** solution.

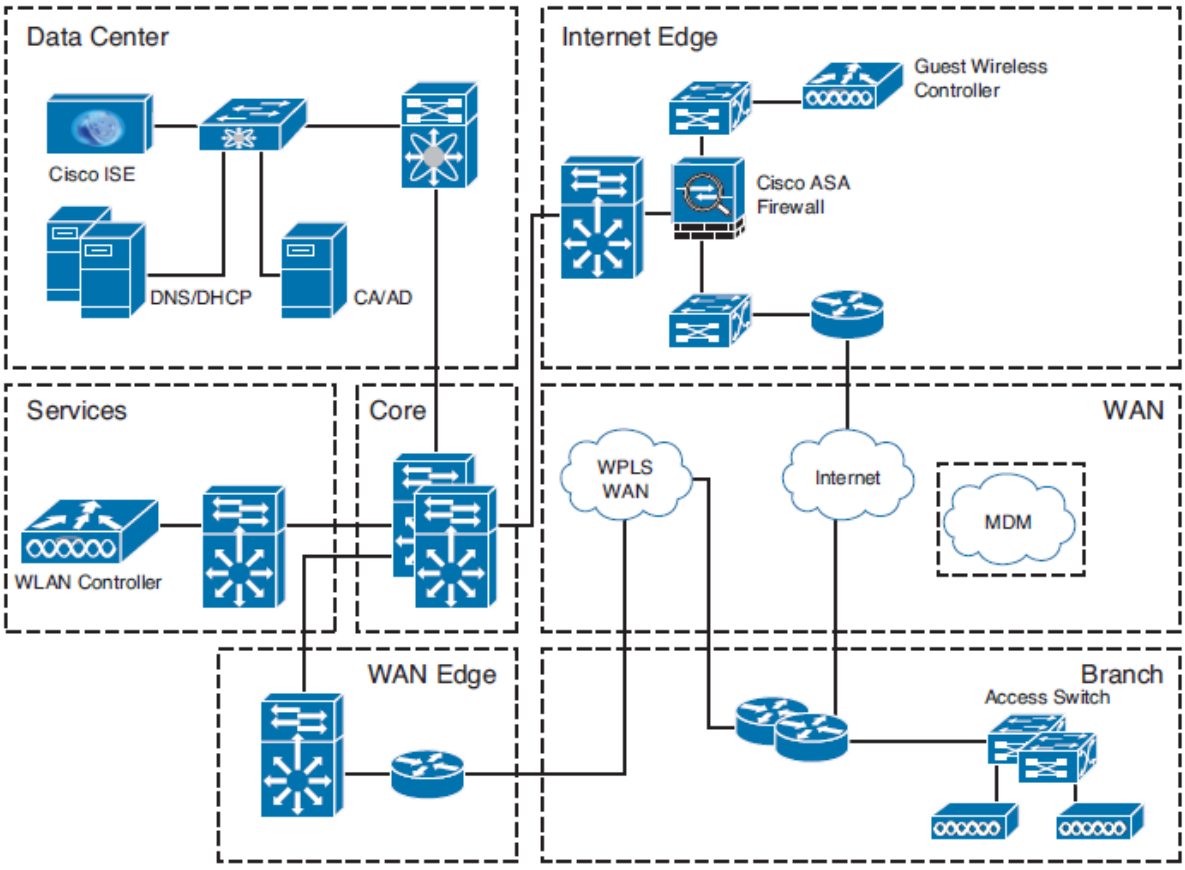
The **benefits** of having a **cloud-based** MDM solution include a much more **simplified solution** from a customer perspective because the customer is no longer responsible for configuring, operating and maintaining the MDM software. Giving up this control, however brings with is some potential **concerns** with the **overall security** of the solution. The cloud-based solution also brings with it greater **scalability**, **flexibility** and **speed** of deployment over an on premise MDM solution.

**WAN:**

1. It provides MPLS VPN connectivity for the branch office back to corporate network.
2. Internet access for the branch office.
3. Access to the cloud-based MDM functionality.

**WAN edge:** Serve as the ingress/egress point for the MPLS WAN traffic entering from and exiting to the branch office environment.

**Branch Office:** Routers provide the main ingress/egress point for all network traffic entering and exiting from the branch environment.



# Chapter 5: Fundamentals of VPN Technology and Cryptography

**What is a VPN?**

A **Virtual Private Network** allows connectivity between two devices. Those two devices could be computer on the same local-area network or could be connected over a wide-area network.

The word **virtual** in VPN refers to a logical connection between two devices.

The letter **P** in VPN refers to private since the connection between the two devices is private.

**VPN Benefits:**

- Cheaper
- Scalability

**Types of VPNs**

**IPsec:** Implements security of IP packets at Layer 3 of the OSI model, and can be used for site-to-site VPNs and remote-access VPNs.

**SSL:** Secure Sockets Layer implements security of TCP sessions over encrypted SSL tunnels of the OSI model and can be used for remote-access VPNs.

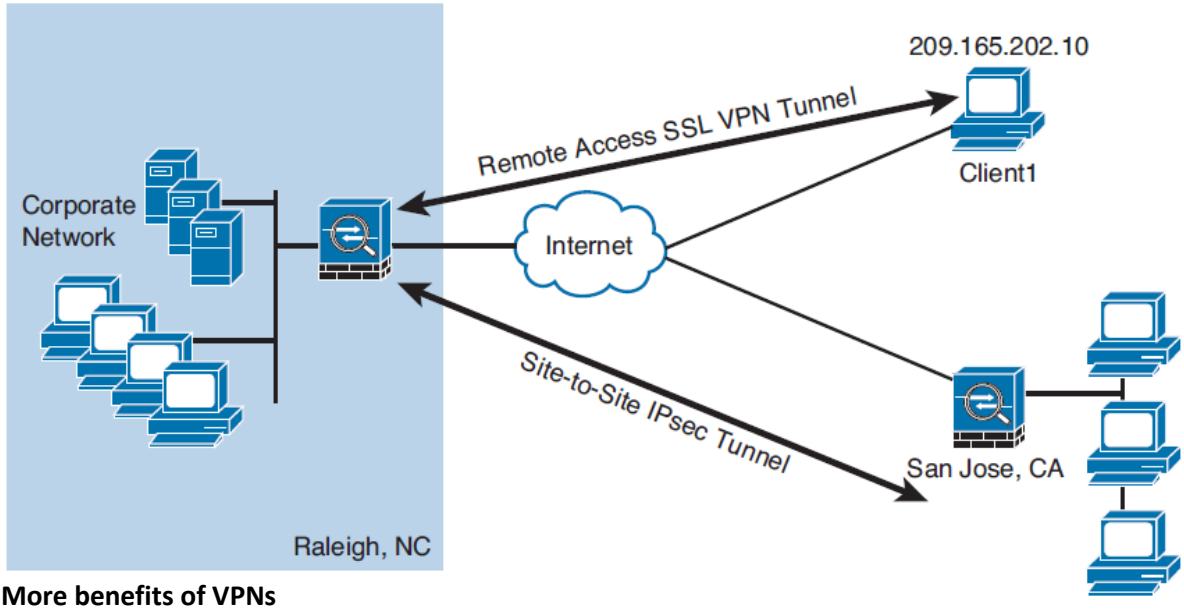
**MPLS:** Multiprotocol Label Switching and MPLS Layer 3 VPNs are provided by a service provider to allow a company with two or more sites to have logical connectivity between the sites using the service provider network for transport. This is also a type of VPN called (MLS L3VPN), but there is no encryption by default.

**IPsec** could be used on top of the MPLS VPN to add confidentiality (through encryption) and the other benefits of IPsec to protect the Layer 3 packets.

**Two Main Types of VPNs**

**Remote-access VPNs:** A VPN connection between a user's computer to the corporate headquarters. Remote-access VPNs can use IPsec or Secure Shell encryption for their VPN. Many Cisco customers use the Cisco AnyConnect client for remote access SSL VPNs.

**Site-to-site VPNs:** Having two or more sites that that want to connect securely together so that each site can communicate with the other site or sites. Use a collection of VPN technologies called IPsec.




**More benefits of VPNs**

**Confidentiality:** Only the intended parties can understand the data that is sent. Any party that eavesdrops may see the actual packets, but the contents of the packet or the payload are scrambled (cipher text) and meaningless to anyone who cannot unlock or decrypt the data.

**Data Integrity:** An important factor about the data is to **not** be **modified** while in **transit**. You can authenticate the peer at the other end of the VPN tunnel is several different ways, including the following:

- Pre-Shared keys used for authentication only
- Public and Private Key pairs used for authentication only.
- User authentication

**Antireplay Protection**

If an attacker watches your VPN traffic and captures it with the intent to replay it back and fool one of the VPN peers into believing that the peer trying to connect is a legitimate peer. An attacker might be able to build a VPN pretending to be a different device. To sole that, most implementation of VPNs have an antireplay functionality built in. This just means that once a VPN packet has been sent and accounted for, that exact same VPN packet is not valid the second time in the VPN session.

**Cryptography Basic Components**

**Ciphers and Keys**

**Ciphers:**

A **Cipher** is a **set** of **rules**, which can also be called an **algorithm** about how to perform **encryption** or **decryption**.

Common **methods** that cipher use include the following:

- **Substitution**: Substitutes one character for another.
- **Polyalphabetic**: Similar to substitution but instead of using a single alphabet is could use multiple alphabets and switch between them by some trigger character in the encoded message.
- **Transposition**: Uses many different options, including the rearrangement of letters, example top to bottom and left to right.

**Keys:**

A one-time pad (OTP) is a good example of a key that is only used **once**. Using this method, if we want to encrypt a 32-bit message, we use a 32-bit key also called the pad, which is used one time only. Each bit is mathematically computed with a corresponding bit from our message, and the results are our cipher text or encrypted content. The **pad** must be **known** also by the **receiver**.

**Block and Stream Ciphers**

**Block Ciphers:**

A **Symmetric** key (same key to encrypt and decrypt) cipher that operates on a group of bits called a block. A block cipher encryption algorithm may take a 64-bit block of plain text and generate a 64-bit block of cipher text. The same key to encrypt is also used to decrypt.

**Examples if symmetrical block cipher algorithms include the following:**

- Advanced Encryption Standard (AES)
- Triple Digital Encryption Standard (3DES)
- Blowfish
- Digital Encryption Standard(DES)
- International Data Encryption Algorithm (IDEA)

**Block ciphers** may add padding in cases there is not enough data to encrypt to make a full block size. This might result in very small amount of wasted overhead, because the small padding would be processed by the cipher along with the real data.

**Stream Ciphers:**

A **Stream cipher** is a **symmetric** key cipher (same key to encrypt and decrypt) where each bit of plaintext data to be encrypted is done 1 bit at a time against the bits of the key stream, also called a cipher digit stream. The resulting output is a cipher text stream. Because a cipher stream does not have to fit in a given block size, there may be slightly less overhead than a block cipher that is requiring padding to complete a block size.

**Symmetric and Asymmetric Algorithm**

**Symmetric:**

Uses to **same key** to encrypt and decrypt the data. Two devices connected via a VPN both need the key or keys to successfully encrypt and decrypt the data that is protected using a symmetric encryption algorithm.

**Common examples of symmetric encryption algorithms include the following:**

- DES
- 3DES
- AES
- IDEA
- RC2, RC4, RC5, RC6
- Blowfish

**Symmetrical** encryption algorithms are used for most of the data that we protect in VPNs today. We use symmetrical to encrypt the **bulk** of our **data** is because it is much **faster** and takes **less CPU processes**.

**Asymmetric:**

An example of **Asymmetric** algorithm is public key algorithms. We use two different keys that mathematically work together as a pair. There is a very high CPU cost when using key pairs to lock and unlock data. This is why we use asymmetric algorithms for things such as authenticating a VPN peer or generating keying material that we could use for our symmetrical algorithms.

One reason this is called **public** key cryptography is that we allow one of these keys to be published and available to anyone who wants to use it. The other key in the key pair is the **private** key that is only known by the device that owns the **public**-**private key pair**.

**Examples of asymmetrical algorithms include the following:**

**RSA:** Used for authentication and has a key length from 512 to 2048.

**DH:** Diffie-Hellman key exchange protocol is an **asymmetrical** algorithm that allows two devices to negotiate and establish shared secret keying material (keys) over an untrusted network. Although the algorithm is asymmetrical, the **keys** generated are **symmetrical** that can then be used with symmetrical algorithms such as **3DES** and **AES**.

**ELGamal:** Based on the DH exchange.

**ECC:** Elliptic Curve Cryptography

**Asymmetrical** algorithms require more **CPU processing** power than a symmetrical algorithm. **Asymmetrical** algorithms however are **more secure**. A typical key-length used in asymmetrical algorithms can be anywhere between **2048** and **4096**.

**Hashes:**

Hashing is a method used to **verify data integrity**. A cryptographic hash function is a process that takes a block of data and creates a small fixed-sized hash value. It is a one-way function meaning that if two different computers take the same data and run the same hash function, they should get the same fixed-sized hash value.

If the hash generated **matches** the hash that was sent, we know that the entire packet is **intact**.

**The three most popular types of hashes are as follows:**

- **Message Digest 5 (MD5):** 128-Bit digest
- **Secure Hash Algorithm 1 (SHA-1):** 160-Bit digest
- **Secure Hash Algorithm (SHA-2):** 224-Bits to 512-Bits

**Hashed Message Authentication Code (HMAC)**

Uses the mechanism of hashing but instead of using a hash that anyone can calculate, it includes in its calculation a **secret key** of some type. Then only the other party who knows the secret key and can calculate the resulting hash can correctly verify the hash. An attacker who is eavesdropping cannot inject or remove data from those packets **without being noticed**.

**Digital Signatures**

This provides:

- Authentication
- Data Integrity
- Nonrepudiation

**Key Management**

Key Management deals with generating keys, verifying keys, exchanging keys, storing keys and destroying keys.

**Next-Generation Encrypting Protocols:**

- Elliptic Curve Cryptography (ECC) replaces RSA signatures with the ECDSA algorithm and replaces the DH key exchange with ECDH.
- AES in the Galois/Counter Mode (GCM) of operation
- ECC Digital Signature Algorithm
- SHA-256, SHA-384 and SHA-512

**IPsec and SSL**

**IPsec**

IPsec is a **collection** of **protocols** and **algorithms** used to protect IP packets at **Layer 3**. IPsec provides that core benefits of confidentiality through encryption, data integrity through hashing and HMAC, and authentication using digital signatures or using pre-shared key (PSK) that is just for the authentication. IPsec also provides antireplay support.

- ESP and AH: Encapsulating Security Payload (ESP) which can do all the features of IPsec and Authentication Header (AH), which can do many parts of the IPsec objectives, except for the important one of encryption of the data.
- Encryption algorithms for confidentiality: DES, 3DES, AES
- Hashing algorithms for integrity: MD5, SHA
- Authentication algorithms: Pre-Shared keys (PSK), RSA digital signatures
- Key management: Diffie-Hellman, PKI, IKE

**SSL**

Transmitting information over a public network needs to be secured through encryption to prevent unauthorized access to that data. To use SSL the user connects to an SSL server by using HTTPS.

**VPN Components**

| Component | Function | Examples |
|---|---|---|
| **Symmetrical encryption algorithms** | Use the same key for encrypting and decrypting data. | DES, 3DES, AES, IDEA |
| **Asymmetrical encryption** | Uses a public and private key. One key encrypts the data and the other key in the pair is used to decrypt. | RSA, Diffie-Hellman |
| **Digital Signature** | Encryption of hash using private key and decryption of hash with the sender's public key. | RSA signatures |
| **Diffie-Hellman key exchange** | Uses a public-private key pair asymmetrical algorithm, but creates final shared secret (keys) that are then used by symmetrical algorithms. | Used as one of the many services of IPsec |
| **Confidentiality** | Encryption algorithms provide this by turning clear text into cipher text | DES, 3DES, AES, RSA, IDEA |
| **Data Integrity** | Validates data by comparing hash values | MD5, SHA-1 |
| **Authentication** | Verifies the peer's identity to the other peer | PSKs, RSA signatures |

**Public Key Infrastructure**

**PKI** uses **asymmetric** encryption.

**Message Confidentiality** – the Message is encrypted with the receiving party's public key. Only the receiving party can decrypt the message with their private key.

**Message authenticity** – The message is encrypted using the senders private key. The message can only be decrypted with the senders public key proving the message is authentic.

A public key infrastructure contains the following parts-

- Certificate Authorities
- Users, people, devices etc
- Storage and Protocols
- Supporting organisational framework
- Supporting legal framework

**Certificates**

A Certificate contains:

1. Public key of the router.
2. Device signature (name) encrypted with the private key. This can only be decrypted using the public key, proves the router is who he says he is.
3. CA Signature. This is the name of the CA encrypted with the CA private key. Only the CA public key can decrypt the signature proving the certificate was signed by the certification authority.

A certificate can have a certificate class to indicate the trustworthiness of the certificate. Typically a number, the higher the number the more trustworthy the certificate. The higher the class the more must be done to prove the authenticity of the requester. Class 0 may require no checks, class 1 may require an email from the domain to prove identity.

**SCEP (Simple Certificate Enrolment Protocol**) – Automated method to send certificates to hosts/routers. A host will request a certificate from the CA. Operates in two modes:

- **Manual** – Administrator approves the request
- **Pre-shared key** – Devices will pass a key to the CA to allow the CA to automatically generate the certificate.

**Certificate Authority**

A certificate authority is a computer or entity that creates and issues digital certificates. Inside of a digital certificate is information about the identity of a device, such as its IP address, *fully qualified domain name (FQDN)*, and the public key of that device.
The CA takes requests from devices that supply all of that information (including the public key generated by the computer that is making the request) and generates a digital certificate, which the CA assigns a serial number to and signs the certificate with its own digital signature (the CA's signature).
Also included in the final certificate is a URL that other devices can check to see whether this certificate has been revoked and the validity dates for the certificate (which is similar to the expiration date of food products).
Also in the certificate is the information about the CA that issued the certificate and several other parameters used by PKI.

A trusted third party which sings the public keys. There are multiple topologies for a PKI system-

- **Single root** – Difficult to scale and vulnerable in that if the root key is compromised all certificates generated are invalid.
- **Hierarchical** – A root CA in turn issues certificates to subordinate CA's. The subordinate CA's then issue certificates to end users. This improves scalability and reduces the impact if a key is compromised.
- **Cross-certifying** – A CA will cross certify with another CA on different PKI installation, in effect creating a trust relationship.

A CA perform many tasks in addition to signing user certificates such as authenticating users when they enrol with the PKI, key generation and distribution of certificates. These tasks can be offloaded to a Registration Authority (RA) enabling the CA to concentrate on signing.

**Root and Identity Certificates**

**Root Certificate**

A root certificate contains the public key of the CA server and the other details about the CA server.

Includes:

- Serial number
- Issuer,
- Validity dates,
- Subject of the certificate,
- Public key
- Thumbprint algorithm
- Thumbprint

**Identity Certificate**

An identity certificate is similar to a root certificate, but it describes the client and contains the public key of an individual host (the client).

**X.500 and X.509v3 Certificates**

X.500 is a series of standards focused on directory services and how those directories are organized. Many popular network operating systems have been based on X.500, including Microsoft Active Directory. A common protocol that is used to do lookups from a directory is called Lightweight Directory Access Protocol (LDAP).

**Authenticating and Enrolling with the CA**

Step 1. Authenticating the CA after downloading the root certificate use an out-of-band method, such as making a telephone call, to validate the root certificate.

Step 2. Involves generating a public-private key pair and including the public key portion in any requests for your own identity certificate; the CA can take all of your information and generate an identity certificate for you, which includes your public key, and then send this certificate back to you.

**Public Key Cryptography Standards**

- **PKCS#10:** This is a format of a certificate request sent to a CA that wants to receive its identity certificate. This type of request would include the public key for the entity desiring a certificate.
- **PKCS#7:** This is a format that can be used by a CA as a response to a PKCS#10 request.
- **PKCS#1:** RSA Cryptography Standard
- **PKCS#12:** A format for storing both public and private keys using a symmetric password-based key to “unlock” the data whenever the key needs to be used or accessed.
- **PKCS#3:** Diffie-Hellman key exchange

**Simple Certificate Enrolment Protocol**

Simple Certificate Enrolment Protocol (SCEP) can automate most of the process for requesting and installing an identity certificate.

**Revoked Certificates**

If a **certificate revocation list** (CRL) is checked, and the certificate from the peer is on that list, the authentication stops at that moment.

To check whether certificates have been **revoked**:

- Certificate revocation list (CRL): A CRL could be very large and can be accessed by LDAP or HTTP.
- Online Certificate Status Protocol (OCSP): a client simply sends a request to find the status of a certificate and gets a response without having to know the complete list of revoked certificates.
- Authentication, authorization, and accounting (AAA)

**Uses for Digital Certificates**

- Can be used when you do online banking from your PC to the bank's website
- if you use SSL technology for your remote-access VPNs you can also use digital certificates for authenticating the peers
- Use digital certificates with the protocol family of IPsec
- Can also be used with protocols such as 802.1X, which involves authentication at the edge of the network

**PKI Topologies**

- **Single root** – Difficult to scale and vulnerable in that if the root key is compromised all certificates generated are invalid.
- **Hierarchical** – A root CA in turn issues certificates to subordinate CA's. The subordinate CA's then issue certificates to end users. This improves scalability and reduces the impact if a key is compromised.
- **Cross-certifying** – A CA will cross certify with another CA on different PKI installation, in effect creating a trust relationship.

A CA perform many tasks in addition to signing user certificates such as authenticating users when they enrol with the PKI, key generation and distribution of certificates. These tasks can be offloaded to a Registration Authority (RA) enabling the CA to concentrate on signing.

**ASA's Default Certificate**

The problem with a self-signed certificate is that no browsers or other devices will have the ASA listed as a trusted CA, and HTTPS connections to the ASA, such as an administrator who wants to run ASDM, will receive a warning message that the certificate is not trusted.

**Generate a new public-private pair**

Keith-asa1(config)# crypto key generate rsa label My-Key-Pair modulus 2048 noconfirm

Keith-asa1(config)# crypto ca trustpoint New-CA-to-Use

Keith-asa1(config-ca-trustpoint)# keypair New-Key-Pair

Keith-asa1(config-ca-trustpoint)# id-usage ssl-ipsec

Keith-asa1(config-ca-trustpoint)# no fqdn

Keith-asa1(config-ca-trustpoint)# subject-name CN=ciscoas

Keith-asa1(config-ca-trustpoint)# enrollment url http://192.168.1.105

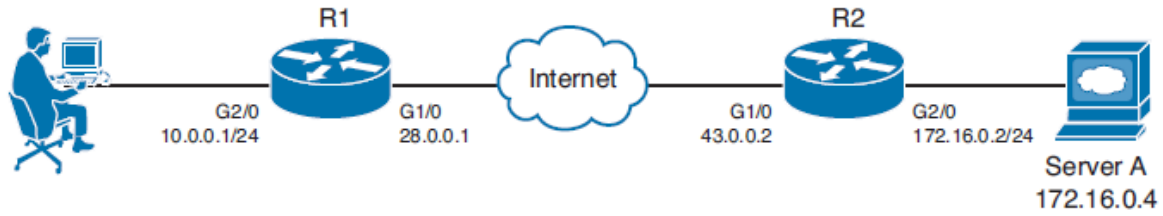
Keith-asa1(config-ca-trustpoint)# exit

Keith-asa1(config)#  crypto ca authenticate New-CA-to-Use nointeractiv

Keith-asa1(config)#  crypto ca enroll New-CA-to-Use noconfirm

# Chapter 6: Fundamentals of IP Security

**IPsec Concepts, Components and Operations**




**The Goal of IPsec**

| Goal | Method That Provides the Feature |
|---|---|
| **Confidentiality** | Encryption |
| **Data Integrity** | Hashing |
| **Peer Authentication** | Pre-Shared keys, RSA digital signatures |
| **Antireplay** | Integrated into IPsec, applying serial numbers to packets |

**Confidentiality:** Provided through encryption changing clear text into cipher text.

**Data Integrity:** Provided through hashing and/or through Hashed Message Authentication Code (HMAC) to verify that data has not been manipulated during its transit across the network.

**Authentication:** Provided through authenticating the VPN peers near the beginning of a VPN session using pre-shared keys (PSK) or digital signatures. Authentication can also be done continuously through the use of an HMAC, which includes a secret known only to the two ends of the VPN.

**Antireplay protection:** When VPNs are established, the peers can sequentially number the packets and if a packet is attempted to be replayed again, the packet will not be accepted because the VPN device believes it has already processes that packet.

**The Internet Key Exchange (IKE) Protocol**
IPsec uses the Internet Key Exchange (IKE) protocol to **negotiate** and **establish** secured site-to-site or remote access virtual private network (VPN) tunnels. In **IKE Phase 1** IPsec peers negotiate and authenticate each other. In **Phase 2** they negotiate keying materials and algorithms for the encryption of the data being transferred over the IPsec tunnel.

- **IKEv2** enhances the function of performing dynamic key exchange and peer authentication.
- Both **IKEv1** and **IKEv2** protocols operate in **two phases**. IKEv2 provides a simpler and more efficient exchange.

Phase 1 in IKEv2 is IKE_SA, consisting of the message pair IKE_SA_INIT (IKEv1 Phase 1).
Phase 2 in IKEv2 is CHILD_SA is the IKE_AUTH message pair (IKEv1 Phase 2).

**The Play by Play for IPsec**

**Step 1: Negotiate the IKEv1 Phase 1 Tunnel**
This tunnel (once established) is not going to be used to forward user packets, but rather only to protect management traffic related to the VPN between the two routers. Five basic items need to be agreed upon between the two VPN devices/gateways (in this case, the two routers) for the IKE Phase 1 tunnel to succeed, as follows:

- **Hash algorithm**, MD5 or SHA
- **Encryption algorithm**, DES (weak), 3DES (better) or AES (best)
- **Diffie-Hellman** (DH) group to use; The DH "group" refers to the modulus size (length of the key). The purpose of DH is to generate shared secret keying material (symmetric keys) that may be used by the two VPN peers for symmetrical algorithms, such as AES.
- **Authentication method**: PSK or RSA signatures
- **Lifetime:** How long until this IKE Phase 1 tunnel should be torn down.

**How to Remember the Five Items Negotiated in IKE Phase 1**
As a handy way to recall the five pieces involved in the negotiation of the IKE Phase 1 tunnel, you might want to remember that the two devices HAGLE over IKE Phase 1:
**H:** Hash
**A:** Authentication method
**G:** DH group (a stretch, but it works)
**L:** Lifetime of the IKE Phase 1 tunnel
**E:** Encryption algorithm to use for the IKE Phase 1 tunnel

**Who Begins the Negotiation?**
The initiator sends over all of its IKE Phase 1 policies, and the other VPN peer looks at all of those policies to see whether any of its own policies match the ones it just received. If there is a matching policy, the recipient of the negotiations sends back information about which received policy matches, and they use that matching policy for the IKE Phase 1 tunnel.

**Step 2: Run the DH Key Exchange**
DH allows two devices that do not yet have a secure connection to **establish shared secret keying material** (keys that can be used with symmetrical algorithms, such as AES).

**Step 3: Authenticate the Peer**
The last piece of IKE Phase 1 is to **validate** or **authenticate** the **peer** on the other side.

IKE Phase 1 tunnel, this tunnel is used only as a management tunnel so that the two routers can securely communicate with each other directly. IKE Phase 1 tunnel is not used to encrypt or protect the end user's packets. The IKE Phase 2 tunnel includes the hashing and encryption algorithms.
So, we could say we have one **IKE Phase 1 bidirectional tunnel** used for management between the two VPN peers and two **IKE Phase 2 unidirectional tunnels** used for encrypting and decrypting end-user packets.

**Configuring and Verifying IPsec**

The first thing to plan is what protocols to use for IKE Phase 1 and IKE Phase 2 and to identify which traffic should be encrypted.

**Implement IPsec VPNs**

```
! This implements our IKE Phase 1 policy. The default policy that CPP
! implements is its policy #1, (which has higher priority than a higher
! numbered policy, including our policy #2.)
R1(config-isakmp)# crypto isakmp policy 2
R1(config-isakmp)# authentication pre-share
R1(config-isakmp)# encr aes 128
R1(config-isakmp)# hash md5
R1(config-isakmp)# group 2
R1(config-isakmp)# lifetime 21600
R1(config-isakmp)# exit
! Note: I like to remove the default policy from CCP for IKE Phase 1, and
! for that reason, I have not replicated it here.
```

```
! This specifies that the PSK of cisco123 should be used as a key for the
! authentication of IKE Phase 1 with peer 43.0.0.2.
R1(config)# crypto isakmp key cisco123 address 43.0.0.2

! Access list that identifies any traffic from the 10.0.0.0/24 network
! and destined for the 172.16.0.0/24 network. An ACL used for cryptography
! is often referred to as a "crypto ACL". This ACL will not be directly
! applied to an interface, but rather it will be called on or "referenced"
! within the crypto map, later in this configuration.
R1(config)# access-list 100 permit ip 10.0.0.0 0.0.0.255 172.16.0.0 0.0.0.255

! The IKE Phase 2 transform set that says SHA and AES 256 should be used.
R1(config)# crypto ipsec transform-set MY-SET esp-sha-hmac esp-aes 256

! Tunnel mode is the default, and means that R1 will take any outbound
! packets matching the access list, encrypt them and then re-encapsulate
! them inside of an IPsec packet, which is then forwarded to the peer (R2)
! on the other side of the VPN tunnel. Whenever customer traffic is going
! through a VPN router, it will need to be in tunnel mode to work.
! Transport mode is the other option, and it is used only when the transit
! traffic is directly from and to the endpoints of the VPN tunnel (such as
! R1 and R2 talking amongst themselves). Because we are encrypting traffic
! for the end users, tunnel mode (the default) will be used.
R1(cfg-crypto-trans)# mode tunnel

R1(cfg-crypto-trans)# exit

! The crypto map is a big "if-then" statement. It is applied to the outside
! (Internet facing) interface, and then it watches for traffic.
! If outbound traffic matches the ACL, then the router knows the packet
! should be encrypted, encapsulated into an IPsec header (usually protocol
! 50, which is ESP and stands for Encapsulating Security Payload), and then
! sent to the IP address of the peer on the other side (R2) who would
! decrypt and forward the plain text packet to the device on network
! 172.16.0.0/24 "ipsec-isakmp" means that we want the router to automatically
! negotiate the IKE Phase 2 tunnel, using isakmp, which stands for Internet
! Security Association Key Management Protocol. In short, it means automate
! the process, so the administrator doesn't manually have to configure all
! keys for encryption. The "1" represents sequence number 1. If we had
! 5 different IPsec peers, we could use 5 different sequence numbers in the
! same crypto map to organize our policies based on the sequence number and
! corresponding peer we would be using IPsec with.
```

```
R1(config)# crypto map SDM_CMAP_1 1 ipsec-isakmp

! This tells the crypto map to pay attention to ACL 100 to see if traffic
! should be encrypted or not
R1(config-crypto-map)# match address 100
! If the traffic matches the ACL, then R1 should use the transform-set
! named MY-SET to negotiate the IKE Phase 2 tunnel, with the peer at
! 43.0.0.2
! If the IKE Phase 1 tunnel isn't present, it will trigger the negotiation
! of that first. If the IKE Phase 2 is already in place, the router will
! use the existing tunnel for the encryption and transmission of the
! customer's packet
R1(config-crypto-map)# set transform-set MY-SET
R1(config-crypto-map)# set peer 43.0.0.2
R1(config-crypto-map)# exit

! Applying the crypto map to the interface, is what activates our policy,
! and tells the router to start paying attention in looking for interesting
! traffic (which is the traffic that matches the ACLs referenced in the
! crypto map).
R1(config)# interface GigabitEthernet1/0
R1(config-if)# crypto map SDM_CMAP_1
R1(config-if)# exit
%CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is ONcom
```

**Example 6-2** *Edited Mirrored VPN Configuration Appropriate for R2*

```
crypto isakmp policy 2
 authentication pre-share
 encr aes 128
 hash md5
 group 2
 lifetime 21600
 exit


crypto isakmp key cisco123 address 23.0.0.1
crypto ipsec transform-set MY-SET esp-sha-hmac esp-aes 256
 mode tunnel
 exit

ip access-list extended SDM_1
 permit ip 172.16.0.0 0.0.0.255 10.0.0.0 0.0.0.255
 exit

crypto map SDM_CMAP_1 1 ipsec-isakmp
 match address SDM_1
 set transform-set MY-SET
 set peer 23.0.0.1
 exit

interface g1/0
crypto map SDM_CMAP_1
end
```

**Verifying the IPsec VPN from the CLI**

```
! Verify the IKE Phase 1 policies in place on the router
R1# show crypto isakmp policy

Global IKE policy
Protection suite of priority 2
        encryption algorithm:   AES - Advanced Encryption Standard (128 bit keys).
        hash algorithm:         Message Digest 5
        authentication method:  Pre-Shared Key
        Diffie-Hellman group:   #2 (1024 bit)
        lifetime:               21600 seconds, no volume limit

! Show the details of the crypto map, and where it is applied, showing
! the contents of the IKE Phase 2 transform sets, learning the ACLs
! involved for the VPN, who the current peer is, and more.
R1# show crypto map
Crypto Map "SDM_CMAP_1" 1 ipsec-isakmp
        Description: Tunnel to43.0.0.2
        Peer = 43.0.0.2
        Extended IP access list 100
            access-list 100 permit ip 10.0.0.0 0.0.0.255 172.16.0.0 0.0.0.255
        Current peer: 43.0.0.2
        Security association lifetime: 4608000 kilobytes/3600 seconds
        Responder-Only (Y/N): N
        PFS (Y/N): N
        Transform sets={
                MY-SET:  { esp-256-aes esp-sha-hmac  } ,
        }
        Interfaces using crypto map SDM_CMAP_1:
                GigabitEthernet1/0

! See the details for the IKE Phase 1 tunnel that is in place

R1# show crypto isakmp sa detail
Codes: C - IKE configuration mode, D - Dead Peer Detection
       K - Keepalives, N - NAT-traversal
       T - cTCP encapsulation, X - IKE Extended Authentication
       psk - Preshared key, rsig - RSA signature
       renc - RSA encryption
IPv4 Crypto ISAKMP SA

C-id  Local       Remote      I-VRF    Status Encr Hash Auth DH Lifetime Cap.

1001  23.0.0.1    43.0.0.2            ACTIVE aes  md5  psk  2  00:04:05
       Engine-id:Conn-id =  SW:1

! See the details for the IKE Phase 2 tunnels that are in place.  There is
```

```
R1# show crypto ipsec sa
<Note: less relevant content removed>
interface: GigabitEthernet1/0
    Crypto map tag: SDM_CMAP_1, local addr 23.0.0.1
! Shows what traffic is being encrypted.  All IP traffic between
! 10.0.0.0/24 and 172.16.0.0/24
   local  ident (addr/mask/prot/port): (10.0.0.0/255.255.255.0/0/0)
   remote ident (addr/mask/prot/port): (172.16.0.0/255.255.255.0/0/0)

! IKE Phase 1 uses UDP port 500 to negotiate and set up the IKE Phase 1
! tunnel
   current_peer 43.0.0.2 port 500

   #pkts encaps: 29, #pkts encrypt: 29, #pkts digest: 29
   #pkts decaps: 29, #pkts decrypt: 29, #pkts verify: 29

! From R1's perspective, the local side is its G1/0, and R2 is at 43.0.0.2
    local crypto endpt.: 23.0.0.1, remote crypto endpt.: 43.0.0.2
    path mtu 1500, ip mtu 1500, ip mtu idb GigabitEthernet1/0

! An SPI is a Security Parameter Index.  It is a fancy way of tracking
! a specific Security Association (SA) between itself and a peer.
! Think of it as a serial number (unique) for each SA.
    current outbound spi: 0x48A3CF57(1218694999)
! PFS stands for Perfect Forward Secrecy, and it is the ability for IKE
! Phase 2 to run the DH algorithm again, instead of using the keys
! generated during the DH from IKE Phase 1.   This feature is off by
! default for most platforms.
    PFS (Y/N): N, DH group: none

! The IPsec or IKE Phase 2 is really two tunnels.   There is one for
! traffic from R1 to R2.   There is another from R2 to R1.   They have
! different SPIs, but together, these two unidirectional tunnels make up
! the "IPsec" tunnel.
! Encapsulating Security Payload (ESP) is the primary method used by IPsec.
! The other option is to use Authentication Header (AH), but it doesn't
! have the ability to encrypt, and isn't often used for that reason.   AH
! also breaks when going through Network Address Translation (NAT).
! Here is the inbound SA used by R1 to receive encrypted user packets from
! R2.
    inbound esp sas:
     spi: 0xE732E3A0(3878871968)
       transform: esp-256-aes esp-sha-hmac ,
       in use settings ={Tunnel, }
       conn id: 1, flow_id: SW:1, sibling_flags 80000046, crypto map:
        SDM_CMAP_1
       sa timing: remaining key lifetime (k/sec): (4388080/3230)
       IV size: 16 bytes
! Here is the built in anti-replay support
       replay detection support: Y
       Status: ACTIVE
! We aren't using AH, so there are no Security Associations (SAs) for AH.
    inbound ah sas:

! Here is the Outbound SA used by R1 to send encrypted user packets to R2.
       conn id: 2, flow_id: SW:2, sibling_flags 80000046, crypto map: SDM_
        CMAP_1
       sa timing: remaining key lifetime (k/sec): (4388079/3230)
       IV size: 16 bytes
       replay detection support: Y
       Status: ACTIVE

   outbound ah sas:

! Another way of seeing that the encryption and decryption is working.
```

```
R1# show crypto engine connections active
Crypto Engine Connections

   ID  Type    Algorithm           Encrypt  Decrypt IP-Address
    1  IPsec   AES256+SHA                 0       29 23.0.0.1
    2  IPsec   AES256+SHA                29        0 23.0.0.1
 1001  IKE     MD5+AES                    0        0 23.0.0.1
```

**Verifying IPsec**

R1# **show crypto isakmp policy =** Verify the IKE Phase 1 policies in place on the route
R1# **show crypto map** = the details of the crypto map
R1# **show crypto isakmp sa *[detail]*** = the details for the IKE Phase 1 tunnel that is in place
R1# **show crypto ipsec sa** = the details for the IKE Phase 2 tunnels that are in place
R1# **show crypto engine connections active** = seeing that the encryption and decryption is working

# Chapter 7: Implementing IPsec Site-to-Site VPNs

**Planning and Preparing an IPsec Site-to-Site VPN**

**IPsec** uses **two methods** for **encryption**: **tunnel** and **transport mode**.
If IPsec **tunnel mode** is used, the **IP header** and the **payload** are **encrypted**.
When **transport mode** is used, only the **packet payload** is **encrypted**.

**Protocols That May Be Required for IPsec:**

- UDP port 500 (IKEv1 Phase 1)
- UDP port 4500 NAT-T (NAT Traversal)
- Layer 4 Protocol 50 ESP
- Layer 4 protocol 51 AH

**Planning IKEv1 Phase 1**

| Function | Strong Method | Stronger Method |
|---|---|---|
| **Hash** | MD5, 128-bit | SHA1, 160-bit |
| **Authentication** | Pre-shared Key (PSK) | RSA-Sigs (digital signatures) |
| **Group # for DH key exchange** | 1,2,5 | IKE Groups 14 and 24 use 2048-bit DH. Groups 15 and 16 use 3072-bit and 4096-bit DH. Groups 19 and 20 support the 256-bit and 384-bit bit ECDH groups respectively. |
| **Lifetime** | 86400 seconds (1 day, default) | Shorter than 1 day, 3600 |
| **Encryption** | 3DES | AES-128 (or 192, or 256) |

Configure above with **crypto isakmp policy** command.

**Planning IKEv1 Phase 2**

| Item to Plan | Implemented By | Notes |
|---|---|---|
| **Peer IP Address** | **Crypto map** | **VPN peer** global IP address |
| **Traffic to encrypt** | **Crypto ACL,**<br>referred to in the Crypto map | **Extended ACL** that is not applied to an interface but is referenced in the crypto map. This should ***only*** **reference outbound** (egress) **traffic**, which should be **protected** by **IPsec**. Traffic not matching the crypto ACL will not be encrypted, but will be sent as a normal packet. |
| **Encryption method** | **Transform set,**<br>referred to in the crypto map | **DES, 3DES, AES** are all options.<br>IKEv1 Phase 2 does **not need to be the same method** as Phase 1. The method **does need to match** the peer's policy (transform sets) for Phase 2. |
| **Hashing (HMAC) method** | **Transform set,**<br>referred to in the crypto map | **MD5** and **SHA HMACs** may be used, and **need to match** the **Phase 2 policy** of the **peer**. |
| **Lifetime** | Global configuration command:<br>**crypto ipsec security-association lifetime** | **Lifetime** for Phase 2 **should match** between the **peers**. If both use the default lifetime (by not specifying a lifetime), both peers would have compatible lifetime policies. The lifetime can be specified as number of seconds or number of kilobytes. |
| **Perfect Forward Secrecy (PFS) (run DH again or not)** | **Crypto map** | DH is run during IKEv1 Phase 1, and Phase 2 reuses that same keying material that was generated. If you want Phase 2 to rerun the DH, it is called Perfect Forward Secrecy (**PFS**), and you must choose a DH group number 1, 2, or 5 for Phase 2 to use. |
| **Which interface used to peer with the other VPN device** | **Crypto map**<br>Applied to the outbound interface | From a routing perspective, this is the interface of a VPN peer that is closest to the other peer, where outbound IPsec packets are leaving the router and inbound IPsec packets are coming into the router. |

Note, that a show running-config, would only show configured items in the ! policy if they were **different from the default**.

**Implementing and Verifying an IPsec Site-to-Site VPN in Cisco IOS Devices**

Because we chose to implement **RSA-Signatures** for this customer, we want to implement **NTP** as one of our first steps. This is because when exchanging **certificates** during IKEv1 Phase 1, if R1 thinks the year is 2040, and the certificate it just received from R2 is listed as being valid from 2012 through 2016, R1 will **reject** the certificate as not being valid, and IKEv1 Phase 1 will not end well. (If IKEv1 Phase 1 does not work, IKEv1 Phase 2 does not have a chance either.)

**Verifying NTP Status**

```
R1# show ntp status
Clock is unsynchronized, stratum 16, no reference clock
nominal freq is 250.0000 Hz, actual freq is 249.9999 Hz, precision is 2**24
reference time is D2C15194.71E5E637 (14:11:32.444 UTC Wed Jan 18 2012)
clock offset is 0.0000 msec, root delay is 0.00 msec
root dispersion is 0.00 msec, peer dispersion is 0.00 msec
loopfilter state is 'CTRL' (Normal Controlled Loop), drift is 0.000000085 s/s
system poll interval is 64, last update was 1518 sec ago.

! Note the above indicates the time isn't synchronized.
! We can check to see if the router has the NTP server configured with the
! following:


R1# show ntp association

  address         ref clock       st   when   poll reach  delay  offset   disp
 ~3.3.3.3         .INIT.          16      -     64     0  0.000   0.000 16000.
 * sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured

! Based on this output, we know that it has information to use the 3.3.3.3
! server
! It may take anywhere from 5 to 15 minutes for
! the synchronization to happen.     After verifying the configuration, and
! waiting about 5 minutes, we can then issue the verification commands
! again and see that the synchronization is complete.
R1# show ntp status
Clock is synchronized, stratum 3, reference is 3.3.3.3
nominal freq is 250.0000 Hz, actual freq is 249.9999 Hz, precision is 2**24
reference time is D2C15854.6F453DAE (14:40:20.434 UTC Wed Jan 18 2012)
clock offset is 0.0029 msec, root delay is 0.01 msec
root dispersion is 0.95 msec, peer dispersion is 0.06 msec
loopfilter state is 'CTRL' (Normal Controlled Loop), drift is 0.000000097 s/s
system poll interval is 64, last update was 251 sec ago.
```

Our next task, in preparation for the IPsec, is to **generate key pairs** on R1 and R2, configure them to use a **CA**, have them **authenticate** the **CA** (get the root certificate), and then **enroll** with the **CA** (request their own identity certificates).

```
! Specify the domain-name that will be included with the key pair you
! are about to generate
! Note: if you have already created a key-pair to be used with SSH
! you don't need to create a separate key-pair.  You can use the same
! key pair for both purposes if desired.
R1(config)# ip domain name cisco.com
R1(config)# crypto key generate rsa
The name for the keys will be: R1.cisco.com
Choose the size of the key modulus in the range of 360 to 2048 for your
  General Purpose Keys. Choosing a key modulus greater than 512 may take
  a few minutes.
! The larger the key the better.  Using a minimum length of 1024 is a best
! practice
How many bits in the modulus [512]: 1024
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]

! Specify the CA that you would like to use, and the URL to be used to
! reach that CA
R1(config)# crypto pki trustpoint CA
R1(ca-trustpoint)# enrollment URL http://3.3.3.3
R1(ca-trustpoint)# exit

! Request the root certificate through "authenticating" the CA
R1(config)# crypto pki authenticate CA
Certificate has the following attributes:
       Fingerprint MD5: B1AF5247 21F35FE3 0200F345 7C20FBA0
       Fingerprint SHA1: F5BB33E3 1CB5D633 0DF720DF 8C72CD48 E744CF5B

% Do you accept this certificate? [yes/no]: yes
Trustpoint CA certificate accepted.

! Request an Identity certificate for this router, via SCEP and the
! "enroll" option
R1(config)# crypto pki enroll CA
%
% Start certificate enrollment ..
% Create a challenge password. You will need to verbally provide this
   password to the CA Administrator in order to revoke your certificate.
   For security reasons your password will not be saved in the configuration.
   Please make a note of it.

! Specifying the challenge password that can be used in the event you need to
! ask the CA to revoke this certificate in the future
Re-enter password: SuperSecret!23

% The subject name in the certificate will include: R1.cisco.com

! The next 2 items are optional elements that may be included in the certificate
% Include the router serial number in the subject name? [yes/no]: no
% Include an IP address in the subject name? [no]: no
Request certificate from CA? [yes/no]: yes
% Certificate request sent to Certificate Authority
% The 'show crypto pki certificate verbose CA' command will show the fingerprint.

CRYPTO_PKI:  Certificate Request Fingerprint MD5: E8E01D26 862C811C 32CB3FCF 858BAF5F
CRYPTO_PKI:  Certificate Request Fingerprint SHA1: E3133B07 07DEA5FD BC6A1D64 DBC9F71F
  3CACA767
%PKI-6-CERTRET: Certificate received from Certificate Authority

! We would repeat this process on the other router, R2
```

**CLI Implementation of the Crypto Policy for R1**

```
R1(config)# crypto isakmp policy 1
R1(config-isakmp)# encr aes 256
R1(config-isakmp)# group 5
R1(config-isakmp)# lifetime 3600
R1(config-isakmp)# authentication rsa-sig
R1(config-isakmp)# hash sha

! To verify the configuration:
R1# show crypto isakmp policy

Global IKEv1 policy
Protection suite of priority 1
        encryption algorithm:   AES - Advanced Encryption Standard (256 bit keys).
        hash algorithm:         Secure Hash Standard
        authentication method:  Rivest-Shamir-Adleman Signature
        Diffie-Hellman group:   #5 (1536 bit)
        lifetime:               3600 seconds, no volume limit
! Note, that a show running-config, would only show configured items in the
! policy if they were different from the default.   Here is a snippet from
! the show run:

crypto isakmp policy 1
 encr aes 256
 group 5
 lifetime 3600

! Because the authentication and hash are using the defaults, they are not
! shown even though we put them in the configuration.  (interesting to know)

! We won't need a pre-shared key, because we are using digital signatures/
! certificates for the IKEv1 Phase 1 authentication.

! Next we can create our transform-set, and crypto ACL, which will be
! placed inside the crypto map.  The crypto map will be applied to the
! interface of the router.


! Transform set details the encryption and HMAC to use
R1(config)# crypto ipsec transform-set MYSET esp-aes esp-sha-hmac
R1(cfg-crypto-trans)# exit

! Crypto ACL identifies which traffic (outbound) should be encrypted
R1(config)# access-list 100 permit ip 172.16.0.0 0.0.255.255 192.168.0.0 0.0.0.255

! The crypto map contains the if/then statement to decide to encrypt or
! not to encrypt a packet on its way out
R1(config)# crypto map MYMAP 1 ipsec-isakmp
R1(config-crypto-map)# match address 100
R1(config-crypto-map)# set peer 209.165.201.1
R1(config-crypto-map)# set transform-set MYSET

! Here is the PFS part that we are adding manually, as the wizard didn't
! support this feature
R1(config-crypto-map)# set pfs group2
R1(config-crypto-map)# exit
```

```
! Applying the crypto map to the interface is what allows the entire IPsec
! function to be triggered.   That is why it is important that the router
! has at least a default route (if not a more specific route) out of this
! interface to reach the remote network for which the router is providing
! IPsec support.
! When the router considers forwarding traffic out the interface, that is
! what triggers the decision to encrypt or not.   If the traffic matches
! the crypto ACL in the crypto map, the router will encrypt the original
! packet, encapsulate the encrypted packet into a new packet with ESP as
! the L4 header, and the peer's global IP address as the new L3 header.
! If no IPsec SA (tunnel) is in place yet, this will also trigger the
! negotiations to build the tunnel, including the IKEv1 Phase 1 if it is not
! already in place.
R1(config)# interface GigabitEthernet1/0
R1(config-if)# crypto map MYMAP
R1(config-if)# exit
```

After the appropriate compatible configuration has been placed on R2, we should be able to encrypt traffic between the two networks using IPsec.

**Troubleshooting IPsec Site-to-Site VPNs in Cisco IOS**

**R1# show crypto isakmp policy** = to verify the configuration
**R1# show crypto map**
**R1# debug crypto isakmp**
**R1# show crypto isakmp sa** = IPSec Phase 1
**R2# show crypto ipsec sa** = IPSec Phase 2

We want to see a state of **QM_IDLE**, meaning the IKEv1 Phase 1 is up.

**R2# show crypto engine connections active**

```
! This verifies the IKEv1 Phase 1 policy or policies in place.   The lower
! the number of the policy, the higher its priority.

R1# show crypto isakmp policy

Global IKEv1 policy
Protection suite of priority 1
        encryption algorithm:   AES - Advanced Encryption Standard (256 bit keys).
        hash algorithm:         Secure Hash Standard
        authentication method:  Rivest-Shamir-Adleman Signature
        Diffie-Hellman group:   #5 (1536 bit)
        lifetime:               3600 seconds, no volume limit

! Next is my favorite command, as it shows virtually all of the rest of the
! config including the transform set and crypto ACLs involved, and where
! the crypto map is applied
```

```
R1# show crypto map
Crypto Map "MYMAP" 1 ipsec-isakmp
        Peer = 209.165.201.1
        Extended IP access list 100
            access-list 100 permit ip 172.16.0.0 0.0.255.255 192.168.0.0
              0.0.0.255
        Current peer: 209.165.201.1
        Security association lifetime: 4608000 kilobytes/3600 seconds
        Responder-Only (Y/N): N
        PFS (Y/N): Y
        DH group:  group2
        Transform sets={
                MYSET:  { esp-aes esp-sha-hmac  } ,
        }
        Interfaces using crypto map MYMAP:
                GigabitEthernet1/0
```

There are other **alternative site-to-site VPN** technologies:

- **Dynamic Multipoint VPN (DMVPN)**
- **FlexVPN**

**DMVPN** is a Cisco solution for deploying highly scalable IPsec site-to-site VPNs. DMVPN uses a centralized architecture to enable the network administrator to deploy granular access controls. It enables branch locations to communicate directly with each other over the Internet without requiring a permanent VPN connection between sites.

**FlexVPN** is a unified VPN solution that can be deployed over either public Internet connections or a private Multiprotocol Label Switching (MPLS) VPN network. FlexVPN is designed for the concentration of both site-to-site and remote access VPNs. One FlexVPN deployment can accept both types of connection requests at the same time. It uses dynamic routing protocols for redundancy and path/head-end selection.

# Chapter 8: Implementing SSL VPNs Using Cisco ASA

Comparison of IPsec versus SSL

| | SSL | IPsec |
|---|---|---|
| **Applications** | Web-based applications, file sharing, e-mail (if not using full client). With the full Cisco AnyConnect Secure Mobility Client, all IP-based applications, similar to IPsec, are available. | All IP-based applications are available to the user. The experience is like being on the local network. |
| **Ease of Use** | Very high | Moderate. Can be challenging for nontechnical users, and deployment is more time-consuming. |
| **Overall Security** | Moderate. Any devices can initially connect | Strong. Only specific devices with specific configurations, such as a VPN client, can connect. |

**SSL and TLS Protocol Framework**

**TLS** and its predecessor **SSL** are **cryptographic protocols** that provide **secure** transactions on the Internet for things such as **e-mail**, **web browsing**, **instant messaging**, and so on. **Both** of these protocols provide **confidentiality**, **integrity**, and **authentication** services. These protocols are considered to be operating at the session layer and higher in the OSI logical model. They both can use the ***public key infrastructure (PKI)*** and **digital certificates** for authentication of the VPN endpoints and for establishing encryption keys that may be used. Similar to IPsec, these protocols use **symmetric** algorithms for bulk encryption, and **asymmetric** algorithms are used for the authentication and for the exchange of keys.

**SSL 3.0** served as the basis of TLS 1.0. Some implementations include the ability to switch to the other protocol if necessary, especially in the case of TLS, which can switch over to SSL if the client connection requires it. Fortunately for us, all of this is done behind the scenes and is transparent to the end user.

Comparison between SSL and TLS

| SSL | TLS |
|---|---|
| Developed by Netscape in the 1990s | Standard developed by the *Internet Engineering Task Force (IETF)* |
| Starts with a secured channel and continues directly to security negations on a dedicated port | Can start with unsecured communications and dynamically switch to a secured channel based on the negotiation with the other side |
| Widely supported on client-side applications | Supported and implemented more on servers, compared to end-user devices |
| More weaknesses identified | Stronger implementation because of the standards process |

**The Play by Play of SSL for VPNs**

SSL VPNs can provide **security**. SSL is used for most online transactions that require security. If a customer was opening up a browser and going to connect to a banking server, or some other type of SSL device, here is what we would expect:

- The **client initiates** a **connection** to the server using the **destination IP** address of the server and the destination TCP port **443**. The source IP address is the IP address of the client, and the source port is some **random unused port** number on the client machine greater than 1023.
- There is the **standard three-way handshake**, which is the normal process for TCP in establishing sessions.
- After the client initiates its request for the connection, the server responds, providing its **digital certificate**, which contains the server's **public key**.
- The client, upon receiving this digital certificate, has a big decision to make. That decision is whether to believe the credibility of the digital certificate that it just received from the SSL VPN server. This is where PKI comes into play. If the digital certificate is signed by a ***certificate authority*** *(CA)* that the client's browser trusts, and the validity dates for that certificate cause the client to believe that the time has not run out on that certificate, and if the client is checking a *certificate revocation list (CRL)* (and the serial number for the certificate is not on the CRL), the client can trust the certificate and extract the public key of the server out of the certificate.
- The client then generates a **shared secret** that it would like to use for **encryption** back and forth between itself and the server. The problem is now how to get this shared secret that the client wants to use sent securely over to the server? The answer is the client uses the public key of the server to encrypt the shared secret and send the encrypted secret to the server.
- The server **decrypts** the sent **symmetric** key using the server's own private key, and now both devices in the session know and can use the shared secret key.
- The key is then used to **encrypt** the **SSL session**. Because SSL VPN provides network access to remote users, you have to consider the placement of the VPN termination devices. Before implementing the SSL VPN feature, ask the following questions:
    - Should the Cisco ASA terminating the VPN be placed behind another firewall? If so, what ports should be opened in that firewall?
    - Should the decrypted traffic be passed through another set of firewalls? If so, what ports should be allowed in those firewalls?
    - Are there any proxy servers between the client and the Cisco ASAs?

**The Play by Play of SSL for VPNs Summary**

- The client initiates a connection to the server on port 443 and uses internally a port > 1023
- There is the standard three-way handshake
- The server responds, providing its digital certificate, which contains the server's public key
- Validate certificate information
- Client sends a shared secret to the server encrypting the key with the server's public key
- Server decrypts the shared secret using its private key
- The key is now used to encrypt data over SSL

**SSL VPN Flavors**

There are **three different types** of **SSL VPN** access methods.

| | Clientless SSL VPN | Clientless SSL VPN with Plug-Ins for Some Port Forwarding | Full Cisco AnyConnect Secure Mobility Client SSL VPN Client |
|---|---|---|---|
| Other names | Web VPN. | Thin client. | Full SSL client. |
| Installed software on client | No client required. | Small applets and/or configuration required. | Full install of Cisco AnyConnect Secure Mobility Client required, but may be installed by initially connecting via the clientless option and securely installing it that way. |
| User experience | Feels like accessing resources (that are on the corporate network) through a specific browser window or hyperlink. | Some applications can be run locally with output redirected through the VPN. Includes the features of the clientless VPN to the left. | Full access to the corporate network. The local computer acts and feels like it is a full participant on the corporate network. |
| Servers that can be used | IOS with the correct software, and ASA with the correct licenses. | IOS with the correct software, and ASA with the correct licenses. | IOS with the correct software, and ASA with the correct licenses. |
| How the user looks from the corporate network | Traffic is proxied (*Port Address Translation [PAT]*) by the SSL server, as the users' packets enter the corporate network. | Traffic is proxied (PAT) by the SSL server as the users' packets enter the corporate network. | Clients are assigned their own virtual IP address to use while accessing the corporate network. Traffic is forwarded from the given IP address of the client into the corporate network. |
| Clients supported | Most SSL-capable computers. | Computers that support SSL and Java. | Most computers that support SSL. |

**Configuring Clientless SSL VPNs on ASA**

```
!  specifies  the creation of a local group
asa1(config)# group-policy NY-Group-Policy internal

! Specifies that it's using its own self signed certificate
! and enabling SSL VPN on the outside interface
asa1(config)# ssl trust-point ASDM_TrustPoint0 outside
asa1(config)# webvpn
asa1(config-webvpn)# enable outside
!  specifies the attributes for this local group, including the bookmarks
asa1(config-webvpn)# group-policy NY-Group-Policy attributes
asa1(config-group-policy)# vpn-tunnel-protocol ssl-clientless
asa1(config-group-policy)# webvpn
asa1(config-group-webvpn)# url-list value IntranetSite
asa1(config-group-webvpn)# exit
asa1(config-group-policy)# exit

!  specifies  a tunnel group for remote access, compared to site to site
asa1(config)# tunnel-group NY-connection-profile type remote-access

!  defines the attributes for this connection profile, including the group
!  policy to be used
asa1(config)# tunnel-group NY-connection-profile general-attributes
asa1(config-tunnel-general)# default-group-policy NY-Group-Policy

!  defines the URL that when connected will trigger what profile to use,
!  and that in turn controls what group profile should be applied

asa1(config-tunnel-general)# tunnel-group NY-connection-profile webvpn-attributes
asa1(config-tunnel-webvpn)# group-alias newyork enable
asa1(config-tunnel-webvpn)# group-url https://209.165.202.129/newyork enable
! The asa uses the outside IP address.
```

**Configuring an SSL Cisco AnyConnect Secure Mobility Client VPN**

```
! For this example, to avoid the wrapping of some of the longer commands
! the firewall prompt has been omitted from the output below
! For use with the nat exemption, at the end of the config
object network NETWORK_OBJ_10.1.1.0_25
subnet 10.1.1.0 255.255.255.128

! Create the pool for the IP addresses it will be handing out
ip local pool anyconnectPool 10.4.4.1-10.0.0.100 mask 255.255.255.0

! Creates an internal group based on the name below
group-policy GroupPolicy_SSL_AnyConnectinternal

! Specifies the attributes of this group, that the protocol for transport
! will be SSL, specifies the DNS and Domain name to hand out.
group-policy GroupPolicy_SSL_AnyConnect attributes
  vpn-tunnel-protocol ssl-client
  dns-server value 10.1.1.23
  wins-server none
  default-domain value example.org
exit

! Specifies that SSL is enabled, and which packages from flash are available
! for client images
webvpn
 enable outside
 anyconnect image disk0:/anyconnect-macosx-4.0-k9.pkg 1

! Enables Anyconnect, and provides the group list (connection profile list)
! to users who are logging on, so they can initially select their "group"
  anyconnect enable
 tunnel-group-list enable

! Creates a tunnel group, and specifies the type of tunnel group it is
tunnel-group SSL_AnyConnect type remote-access

! Specifies what group policy should be used by this tunnel group,
! and what pool of IP addresses should be used for the users
tunnel-group SSL_AnyConnect general-attributes
  default-group-policy GroupPolicy_SSL_AnyConnect
  address-pool anyconnectPool

! Enables this URL (Alias) to be used to access the server
tunnel-group SSL_AnyConnectwebvpn-attributes
  group-alias SSL_AnyConnectenable

! provides the exception for NAT (if present) for VPN traffic from the inside
! network if it is going to the address range used by the AnyConnect clients.
! Note: the following is a single line that is shown as wrapped because
! is longer than the width of this page.
nat (inside,outside) 3 source static inside interface destination static
NETWORK_OBJ_10.1.1.0_25 NETWORK_OBJ_10.1.1.0_25 no-proxy-arp route-lookup
```

**Groups, Connection Profiles, and Defaults**
The connection profiles are responsible for the initial connection of the user.

**Split Tunneling**
Without split tunneling, all IP traffic leaving the client's machine goes through the tunnel to the ASA. A split tunnel addresses this issue by sending traffic down the VPN only if it is destined for specific networks located at the headquarter site.

**Troubleshooting SSL Negotiations**

- **Step 1:** Verify that the user's computer can **ping** the **Cisco ASA's outside IP** address
- **Step 2:** If the user's workstation can ping the address, issue the **show running all** | **include ssl** command on the Cisco ASA and verify that **SSL encryption** is **configured**.
- **Step 3:** If SSL encryption is properly configured, use an **external sniffer** to verify whether the **TCP three-way handshake** is **successful**

**Troubleshooting AnyConnect Client Issues**

```
ASA1# debug webvpn svc
CSTP state = HEADER_PROCESSING

http_parse_cstp_method()
...input: 'CONNECT /CSCOSSLC/tunnel HTTP/1.1'
webvpn_cstp_parse_request_field()
...input: 'Host: 209.165.200.226'
<snip>
Processing CSTP header line: 'X-DTLS-CipherSuite: AES256-SHA:AES128-SHA:DES-CBC3-SHA:
  DES-CBC-SHA'
Validating address: 0.0.0.0
CSTP state = WAIT_FOR_ADDRESS
webvpn_cstp_accept_address: 0.0.0.0/0.0.0.0
webvpn_cstp_accept_address: no address?!?
CSTP state = HAVE_ADDRESS
No assigned address
webvpn_cstp_send_error: 503 Service Unavailable
CSTP state = ERROR
```

```
ASA1(config)# logging on
ASA1(config)# logging class svc buffered debugging
ASA1(config)# exit
ASA1# show logging
%ASA-3-722020: TunnelGroup <SSLVPNTunnel> GroupPolicy <AnyConnectGroupPolicy> User
  <sslvpnuser> IP <209.165.200.231> No address available for SVC connection
```

**Traffic-Specific Issues**

- Routing issues behind the ASA—internal network unable to route packets back to the assigned IP addresses and VPN clients
- Access control lists blocking traffic
- Network Address Translation not being bypassed for VPN traffic

# Chapter 9: Securing Layer 2 Technologies

**Port Security**

**Attack type:** CAM/MAC table overflow
**Mitigation method:** Port Security

CAM overflow attacks are caused by a client connected to a switch sending out thousands of frames, each with a different MAC address. The the MAC address table storage depletes and floods all new traffic out every port within that VLAN. A malicious user could use this method to sniff traffic not destined to him.

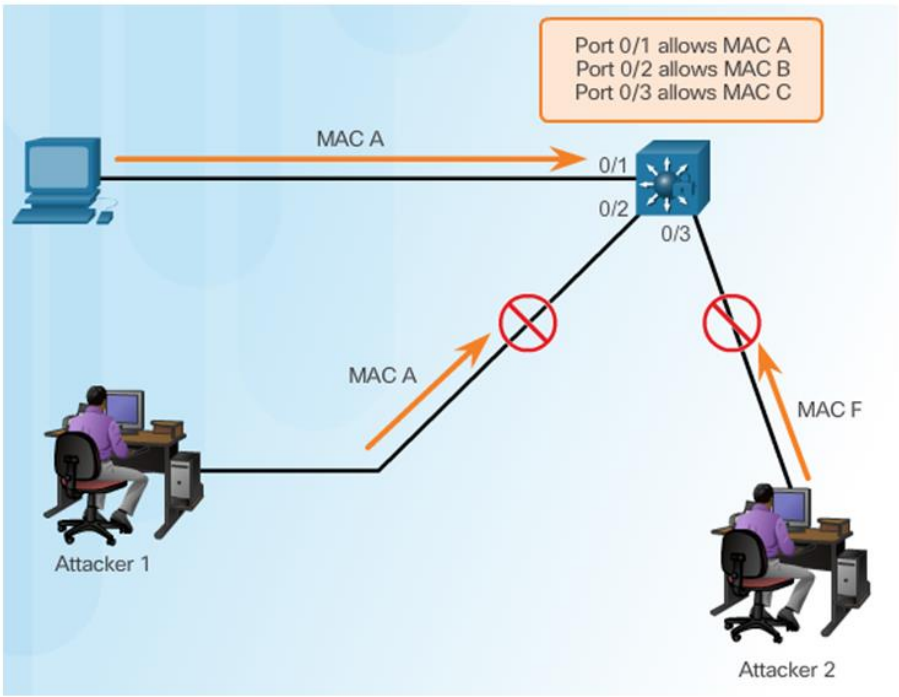
Port Security is used by access layer switches to mitigate an administratively defined limit on the amount of MAC addresses which can be sent to a switch port.

Port security is enabled on an interface by interface basis and is not globally enabled. To enable port security, enter **switchport port-security** on the interface To limit the amount of MAC addresses through the port: **switchport port-security maximum *<num>*** *(Default is one address)* The default violation of a port is to shut the port down, which requires an administrator to shut the port down, and bring it back online with no shutdown. There is three violation modes; Shutdown, Protect and Restrict.

Shutdown was already covered, protect allows the original mac address (up to its defined limit) to be transmitted without restriction. Restrict is the same as protect, with only change is that a SNMP trap is sent each time a violation is detected, whereas protect will not alert.

If you don't want to statically assign which MAC addresses are allowed on a switchport, you can enable the switch to dynamically learn the maximum addresses with *sticky mac addresses*.
This is configured with **switchport mac-address sticky.**

## Countermeasure for CAM Table Attacks



```
SW2(config-if)# interface fa 0/2

! Enable the feature per interface
SW2(config-if)# switchport port-security

! Set the maximum to desired number.  Default is 1. If we administratively
! set the maximum to 1, the command won't show in the running configuration
! because the configuration matches the default value. It is handy to know
! this behavior, so you won't be surprised by what may seem to be a missing
! part of your configuration.
SW2(config-if)# switchport port-security maximum 5

! Set the violation action.  Default is err-disable. Protect will simply
! not allow
! frames from MAC addresses above the maximum.
SW2(config-if)# switchport port-security violation protect
! This will cause the dynamic mac addresses to be placed into running
! -config to save them to startup config, use copy run start
SW2(config-if)# switchport port-security mac-address sticky

! To verify settings, use this command
SW2# show port-security
Secure Port  MaxSecureAddr  CurrentAddr  SecurityViolation  Security Action
                (Count)       (Count)          (Count)
---------------------------------------------------------------------------
      Fa0/2              5            1                  0          Protect
---------------------------------------------------------------------------
Total Addresses in System (excluding one mac per port)     : 0
Max Addresses limit in System (excluding one mac per port) : 6144

! This can also provide additional information about port security:

SW2# show port-security interface fa0/2
Port Security              : Enabled
Port Status                : Secure-up
Violation Mode             : Protect
Aging Time                 : 0 mins
Aging Type                 : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses      : 5
Total MAC Addresses        : 1
Configured MAC Addresses   : 0
Sticky MAC Addresses       : 1
Last Source Address:Vlan   : 0000.2222.2222:10
Security Violation Count   : 0
```

**DHCP Snooping**

**Attack type:** Rouge DHCP Server
**Mitigation method:** DHCP Snooping

An attacker could set up their own DHCP server and start sending clients on the network segment **invalid network information** causing a DoS situation, or could route all traffic to itself and perform a man in the middle attack.

DHCP Snooping acts like a firewall for DHCP requests to verify legitimate DHCP messages and to filter non-legitimate traffic.

**DHCP Snooping:**

- Validates DHCP messages that are received from untrusted sources, and filters invalid messages.
- Rate limits DHCP traffic from trusted and untrusted sources
- Builds and maintains the DHCP Snooping binding database, which contains information about untrusted hosts with leased IP addresses
- Utilizes the DHCP snooping binding database to validate subsequent requests from untrusted hosts.

**All ports are untrusted** by **default** and all trusted ports must be configured individually. It is important to note that **DHCP Snooping** is **enabled globally**, and will ignore all DHCP requests if the trusted ports are not configured first.

Enabling a trusted interface to the DHCP Snooping binding database is done by the adding 'ip dhcp snooping trust' to the interface.

**The following steps are required to implement DHCP snooping on your network:**

**Step 1:** Define and configure the DHCP server. Configuration of this step does not take place on the switch or router and is beyond the scope of this course.

**Step 2:** Enable DHCP snooping on at least one VLAN. By default, DHCP snooping is inactive on all VLANs.

**Step3:** Ensure that DHCP server is connected through a trusted interface. By default, the trust state of all interfaces is untrusted.

**Step 4:** Configure the DHCP snooping database agent. This step ensures that database entries are restored after a restart or switchover.

**Step 5:** Enable DHCP snooping globally.

```
! Enable DHCP Snooping Globally
sw2(config)# ip dhcp snooping
! Enable DHCP Snooping on VLAN 10
sw2(config)# ip dhcp snooping vlan 10
! Configure Interface Fa1/0/24 as a Trusted interface
sw2(config)# interface fa1/0/24
sw2(config-if)# ip dhcp snooping trust
! Configure the DHCP snooping database agent to store the bindings at a given location
sw2(config)# ip dhcp snooping database tftp://10.1.1.1/directory/file
sw2(config)# exit
sw2#
! Verify DHCP Snooping Configuration
sw2# show ip dhcp snooping
Switch DHCP snooping is enabled
DHCP snooping is configured on following VLANs:
10
DHCP snooping is operational on following VLANs:
none
DHCP snooping is configured on the following L3 Interfaces:

Insertion of option 82 is enabled
   circuit-id default format: vlan-mod-port
   remote-id: 000f.90df.3400 (MAC)
Option 82 on untrusted port is not allowed
Verification of hwaddr field is enabled
Verification of giaddr field is enabled
DHCP snooping trust/rate is configured on the following Interfaces:

Interface                  Trusted    Allow option    Rate limit (pps)
-----------------------    -------    ------------    ----------------
FastEthernet1/0/24         yes        yes             unlimited
  Custom circuit-ids:
```

**ARP Inspection**

**Attack type:** Man in the Middle (MITM)
**Mitigation method:** ARP Inspection

**ARP** is a **layer 2** protocol used to **discover** the **layer 2 address** of a host and **map** it to its **layer 3 address**. This can be exploited by a malicious user by sending out ARP messages as the layer 3 IP gateway, but using their own MAC address in the frames.

In a typical network configuration, you configure all switch ports connected to host ports as untrusted and configure all switch ports connected to switches as trusted.
You configure the trust setting by using the **ip arp inspection trust** interface configuration command.

```
! Enable DAI on VLAN 10
sw2(config)# ip arp inspection vlan 10
sw2(config)# exit
! Verify DAI Configuration for VLAN 10
sw2# show ip arp inspection vlan 10

Source Mac Validation      : Disabled
Destination Mac Validation : Disabled
IP Address Validation      : Disabled

 Vlan     Configuration    Operation   ACL Match          Static ACL
 ----     -------------    ---------   ---------          ----------
   10     Enabled          Inactive

 Vlan     ACL Logging      DHCP Logging      Probe Logging
 ----     -----------      ------------      -------------
   10     Deny             Deny              Off
! Configure Interface Fa1/0/24 as a Trusted DAI Interface
sw2(config)# interface fa1/0/24
sw2(config-if)# ip arp inspection trust
sw2(config-if)# exit
sw2(config)# exit
sw2# show ip arp inspection interfaces
 Interface        Trust State     Rate (pps)     Burst Interval
 ---------------  -----------     ----------     --------------
 Fa1/0/1          Untrusted               15                  1
 Fa1/0/2          Untrusted               15                  1
! output removed for brevity
 Fa1/0/23         Untrusted               15                  1
 Fa1/0/24         Trusted               None                N/A
```

**Root Guard**

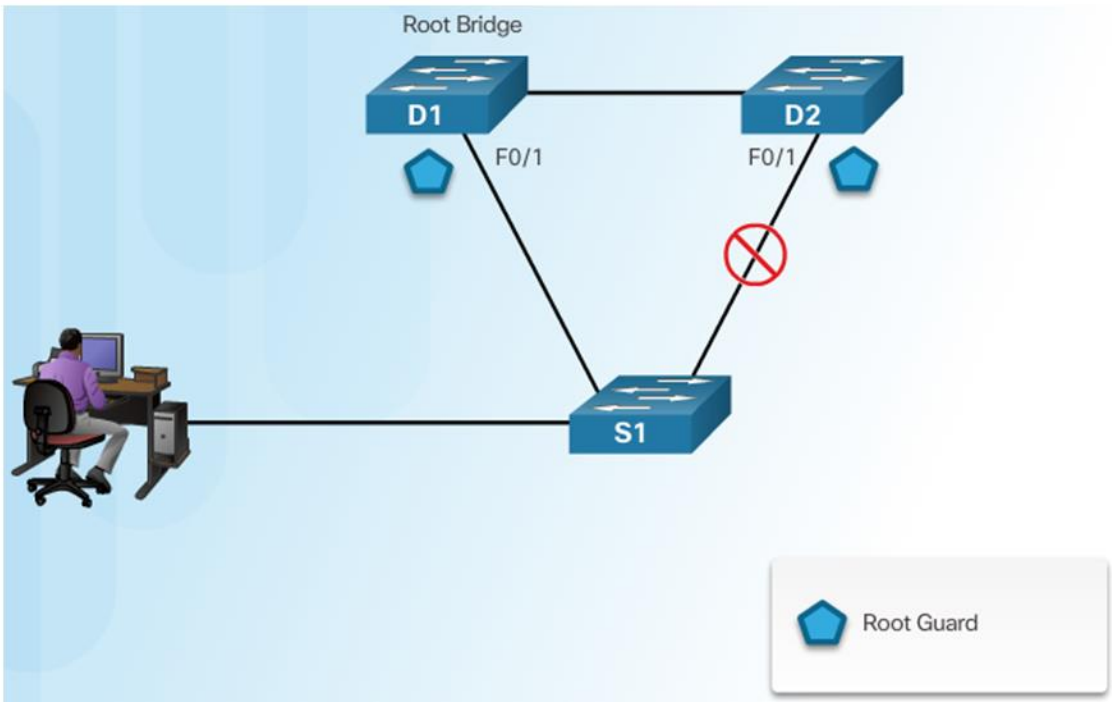
**Attack type:** Spanning Tree MITM
**Mitigation method:** Root Guard

If you connect your switch network to an untrusted network, you may not want them to be able to become the root bridge. You would enable root guard on the interface(s) towards the unmanaged switches to ensure this doesn't occur. This same method can be used on access ports where no switches should exist, stopping a user from advertising a lower root bridge ID than your current root bridge.

Root guard is enabled on an interface basis, and is done so by entering the interface you wish to protect, and configuring ***spanning-tree root guard***.

```
SW1(config)# interface fa 0/24
SW1(config-if)# spanning-tree guard root
%SPANTREE-2-ROOTGUARD_CONFIG_CHANGE: Root guard enabled on port
  FastEthernet0/24.
```

# Configuring Root Guard



**BPDU Guard**

**Attack type:** Spanning Tree MITM
**Mitigation method:** BPDU Guard

BPDU Guard stops ports from being allowed to send BPDU's into the switch network. A user could send a BPDU with the bridge ID lower than the current root bridge and impact the STP toplogy.

BPDU Guard will by default, shut the port down if a BPDU is detected on a port with BPDU guard enabled.
To enable BPDU Guard, enter **spanning-tree portfast bpduguard** in global configuration mode.
To have the port recover automatically, enter **errdisable recovery cause bpduguard** followed by **errdisable recovery interval *<time in seconds>***.

```
SW2(config-if)# interface fa 0/2
SW2(config-if)# spanning-tree bpduguard enable

! Verify the status of the switchport
SW2# show interface fa0/2 status

Port      Name          Status       Vlan       Duplex  Speed  Type
Fa0/2                   connected    10         a-full  a-100  10/100BaseTX
SW2#
```

```
SW2(config)# errdisable recovery cause bpduguard

! err-disabled ports will be brought back up after 30 seconds of no bpdu
! violations
SW2(config)# errdisable recovery interval 30

! You can also see the timeouts for the recovery

SW2# show errdisable recovery
ErrDisable Reason            Timer Status
-----------------            --------------
arp-inspection               Disabled
bpduguard                    Enabled
<snip>

Timer interval: 30 seconds
Interfaces that will be enabled at the next timeout:
```

**Switchport nonegotiate**

**Attack type:** Trunk port negotiation
**Mitigation method:** Disable trunk negotiations

A user connected to a switchport not assigned as an access port may negotiate it's port status to a trunk, sending information for all VLAN traffic to the malicious user

This is a simple one:

- Configure ports towards users as access ports with **switchport mode access.**
- Configure ports towards other switches with **switchport mode trunk.**
- Disable automatic trunk negotiation with **switchport nonegotiate**

**Best Practices**

- Change the **default VLAN**, and place **unused ports** into this **VLAN**
- Avoid using **VLAN 1** anywhere
- Configure **access ports** so users cannot negotiate trunks
- Disable **dynamic trunking protocol**
- **Limit** the amount of **MAC addresses** on an access port
- Enable **root guard** to stop other switches from becoming the **root**
- Turn off **CDP** on **access** ports
- Shut down **all ports** not in use

| Tool | Description |
|---|---|
| **Port Security** | Limits the number of MAC addresses to be learned on an access switch port. |
| **BPDU Guard** | If BPDUs show up where they should not, the switch protects itself. |
| **Root Guard** | Controls which ports are not allowed to become root ports to remote root switches. |
| **Dynamic ARP Inspection** | Prevents spoofing of Layer 2 information by hosts |
| **IP Source Guard** | Prevents spoofing of Layer 3 information by hosts |
| **802.1X** | Authenticates users before allowing their data frames into the network |
| **DHCP Snooping** | Prevents rouge DHCP servers from impacting the network |
| **Storm Control** | Limits the amount of broadcast or multicast traffic flowing through the switch |
| **Access Control Lists** | Traffic control to enforce policy |

# Chapter 10: Network Foundation Protection

**The Network Foundation Protection Framework**

- **Management plane:** Includes this protocols and traffic that an administrator uses between his workstation and the router or switch. Example using SSH to monitor or configure the router. If a failure occurs in the management plane, it may result in losing the ability to manage the network device altogether.

- **Control plane:** This includes protocols and traffic that the network devices use on their own without direct interaction from an administrator. An example is a routing protocol. A Routing protocol can dynamically learn and share routing information that the router can then use to maintain an updated routing table. If a failure occurs in the control plane, a router might lose the capability to share or correctly learn dynamic routing information.

- **Data plane:** This includes traffic that is being forwarded through the network. A failure of some component in the data plane results in the customer's traffic not being able to be forwarded.

**Interdependence**
Some interdependence exists between these three planes. For example, if the control plane fails, and the router does not know how to forward traffic, this scenario impacts the data plane because user traffic cannot be forwarded. Another example is a failure in the management plane that might allow an attacker to configure devices and as a result could cause both a control plane and data plane failure.

**Implementing NFP**
NFP is not a single feature but rather is a holistic approach that covers the three components (that is, planes) of the infrastructure, with recommendations about protecting each one using a suite of features.

**Components of a Threat Control and Mitigation Strategy**

| Plane | Security Measures | Protection Objectives |
|---|---|---|
| **Management Plane** | • Authentication, authorization, accounting (AAA)<br>• Authenticated Network Time Protocol (NTP)<br>• Secure Shell (SSH)<br>• Secure Sockets Layer/Transport Layer Security (SSL/TLS)<br>• Protected syslog<br>• Simple Network Management Protocol Version 3 (SNMPv3)<br>• Parser views | • Authenticate and authorize and administrators.<br>• Protect time synchronization by using authenticated NTP.<br>• Use only encrypted remote-access protocols such as SSH for CLI/TLS for GUI tools and use secure versions of SNMP. If plaintext tools are used (Syslog, Telnet) they should be protected by encryption such as IPsec. |
| **Control Plane** | • Control Plane Policing (CoPP) and Control Plane Protection (CPPr)<br>• Authenticated routing protocol updates | • Limit the damage an attacker can attempt to implement directly at one of the router's IP addresses (traffic addressed directly to the router, which the router must spend CPU resources to process).<br>• Routing protocol updates should be authenticated to remove the possibility of an attacker manipulating routing tables by putting a rogue router running the same routing protocol on your network. The attacker could be doing reconnaissance to learn the routes, or the attacker could be attempting to manipulate the resulting data plane by changing the routing on the network. |
| **Data Plane** | Access control lists (ACL)<br>Layer 2 controls, such as private VLANs, Spanning Tree Protocol guards.<br>IOS IPS, zone-based firewall | • ACLs, when applied as filters on interfaces, can control which traffic (transit traffic) is allowed on the data plane.<br>• At Layer 2, by protecting the infrastructure there, you can avoid a rogue switch from becoming the root of your spanning tree, which would affect the data plane at Layer 2.<br>• Firewall filtering and services can also control exactly what traffic is flowing through your network. An example is using an IOS zone-based firewall to implement policy about the data plane and what is allowed. |

**Understanding the Management Plane**

**Best Practices for Securing the Management Plane:**

- Enforce **password policy**, including features such as maximum number of login attempts and minimum password length
- Implement **role-based access control** (**RBAC**). This concept has been around for a long time in relation to groups; using Access Control Server (ACS) and CLI parser views.
- Use **AAA services**, and centrally manage those services on an ACS server
- Keep **accurate time** across all network devices using secure **Network Time Protocol** (**NTP**)
- Use **encrypted** and **authenticated** versions of **SNMP**, which includes Version 3
- **Control** which **IP** addresses are allowed to initiate management sessions with the network device
- Lock down **syslog**. Use separate VLAN for management.

- **Disable** any **unnecessary services**, especially those that use User Datagram Protocol (UDP):
    - TCP and UDP small services
    - Finger
    - BOOTP
    - DHCP
    - Maintenance Operation Protocol
    - DNS resolution
    - Packet assembler/disassembler
    - HTTP server and Secure HTTP (HTTPS) server
    - CDP
    - LLDP

**Understanding the Control Plane**

**Control plane** security is primarily **guarding against attacks** that might otherwise negatively impact the **CPU**, including routing updates (which are also processed by the CPU).

**Best Practices for Securing the Control Plane:**

- **CoPP Control plane policing:**
  You can configure this as a filter for any traffic destined to an IP address on the router itself. This is applied to a logical control plane interface (not directly to any Layer 3 interface) so that the policy can be applied globally to the router.

- **CPPr Control plane protection:**
  This allows for a more detailed classification of traffic (more than CoPP) that is going to use the CPU for handling.

- **Routing protocol authentication:**
  Using CoPP or CPPr, you can specify which types of management traffic are acceptable at which levels. Traffic that exceeds the thresholds can be safely dropped if it is not from one of your specific management stations.

**Understanding the Data Plane**

For the data plane, this discussion concerns traffic that is going **through** your **network device** rather than to a network device.

**Protecting the Data Plane:**

- **ACLs used for filtering**
- **IOS firewall support**
- **IOS IPS:** software implementation of an intrusion prevention system (IPS)
- **TCP Intercept:** enables the router to look at the number of half-formed sessions that are in place and intervene on behalf of the destination device
- **Unicast Reverse Path Forwarding**: When this feature is enabled on an interface, as packets enter that interface the router spends an extra moment considering the source address of the packet.

**Best Practices for Protecting the Data Plane:**

- Block unwanted traffic at the router: placing the ACL closer to the source saves resources
- Reduce the chance of DoS attacks
- Reduce spoofing attacks
- Provide bandwidth management
- When possible, use an IPS to inhibit the entry of malicious traffic into the network

**Additional Data Plane Protection Mechanisms Layer2:**

- **Port security** to protect against MAC addresses flooding and CAM overflow attacks.
- **Dynamic Host Configuration Protocol (DHCP)** snooping to prevent a rogue DHCP server
- **Dynamic ARP inspection (DAI)** can protect against Address Resolution Protocol (ARP) spoofing
- **IP Source Guard**, when implemented on a switch, verifies that IP spoofing is not occurring by devices on that switch

# Chapter 11: Securing the Management Place on IOS Devices

**What Is Management Traffic and the Management Plane?**

By requiring a **username** or **password**, you are taking the first steps toward improving what is called the management plane on this router or switch.

The management plane includes not only configuration of a system, but also who may access a system and what they are allowed to do while they are logged in to the system.

The management plane also includes messages to or from a Cisco router or switch that is used to maintain or report on the current status of the device, such as a management protocol like Simple Network Management Protocol (**SNMP**).

**Management Plane Best Practices**

- **Strong passwords**: Make passwords very difficult to break.
- **User authentication and AAA**: you can control which administrators are allowed to connect to which devices and what they can do while they are there
- **Login Password Retry Lockout**: allows system administrators to lock out a local AAA user account after a configured number of unsuccessful attempts
- **Role-based access control** (**RBAC**): you can control access through AAA and customize privilege levels/parser views
- **Encrypted management protocols**: encrypted communications should be used, such as Secure Shell (**SSH**) or Hypertext Transfer Protocol Secure (**HTTPS**)
- **Out-of-band** (**OOB**) management implies that there is a completely separate network just for management protocols and a different network for end users and their traffic. In-band management is when the packets used by your management protocols may intermingle with the user packets (considered less secure than OOB).
- **Logging and monitoring**: includes not only what administrators have changed or done but also system events that are generated by the router or switch because of some problem that has occurred or some threshold that has been reached; the storage of the logs and the transmission of the logs should be protected. If SNMP is used, preferably use Version 3 because of its authentication and encryption capabilities. An SNMP trap is a message generated by the router or switch to alert the manager or management station of some event.
- **Network Time Protocol** (**NTP**): to synchronize the clocks on network devices so that any logging that includes time stamps may be easily correlated. Preferably, use NTP Version 3 with authentication.
- **Secure system files**: Make it difficult to delete, whether accidentally or on purpose, the startup
- Configuration files and the IOS images that are on the file systems of the local routers and switches.

**Password Recommendations**
**# Security passwords min-length X** – set the minimum password length

- It is best to have a minimum of eight characters for a password

- Passwords can include any alphanumeric character, a mix of uppercase and lowercase characters, and symbols and spaces. Leading spaces in a password are ignored, but any subsequent spaces, including in the middle or at the end of a password, literally become part of that password and are generally a good idea.

**Using AAA to Verify Users**
In a nutshell, the goal of AAA is to **identify** who **users** are before giving them any kind of access to the network, and once they are identified, only give them access to the part they are authorized to use, see, or manage.

**AAA Components**
- **Authentication**: Authentication is the process by which individuals prove that they are who they claim to be. To specify the method to use, you create an authentication "method list" that specifies how to authenticate the user.

- **Authorization**: After the user or administrator has been authenticated, authorization can be used to determine which resources the user or administrator is allowed to access, and which operations may be performed.

- **Accounting and auditing**: record what the user or administrator actually does with this access, what he accesses, and how long he accesses it.

**Options for Storing Usernames, Passwords, and Access Rules**
Uses a centralized service to keep usernames, passwords, and configured rules about who can access which resources.

- **Cisco Secure ACS Solution Engine**: This is a dedicated server that contains the usernames, their passwords, and other information about what users are allowed to access and when they are allowed to access. TACACS+ for an administrator who is seeking command-line access to the network device, and RADIUS if you are authenticating an end user that is requesting access to the network.

- **Cisco Secure ACS for Windows Server**: This software package may be used for user and administrator authentication.

- **Current flavors of ACS functionality**: The most common way that ACS services are implemented today is through a virtual machine running on some flavor of VMware. Cisco **Identity Services Engine** (**ISE**), which can be bundled in a single physical or logical device or appliance.

- **Self-contained AAA**: AAA services may be self-contained in the **router itself**. The database that contains the usernames and passwords is the running configuration of the router or IOS device, and from an AAA perspective is referred to as the local database on the router.

**Authorizing VPN Users**
One common implementation of **AAA** is its use in authenticating users accessing the corporate LAN through a remote-access IPsec VPN. We authenticate the users by asking for their **username** and **password**, and then **check** the **rules** to see what they are authorized to access.

If we use the remote Access Control Server (**ACS**) server for the authentication and authorization for an end user, we would very likely use the RADIUS protocol between the router and the AAA server.

**Router Access Authentication**
We must choose authentication first if we want to also use authorization for a user or administrator. We cannot choose authorization for a user without knowing who that user is through authentication first. When an administrator is at the CLI, that interface is provided by something called an **EXEC shell**. This type of access (CLI) could also be referred to as **character mode**.

| Access Type Mode | Mode | Where These Are Likely to Be Used | AAA Command Element |
|---|---|---|---|
| **Remote administrative access Usually TACACS+ between the router and the ACS** | Character (line or EXEC mode) | **Lines:** vty, AUX console, and tty | **login**, **enable**, **exec** |
| **Remote network access end users Usually RADIUS between the router and the ACS** | Packet (interface mode) such as an interface with PPP requiring authentication | **Interfaces:** async, group-async, BRI, PRI Other functionality: VPN user authentication | **ppp**, **network**, **vpn groups** |

**The AAA Method List**

To make implementing AAA modular, we can specify individual lists of ways we want to authenticate, authorize, and account for the users. We can create method lists that define the authentication methods to use, authorization method lists that define which authorization methods to use, and accounting method lists that specify which accounting method lists to use.

**aaa type {default | list-name} method-1 [method-2 method-3 method-4]**

| Command Element | Description |
|---|---|
| **Type** | Identifies the type of list being created.<br>**authentication**, **authorization**, or **accounting**. |
| **Default** | Specifies the default list of methods to be used based on the methods that follow this argument. If you use the keyword **default**,<br>a custom name is not used. |
| **List-Name** | Used to create a **custom method list**. This is the name of this list, and is used when this list is applied to a line, such as to **vty** lines 0–4. |
| **Method** | At least one method must be specified.<br>To use the local user database, use the **local** keyword.<br>A **single list** can contain up to **four methods**, which are tried in order, from left to right.<br><br>In the case of an authentication method list, methods include the following:<br>• **enable:** The enable password is used for authentication. This might be an excellent choice as the last method in a method list. This way, if the previous methods are not available (such as the AAA server, which might be down or not configured), the router times out on the first methods and eventually prompts the user for the enable secret as a last resort .<br>• **krb5:** Kerberos 5 is used for authentication.<br>• **krb5-telnet:** Kerberos 5 Telnet authentication protocol is used when using Telnet to connect to the router.<br>• **line:** The line password is used for authentication.<br>• **local:** The local username database (running config) is used for authentication.<br>• **local-case:** Requires case-sensitive local username authentication.<br>• **none:** No authentication is used.<br>• **group radius:** A RADIUS server (or servers) is used for authentication.<br>• **group tacacs+:** A TACACS+ server (or servers) is used for authentication.<br>• **group** *group-name***:** Uses either a subset of RADIUS or TACACS+ servers for authentication as defined by the **aaa group server radius** or **aaa group server tacacs+** command. |

**Role-Based Access Control**

The concept of **role-based access control (RBAC)** is to create a **set** of **permissions** or **limited access** and **assign** that set of permissions to **users** or **groups**.

**Custom Privilege Levels**
When you first connect to a console port on the router, you are placed into user mode. User mode is really privilege level 1. This is represented by a prompt that ends with >.

**Limiting the Administrator by Assigning a View**
A solution to this is to use parser views, also referred to as simply a view. You can create a view and associate it with a subset of commands. When the user logs in using this view, that same user is restricted to only being able to use the commands that are part of his current view. You can also associate multiple users with a single view.

**Encrypted Management Protocols**
The problem with **Telnet** is that it uses **plain text**, and anyone who gets a copy of those packets can identify our usernames and passwords used for access and any other information that goes between administrator and the router being managed (over the management plane).

**Secure Shell** (**SSH**) provides the same functionality as Telnet, in that it gives you a CLI to a router or switch; unlike Telnet, however, **SSH encrypts all** the **packets** that are used in the session.
For graphical user interface (GUI) management tools such as CCP, use HTTPS rather than HTTP because, like SSH, it encrypts the session, which provides confidentiality for the packets in that session.

**Using Logging Files**
Administrators should, on a regular basis, analyze logs, especially from their routers, in addition to logs from other network devices. Log output sent to a variety of destinations:

- **Console:** send log messages to an attached terminal
- **VTY lines:** However, the **terminal monitor** command should be issued to cause log messages to be seen by the user on that vty line.
- **Buffer:** log messages can be stored in router memory; when the router is rebooted, these messages in the buffer memory are lost.
- **SNMP server:** When configured as an SNMP device, a router or switch can generate log messages in the form of SNMP traps and send them to an SNMP manager (server).
- **Syslog server:** Easily configured and can store a large volume of logs. Syslog messages can be directed to one or more syslog servers from the router or switch.

A **syslog** logging solution consists of two primary components: **syslog servers** and **syslog clients**. A syslog server receives and stores log messages sent from syslog clients such as routers and switches.

| Level Number | Security Level | Description | Example |
|---|---|---|---|
| **0** | Emergencies | System is unstable | Could not load IOS software |
| **1** | Alert | Immediate action is needed | Temperature too high |
| **2** | Critical | Critical conditions | Unable to allocate memory |
| **3** | Error | Error conditions | Invalid memory size |
| **4** | Warning | Warning conditions | Crypto operation failed |
| **5** | Notification | Normal but significant conditions | Interface changed state |
| **6** | Informational | Informational messages only | Packet denied by ACL |
| **7** | Debugging | Debugging messages only | Packet type invalid |

**Understanding NTP**

Network Time Protocol (**NTP**) uses UDP port 123, and it allows network devices to **synchronize** their **time**. One benefit of having reliable synchronized time is that **log files** and **messages** generated by the router can be **correlated**.

**Protecting Cisco IOS Files**

To help protect a router from accidental or malicious tampering of the IOS or startup configuration, Cisco offers a resilient configuration feature. The secure files are referred to as a **secure bootset**. The administrator cannot disable the features remotely.

**Implementing Strong Passwords**

```
! Use the "secret" keyword instead of the "password" for users
! This will create a secured password in the configuration by default
! The secret is hashed using the MD5 algorithm as it is stored in the
! -configuration
R1(config)# username admin secret CeyeSc01$24

! At a minimum, require a login and password for access to the console port
! Passwords on lines, including the console, are stored as plain text, by
! default, in the configuration
R1(config)# line console 0
R1(config-line)# password k4(1fmMsS1#
R1(config-line)# login
R1(config-line)# exit
```

```
! At a minimum, require a login and password for access to the VTY lines which
! is where remote users connect when using Telnet
! Passwords on lines, including the vty lines, are stored as plain text, by
! default, in the configuration
R1(config)# line vty 0 4
R1(config-line)# password 8wT1*eGP5@
R1(config-line)# login

! At a minimum, require a login and password for access to the AUX line
! and disable the EXEC shell if it will not be used
R1(config-line)# line aux 0
R1(config-line)# no exec
R1(config-line)# password 1wT1@ecP27
R1(config-line)# login
R1(config-line)# exit
```

```
! Encrypt the plain text passwords so that someone reading the configuration
! won't know what the passwords are by simply looking at the configuration.
R1(config)# service password-encryption
```

**User Authentication with AAA**

```
! Enable aaa features, if not already present in the running configuration
R1(config)# aaa new-model

! Identify a AAA server to be used, and the password it is expecting with
! requests from this router. This server would need to be reachable and
! configured as a TACACS+ server to support R1's requests
R1(config)# tacacs-server host 50.50.4.101
R1(config)# tacacs-server key ToUgHPaSsW0rD-1#7

! configure the default method list for the authentication of character
! mode login (where the user will have access to the CLI)
! This default method list, created below has two methods listed "local"
! and "enable"
! This list is specifying that the local database (running-config) will
!  be used first to look for the username.  If the username isn't in the
! running-config, then it will go to the second method in the list.
! The second method of "enable" says that if the user account isn't found
! in the running config, then to use the enable secret to login.
! This default list will apply to all SSH, Telnet, VTY, AUX and Console
! sessions unless there is another (different) custom method list that is
! created and directly applied to one of those lines.
R1(config)# aaa authentication login default local enable

! The next authentication method list is a custom authentication
! method list named MY-LIST-1.This method list says that the first attempt
! to verify the user's name and password should be done through one of the
! tacacs servers (we have only configured one so far), and then if that server
! doesn't respond, use the local database (running-config), and if the
! username isn't in the running configuration to then use the enable secret
! for access to the device.  Note: this method list is not used until
! applied to a line elsewhere in the configuration, i.e. the default list
! configured previously is used unless MY-LIST-1 is specifically configured.
R1(config)# aaa authentication login MY-LIST-1 group tacacs local enable
```

```
! We could create a default one as well, using the key
! word "default" instead of a name.    These custom method lists for
! authorization won't be used until we apply them
! elsewhere in the configuration, such as on a VTY line.
! The first method list called TAC1 is an authorization
! method list for all commands at user mode (called privilege level 1).
! The second method list called TAC15 is an
! authorization method list for commands at level 15 (privileged exec mode).
! If these method lists are applied to a line, such as the
! console or VTY lines, then before any commands
! are executed at user or privileged mode, the router will check
! with an ACS server that is one of the "tacacs+" servers, to see if the user
! is authorized to execute the command. If a tacacs+ server isn't
! reachable, then the router will use its own database of users (the local
! database) to determine if the user trying to issue the command
! is at a high enough privilege level to execute the command.
R1(config)# aaa authorization commands 1 TAC1 group tacacs+ local
R1(config)# aaa authorization commands 15 TAC15 group tacacs+ local
```

```
! The next 2 method lists are accounting method lists that will record the
! commands issued at level 1 and 15 if the lists are applied to a line, and
! if an administrator connects to this device via that line.
! Accounting method lists can have multiple methods, but can't log to the
! local router.
R1(config)# aaa accounting commands 1 TAC-act1 start-stop group tacacs+
R1(config)# aaa accounting commands 15 TAC-act15 start-stop group tacacs+

! Creating a user with level 15 access on the local router is a good idea,
! in the event the ACS server can't be
! reached, and a backup method has been specified as the local database.
R1(config)# username admin privilege 15 secret 4Je7*1swEsf

! Applying the named method lists is what puts them in motion.
! By applying the method lists to the VTY lines
! any users connecting to these lines will be authenticated by the
! methods specified by the lists that are applied
! and also accounting will occur, based on the lists that are applied.
R1(config)# line vty 0 4
R1(config-line)# login authentication MY-LIST-1
R1(config-line)# authorization commands 1 TAC1
R1(config-line)# authorization commands 15 TAC15
R1(config-line)# accounting commands 1 TAC-act1
R1(config-line)# accounting commands 15 TAC-act15

! Note: on the console and AUX ports, the default list will be applied,
! due to no custom method list being applied
! directly to the console or AUX ports.
```

**Another Example of Creating and Applying a Custom Method List to VTY Lines**

```
! Creating the method list, which in this example has 3 methods.
! First the local database
! (if the username exists in the configuration, and if not
! then the enable secret (if configured),  and if not then no
! authentication required
! (none)
R2(config)# aaa authentication login MY-AUTHEN-LIST-1 local enable none

! Applying the method list to the VTY lines 0-4
R2(config)# line vty 0 4
R2(config-line)# login authentication MY-AUTHEN-LIST-1
R2(config-line)# exit

! Creating a local username in the local database (running-config)
R2(config)# username bob secret ciscobob

! Setting the password required to move from user mode to privileged mode
R2(config)# enable secret ciscoenable
R2(config)# interface loopback 0

! Applying an IP address to test a local telnet to this same local router
! Not needed if the device has another local IP address that is in use
R2(config-if)# ip address 2.2.2.2 255.255.255.0
R2(config-if)# exit

! Enable logging so we can see results of the upcoming debug
R2(config)# logging buffered 7
R2(config)# end

! Enabling debug of aaa authentication, so we can see what the router is
! thinking regarding aaa authentication
R2# debug aaa authentication
AAA Authentication debugging is on
```

**Using the CLI to Troubleshoot AAA for Cisco Routers**

You may use three separate **debug** commands to troubleshoot the various aspects of AAA:

- **Debug aaa authentication:** Use this command to display debugging messages for the authentication functions of AAA.
- **Debug aaa authorization:** Use this command to display debugging messages for the authorization functions of AAA.
- **Debug aaa accounting:** Use this command to display debugging messages for the accounting functions of AAA.

**#test aaa group tacacs+ username password legacy** – test user connectivity and rights

Each of these commands is executed from **privileged EXEC** mode. To disable debugging for any of these functions, use the **no** form of the command, such as **no debug aaa authentication**. If you want to disable all debugging, issue the **undebug all** command.

```
! R4 will have a loopback, so we can telnet to ourselves to test
R4(config-if)# ip address 4.4.4.4 255.255.255.0
R4(config-if)# exit

! Local user in the database has a privilege level of 15
R4(config)# username admin privilege 15 secret cisco

! This method list, if applied to a line, will specify local authentication
R4(config)# aaa authentication login AUTHEN_Loc local

! This next method list, if applied to a line, will require authorization
! before giving the administrator an exec shell.   If the user has a valid
! account in the running configuration, the exec shell will be created for
! the authenticated
! user, and it will place the user in their privilege level automatically
R4(config)# aaa authorization exec AUTHOR_Exec_Loc local

! This method list, if applied to a line, will require authorization for
! each and every level 15 command issued.   Because the user is at -
! privilege level 15 the router will say "yes" to any level 15 commands
! that may be issued by the user
R4(config)# aaa authorization commands 15 AUTHOR_Com_15 local

! Next we will apply the 3 custom method lists to vty lines 0-4, so that
! when anyone connects via these vty lines, they will be subject to the
! login authentication, the exec authorization, and the level 15 command
! authorizations for the duration of their session.

R4(config)# line vty 0 4
R4(config-line)# login authentication AUTHEN_Loc
R4(config-line)# authorization exec AUTHOR_Exec_Loc
R4(config-line)# authorization commands 15 AUTHOR_Com_15
R4(config-line)# exit
R4(config)#
R4(config)# do debug aaa authentication
AAA Authentication debugging is on
R4(config)# do debug aaa authorization
AAA Authorization debugging is on
R4(config)# exit

! Now test to see it all in action.
R4# telnet 4.4.4.4
Trying 4.4.4.4 ... Open
User Access Verification
```

```
Username: admin
Password: [cisco] password not displayed when entering

! It  picked the login authentication list we specified
AAA/BIND(00000071): Bind i/f
AAA/AUTHEN/LOGIN (00000071): Pick method list 'AUTHEN_Loc'

! It picked the authorization list we specified for the exec shell
R4#
AAA/AUTHOR (0x71): Pick method list 'AUTHOR_Exec_Loc'
AAA/AUTHOR/EXEC(00000071): processing AV cmd=
AAA/AUTHOR/EXEC(00000071): processing AV priv-lvl=15
AAA/AUTHOR/EXEC(00000071): Authorization successful

! It picked the command level 15 authorization list, when we issued the
! configure terminal command, which is a level 15 command.
R4# config t
Enter configuration commands, one per line.  End with CNTL/Z.
R4(config)#
AAA/AUTHOR: auth_need : user= 'admin' ruser= 'R4' rem_addr= '4.4.4.4' priv= 15 list=
'AUTHOR_Com_15' AUTHOR-TYPE= 'command'
AAA: parse name=tty2 idb type=-1 tty=-1
AAA: name=tty2 flags=0x11 type=5 shelf=0 slot=0 adapter=0 port=2 channel=0
AAA/MEMORY: create_user (0x6A761F34) user='admin' ruser='R4' ds0=0 port='tty2'
rem_addr='4.4.4.4' authen_type=ASCII service=NONE priv=15 initial_task_id='0',
vrf= (id=0)
tty2 AAA/AUTHOR/CMD(1643140100): Port='tty2' list='AUTHOR_Com_15' service=CMD
AAA/AUTHOR/CMD: tty2(1643140100) user='admin'
tty2 AAA/AUTHOR/CMD(1643140100): send AV service=shell
tty2 AAA/AUTHOR/CMD(1643140100): send AV cmd=configure
tty2 AAA/AUTHOR/CMD(1643140100): send AV cmd-arg=terminal
tty2 AAA/AUTHOR/CMD(1643140100): send AV cmd-arg=<cr>
tty2 AAA/AUTHOR/CMD(1643140100): found list "AUTHOR_Com_15"
tty2 AAA/AUTHOR/CMD(1643140100): Method=LOCAL
AAA/AUTHOR (1643140100): Post authorization status = PASS_ADD
AAA/MEMORY: free_user (0x6A761F34) user='admin' ruser='R4' port='tty2'
rem_addr='4.4.4.4' authen_type=ASCII service=NONE priv=15 vrf= (id=0)
R4(config)#
! It made a big splash, with lots of debug output, but when you boil it all
! down it means the user was authorized to issue the configure terminal
! command.
```

**RBAC Privilege Level/Parser View**

You may implement **RBAC** through **AAA**, with the rules configured on an ACS server, but you may implement it in other ways, too, including creating **custom privilege levels** and having users enter those custom levels where they have a limited set of permissions, or creating a ***parser view***, which also limits what the user can see or do on the Cisco device.

**Each option** can be **tied** directly to a **username**, so that once users authenticate they may be placed at the custom privilege level, or in the view that is assigned to them.

**Creating and Assigning Commands to a Custom Privilege Level**

```
! By default,  we use privilege level 1 (called user mode), and privilege
! level 15 (called privileged mode).   By creating custom levels, (between
! 1-15) and assigning commands to those levels, we are creating custom
! privilege levels
! A user connected at level 8, would have any of the new commands -
! associated with level 8, as well as any commands that have been custom
! assigned or defaulted to levels 8 and below.    A user at level 15 has
! access to all commands at level 15 and below.
! This configuration assigns the command "configure terminal" to privilege
! level 8
R2(config)# privilege exec level 8 configure terminal

! This configuration command assigns the password for privilege level 8
! the keyword "password" could be used instead of secret, but is less secure
! as the "password" doesn't use the MD5 hash to protect the password
! The "0" before the password, implies that we are inputting a non-hashed
! (to begin with) password.  The system will hash this for us, because we
! used the enable "secret" keyword.
R2(config)# enable secret level 8 0 NewPa5s123&
R2(config)# end
R2#
%SYS-5-CONFIG_I: Configured from console by console

! To enter this level, use the enable command, followed by the level you want
! to enter.   If no level is specified, the default level is 15
```

```
R2> show privilege
Current privilege level is 1
! Context sensitive help shows that we can enter a level number after the
! word enable
R2> enable ?
  <0-15>  Enable level
  view    Set into the existing view
  <cr>

R2> enable 8
Password: [NewPa5s123&] ! note: password doesn't show when typing it in
R2# show privilege
Current privilege level is 8
! We can go into configuration mode, because "configure terminal" is at our
! level
R2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
! Notice we don't have further ability to configure the router, because
! level 8 doesn't include the interface configuration or other router -
! configuration commands.
```

**Creating a Local User and Associating That User with Privilege Level 8 and Assigning Login Requirements on the VTY Lines**

```
! Create the user account in the local database (running-config) and
! associate that user with the privilege level you want that user to use.
R2(config)# username Bob privilege 8 secret Cisco123
R2(config)# line vty 0 4

! "login local" will require a username and password for access if the "aaa
! new-model" command is not present.   If we have set the aaa new-model,
! then we would also want to create a default or named method list that
! specifies we want to use the local database for authentication.
R2(config-line)# login local

! Note:  Once bob logs in, he would have access to privilege level 8 and
! below, (including all the normal show commands at level 1)
```

**Implementing Parser Views**

To **restrict users** without having to create custom privilege levels, you can use a ***parser* view**. A view can be created with a subset of **privilege level 15** commands, and when the user logs in using this view, that same user is restricted to only being able to use the commands that are part of his current view.

To **create** a view, an **enable secret** password must first be configured on the router. **AAA** must also be enabled on the router ( **aaa new-model** command).

**Create a Parser View**

```
! Set the enable secret, and enable aaa new-model (unless already in
! place)
R2(config)# enable secret aBc!2#&iU
R2(config)# aaa new-model
R2(config)# end

! Begin the view creation process by entering the "default" view, using the
! enable secret
R2# enable view
Password: [aBc!2#&iU] note password not shown when typed

R2#
%PARSER-6-VIEW_SWITCH: successfully set to view 'root'.
R2# configure terminal

! As the administrator in the root view, create a new custom view
R2(config)# parser view New_VIEW
%PARSER-6-VIEW_CREATED: view 'New_VIEW' successfully created.

! Set the password required to enter this new view
R2(config-view)# secret New_VIEW_PW

! Specify which commands you want to include as part of this view.
! commands "exec" refer to commands issued from the command prompt
! commands "configure" refer to commands issued from privileged mode
R2(config-view)# commands exec include ping
R2(config-view)# commands exec include all show
R2(config-view)# commands exec include configure

! This next line adds the ability to configure "access-lists" but nothing
! else
R2(config-view)# commands configure include access-list
R2(config-view)# exit
R2(config)# exit
```

**Testing the Parser View**

```
! Test the view, by going to user mode, and then back in using the new view
R2# disable
R2>enable view New_VIEW
Password: [New_VIEW_PW] Password not shown when typed in

! Console message tells us that we are using the view
%PARSER-6-VIEW_SWITCH: successfully set to view 'New_VIEW'.

! This command reports what view we are currently using
R2# show parser view
Current view is 'New_VIEW'

! We can verify that the commands assigned to the view work
! Note: we only assigned configure, not configure terminal so we have to
! use the configure command, and then tell the router we are configuring
! from the terminal.   We could have assigned the view "configure terminal"
! to avoid this
R2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.

! Notice that the only configuration options we have are for access-list,
! per the view
R2(config)# ?
Configure commands:
  access-list  Add an access list entry
  do           To run exec commands in config mode
  exit         Exit from configure mode
```

We could also **assign** this **view** to a **user account**, so that when users log in with their username and password, they are automatically placed into their view. This needs to be done by someone in privilege level 15.

**Associate User Account with a Parser View**

```
R2(config)# username Lois view New_VIEW secret cisco123
```

**SSH**

To enable SSH on a router or switch, the following items need to be in place:

- **Hostname**
- **Domain name**
- Generating a **public/private key pair**, used behind the scenes by SSH.
- Requiring **user login** via the **vty lines**, instead of just a password. Local authentication or authentication using an ACS server are both options.
- Having at least one **user account** to log in with, either locally on the router, or on an ACS server.

**Preparing for SSH**

```
! To create the public/private key pair used by SSH, we would issue the
! following command.   Part of the key pair, will be the hostname and the
! domain name.
! If these are not configured first, the crypto key generate command will
! tell you as shown in the next few lines.
Router(config)# crypto key generate rsa
% Please define a hostname other than Router.
Router(config)# hostname R1
R1(config)# crypto key generate rsa
% Please define a domain-name first.
R1(config)# ip domain-name cisco.com

! Now with the host and domain name set, we can generate the key pair
R1(config)# crypto key generate rsa
The name for the keys will be: R1.cisco.com
How many bits in the modulus [512]: 1024
% Generating 1024 bit RSA keys, keys will be non-exportable...

R1(config)#
%SSH-5-ENABLED: SSH 1.99 has been enabled
! Note the "1.99" is based on the specifications for SSH from RFC 4253
! which indicate that an SSH server may identify its version as 1.99 to
! identify that it is compatible with current and older versions of SSH.

! Create a user in the local database
R1(config)# username Keith secret Ci#kRk*ks

! Configure the vty lines to require user authentication
R1(config)# line vty 0 4
R1(config-line)# login local
```

```
! Alternatively, we could do the following for the requirement of user
! authentication
! This creates a method list which points to the local database, and then
! applies that list to the VTY lines
R1(config)# aaa new-model
R1(config)# aaa authentication login Keith-List-1 local
R1(config)# line vty 0 4
R1(config-line)# login authentication Keith-List-1

! To test this we could SSH to ourselves from the local machine, or from
! another router that has IP connectivity to this router.

R1# ssh ?
  -c    Select encryption algorithm
  -l    Log in using this user name
  -m    Select HMAC algorithm
  -o    Specify options
  -p    Connect to this port
  -v    Specify SSH Protocol Version
  -vrf  Specify vrf name
  WORD  IP address or hostname of a remote system

! Note: one of our local IP addresses is 10.1.0.1
R1# ssh -l Keith 10.1.0.1

Password: <password for Keith goes here>

R1>
! to verify the current SSH session(s)
R1> show ssh
Connection Version Mode Encryption  Hmac       State            Username
0          2.0     IN   aes128-cbc  hmac-sha1  Session started  Keith
0          2.0     OUT  aes128-cbc  hmac-sha1  Session started  Keith
%No SSHv1 server connections running.
```

**HTTPS**

Can be used to **manage** a **router** via **HTTPS**. You can use **CCP** or a **similar tool** and implement HTTPS functionality.

**Preparing for HTTPS**

```
! Enable the SSL service on the local router.   If it needs to generate
! keys for this feature, it will do so on its own in the background.
R1(config)# ip http secure-server

! Specify how you want users who connect via HTTPS to be authenticated
R1(config)# ip http authentication ?
  aaa     Use AAA access control methods
  enable  Use enable passwords
  local   Use local username and passwords

R1(config)# ip http authentication local

! If you are using the local database, make sure you have at least one user
! configured in the running-config so that you can login.  To test, open
! a browser to HTTPS://a.b.c.d where a.b.c.d is the IP address on the
! router.
```

**Logging**

**Implementing Logging Features**

Logging is important as a tool for **discovering events** that are happening in the network and for **troubleshooting**.

```
logging 10.1.1.200
logging trap notifications
logging buffered 4096 debugging
```

```
R4(config)# interface fa0/0
R4(config-if)# shut
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to administratively down
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to down
R4(config-if)#

! If we add time stamps to the syslog messages, those time stamps can assist it
! correlating events that occurred on multiple devices

R4(config)# service timestamps log datetime
R4(config)# int fa0/0
R4(config-if)# no shutdown

! These syslog messages have the date of the event, the event (just after
! the %) a description, and also the level of the event (the first event in
! the example below is level 3 with the second event being level 5).
*Nov 22 12:08:13: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Nov 22 12:08:14: %LINEPROTO-5-UPDOWN: Line protocol on Interface
  FastEthernet0/0, changed state to up
```

**SNMP Features**

***Simple Network Management Protocol (SNMP)*** is to **manage** network nodes, such as network servers, routers, switches, and so on. **SNMP versions** range from **version 1** to **3**, with some intermediate steps in between. The later the version, the more **security** features it has.

- SNMP manager: runs a network management application, network management server (**NMS**).
- SNMP agent: is a piece of software that runs on a managed device
- Management Information Base (**MIB**): Information about a managed device's resources and activity is defined by a series of objects

| Component | Description |
|---|---|
| **SNMP Manager** | Runs a **network management application**, network management server (**NMS**). |
| **SNMP Agent** | A piece of **software** that runs on a **managed device** |
| **Management Information Base (MIB)** | **Information** about a **managed device's resources** and activity is defined by a series of objects |

An **SNMP manager** can send information to, receive request information from, or receive unsolicited information (called a trap) from a managed device (a router). The managed device runs an SNMP agent and contains the MIB.

**SNMP messages**

- **GET**: An SNMP GET message is used to **retrieve information** from a managed device
- **SET**: An SNMP SET message is used to **set** a **variable** in a managed device or to **trigger** an **action** on a managed device
- **Trap**: An SNMP trap message is an **unsolicited message** sent from a managed device to an SNMP manager

Potential **security vulnerability** if an attacker introduces a **rogue NMS** into the network, the attacker's NMS might be able to gather information about network resources by polling the MIBs of managed devices.

In addition, the attacker might launch an attack against the network by **manipulating** the **configuration** of managed devices by sending a series of **SNMP SET** messages.

SNMPv1 and SNMPv2c use *community strings* to gain read-only access/read-write access to a managed device. Default read-only community string of "public" and a default read-write community string of "private."

SNMPv3 uses the concept of a security model and a security level:

- **Security model:** An approach for user and group authentications.
- **Security level:** Type of security algorithm performed on SNMP packets. Three security levels are discussed here**:**
    - **noAuthNoPriv:** Uses Community strings and has no authentication and privacy
    - **authNoPriv:** Has authentication but no privacy). Provides authentication using *(HMAC)* with *(MD5)* or *(SHA).* However, no encryption is used.
    - **authPriv:** Has authentication and privacy) and offers HMAC MD5/SHA with DES-56

| Security Model | Security Level | Authentication Strategy | Encryption Type |
|---|---|---|---|
| **SNMPv1** | noAuthNoPriv | Community string | None |
| **SNMPv2c** | noAuthNoPriv | Community String | None |
| **SNMPc3** | noAuthNoPriv | Username | None |
| | authNoPriv | MD5 or SHA | None |
| | authPriv | MD5 or SHA | CBC-DES (DES-56) |

**SNMPv3** offers three primary security enhancements:

- ■ **Integrity:** Using hashing algorithms, Packet not modified in transit
- **Authentication:** Hashing allows SNMPv3 to validate the source of an SNMP message
- **Encryption:** Using the CBC-DES (DES-56) encryption algorithm, SNMPv3 provides privacy for SNMP messages, making them unreadable by an attacker who might capture an SNMP packet.

**Output Created by CCP for impleading SNMPv1**

```
snmp-server location 192.168.1.96
snmp-server contact Bubba Jones
snmp-server community CCNA RO
snmp-server host 10.1.0.26 trap cisK0tRap^
```

**SNMPv3 Configuration**

```
! Enter global configuration mode
CCNA-Router# configure terminal
! Configure the community string along with an access-list to restrict access
CCNA-Router(config)# snmp-server community CCNA RO 99
! Create the IP Standard Access List defined in the previous step
CCNA-Router(config)# access-list 99 permit 192.168.1.0 /24
! Configure the v3 for no authentication (noauth)
CCNA-Router(config)# snmp-server group CCNA-group v3 noauth
! Configure a v3 user that resides in the v3 group
CCNA-Router(config)# snmp-server user CCNA-user CCNA-group v3
! Configure the community string and access-list to restrict SNMP to hosts in the
! 192.168.1.0/24 subnet
CCNA-Router(config)# snmp-server community CCNA RO 99
! Specify interface to be used for SNMP traps
CCNA-Router(config)# snmp-server trap-source FastEthernet0/1
! Specify the SNMP v3 server that will be allowed SNMP access
CCNA-Router(config)# snmp-server host 192.168.1.96 version 3 noauth CCNA-user
!
```

**Configuring NTP**

Because time is such an important factor, you should use Network Time Protocol (NTP) to **synchronize the time** in the network so that events that generate messages and time stamps can be correlated.

**Using Authentication via Keys with NTPv3**

```
ntp authentication-key 1 md5 141411050D 7
ntp authenticate
ntp trusted-key 1
ntp update-calendar
ntp server 192.168.1.96 key 1 prefer source FastEthernet0/1
```

**Verifying Synchronization via Client**

```
CCNA-Router# show ntp status
Clock is synchronized, stratum 4, reference is 192.168.1.96
nominal freq is 250.0000 Hz, actual freq is 249.9980 Hz, precision is 2**24
reference time is D8147295.4E6FD112 (13:11:49.306 UTC Mon Nov 17 2014)
clock offset is -0.3928 msec, root delay is 83.96 msec
root dispersion is 94.64 msec, peer dispersion is 2.22 msec
loopfilter state is 'CTRL' (Normal Controlled Loop), drift is 0.000007749 s/s
system poll interval is 64, last update was 126 sec ago.
CCNA-Router#

CCNA-Router# show ntp association

  address         ref clock       st   when   poll reach  delay  offset   disp
*~192.168.1.96    208.75.89.4      3     49     64   377  1.341  -0.392  2.424
 * sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
CCNA-Router#
```

**Secure Copy Protocol**

**The Secure Copy (SCP)** feature provides a **secure** and **authenticated** method for **copying device configurations** or **device image** files. SCP requires that authentication, authorization, and accounting (**AAA**) authorization be configured so that the device can determine whether the user has the **correct privilege level**.

**SCP Configuration**

```
CCNA-Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
CCNA-Router(config)# ip scp server enable
CCNA-Router(config)# exit
```

**Securing the Cisco IOS Image and Configuration Files**

The **Cisco Resilient Configuration** feature is intended to improve the recovery time by making a secure working copy of the IOS image and startup configuration files (which are referred to as the primary bootset) that **cannot be deleted** by a **remote user**.

**Creating a Secure Bootset**

```
! Secure the IOS image
R6(config)# secure boot-image
%IOS_RESILIENCE-5-IMAGE_RESIL_ACTIVE: Successfully secured running image

! Secure the startup-config
R6(config)# secure boot-config
%IOS_RESILIENCE-5-CONFIG_RESIL_ACTIVE: Successfully secured config archive
   [flash:.runcfg-20111222-230018.ar]

! Verify the bootset
R6(config)# do show secure bootset
IOS resilience router id FTX1036A13J

IOS image resilience version 12.4 activated at 23:00:10 UTC Thu Dec 22 2011
Secure archive flash:c3825-advipservicesk9-mz.124-24.T.bin type is image (elf) []
  file size is 60303612 bytes, run size is 60469256 bytes
  Runnable image, entry point 0x80010000, run from ram

IOS configuration resilience version 12.4 activated at 23:00:18 UTC Thu Dec 22 2011
Secure archive flash:.runcfg-20111222-230018.ar type is config
configuration archive size 1740 bytes

! Note: to undo this feature, (using the "no" option in front of the command)
! you must be connected via the console.  This prevents remote users from
! disabling the feature.
```

# Chapter 14: Understanding Firewall Fundamentals

**Firewall Technologies**

A **Firewall** is a concept that can be implemented by a single device, a group of devices, or even simply software running on a device such as a host or a server. The function of a firewall primarily is to **deny unwanted traffic** from crossing the boundary of the firewall. For network traffic, this means that a firewall could be implemented by the following:

A Router or other Layer 3 forwarding device that has an access list or some other method used to filter traffic that is trying to go between two of its interfaces. This is the primary method that is implemented by an IOS router using firewall features or the Adaptive Security Appliance ASA firewall.

A Switch that has two virtual LANs (VLAN) without any routing in between them, which would absolutely keep traffic from the two different networks separate.

Hosts or servers that are running software that prevents certain types of received traffic from being processed and controls which traffic can be sent. (Software Firewall)

**Objectives of a Good Firewall**

- It must be resistant to attacks.
- Traffic between networks must be forced through the firewall.
- The Firewall enforces the access control policy of the organization

| Reduces the Risk Of | Explanation |
|---|---|
| **Exposure of sensitive systems to untrusted individuals** | By hiding the functionality of a host or network device and permitting only the minimum required connectivity to tat given system, the firewall reduces the exposure for that system. |
| **Exploitation of Protocol flaws** | You can configure a firewall to inspect protocols to ensure compliance with the standards for that protocol at multiple layers of the protocol stack. It can also control the amount of time it will allow for a normal connection sequence before saying enough is enough. |
| **Unauthorized users** | By using authentication methods a firewall could control which user's traffic is allowed through the firewall and be configured to block on all traffic based on policy. |
| **Malicious data** | A firewall can detect and block malicious data which could stop traffic from reaching the intended destination. This function could also be provided by an intrusion prevention system (IPS). |

**Potential Firewall Limitations**

| Limitation | Explanation |
|---|---|
| **Configuration mistakes have serious consequences** | Incorrect ACLs, NAT and authentication. |
| **Not all network applications were written to survive going through the firewall** | If an application will not work based on the combination of what the application is doing and current firewall rules, the choice is to rewrite the application or make an exception in the firewall policy for the application. |
| **Individuals who are forced to go through a firewall might try to engineer a way around it** | Using tunnelling to hide a message into another protocol |
| **Latency being added by the firewall** | It might take a few milliseconds or more per packet for the analysis and as a result some slight delay may be added to the network. |

**Firewall Methodologies**

**Network-based firewalls** provide key features used for **perimeter** security. The primary task of a network firewall is to deny or permit traffic that attempts to enter or leave the network based on explicit preconfigured policies and rules. Firewalls are often deployed in several other parts of the network to provide network segmentation within the corporate infrastructure and also data centres. The **Processes** used to allow or block traffic may include the following:

- Simple packet-filtering techniques
- Proxy servers
- NAT
- Stateful inspection firewall
- Transparent firewalls
- Next-Generation context and application-aware firewalls

**Static Packet Filtering**

Static packet filtering is based on **Layer 3 and Layer 4** of the OSI model. An example of a firewall technology that uses static packet filtering is a router with an ACL applied to one or more of its interfaces for the purpose of permitting and denying specific traffic. The admin needs to know what traffic needs to be allowed through the firewall.

**Advantages and Disadvantages of Packet Filters**

| Advantages | Disadvantages |
|---|---|
| **Based on simple sets of permit or deny entries** | Susceptible to IP spoofing. If the ACL allows traffic from a specific IP address and someone is spoofing the source IP address, the ACL permits that individual packet |
| **Have a minimal impact of network** | Does not filter fragmented packets with the same accuracy as nonfragmented packets |
| **Simple to implement** | Extremely long ACLs are difficult to maintain |
| **Configurable on most routers** | Stateless (Does not maintain session information for current flows of traffic going through the router |
| **Can perform many of the basic filtering needs without requiring the expense of a high-end firewall** | Some applications jump around and use many ports, some of which are dynamic. A Static ACL may be required to open a very large range of ports to support application that may only use a few of them |

**Application Layer Gateway**

Application Layer firewalls which are also called **proxy firewalls** or **application gateways** can operate at Layer 3 and higher in the OSI reference model. Most of these proxy servers include specialized application software that takes requests from a client, puts that client on hold for a moment and then turns around and makes the requests as if it is its own request out to the final destination.

A Proxy firewall acts as an **intermediary** between the original client and the server. No direct communication occurs between the client and the destination server. Because the application layer gateway can operate all the way up to Layer 7, it has the potential to be very granular and analytical about every packet that the client and server exchange and can enforce rules based on anytime the firewall sees.

**Advantages and Disadvantages of Application Layer Gateways**

| Advantages | Disadvantages |
|---|---|
| **Very tight control is possible, due to analysing the traffic all the way to the application layer** | Is processor intensive because most of the work is done via software on the proxy server |
| **It is more difficult to implement an attack against an end device because of the proxy server standing between the attacker and potential victim** | Not all applications are supported and in practice it might support a specific few applications |
| **Can provide very detailed logging** | Special client software may be required |
| **May be implemented on common hardware** | Memory and disk intensive at the proxy server. Could potential be a single point of failure in the network, unless fault tolerance is also configured. |

**Stateful Packet Filtering**

It **remembers** the state of sessions that are going through the firewall.

With **stateful** packet-filtering device, for customers' on the inside of the corporate network as they are trying to reach resources on the outside public networks, their packets go to the firewalls on the way out. The firewalls take a look at the source IP address, destination IP address, the ports in use and other layers of information in the stateful database.

By **default**, this same firewall does not allow any traffic from the outside and untrusted networks back into the private trusted inside network. The exception to this is for return traffic that exactly matches the expected return traffic based on the stateful database information on the firewall. In short, the reply traffic goes back to the users successfully, but attackers on the outside trying to initiate sessions are denied by default.

| Advantages | Disadvantages |
|---|---|
| **Can be used as a primary means of defence by filtering unwanted or unexpected traffic** | Might not be able to identify or prevent an application layer attack |
| **Can be implemented on routers and dedicated firewalls** | Not all protocols contain tightly controlled state information, such as User Datagram Protocol (UDP) and Internet Control Message Protocol (ICMP) |
| **Dynamic in nature compared to static packet filtering** | Some applications may dynamically open up new ports from the server, which if a firewall is not analysing specific applications or prepared for this server to open up a new port, it could cause a failure of that application for the end user. If a firewall also supports application layer inspection, it may be able to predict and allow this inbound connection. |
| **Provides a defence against spoofing and denial of service attack (DoS)** | Stateful technology, by itself does not support user authentication. This however does not prevent a firewall that implements stateful packet filtering from also implementing authentication as an additional feature |

**Application Inspection**

An Application inspection firewall can **analyse** and **verify protocols** all the way up to **Layer 7** of the OSI reference model but does not act as a proxy between the client and the server being accessed by the client.

| Feature | Explanation |
|---|---|
| **Can see deeper into conversations, to see secondary channels that are about to be initiated from the server** | If an application is negotiating dynamic ports, and the server is about to initiate one of these dynamic ports to the client, the application inspection could have been analysing that conversation and dynamically allowed that connection from the server to allow it through the firewall and to the client.<br>This would allow the application to work for the client (Through Firewall) |
| **Awareness of the details at the application layer** | If there is a protocol anomaly that is a deviation from the standard, an application layer firewall could identify this and either correct the packet or deny the packet from reaching the destination |
| **Can prevent more kinds of attacks than stateful filtering on its own** | Current firewalls have packet filtering, stateful filtering and application inspection capabilities in a single device. With the additional features, more types of traffic can be classified and then permitted or denied based on policy. |

**Transparent Firewalls**

A transparent firewall is more about how we inject the firewall into the network as opposed to what technologies it uses for filtering. A transparent firewall can use **packet**-**based filtering**, **stateful filtering**, **application** but with transparent firewalls is that they are **implemented** at **Layer 2.**

Most traditional firewalls are implemented as a Layer 3 hop in the network (similar to a router hop), meaning that packets have to go through this device at Layer 3. In a Layer 3 firewall, each of the interfaces has an IP address on a different network, and traffic from one subnet to another that goes through the firewall has to pass the rules on the firewall.

With a transparent firewall, we still have two interfaces, but we do not assign IP addresses to those interfaces, and those two interfaces act more like a bridge (or a switch with two ports in the same VLAN). Traffic from one segment of a given subnet is going to be forced through the transparent firewall if those frames want to reach the second segment behind the firewall. A transparent firewall has a management IP address so that we can remotely access it, but that is all.

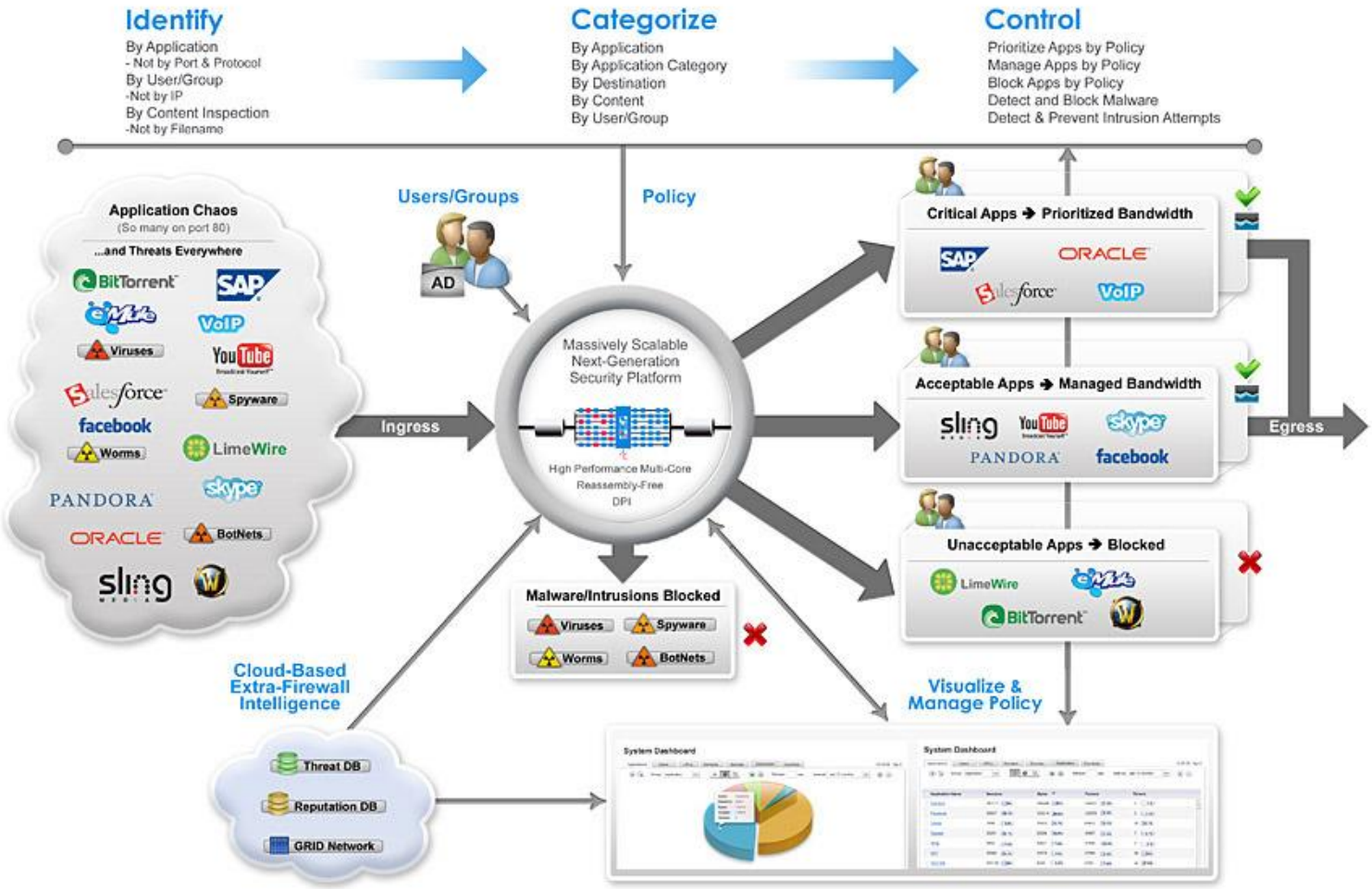
**Users** accessing resources through the firewall will **not be aware** that it is even present, and one of the biggest advantages of using a transparent firewall is that we do not have to re-address our IP subnets to put a transparent firewall in-line on the network.

Even though this is implemented as a Layer 2 device, it **still sees all packets** that go between its interfaces, and it can still apply all the rules of a normal Layer 3 firewall related to permitting traffic, building a stateful database, and performing application inspection.

**Next-Generation Firewalls**

***Next-generation firewalls (NGFW)*** provide **threat-focused security** services allowing for more comprehensive protection from known and advanced threats, including protection against targeted and persistent malware attacks.

An example of an NGFW is the **Cisco ASA** with **FirePOWER** Services. It combines the classic ASA firewall with Sourcefire threat prevention and advanced malware protection in a single device. The goal of NGFW is to maintain comprehensive visibility into users, mobile devices, clientside apps, *virtual machine (VM)*-to-VM communications, vulnerabilities, threats, and *uniform resource locato rs (URL)*.



**Firewall Summary**

**Stateless** – Use of static packet filters (ACLs) to control what traffic can enter a network. As much network traffic uses random port numbers (FTP, in bound HTTP traffic etc), this method is not optimum.

**Stateful** – Monitors the state of connections storing them in a session/state table. Storing open connections allows the firewall to detect attacks by examining the sequence numbers (TCP Only) and allows return traffic for outbound connections. A Stateful firewall will not allow a TCP packet with the SYN bit set and only allows packets with the ACK bit set if there is an entry in the session table indicating an inside user initiated the connection. Operates at OSI layers 3, 4 & 5.

**Application Layer Gateway** – Acts as proxy. Operates at OSI layers 3, 4, 5 & 7. An ALG can enforce user authentication rather than devices

**Transparent Firewalls** – Transparent firewalls are layer 2 devices which act like a network bridge. They are easily introduced as IP addressing of the existing networks do not need to be changed. Extended ACLs can be created for IP traffic and EtherType ACLs for non IP traffic. By default only ARP traffic can pass. Transparent Firewalls do not pass traffic with an EtherType greater than or equal to 0x600 (CDP, IS-IS etc.). Spanning Tree BPDUs, EIGRP, OSPF etc are supported.

**Creating and Deploying Firewalls**

**Firewall Design Considerations**

- Firewalls should be **placed at security boundaries**, such as between two networks that have different levels of trust.
- Firewalls should be a **primary security device**, but not the only security device or security measure on the network.
- A Policy that stars with a "**Deny All**" attitude and then specifically only permits traffic that is required is a better security posture than a default "Permit All" attitude first and then denying traffic.
- Leverage the firewall feature that **best suit** the **need**. Example if you know you have thousands of users who need access to the internet, you can implement dynamic NAT/PAT for those users along with stateful filtering and deny all inbound traffic coming from the internet. This stops users on the internet from initializing sessions to your users.
- Make sure that **physical security** controls and management access to the firewall devices are secure.
- Have regulatory structured **review** process looking at the firewall logs.
- Practice change management for any configuration modification on the firewalls.

**Firewall Access Rules**

| Rule | Description |
|---|---|
| **Rules based on service control** | These rules are based on the types of services that may be accessed through the firewall, **inbound** or **outbound**. Example access to web servers is allowed while all other traffic is denied. |
| **Rules based on address control** | These rules are based on the **source/destination addresses** involved, usually with a permit or deny based on specific entries in an access control list. |
| **Rules based on direction control** | These rules specify where the **initial traffic can flow**. |
| **Rules based on user control** | These rules control **access based** on knowing who the **user** is and what that user is **authorized** to do. This can be implemented via **AAA services**. |
| **Rules based on behaviour control** | These rules control how a particular **service is used**. Example a firewall may implement an e-mail filter to protect against spam. |

**Firewall Rule Design Guidelines**

- Use a **restrictive approach** as opposed to a permissive approach for all interfaces and all directions of traffic.
- Presume that your **internal user's** machines may be part of the **security problem**.
- Be **specific as possible** in your **permit** statements such as avoiding the use of the keyword "ANY" or "ALL" IP protocols if possible.
- Recognize the necessity of a **balance** between **functionality** and **security**.
- Filter **bogus traffic** and perform **logging** on that traffic.
- Periodically **review the policies** that are implemented on the firewall to verify that they are current and correct.

**Rule Implementation Consistency**

For any changes that will be made to a firewall, a **change control procedure** should identify exactly what is going to be done, why it is going to be done and the approval of the person in charge.

| Rule | Description |
|---|---|
| **Rules that are too promiscuous** | Allow **more access** than is necessary |
| **Redundant rules** | If a **rule** is **already in place** as allowing a specific flow of traffic, a second rule for that does not need to be added |
| **Shadowed rules** | An **incorrect order** placement in the ACL |
| **Orphaned rules** | **Configuration error** that is referencing **incorrect** IP address |
| **Incorrectly planned rules** | An **error** made as the **business requirements** are being translated to the technical and logical controls that the firewall will implement |
| **Incorrectly implemented rules** | An administrator implanting the **incorrect port**, **protocol** or **IP** information on the firewall |

# Chapter 15: Implementing Cisco IOS Zone-Based Firewalls

With **ZBF**s, **interfaces** are **placed** into **zones**. Zones are created by the network administrator using any naming convention that makes sense. Then **policies** are **specified** as to what transit traffic is allowed to be initiated and what **action** the firewall should take such as **inspection**.

After **traffic** is **inspected**, the **replay traffic** is **allowed** back through the firewall because of the **stateful** filtering feature. The policies are implemented in a **single direction**.

If you want to allow initial traffic in **both directions**, you create **two unidirectional policies** for traffic to be allowed and inspected from the inside to the outside, and from outside to inside.
You implement two separate policies because the **policies** themselves are **unidirectional**.

One **benefit** of this modular approach is that after policies are in place, if you add **additional interfaces** all you need to do is **add** those interfaces to **existing zones** and your policies will **automatically** be in place.

**Specific Features of Zone-Based Firewalls**

- Stateful inspection
- Application Inspection
- Packet Filtering
- URL Filtering
- Transparent firewall (Implementation method)
- Support for virtual routing and forwarding (VRF)
- Access control lists (ACL) are not required as a filtering method to implement the policy

**Zones and Why We Need Pairs of Them**

A **zone** is a **logical area** where devices with **similar trust levels** reside. A zone is created by the administrator and then interfaces can be assigned to zones. A Zone can have one or more interfaces assigned to it. Any given interface can belong to **only** a **single zone**.

There is a **default zone** called the **self-zone**, which is a logical zone. For any packets directed to the router directly, the router automatically considers that traffic to be entering the self-zone. In addition, any traffic initiated by the router is considered as leaving the self-zone. By **default**, any traffic to or from the self-zone is **allowed**, but can change the policy.

For the rest of the administrator-created zones, no traffic is allowed between interfaces in different zones. For interfaces that are members of the same zone, all traffic is permitted by default.

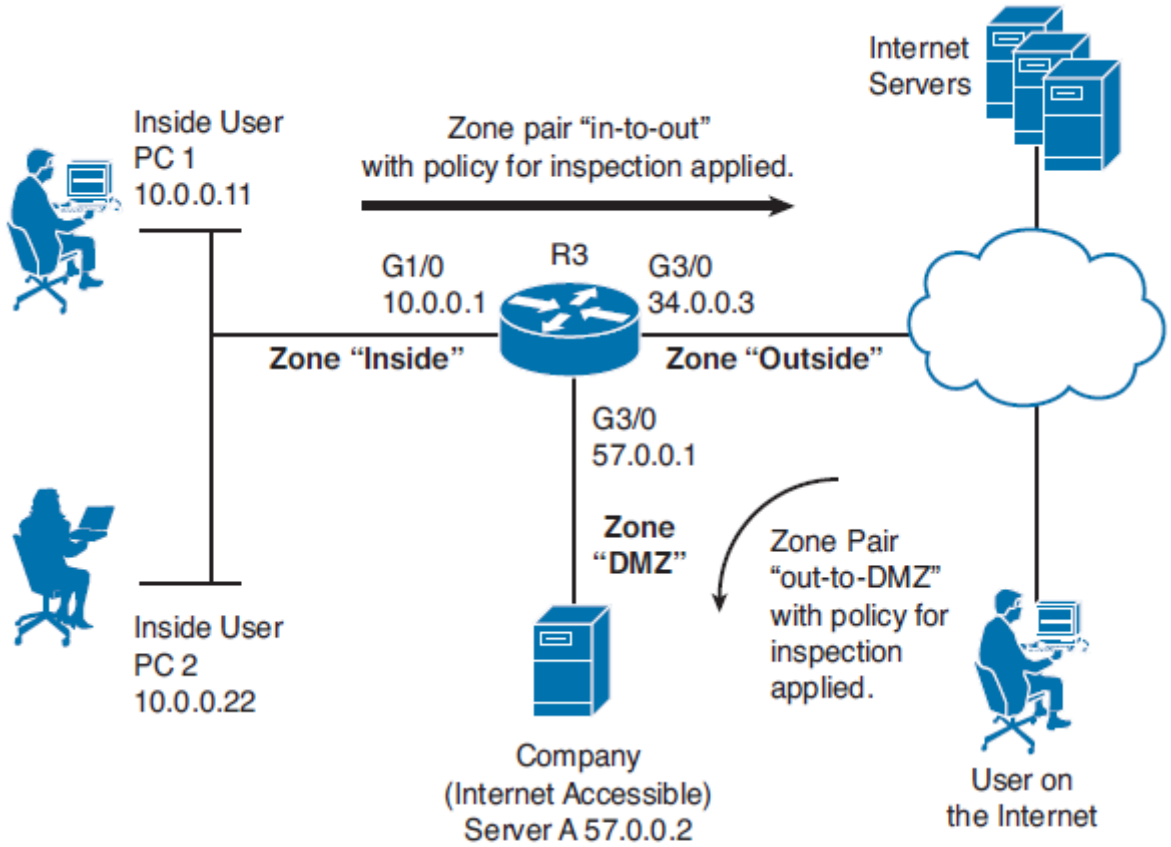
A small company, with users on the inside network, with the only other connection being the Internet, might want to create two zones, one for the inside and one for the outside.

Then they would assign the **inside interface** to the **inside zone**, and the **outside interface** to the **outside zone**. Then, a **policy** could be created that specifies that **traffic** that is **initiated** from the **inside users** and **going out** to the **Internet** should be inspected and that information should be placed in the **stateful database**.

A **zone pair** identifying traffic from the **inside** to the **outside** would have the policy applied to it, letting it know that the **stateful inspection** should be done.

A larger company that has a public-facing server may have three interfaces and three zones. The zones may be inside, outside, and DMZ. Compared to the small company, this mediumsized company creates an additional zone pair (from outside to DMZ) and then applies a policy to that zone pair to allow outside users to access the servers on the DMZ.

**Example of a medium-sized company with a DMZ**




Any time traffic is inspected, it is allowed between the zones and the session information is placed into the stateful database on the zone-based firewall. This allows the return traffic to be allowed, even without a zone pair and policy in the direction of the return traffic.

**Putting the Pieces Together**

**Class Maps:**
These are **used** to **identify traffic**, such as traffic that should be inspected. Traffic can be **matched** based on **Layer 3** through **Layer 7** including **application-based matching**. Class maps can also refer to *access control lists (ACL)* for the purpose of identifying traffic or even call upon other class maps. Class maps can have multiple **match** statements. A class map can specify that all **match** statements have to match (which is a *match-all condition*) or can specify that matching any of the entries is considered a match (which is a *match-any condition*). A system-defined class map named class-default can be used that represents all traffic not matched in a more specific (administratively configured) class map.

**Policy Maps:**
These are the **actions** that should be **taken on the traffic**. Policy maps **call** on the **class maps** for the classification or traffic. Policy maps with multiple sections are processed in order. The primary actions that can be implemented by the policy map are **inspect**, **permit**, **drop** or **log**.

**Service Policies**:
This is where you **apply** the **policies**, identified from a policy map to zone pair.

**Policy Map Actions**

| Policy Action | Description | When to Use It |
|---|---|---|
| **Inspect** | Permit and stateful inspect the traffic | This should be used on **transit traffic initiated** by **users** who expect to **get replies** from devices on the other side of the firewall. |
| **Pass** | Permit/allow the traffic but do not create an entry in the stateful database | Traffic that **does not need** a **reply**. Also in the case of **protocols** that do **not support inspection**, this policy could be applied to the zone pair for specific, outbound traffic and be applied to a second zone pair for inbound traffic. |
| **Drop** | Deny the packet | Traffic you do **not want to allow** between the zones where this policy map is applied. |
| **Log** | Log the packets | If you want to **see log information** about **packets** that were **dropped** because of policy, you can add this option. |

**Service Policies:**

A **service policy** is **applied** to a **zone pair**. The zone pair represents a **unidirectional** flow of traffic **between two zones**. A specific **zone pair** can have only a **single service policy** assigned to it. Because the zone is unidirectional, the policy map applied to the zone pair (Using service-policy command) applies to traffic initiated in one zone going to the other zone in one direction.

If reply traffic is desired, the inspect action in the policy map should be applied, which will allow stateful inspection and the reply traffic from the servers will be dynamically allowed.

When a router receives a packet, it normally makes a routing decision and then forwards that packet on its way. If ZBF is configured, the router may or may not forward the packet, based on the stateful table and the policies that are in place.

| Ingress Interface Member of Zone | Egress Interface Member of Zone | Zone Pair Exists with Applied Policy | Result |
|---|---|---|---|
| **No** | No | Does not matter | Traffic is forwarded. |
| **No** | Yes (any zone) | Does not matter | Traffic is dropped. |
| **Yes (zone A)** | Yes (zone A) | Does not matter | Traffic is forwarded |
| **Yes (zone A)** | Yes (zone B) | No | Traffic is dropped |
| **Yes (zone A)** | Yes (zone B) | Yes | Policy is applied. If policy is inspect or passes, the initial traffic is forwarded.<br>If the policy is drop the initial traffic is dropped. |

If there is a zone pair that identifies traffic between two zones and the policy is not applied to the zone pair, the default behaviour is to **drop** traffic as if **no zone pair** even **existed**.

**Configuration that includes the following ZBF components:**

- Zones
- Interfaces that are members of zones
- Class maps that identify traffic
- Policy maps that use class maps to identify traffic and then specify the actions which should take place
- Zone pairs, which identify a unidirectional traffic flow, beginning from devices in one zone and being routed out an interface in a second zone
- Service policy, which associates a policy map with a zone pair

```
! The class map "classifies" or "identifies" the traffic
! In this example, this class map will match on either TELNET traffic or
! any type of ICMP traffic
R3(config)# class-map type inspect match-any MY-CLASS-MAP
R3(config-cmap)# match protocol telnet
R3(config-cmap)# match protocol icmp
R3(config-cmap)# exit
```

```
! policy action.   In this example, it is to inspect the traffic
R3(config)# policy-map type inspect MY-POLICY-MAP
R3(config-pmap)# class type inspect MY-CLASS-MAP
R3(config-pmap-c)# inspect
R3(config-pmap-c)# exit
R3(config-pmap)# exit

! Next we create the security zones, they can be named whatever you want to
! name them.  In this example, I named them inside and outside.
R3(config)# zone security inside
R3(config-sec-zone)# exit
R3(config)# zone security outside
R3(config-sec-zone)# exit

! Create the zone-pair, specifying the zones and the direction (from where
! to where)
R3(config-sec-zone)# zone-pair security in-to-out source inside destination outside

! Use the service-policy command in zone-pair configuration mode to apply
! the policy map you want to use for traffic that matches this zone-pair
R3(config-sec-zone-pair)# service-policy type inspect MY-POLICY-MAP
R3(config-sec-zone-pair)# exit

! Configure the interfaces, so they become members of the respective zones
R3(config)# interface GigabitEthernet3/0
R3(config-if)# description Belongs to outside zone
R3(config-if)# zone-member security outside
R3(config-if)# exit
R3(config)# interface GigabitEthernet1/0
R3(config-if)# description  Belongs to inside zone
R3(config-if)# zone-member security inside
R3(config-if)# exit
R3(config)#
```

The preceding policy performs **stateful inspection** for **traffic** from the **inside users** for traffic **going** to the **Internet** if that traffic is **Telnet** traffic (which is TCP port 23) or is ***Internet Control Message Protocol*** *(ICMP)* traffic.

**ACLs** can be used by the **class map** for matching and generic protocol matches such as *User Datagram Protocol (UDP)* or *Transfer Control Protocol (TCP)*. **Application-specific matching** adds the ability for the firewall to detect additional communication channels that may be initialized by the outside devices, such as in the case of inspecting **FTP**, where the server may initiate the data connection on a port mutually agreed to by the c lient and the FTP server.

**The Self-Zone**

**Traffic directed** to the **router** itself (as opposed to traffic going through the router as transit traffic that is not destined directly to the router) **involves** the **self zone**.

**Traffic destined** to **Unknown** the **router**, regardless of which interface is used, is considered to be **going** to the **self zone**.

**Traffic** being **sourced from** the **router** is considered to be **coming from** the **self zone**. By **default**, **all traffic** to the **self zone** or from the **self zone** (which really means all traffic from the router or to the router) is **allowed**. However, if you want to create policies related to traffic to or from this self zone, you do it the same way by creating zone pairs and assigning a policy to the zone pair.

| Source Traffic Member of Zone | Destination Traffic Member of Zone | Zone Pair Exists, With a Policy Applied | Result |
|---|---|---|---|
| **Self** | Zone A | No | Traffic is passed |
| **Zone A** | Self | No | Traffic is passed |
| **Self** | Zone A | Yes | Policy is applied |
| **Zone A** | Self | Yes | Policy is applied |

Regarding the **self zone**, if there is a **zone pair** but **no policy** is **applied**, the **default** behavior is to **forward all traffic** (which is different from the traffic between manually created zones). When configuring a zone pair that includes the self zone, the **administrator must allow management traffic** to be allowed so as to prevent administrative connections from being denied.

# Chapter 16: Configuring Basic Firewall Policies on Cisco ASA

**ASA Features and Services**

- **Packet Filtering:** Simple packet filtering normally represents an access list.
- **Stateful Filtering:** By Default, the ASA enters stateful tracking information about packets that have been initially allowed through the firewall.
- **Application inspection/awareness:** With application layer inspection, the ASA learns about the dynamic ports that were agreed to and dynamically allows the data connection to be initiated from the server that is on the outside going to the client on the inside.

**Network Address Translation (NAT):**

**DHCP:** The ASA can act as a DHCP server or client or both.

**Routing:** The ASA supports most of the interior gateway routing protocols, including RIP, EIGRP and OSPF. It also supports static routing.

**Layer 3 or Layer 2 Implementation:** Can be implemented as a Layer 3 firewall or as a transparent firewall.

**VPN Support:** The ASA can operate as either the head-end or remote-end device for VPN tunnels.

**Object Groups:** An object group is a configuration item on the ASA that refers to one or more items. In the case of a network object group, it refers to one or more IP addresses or network address ranges.

**Botnet Traffic Filtering:** The ASA works with an external system at Cisco that provides information about the Botnet Traffic Filter Database and so can protect against.

**Advanced Malware Protection (AMP):** The Cisco ASA provides next-generation firewall (NGFW) capabilities that combine traditional firewall features with thread and advanced malware protection in a single device.

**High Availability:** By using two firewalls in a high-availability failover combination, you can implement protection against a single system failure.

**AAA Support:** The use of AAA services either locally or from an external server such as ACS, is supported.

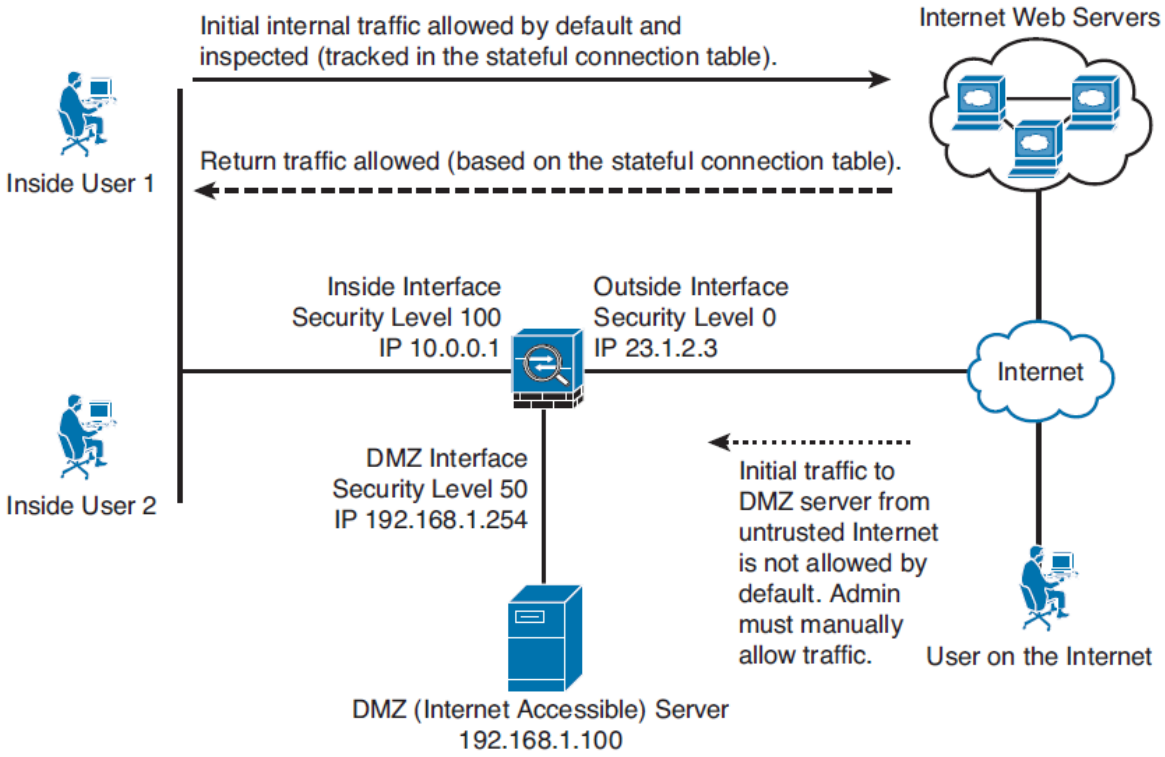
**ASA Firewall Fundamentals**

**ASA Security Levels**

The ASA uses security levels associated with **each routable interface**. The security level is a number between **0** and **100**. The bigger the number, the more trust you have for the network that the interface is connected.

**Making ASA interfaces operational:**

- Assign a security level to the interface
- Assign a name to the interface
- Bring up the interface with the no shutdown command




**The Default Flow of Traffic**

By default, the ASA forwards traffic if the initial traffic is sourced from a device that lives off its high-security interface and if the destination of the packet is being routed out of an interface that has a lower security level.

By default, the firewall is stopping all initial traffic that is trying to go from lower security to higher security levels.

By Default, if two interfaces are both at the exact same security level, traffic is not allowed between those two interfaces.

**Tools to Manage the ASA**

- Command-Line Interface(CLI)
- ASA Security Device Manager (ASDM)
- Cisco Security Manager (CSM)

**Packet Filtering on the ASA**

- Inbound to an interface:
- Inbound from a security level perspective:
- Outbound to an interface:
- Outbound from a security level perspective:
- Implementing Additional Firewall Interfaces

**Configuring the ASA as a DHCP Server for Inside Clients**

```
!  specifies the pool range, enables the feature and specifies the
!  interface
ASA1(config)# dhcpd address 10.0.0.101-10.0.0.132 inside
ASA1(config)# dhcpd enable inside
ASA1(config)# dhcpd dns 10.8.8.8  interface inside
ASA1(config)# dhcpd domain example.org interface inside
```

**Adding a Static or a Default Route**

```
!  this tells the  ASA  that the default route will use the next hop of
!  23.1.2.7
!  which is located off of the outside interface (on that same subnet)
ASA1(config)# route outside 0.0.0.0 0.0.0.0 23.1.2.7
```

**Dynamic NAT**

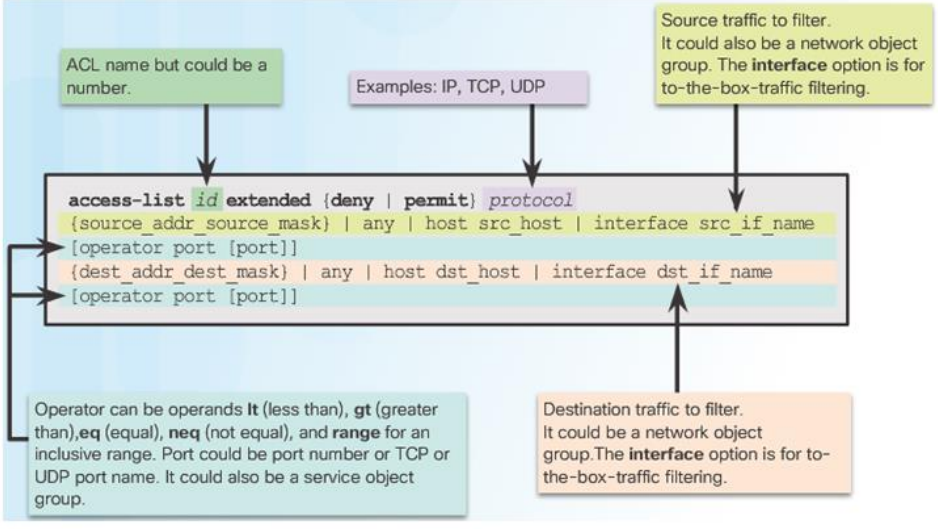
```
!  creates a network object that refers to the 10.0.0.0/24 network
ASA1(config)# object network Inside_Hosts
ASA1(config-network-object)# subnet 10.0.0.0 255.255.255.0
ASA1(config-network-object)# description Inside_Hosts
ASA1(config-network-object)# exit
!  creates a NAT rule that says any traffic sourced from devices
!  from the Inside_Hosts object group (network the 10.0.0.0/24 network),
!  and coming in on the inside interface, as well as exiting (being routed
!  through) the outside interface (based on the routing table of the ASA),
!  it would then translate the source address of these packets,  and
!  substitute the source address of the outside interface of the ASA.
!  Additionally it would track this in a NAT/PAT  table,  that is separate
!  from the stateful  database, and the ASA would manage both of these
!  tables.
ASA1(config)# nat (inside,outside) 1 source dynamic Inside_Hosts interface
!  With the NAT on version 8.3 and newer, there are multiple options of
!  configuring the NAT, including a NAT command done within object group
!  configuration mode.  These additional options, including advanced ASA NAT
!  configuration are covered in the CCNP Security curriculum.
```

**Creating and Applying an ACL at the CLI**

```
ASA1(config)# access-list inside_access_in deny tcp any any eq telnet
ASA1(config)# access-list inside_access_in permit ip any any
ASA1(config)# access-group inside_access_in in interface inside
! Note: the optional elements of line number, and extended are optional.
! The ASA assumes the ACL as an extended (if the keyword "standard" isn't
! used)
! In the absence of a "line" command, the ASA adds new entries to the end
! of the ACL
! To apply the ACL, the ASA uses a global access-group command, which is
! different than on an IOS router, where applying an ACL is done in
! interface configuration mode.
```

## Configuring ACLs (Cont.)

Condensed Extended ACL Syntax



# Chapter 17: Cisco IDS/IPS Fundamentals

**IPS vs IDS**

**What Sensors Do**

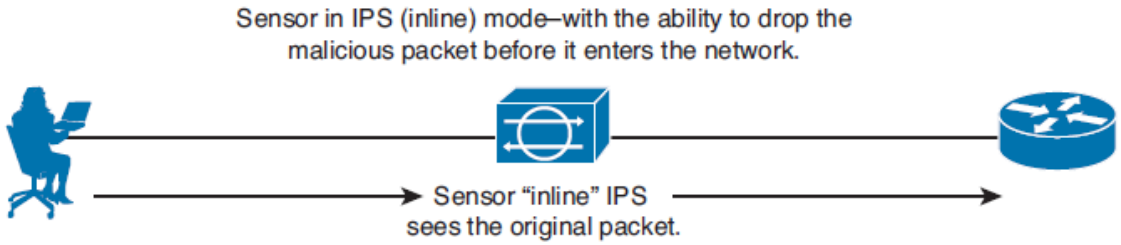
A **Sensor** is a **device** that looks at **traffic** on the network and then makes a decision based on a set of **rules** to indicate wheatear that traffic is okay or wheatear it is malicious.

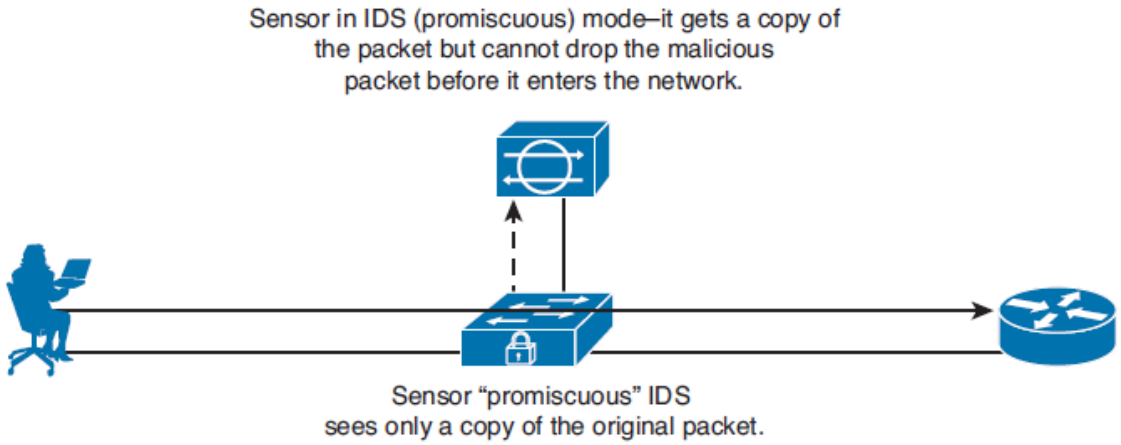
**Difference between IPS and IDS**

You can place a sensor in the network to analyse network traffic in one of two ways:

**Inline with the traffic:** Any traffic going through your network is forced to go in one physical or logical port on the sensor. Ex Intrusion Prevention System (IPS). If the sensor fails and you do not have an alternative path, the entire network could fail as a result of the sensor problems.

The IDS is similar to IPS but instead of placing it inline it operates in promiscuous mode and send copies of the packets that are going through a network to the IDS sensor. This sill generates alerts but we cannot deny the packet but only detect it. One benefit of IDS is that is has no delay.



**IDS versus IPS**

| | IDS | IPS |
|---|---|---|
| **Position in the network flow** | Not inline with the flow of network traffic, the IDS is sent copies of the original packets | Directly inline with the flow of network traffic and every packet goes through the sensor on its way through the network |
| **Mode** | Promiscuous mode, out of band | Inline mode |
| **Latency or Delay** | Does not add delay to the original traffic because it is not inline | Adds a small amount of delay before forwarding it through the network |
| **Impact caused by the sensor failing to forward packets** | There is no negative impact | Traffic that would normally flow through the sensor could be impacted, dependent of “fail open” or “fail closed” configuration |
| **Ability to prevent malicious traffic from going into the network** | By itself, a promiscuous mode IDS cannot stop the original packet. Options exist for a sensor in promiscuous mode to request assistance from another device that is inline witch may block packets. An IDS can send TCP Reset packets to break malicious connections but there is no guarantee | The IPS can drop the packet on its own because it is inline. The IPS can also request assistance from another device to block future packets just a the IDS. |
| **Normalization ability** | Since IDS does not see the original packet, it cannot manipulate any original inline traffic | Since IPS is inline, it can normalize traffic inline based on a current set of rules |

**Positive/Negative Terminology**

- False positive
- False negative
- True positive
- Trust negative

**Identifying Malicious Traffic on the Network**

There are several different methods that sensors can be configured to use to identify malicious traffic, including the following:

- Signature-Based IPS/IDS
- Policy-Based IPS/IDS
- Anomaly-Based IPS/IDS
- Reputation-Based IPS/IDS

**Signature-Based IPS/IDS**

A Signature is just a **set of rules** looking for some **specific pattern** or **characteristic** in either a single packet or a stream of packets. A new sensor may have thousands of default signatures provided by Cisco. Not all signatures are enabled but the administrator can enable, disable, customize and create new signatures.

**Policy-Based IPS/IDS**

This type of **traffic matching** can be implemented based on the security policy for your network. Example creating a custom rule that states that if TCP traffic is seen destined to port 23 to a device in the network, the IPS can generate an alert and drop the packet.

**Anomaly-Based IPS/IDS**

Examples, creating a **baseline** of how many TCP sender requests are generated on average each minute that do not get a response.

**Reputation-Based IPS/IDS**

**Collects input** from **systems** all over the world that are participating in global correlation, so what other sensors have learned collectively, your local sensor can use locally.

**IPS/IDS Method Advantages and Disadvantages**

| | Advantages | Disadvantages |
|---|---|---|
| **Signature-based** | Easy to configure, simple to implement | Does not detect attacks outside of the rules.<br>May need to disable signatures that are creating false positives.<br>Signatures must be updated periodically to remain current and effective against new threats. |
| **Policy-based** | Simple and reliable, very customizable, only allows policy based traffic that could dent unknown attacks, which by default are outside of the policy being allowed | Policy must be manually crated.<br>Implementation of the policy is only as good as the signatures you manually create. |
| **Anomaly-based** | Self-configuring baselines, detect worms based on anomalies, even if specific signatures have not been created yet for that type of traffic. | Difficult to accurately profile extremely large networks.<br>May cause false positives based on significant changes in valid network traffic. |
| **Reputation-based** | Leverages enterprise and global correlation, providing information based on the experience of other systems. Early-warning system. | Requires timely updates and required participation in the correlation process. |

**Possible Sensor Responses to Detected Attacks**

- **Deny attacker inline**: denies packets from the source IP address of the attacker for a configurable duration of time, after which the deny action can be dynamically removed.

- **Deny connection inline**: terminates the packet that triggered the action and future packets that are part of the same TCP connection

- **Deny packet inline**: terminates the packet that triggered the event

- **Log attacker (source) packets**: begins to log future packets based on attacker's source IP address

- **Log victim (destination) packets**: begins to log all IP packets with a destinations address of the victim

- **Log pair (source, destination) packets**

- **Produce alert**: This is the default behavior for most signatures enabled on a sensor.

- **Produce verbose alert**: same as above plus it includes a copy of the packets that triggered the alert

- **Request block connection**: This action causes the sensor to request a blocking device to block based on the source IP address of the attacker, the destination IP address of the victim, and the ports involved in the packet that triggered the alert.

- **Request block host**: blocks the attacker's/destination's IP address regardless of the port used

- **Request SNMP trap**

- **Reset TCP connection**: send a proxy TCP reset to the attacker.

**Controlling Which Actions the Sensors Should Take**

This is implemented using a calculated result called a **risk rating**. The **maximum value** for risk rating is **100**. As the administrator, you can choose which countermeasure to take based on the risk rating that triggers an alert.

There are **three primary factors**, or influencers, of the final risk rating value:

- **Signature fidelity rating (SFR)**: The accuracy of the signature as determined by the person who created that signature
- **Attack severity rating (ASR)**: The criticality of the attack as determined by the person who created that signature
- **Target value rating (TVR)**: The value that you, as an administrator, have assigned to specific destination IP addresses or subnets where the critical servers/ devices live.

**Additional factors:**

- **Attack relevancy (AR):** A signature match that is destined to a host where the attack is relevant, such as a Windows server–based attack, which is going to the destination address of a known Windows server, is considered a relevant attack, and the risk rating increases slightly as a result.

- **Global correlation:** If the sensor is participating in global correlation and receives information about specific source addresses that are being used to implement large-scale attacks, attacks coming from these source IP addresses are also given a slightly increased risk rating value.

**Implementing Actions Based on the Risk Rating**

Although it is true that you can implement actions as properties of individual signatures, it makes the most sense, and it is much more scalable to manage, to configure actions based on the risk rating that is created as a result of the signature matches.

For example, you can specify severe countermeasures if a risk rating is generated that is 90 or higher. A risk rating of 50 or lower may simply be configured to generate an alert but not cause a severe countermeasure, such as deny attacker, to be implemented.

**IPS/IDS Evasion Techniques**

- **Traffic fragmentation:** the attacker splits malicious traffic into multiple parts; IPS/IDS does complete session reassembly to see the entire traffic
- **Traffic substitution and insertion:** The attacker substitutes characters in the data using different formats that have the same final meaning
- **Protocol level misinterpretation: Cisco** does TTL analysis and TCP checksum validation
- **Timing attacks** (for example, "low and slow" attacks): attacker sending packets at lower packets per second
- **Encryption and tunneling**
- **Resource exhaustion:** If thousands of alerts are being generated by distracter attacks; dynamic and configurable summarization.

**Managing Signatures**

The most effective way to **identify malicious traffic** in the Cisco IPS/IDS systems is through the use of **signature-based matching**. Cisco organizes its signatures into groups that have similar characteristics. For each of its groups, a signature micro-engine is used to govern that set of signatures.

**Micro-Engines (Groupings of Signatures)**

- **Atomic:** Signatures that can match on a single packet, as compared to a string of packets
- **Service**: Signatures that examine application layer services, regardless of the operating system
- **String or Multistring:** Supports flexible pattern matching and can be identified in a single packet or group of packets, such as a session
- **Other:** Miscellaneous signatures that may not specifically fit into the previously mentioned other categories

**Signature or Severity Levels**

Instead of having to set a numeric value for the severity, the interface for IPS/IDS prompts us for one of four levels:

- Informational
- Low
- Medium
- High

**Monitoring and Managing Alarms and Alerts**

**Three main protocols** are used in delivering alerts. They are **Security Device Event Exchange (SDEE)**, **syslog**, and **SNMP**. SDEE is used for real-time delivery of alerts, and is the most secure method for delivering alerts. Applications: IPS Manager Express (IME), Cisco Security Manager (CSM).

**Security Intelligence**

So, in short, the more sensors you have reporting, the more granular and complete the information is going to be about the attacks and the patterns that exist in the network. With global correlation, we can increase the risk rating for specific attacks if they are from source addresses that we identified as suspect in information learned from external sensors through the global correlation process. Global correlation is available on the sensor appliances but does not have to be enabled – Cisco Security Intelligence Operations (SIO)

**IPS/IDS Best Practices**

- Implement an IPS so that you can analyze traffic going to your critical servers and other mission-critical devices, or the "crown jewels" for your organization.
- If you cannot afford dedicated appliances, use modules or IOS software-based IPS/IDS
- Take advantage of global correlation to improve your resistance against attacks that may be targeting your organization
- Use a risk-based approach, where countermeasures occur based on the calculated risk rating as opposed to manually assigning countermeasures to individual signatures
- Use automated signature updates when possible instead of manually installing updates
- Continue to tune the IPS/IDS infrastructure as traffic flows and network devices and topologies change

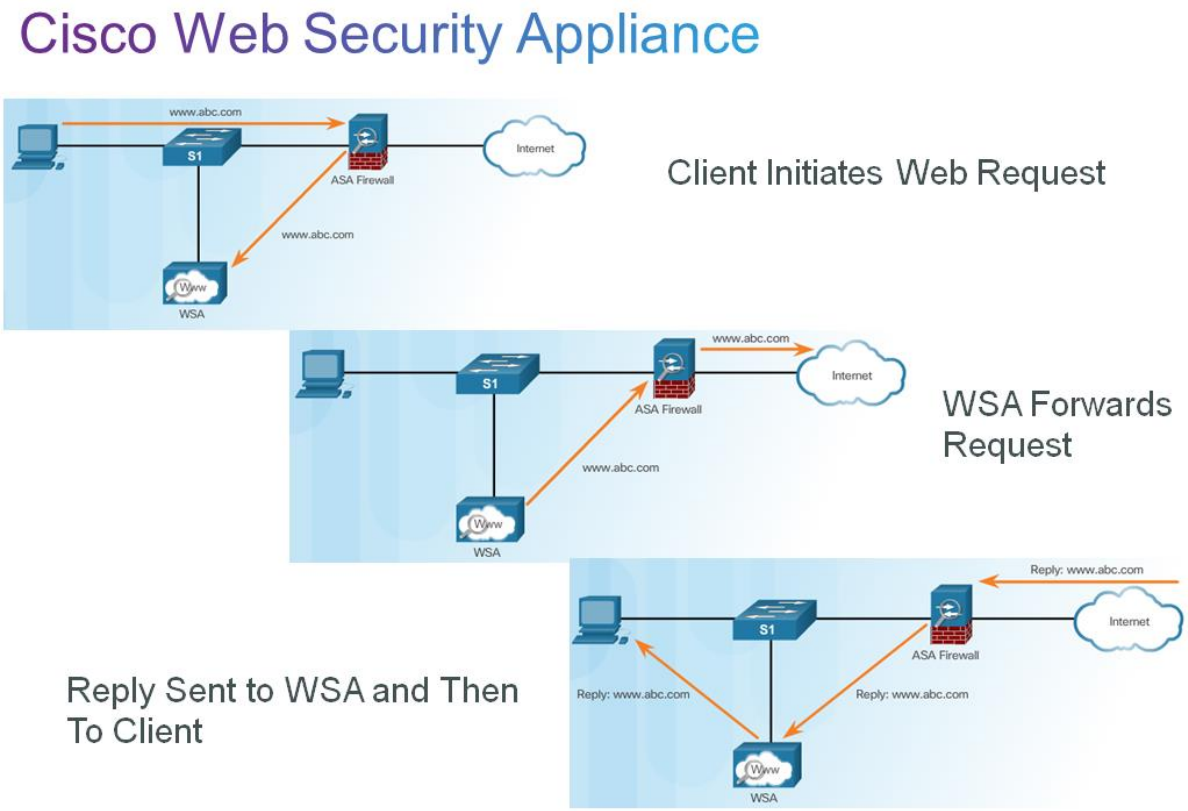
**Cisco Next-Generation IPS Solutions**

- Cisco FirePOWER 8000/7000 series appliances
- Virtual Next-Generation IPS (NGIPSv) for VMware
- ASA with FirePOWER Services
- FireSIGHT Management Center

# Chapter 18: Mitigation Technologies for E-Mail-Based and Web-Based Threats

**Mitigation Technologies for E-mail Based and Web-Based Threats**

The Cisco **E-mail Security Appliances** (**ESA**) and the Cisco **Web Security Appliance** (**WSA**) provide a great solution designed to protect corporate users against these threats. Cisco has added **Advanced Malware Protection** (**AMP**) to the ESA and WSA to allow security administrators to detect and block malware and perform continuous analysis and retrospective alerting.



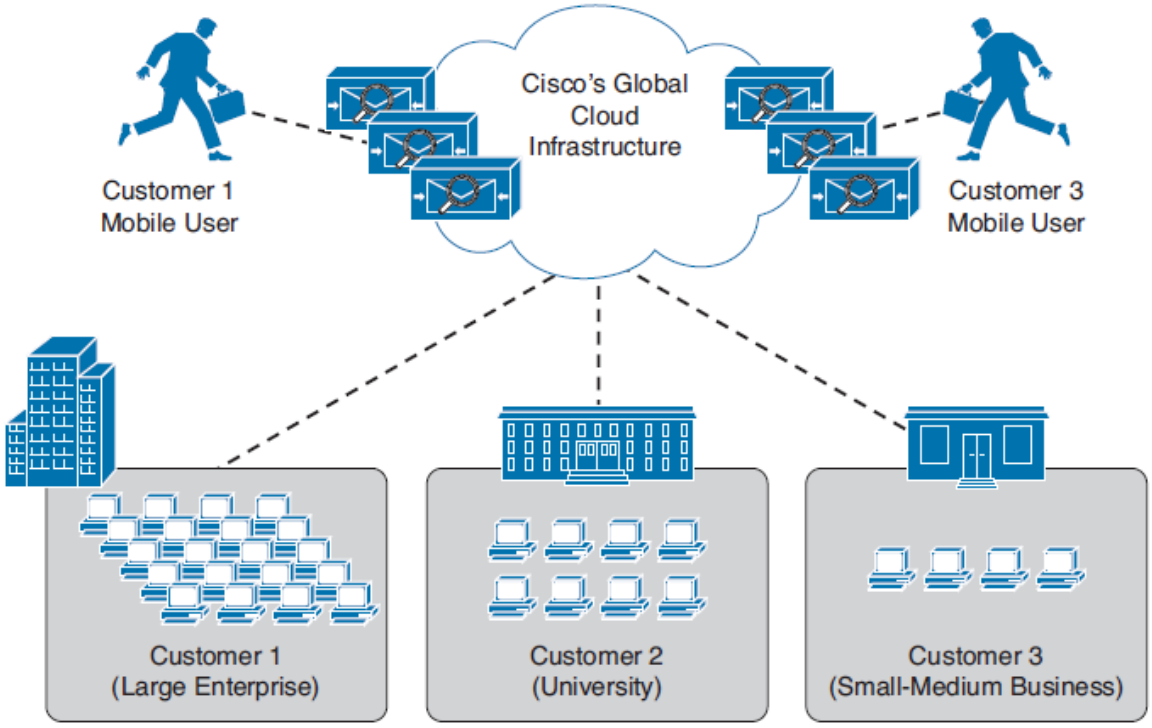

**Mitigation Technology for E-mail-Based Threats**

**E-mail-Based Threats**

- **Spam**: unsolicited e-mail messages that can be advertising a service or (typically) a scam or a message with malicious intent
- **Malware attachments**: mail messages containing malicious software
- **Phishing**: an attacker's attempt to fool a user that such e-mail communication comes from a legitimate entity or site, such as banks, social media websites, online payment processors, or even corporate IT communications.
- **Spear phishing**: These phishing e-mails are directed to specific individuals or organizations

**Cisco Cloud E-mail Security**

Cisco cloud e-mail security provides a cloud-based solution that allows companies to outsource the management of their e-mail security management.




**Cisco Hybrid E-mail Security**

The Cisco hybrid e-mail security solution combines both cloud-based and on-premises ESAs. This hybrid solution helps Cisco customers reduce their on-site e-mail security footprint, outsourcing a portion of their e-mail security to Cisco, while still allowing them to maintain control of confidential information within their physical boundaries.

**Cisco E-mail Security Appliance**

- **Cisco X-Series E-mail Security Appliances**
  - Cisco X1070: High-performance ESA for service providers and large enterprises
- **+ Cisco C-Series E-mail Security Appliances**
  - Cisco C680: The high-performance ESA for service providers and large enterprises
  - Cisco C670: Designed for medium-size enterprises
  - Cisco C380: Designed for medium-size enterprises
  - Cisco C370: Designed for small- to medium-size enterprises
  - Cisco C170: Designed for small businesses and branch offices

**Features supported by the Cisco ESA:**

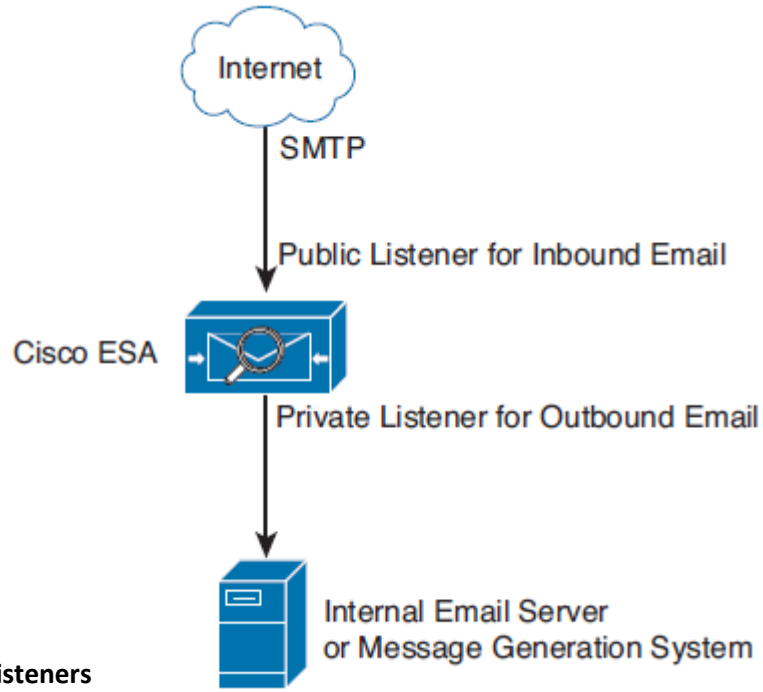
- **Access control:** Controlling access for inbound senders according to the sender's IP address, IP address range, or domain name.
- **Antispam:** Multilayer filters based on Cisco Sender Base reputation and Cisco Antispam integration
- **Network Antivirus**
- **Advanced malware protection (AMP**): Allows security administrators to detect and block malware and perform continuous analysis and retrospective alerting
- **DLP:** The ability to detect any sensitive e-mails and documents leaving the corporation

- **E-mail encryption:** The ability to encrypt outgoing mail to address regulatory requirements. The administrator can configure an encryption policy on the Cisco ESA and use a local key server or hosted key service to encrypt the message.
- **E-mail authentication:** Email authentication mechanisms such as: Sender Policy Framework (SPF), Sender ID Framework (SIDF), and DomainKeys Identified Mail (DKIM) verification of incoming mail, as well as DomainKeys and DKIM signing of outgoing mail.
- **Outbreak filters:** Preventive protection against new security outbreaks and e-mail-based scams with SIO

The **Cisco ESA** acts as the e-mail gateway to the organization, handling all e-mail connections, accepting messages, and relaying them to the appropriate systems. The Cisco ESA uses listeners to handle incoming SMTP connection requests.

A listener defines an e-mail processing service that is configured on an interface in the Cisco ESA. The following listeners can be configured:

- Public listeners for e-mail coming in from the Internet
- Private listeners for e-mail coming from hosts in the corporate (inside) network




**Cisco ESA Listeners**

Cisco ESA listeners are often referred to as ***SMTP daemons*** running on a specific Cisco ESA interface. When a listener is configured, the following information must be provided:

- Listener properties such as a **specific interface** in the Cisco ESA and the **TCP port** that will be used. The listener properties must also indicate whether it is a **public** or a **private** listener.
- The hosts that are allowed to **connect** to the **listener** using a combination of **access control rules**. An administrator can specify which remote hosts can connect to the listener.
- The **local domains** for which public listeners **accept** messages.

**Cisco ESA Initial Configuration**

- Step 1. Log in to the Cisco ESA. The default username is admin, and the default password is ironport
- Step 2. Use the **systemsetup** command in CLI of the Cisco ESA to initiate the System Setup Wizard

In this example, the **inside** (private) and **outside** (public) listeners are configured. The domain name of securemeinc.org is used.

```
IronPort> systemsetup
WARNING: The system setup wizard will completely delete any existing
'listeners' and all associated settings including the 'Host Access Table' - mail
operations may be interrupted.
Are you sure you wish to continue? [Y]> Y

You are now going to configure how the IronPort C60 accepts mail by
creating a "Listener".

Please create a name for this listener (Ex: "InboundMail"):
[]> InboundMail

Please choose an IP interface for this Listener.
1. Management (192.168.42.42/24: mail3.example.com)
2. PrivateNet (192.168.1.1/24: mail3.example.com)
3. PublicNet (192.168.2.1/24: mail3.example.com)

[1]> 3
Enter the domains or specific addresses you want to accept mail for.
Hostnames such as "example.com" are allowed.

Partial hostnames such as ".example.com" are allowed.

Usernames such as "postmaster@" are allowed.

Full email addresses such as "joe@example.com" or "joe@[1.2.3.4]" are allowed.
Separate multiple addresses with commas

[]> securemeinc.org
Would you like to configure SMTP routes for example.com? [Y]> y

Enter the destination mail server which you want mail for example.com to be delivered.

Separate multiple entries with commas.
[]> exchange.securemeinc.org
```

```
Do you want to enable rate limiting for this listener? (Rate limiting defines the
maximum

number of recipients per hour you are willing to receive from a remote domain.) [Y]> y

Enter the maximum number of recipients per hour to accept from a remote domain.
[]> 4500

Default Policy Parameters
==========================
Maximum Message Size: 100M
Maximum Number Of Connections From A Single IP: 1,000
Maximum Number Of Messages Per Connection: 1,000
Maximum Number Of Recipients Per Message: 1,000
Maximum Number Of Recipients Per Hour: 4,500
Maximum Recipients Per Hour SMTP Response:
 452 Too many recipients received this hour
Use SenderBase for Flow Control: Yes
Virus Detection Enabled: Yes
Allow TLS Connections: No
Would you like to change the default host access policy? [N]> n
Listener InboundMail created.
Defaults have been set for a Public listener.

Use the listenerconfig->EDIT command to customize the listener.
*****

Do you want to configure the C60 to relay mail for internal hosts? [Y]> y

Please create a name for this listener (Ex: "OutboundMail"):
[]> OutboundMail

Please choose an IP interface for this Listener.
1. Management (192.168.42.42/24: mail3.example.com)
2. PrivateNet (192.168.1.1/24: mail3.example.com)
3. PublicNet (192.168.2.1/24: mail3.example.com)

[1]> 2

Please specify the systems allowed to relay email through the IronPort C60.

Hostnames such as "example.com" are allowed.

Partial hostnames such as ".example.com" are allowed.
```

```
Separate multiple entries with commas.
[]> .securemeinc.org

Do you want to enable rate limiting for this listener? (Rate limiting defines the
maximum number of recipients per hour you are willing to receive from a remote
domain.)
[N]> n

Default Policy Parameters
==========================
Maximum Message Size: 100M
Maximum Number Of Connections From A Single IP: 600
Maximum Number Of Messages Per Connection: 10,000
Maximum Number Of Recipients Per Message: 100,000
Maximum Number Of Recipients Per Hour: Disabled
Use SenderBase for Flow Control: No
Virus Detection Enabled: Yes
Allow TLS Connections: No
Would you like to change the default host access policy? [N]> n
Listener OutboundMAil created.
Defaults have been set for a Private listener.
Use the listenerconfig->EDIT command to customize the listener.
*****

Congratulations! System setup is complete. For advanced configuration, please refer to
  the User Guide.
mail3.securemeinc.org >
```

**Verifying the Configuration with the mailconfig Command**

```
mail3.securemeinc.org> mailconfig

Please enter the email address to which you want to send
the configuration file. Separate multiple addresses with commas.

[]> admin@securemeinc.org

The configuration file has been sent to admin@securemeinc.org.

mail3.securemeinc.org>
```

**Mitigation Technology for Web-Based Threats**

The core solutions for **mitigating web-based threats** are the **Cisco Cloud Web Security** (**CWS**) offering and the integration of **advanced malware protection** (**AMP**) to the **Cisco Web Security Appliance** (**WSA**).

**Cisco CWS**

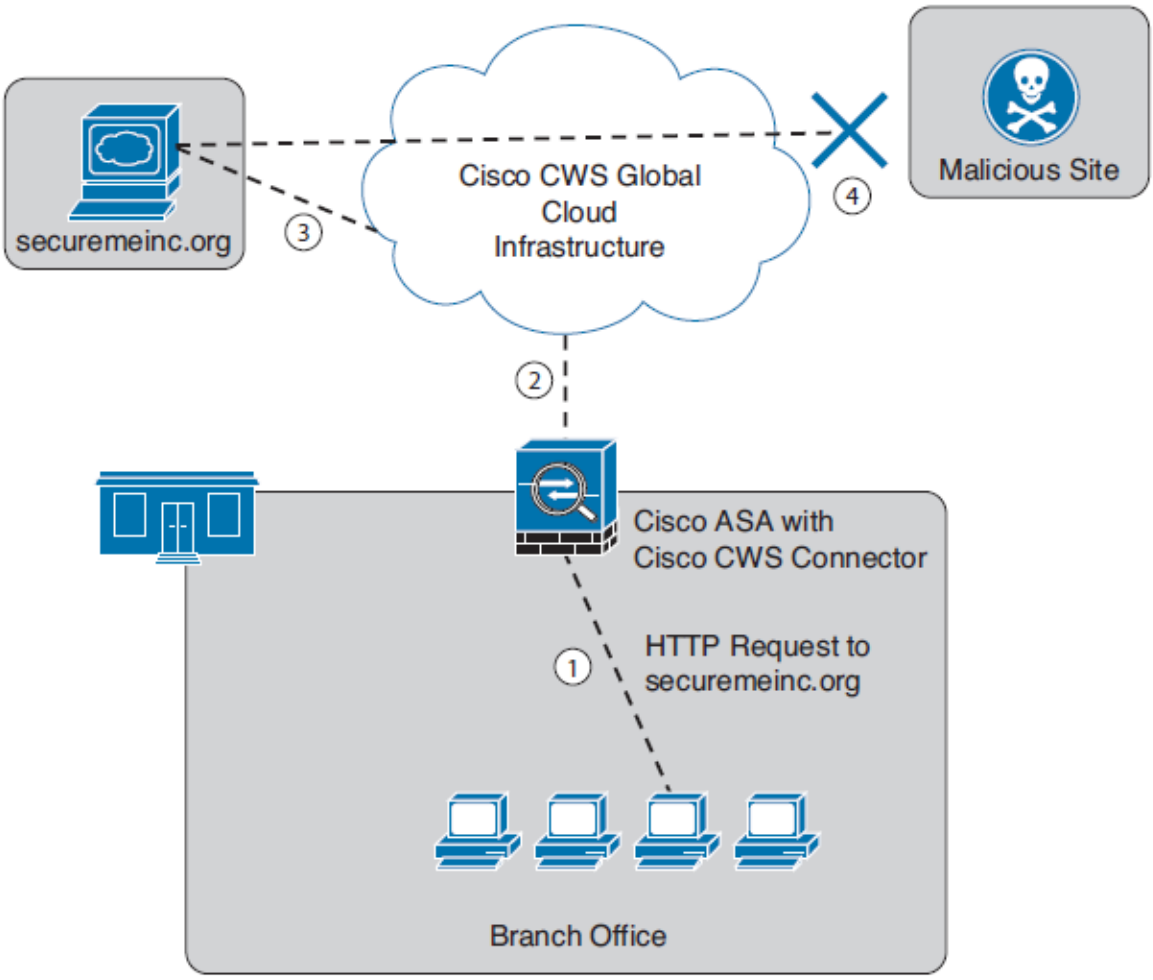
Cisco CWS is a **cloud-based security service** from Cisco that provides worldwide threat intelligence, advanced threat defense capabilities, and roaming user protection. Cisco customers can connect to the Cisco CWS service directly by using a proxy autoconfiguration (PAC) file in the user endpoint or through connectors integrated into the following Cisco products:

- Cisco ISR G2 routers
- Cisco ASA
- Cisco WSA
- Cisco AnyConnect Secure Mobility Client

Organizations using the transparent proxy functionality through a connector can get the most out of their existing infrastructure. In addition, the scanning is offloaded from the hardware appliances to the cloud, reducing the impact to hardware utilization and reducing network latency.

The picture below shows the Cisco ASA is enabled with the Cisco CWS connector at a branch office. The following steps explain how Cisco CWS protects the corporate users at the branch office:

1. An internal user makes an HTTP request to an external website (securemeinc.org).
2. The Cisco ASA forwards the request to Cisco CWS global cloud infrastructure.
3. It notices that securemeinc.org had some web content (ads) that were redirecting the user to a known malicious site.
4. Cisco CWS blocks the request to the malicious site.



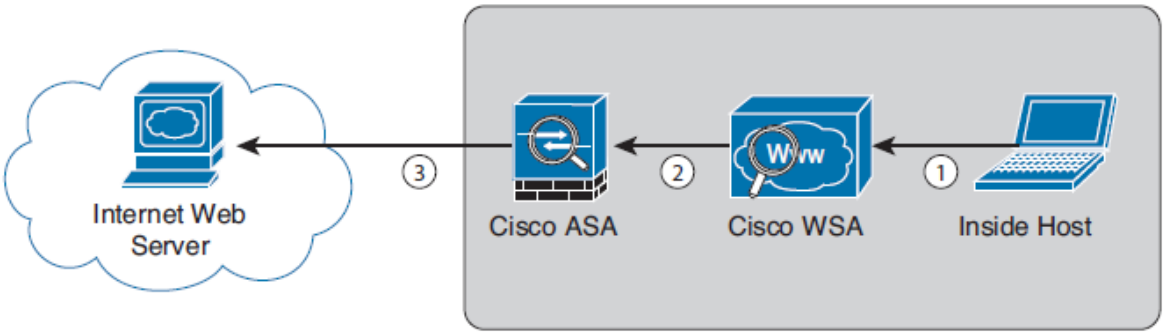
**Cisco WSA**

The **Cisco WSA** uses **cloud-based intelligence** from Cisco to help protect the organization before, during, and after an attack. This threat intelligence helps security professionals to stop threats **before** they enter the corporate network, while also enabling file reputation and file sandboxing to identify threats during an attack. Retrospective attack analysis allows security administrators to investigate and provide protection after an attack when advanced malware might have evaded other layers of defense.

The **Cisco WSA** can be deployed in **explicit proxy mode** or as a **transparent proxy** using the *Web Cache Communication Protocol (WCCP)*.

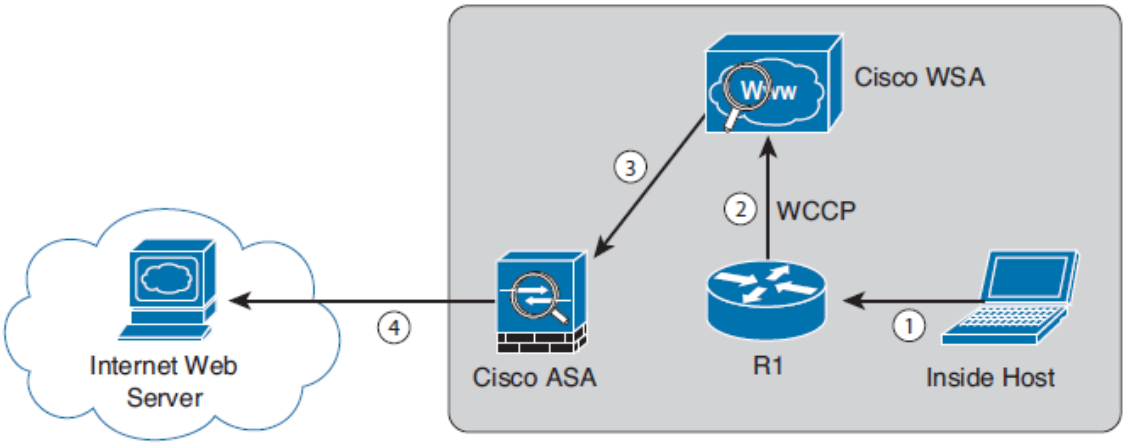
**Explicit Proxy Configuration**

1. An internal user makes an HTTP request to an external website. The client browser is configured to send the request to the Cisco WSA.
2. The Cisco WSA connects to the website on behalf of the internal user.
3. The firewall (Cisco ASA) is configured to only allow outbound web traffic from the Cisco WSA, and it forwards the traffic to the web server.



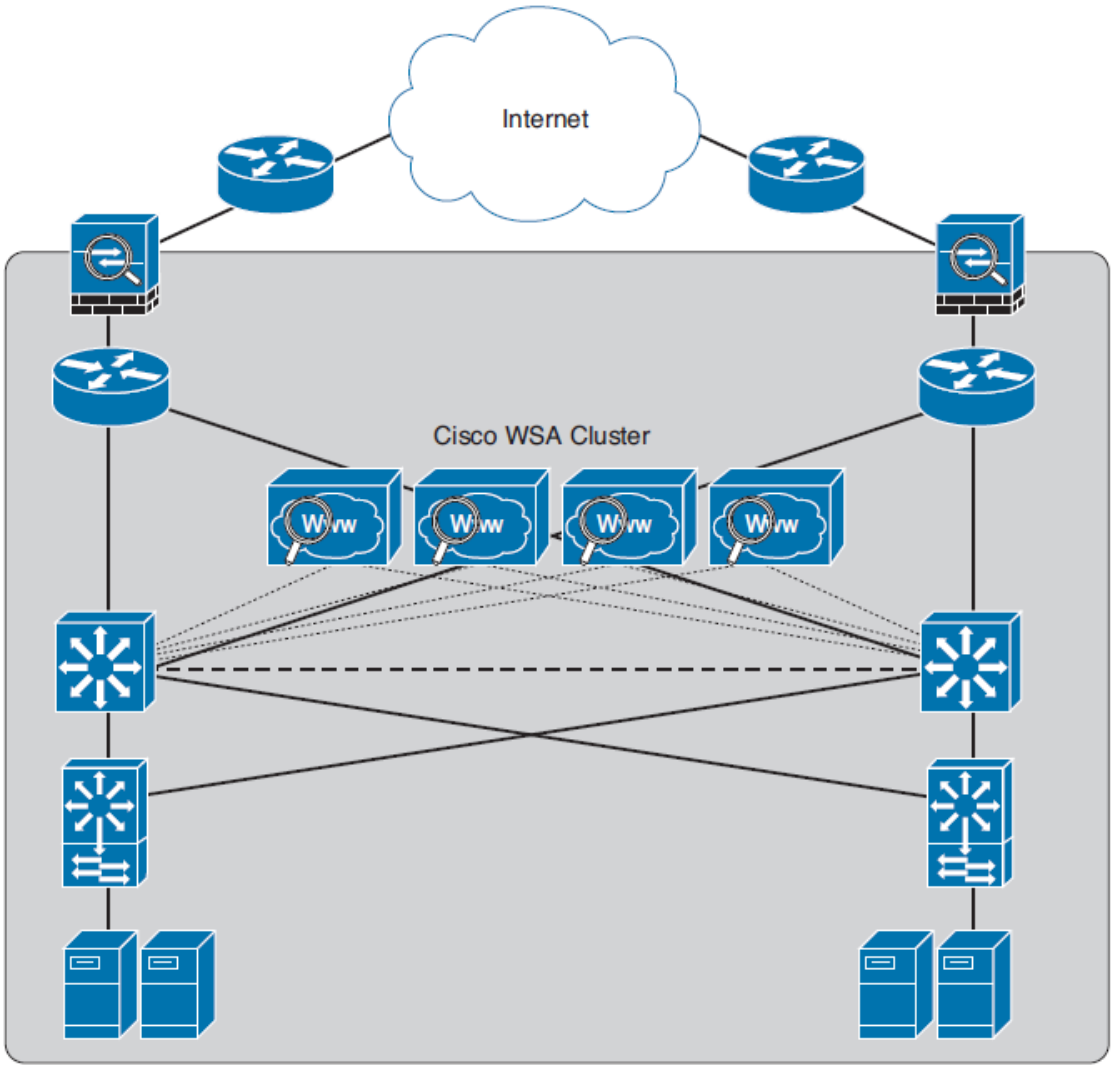

**Transparent Proxy Configuration**

1. An internal user makes an HTTP request to an external website.
2. The internal router (R1) redirects the web request to the Cisco WSA using WCCP.
3. The Cisco WSA connects to the website on behalf of the internal user.
4. Also in this example, the firewall (Cisco ASA) is configured to only allow outbound web traffic from the WSA. The web traffic is sent to the Internet web server.



**WCCP Registration**

During the WCCP registration process:
The **WCCP client** sends a **registration announcement** ("Here I am") every 10 seconds.
The **WCCP server** (the Cisco router in this example) accepts the **registration request** and **acknowledges** it with an "I See You" WCCP message.
The **WCCP server** waits **30 seconds** before it declares the client as "**inactive**" (engine failed).
WCCP can be used in large-scale environments. The picture below shows a cluster of Cisco WSAs, where internal Layer 3 switches redirect web traffic to the cluster.




The following are the different Cisco WSA models:
**Cisco WSA S680**: 6-12k users, 2 rack unit (RU), 2 octa core CPUs, 32GB of memory, 4,8TB of space
**Cisco WSA S670**
**Cisco WSA S380**: 1, 5 to 6k users
**Cisco WSA S370**
**Cisco WSA S170**: up to 1,5k users, 1 RU, 1 dual core CPUs, 4GB of memory, 500GB of space

The Cisco WSA runs Cisco **AsyncOS operating system**. The Cisco AsyncOS supports numerous features that will help mitigate web-based threats.
The following are examples of these features:

■ **Real-time antimalware adaptive scanning:** The Cisco WSA can be configured to dynamically select an antimalware scanning engine based on URL reputation, content type, and scanner effectiveness. Adaptive scanning is a feature designed to increase the "catch rate" of malware that is embedded in images, JavaScript, text, and Adobe Flash files. Adaptive scanning is an additional layer of security on top of Cisco WSA Web Reputation Filters that include support for Sophos, Webroot, and McAfee.

■ **Layer 4 traffic monitor:** Used to detect and block spyware. It dynamically adds IP addresses of known malware domains to a database of sites to block.

■ **Third-party DLP integration:** Redirects all outbound traffic to a third-party DLP appliance, allowing deep content inspection for regulatory compliance and data exfiltration protection. It enables an administrator to inspect web content by title, metadata, and size and to even prevent users from storing files to cloud services, such as Dropbox, Google Drive, and others.

■ **File reputation:** Using threat information from Cisco Talos. This file reputation threat intelligence is updated every 3 to 5 minutes.

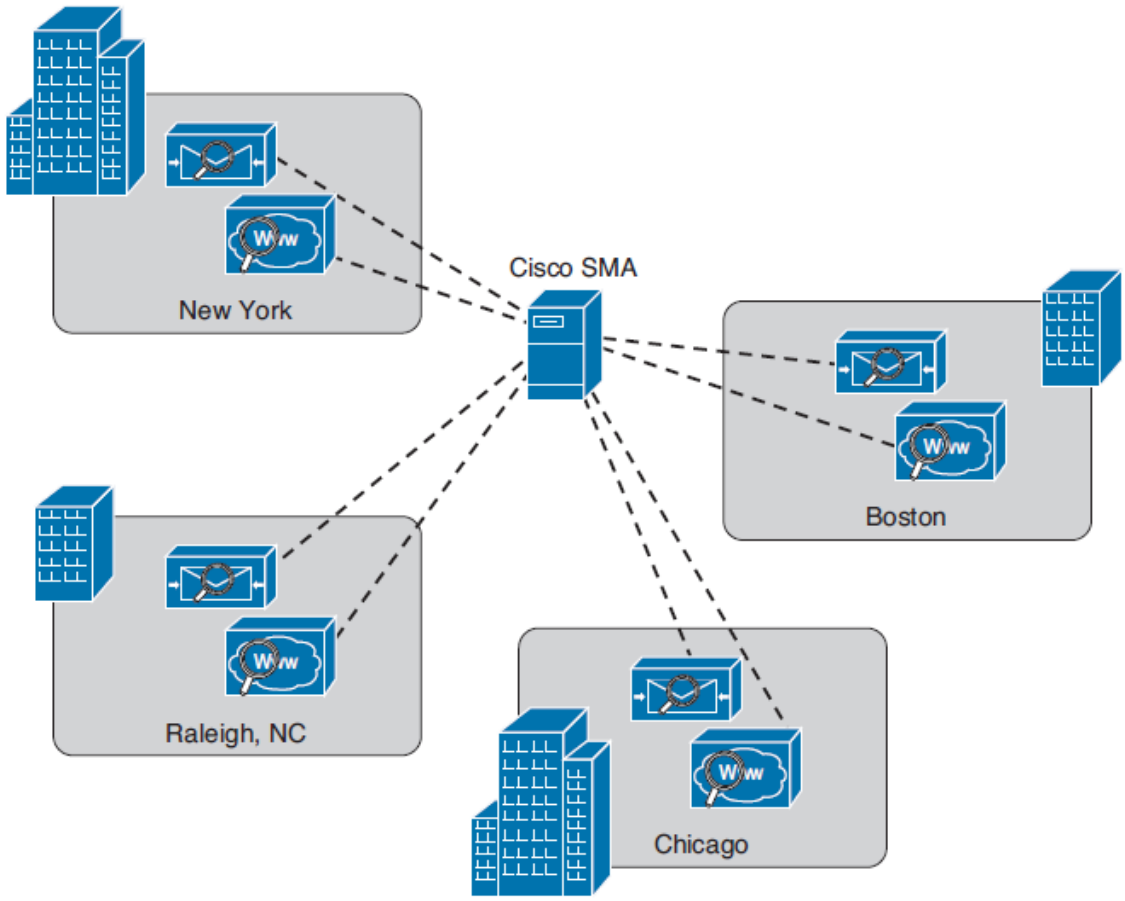
■ **File sandboxing:** If malware is detected, the Cisco AMP capabilities can put files in a sandbox to inspect its behavior, combining the inspection with machine-learning analysis to determine the threat level. Cisco Cognitive Threat Analytics (CTA) uses machinelearning algorithms to adapt over time.

■ **File retrospection:** After a malicious attempt or malware is detected, the Cisco WSA continues to cross-examine files over an extended period of time.

■ **Application visibility and control:** Allows the Cisco ASA to inspect and even block applications that are not allowed by the corporate security polity. For example, an administrator can allow users to use social media sites like Facebook but block micro-applications such as Facebook games.

**Cisco Content Security Management Appliance**

Cisco Security Management Appliance (SMA) is a Cisco product that **centralizes** the **management** and **reporting** for one or more **Cisco ESAs** and **Cisco WSAs**. Cisco SMA has consistent enforcement of policy, and enhances threat protection.




The **Cisco SMA** comes in different models. These models are physical appliances or the Cisco Content ***Security Management Virtual Appliance (SMAV)***.

The following are the different **Cisco SMA models**:

- **Cisco SMA M680:** Designed for large organizations with over 10,000 users
- **Cisco SMAV M600v:** Designed for large enterprises or service providers
- **Cisco SMA M380:** Designed for organizations with 1000 to 10,000 users
- **Cisco SMAV M300v:** Designed for organizations with 1000 to 5000 users
- **Cisco SMA M170:** Designed for small business or branch offices with up to 1000 users
- **Cisco SMAV M100v:** Designed for small business or branch offices with up to 1000 users

# Chapter 19: Mitigation Technologies for Endpoint Threats

**Mitigation Technologies for Endpoint Threats**

**Antivirus and Antimalware Solutions**

The following are the most common types of **malicious software**:

- **Computer viruses:** A malicious software that infects a host file or system area to perform undesirable outcomes such as erasing data, stealing information, or corrupting the integrity of the system
- **Worms:** Viruses that replicate themselves over the network infecting numerous vulnerable systems
- **Mailers and mass-mailer worms:** A type of worm that sends itself in an e-mail message
- **Logic bombs:** A type of malicious code that is injected into a legitimate application
- **Trojan horses:** A type of malware that executes instructions determined by the nature of the Trojan to delete files, steal data, and compromise the integrity of the underlying operating system
- **Back doors:** A piece of malware or configuration change that allows attackers to control the victim's system remotely
- **Exploits:** A malicious program designed to "exploit" or take advantage of a single vulnerability or set
- **Downloaders:** A piece of malware that downloads and installs other malicious content from the Internet to perform additional exploitation on an affected system
- **Spammers:** the act of sending unsolicited messages via e-mail, instant messaging, newsgroups, or any other kind of computer or mobile device communications
- **Key loggers:** A piece of malware that captures the user's keystrokes on a compromised computer or mobile device
- **Rootkits:** A set of tools that are used by an attacker to elevate their privilege to obtain root-level access to be able to completely take control of the affected system
- **Ransomware:** A type of malware that compromises a system and then demands a ransom from the victim to often pay the attacker in order for the malicious activity to cease or for the malware to be removed from the affected system – ex: Crypto Locker and Crypto Wall

**Known antivirus programs:**

- Avast
- AVG Internet Security
- Bitdefender Antivirus Free
- ZoneAlarm PRO Antivirus + Firewall and ZoneAlarm Internet Security Suite
- F-Secure Antivirus
- Kaspersky Anti-Virus
- McAfee Antivirus
- Panda Antivirus
- Sophos Antivirus
- Norton AntiVirus
- ClamAV: sponsored and maintained by Cisco and non-Cisco engineers
- Immunet: a free community-based antivirus software maintained by Cisco Sourcefire

**Personal Firewalls and Host Intrusion Prevention Systems**
Are software applications that you can install on end-user machines or servers to protect them from external security threats and intrusions.

**Advanced Malware Protection for Endpoints**
Cisco AMP for Endpoints provides mitigation capabilities that go beyond point-in-time detection. It uses **threat intelligence** from Cisco to perform retrospective analysis and protection. Cisco AMP for Endpoints also provides device and file trajectory capabilities to allow the security administrator to analyze the full spectrum of the attack.

Cisco acquired a security company called ThreatGRID that provides cloud-based and on-premise malware analysis solutions. Cisco integrated Cisco AMP and ThreatGRID to provide a solution for advanced malware analysis with deep threat analytics.

**Hardware and Software Encryption of Endpoint Data**

**E-mail Encryption**
When people refer to e-mail encryption, they often are referring to encrypting the actual e-mail message so that only the intended receiver can decrypt and read the message.
To **effectively protect** your **e**-**mails**, you should make sure of the following:

- The connection to your e-mail provider or e-mail server is actually encrypted
- Your actual e-mail messages are encrypted
- Your stored, cached, or archived e-mail messages are also protected

**The following are examples of e-mail encryption solutions:**

- Pretty Good Privacy (PGP): requires you to generate a public and private key
- GNU Privacy Guard (GnuPG)
- Secure/Multipurpose Internet Mail Extensions (S/MIME): requires you to install a security certificate on your computer
- Web-based encryption e-mail service like Sendinc or JumbleMe

**Encrypting Endpoint Data at Rest**
Much commercial and free software enables you to encrypt files in an end-user workstation or mobile device. The following are a few examples of free solutions:

- GPG: GPG also enables you to encrypt files and folders on a Windows, Mac, or Linux system
- The built-in MAC OS X Disk Utility: enables you to create secure disk images by encrypting files with AES 128-bit or AES 256-bit encryption
- TrueCrypt: free encryption tool for Windows, Mac, and Linux systems
- AxCrypt: free Windows-only file encryption tool
- BitLocker: Full disk encryption feature included in several Windows operating systems
- Many Linux distributions such as Ubuntu
- MAC OS X FileVault: Supports full disk encryption on Mac OS X systems

**Virtual Private Networks**

Many organizations deploy virtual private networks (VPN) to provide data integrity, authentication, and data encryption to ensure confidentiality of the packets sent over an unprotected network or the Internet.

Many different **protocols** are **used for VPN** implementations, including the following:

- Point-to-Point Tunneling Protocol (PPTP) – very weak security
- Layer 2 Forwarding (L2F) Protocol
- Layer 2 Tunneling Protocol (L2TP)
- Generic routing encapsulation (GRE)
- Multiprotocol Label Switching (MPLS) VPN
- Internet Protocol Security (IPsec)
- Secure Sockets Layer (SSL)

VPN implementations can be categorized into two distinct groups:

- **Site-to-site VPNs:** Enable organizations to establish VPN tunnels between two or more network infrastructure devices in different sites so that they can communicate over a shared medium such as the Internet. Many organizations use IPsec, GRE, or MPLS VPN as site-to-site VPN protocols.

- **Remote-access VPNs:** Enable users to work from **remote locations** such as their homes, hotels, and other premises as if they were directly connected to their corporate network. Many organizations use IPsec and SSL VPN for remote access VPNs.

# References


https://www.certificationkits.com/cisco-certification/ccna-security-certification-topics/

1.2: https://itprotv-notes-bucket.s3.amazonaws.com/cisco-ccnasec210260-1-2-2-identify_common_security_threats_pt2-060116.pdf

2.1: https://itprotv-notes-bucket.s3.amazonaws.com/cisco-ccnasec210260-2-1-5-secure_management_pt5-060616.pdf

5.5: https://itprotv-notes-bucket.s3.amazonaws.com/cisco-ccnasec210260-5-5-2-configure_ASA_management_pt2-062416.pdf

7.1: https://itprotv-notes-bucket.s3.amazonaws.com/cisco-ccnasec210260-7-0-content_and_endpoint_security-062416.pdf

https://www.certificationkits.com/cisco-certification/ccna-security-certification-topics/ccna-security-implement-ips-with-sdm/ccna-security-network-based-vs-host-based-intrusion-detection-a-prevention/

http://www.cs.rpi.edu/~kotfid/secvoice10/powerpoints/

http://sclabs.blogspot.com.mt/2012/09/chapter-5-implementing-intrusion.html

http://sclabs.blogspot.com.mt/2013/01/chapter-10-implementing-cisco-adaptive.html

https://learningnetwork.cisco.com/docs/DOC-15381

http://www.mustbegeek.com/configure-site-to-site-ipsec-vpn-tunnel-in-cisco-ios-router/

http://sclabs.blogspot.com.mt/2013/04/ccna-security-ipsec-site-to-site.html

https://www.youtube.com/watch?v=xSQo8NVH4rk

https://www.youtube.com/watch?v=_JasBNJXMFQ