Evropský polytechnický institut s.r.o.

BACHELOR WORK

2005 Jaroslav Kafka

Evropský polytechnický institut, s. r. o.

v Kunovicích

Course: Electronic computers

D E S I G N I N G S E C U R E C O M P U T E R N E T W O R K S

(bachelor work)

Author: Jaroslav Kafka

Praha, 18th March 2005

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně pod vedením ing. Leoše Boháče a uvedl v seznamu literatury všechny použité literární a odborné zdroje.

Kunovice, březen 2005 .....

Děkuji panu ing. Leoši Boháčovi za velmi užitečnou metodickou pomoc, kterou mi poskytl při zpracování mé bakalářské práce.

Kunovice, březen 2005 Jaroslav Kafka

CONTENTS

Introducion

New business practices and opportunities are driving a multitude of changes in all areas of enterprise networks, and as such, enterprise security is becoming more and more prevalent as enterprises try to understand and manage the risks associated with the rapid development of business applications deployed over the enterprise network. This coupled with the exponential growth of the Internet has presented a daunting security problem to most enterprises: How does the enterprise implement and update security defenses and practices in an attempt to reduce its vulnerability to exposure from security breaches?

In my bachelor work, I will attempt to bridge the gap between the theory and practice of network security and place much of its emphasis on securing the enterprise infrastructure, but first let me emphasize that there is no such thing as absolute security. The statement that a network is secure is more often than not, misunderstood to mean that there is no possibility of a security breach. However – having a secure network means that the proper security mechanisms have been put in place in an attempt to reduce most of the risks enterprise assets are exposed to. Primary focus of this book is on the Cisco product offering, the principles apply to many other environments as well – securing every enterprise network is based on common basis.

As sources for my bachelor work I will use technical literature, especially books from CiscoPress. These books have high standard and contains current knowledge. Specific role as a source plays self-study books for Cisco Certified Network Professional exam. Second large source used especially in sixth chapter is Contactel's internal knowledge base database. This database contains description of known problem with software, hardware, configurations used in Contactel's network and combinations of these elements.

My bachelor work is divided into six chapters. First chapter describe common network attacks and potential security threads. End of this chapter is focused on active Cisco network equipments used in secured enterprise networks. Second chapter describes security on first three layers of the network – physical security (e.g. physical access to equipment or how secure console port etc.) and security on data-link (especially VLANs) and network layer (securing routing updates and route filtering). Third chapter describe security on transport and application layer. Fourth chapter is focused on firewall design. This chapter describes widely used technology – PIX Firewall. Fifth chapter is light introduction into virtual private networks.

The goal of my bachelor work is described in the last chapter. This chapter is about designing Contactel's WAN. I am working in Contactel since 1999 and my goal is to design private network between branches and headquarter in Prague. This enterprise VPN should be able to carry voice over IP and data. This chapter contains figures with topologies and configuration examples. In accordance to this project will be realized real Contactel's private network.

After designing VPN topology I will build "copy" of designed network in a lab. This lab verifies correct function of my project and helps solving possible problems before breaking-in. This testing phase is planned in the June/July 2005 therefore results of this testing are not included in my bachelor work.

1. Network security and hardware

1.1. Overview of network security

Security has one purpose, to protect assets. For most of history, this meant building strong walls to stop the bad guys, and establishing small, well-guarded doors to provide secure access for the good guys. This strategy worked well for the centralized, fortress-like world of mainframe computers and closed networks. The closed network typically consists of a network designed and implemented in a corporate environment, and provides connectivity only to known parties and sites without connecting to public networks. Networks were designed this way in the past and thought to be reasonably secure because of no outside connectivity.

With the advent of personal computers, LANs, and the wide-open world of the Internet, the networks of today are more open. As e-business and Internet applications continue to grow, finding the balance between being isolated and being open will be critical, along with the ability to distinguish the good guys from the bad guys. Furthermore, the rise of mobile commerce and wireless networks will be as the cannon was to the castle walls, exploding the old model and demanding that security solutions become seamlessly integrated, more transparent, and more flexible.

With the increased number of LANs and personal computers, the Internet began to create untold numbers of security risks. Firewall devices, which are software or hardware that enforce an access control policy between two or more networks, were introduced. This technology gave businesses a balance between security and simple outbound access to the Internet which was mostly used for e-mail and Web surfing.

This balance was short lived however as the use of extranets began to grow, which connected internal and external business processes. Businesses were soon realizing tremendous cost savings by connecting supply-chain management and enterprise resource planning systems to their business partners, and by connecting sales-force automation systems to mobile employees, and by providing electronic commerce connections to business customers and consumers. The firewall began to include intrusion detection, authentication, authorization, and vulnerability assessment systems. Today, successful companies have once again struck a balance by keeping the bad guys out with increasingly complex ways of letting the good guys in.

Most people expect security measures to ensure the following:

- Users can perform only authorized tasks.
- Users can obtain only authorized information.
- Users cannot cause damage to the data, applications, or operating environment of a system.

1.1.1. Trends that affect network security

Legal issues and privacy concerns

For many businesses today, one of the biggest reasons to create and follow a security policy is compliance with the law. Any business is potentially liable should a hacker or a virus take down the operation. Similarly, if a business is running a publicly held e-business and a catastrophic attack seriously impairs the business, a lawsuit is possible.

Legal liability in such cases is likely to depend on what prevention technologies and practices are available and on whether these technologies and practices are reasonably cost-effective to implement. As a result, showing due diligence will mean everything from implementing technologies such as firewalls, intrusion-detection tools, content filters, traffic analyzers and virtual private networks to having best practices for continuous risk assessment and

vulnerability testing. Of course, litigation isn't the only legal consideration that e-businesses are facing today. Lawmakers concern over the lack of Internet security, particularly where it hampers rights to privacy, is growing.

Wireless access

The increasing use of wireless local area network (WLAN) connections and the rapid rise of Internet access from cell phones in Europe and Asia are requiring entirely whole new approaches to security. RF connections do not respect firewalls the way wired connections do. Moreover, the slow processors, small screens, and nonexistent keyboards on cell phones and personal digital assistants (PDAs) break many of the standard approaches to access, authentication, and authorization.

The need for speed

The number of broadband connections to the Internet from homes is exceeding projections. Many businesses are finding that multiple T1 or E1 connections to the Internet are no longer sufficient. Current software-based security approaches have problems scaling to OC-1 and higher rates.

IT staffing shortages

The IT staffing shortage is especially evident in the security field. To solve this problem, many enterprises are increasingly outsourcing day-to-day security management tasks. The application service provider (ASP) business model will become increasingly common in the security world. Therefore, security solutions will need to be more manageable in this outsourced model.

1.1.2. Goals of network security

Confidentiality

Confidentiality protects sensitive information from unauthorized disclosure or intelligible interception. Cryptography and access control are used to protect confidentiality. The effort applied to protecting confidentiality depends on the sensitivity of the information and the likelihood of it being observed or intercepted. Network encryption can be applied at any level in the protocol stack. Applications can provide end-to-end encryption, but each application must be adapted to provide this service. Encryption at the transport layer is used frequently today. Virtual private networks (VPNs) can be used to establish secure channels of communication between two sites or between an end user and a site. Encryption can be used at the OSI data-link layer, but doesn't scale easily; every networking device in the communication pathway would have to participate in the encryption scheme. Datalink layer encryption is making a comeback in the area of wireless security, such as in IEEE 802.11. Physical security, meanwhile, is used to prevent unauthorized access to network ports or equipment rooms. One of the risks at the physical level is violation of access control through the attachment of promiscuous packet capture devices to the network, particularly with the widespread use of open source tools such as Ethereal (www.ethereal.com) and tcpdump (www.tcpdump.org) that permits nearly any host to become a packet decoder.

Integrity

Integrity ensures that information or software is complete, accurate, and authentic. We want to keep unauthorized people or processes from making any changes to the system, and keep authorized users from making changes that exceed their authority. These changes may be intentional or unintentional, and similar mechanisms can protect a system from both. For network integrity, we need to ensure that the message received is the same message that was sent. The content of the message must be complete and unmodified, and that the link is between a valid source and destination nodes. Connection integrity can be provided by cryptography and routing control. Simple integrity assurance methods to detect incidental changes, like adding up all the bytes in a message and recording that as an element in the packet, are used in everyday IP flows. More robust approaches, such as taking the output from a

hash function like message digest (version) 5 (MD5) or secure hash algorithm (SHA) and adding that to the message, as is used in IPSec, can detect attempted malicious changes to a communication.

For host integrity, cryptography can also come to the rescue. Using a secure hash can identify whether an unauthorized change has occurred. However, of fundamental importance are careful use of audit trails to determine what changed, when the change occurred, and who made the change. Sound security design includes a centralized log server, and policy and procedure around safe handling of audit data. Integrity also extends to the software images for network devices that are transporting data. The images must be verified as authentic, and that they have not been modified or corrupted. Just as a transported IP packet has a checksum to verify it wasn't accidentally damaged in transit, Cisco provides a checksum for IOS images.

Availability

Availability ensures that information and services are accessible and functional when needed. Redundancy, fault tolerance, reliability, failover, backups, recovery, resilience and load balancing are the network design concepts used to assure availability. If systems aren't available, then integrity and confidentiality won't matter. Build networks that provide high availability. Customers or end users will perceive availability as being the entire system – application, servers, network and workstation. If they can't run their applications, then it is not available. To provide high availability, ensure that security processes are reliable and responsive. Modular systems and software, including security systems, need to be interoperable.

Cisco makes many products designed for high hardware availability. These devices are characterized by a long mean time between failure (MTBF) with redundant power supplies, and hot-swappable cards or modules. For example, devices that provide 99.999 percent availability would have about five minutes downtime[1] per year.

Availability of individual devices can be enhanced by their configuration. Using features such as redundant uplinks with Hot Standby Router Protocol (HSRP), fast convergent Spanning Tree, or Fast EtherChannel provide a failover if one link should fail. Uninterruptible power supplies (UPSs) and backup generators are used to protect mission-critical equipment in the event of a power outage. These are not security features per se- and in some instances may work against security, such as using HSRP to force a router offline to allow the bypassing of access controls – but are a valid part of a security design. Cisco IOS includes reliability features such as:

- Hot Standby Router Protocol (HSRP)
- Simple Server Redundancy Protocol (SSRP)
- Deterministic Load Distribution (DLD)

1.1.3. Key elements of network security

The successful use of Internet technologies requires an increased need to protect valuable data and network resources from corruption and intrusion. A security solution contains five key elements:

Identity

Identity refers to the accurate and positive identification of network users, hosts, applications, services, and resources. Standard technologies that enable identification include authentication protocols such as Remote Access Dial-In User Service (RADIUS) and Terminal Access Controller Access Control System Plus (TACACS+), Kerberos, and one-time password (OTP) tools. New technologies such as digital certificates, smart cards, and directory services are beginning to play increasingly important roles in identity solutions.

---

[1] There is big difference between HW and SW downtimes. About 95% of all problems is caused by software.

Perimeter security

This element provides the means to control access to critical network applications, data, and services so that only legitimate users and information can pass through the network. Routers and switches with packet filtering or stateful firewalling, as well as dedicated firewall appliances, provide this control. Complementary tools, including virus scanners and content filters, also help control network perimeters.

Data privacy

When information must be protected from eavesdropping, the ability to provide authenticated, confidential communication on demand is crucial. Sometimes, data separation using tunneling technologies, such as generic routing encapsulation (GRE) or Layer 2 Tunneling Protocol (L2TP), provides effective data privacy. However, additional privacy requirements often call for the use of digital encryption technology and protocols such as IP Security (IPSec). This added protection is especially important when implementing Virtual Private Networks (VPNs).

Security monitoring

To ensure that a network remains secure, it is important to regularly test and monitor the state of security preparation. Network vulnerability scanners can proactively identify areas of weakness, and intrusion detection systems can monitor and respond to security events as they occur. By using security-monitoring solutions, organizations can obtain significant visibility into both the network data stream and the security posture of the network.

Policy management

As networks grow in size and complexity, the requirement for centralized policy management tools grows as well. Sophisticated tools that can analyze, interpret, configure, and monitor the state of security. Browser-based user interfaces and tools enhance the usability and effectiveness of network security solutions.

1.2. Vulnerabilities and Threats

1.2.1. Network security weaknesses

There are three primary reasons for network security threats:

- Technology weaknesses.
- Configuration weaknesses.
- Security policy weaknesses.

There are people eager, willing, and qualified to take advantage of each security weakness, and they continually search for new exploits[2] and weaknesses.

Technology weaknesses

Computer and network technologies have intrinsic security weaknesses. These include TCP/IP protocol weaknesses, operating system weaknesses, and weaknesses in configuration and security policy. Technology weaknesses are listed in following table:

TCP/IP protocol weaknesses	Protocols HTTP, FTP and ICMP are inherently insecure. SNMP, SMTP and Syn Floods are relaeat to the inherntly insecure structure upon which TCP was designed.
Operating system weaknesses	Each operating system has security problems that must be addressed.
Network equipment weaknesses	Various types of network equipment (such routers, firewalls and switches) that must be recognized and protected against. These weaknesses include password protection, lack of authentication, routing protocols and firewall holes.

Table 1: Technology weaknesses

[2] Very useful website is www.cert.org

Configuration weaknesses

Network administrators or network engineers need to learn what the configuration weaknesses are and correctly configure their computing and network devices to compensate. Some common configuration weaknesses are listed in Appendix 1.

Security policy weaknesses

Security policy weaknesses can create unforeseen security threats. The network may pose security risks to the network if users do not follow the security policy. Some common security policy weaknesses and how those weaknesses are exploited are listed in Appendix 1.

1.2.2. Primary network threats

There are four primary classes of threats to network security:

Unstructured threats

Unstructured threats consist of mostly inexperienced individuals using easily available hacking tools such as shell scripts and password crackers. Even unstructured threats that are only executed with the intent of testing and challenging a hacker's skills can still do serious damage to a company. For example, if an external company Web site is hacked, the integrity of the company is damaged. Even if the external Web site is separate from the internal information that sits behind a protective firewall, the public does not know that. All the public knows is that the site is not a safe environment to conduct business.

Structured threats

Structured threats come from hackers that are more highly motivated and technically competent. These people know system vulnerabilities, and can understand and develop exploit-code and scripts. They understand, develop, and use sophisticated hacking techniques to penetrate unsuspecting businesses. These groups are often involved with the major fraud and theft cases reported to law enforcement agencies.

External threats

External threats can arise from individuals or organizations working outside of a company. They do not have authorized access to the computer systems or network. They work their way into a network mainly from the Internet or dialup access servers.

Internal threats

Internal threats occur when someone has authorized access to the network with either an account on a server or physical access to the network. According to the networks administrators, internal access and misuse account for 60 to 80 percent of reported incidents.

Following picture shows how threats continue to become more sophisticated as the technical knowledge required to implement attacks diminishes.

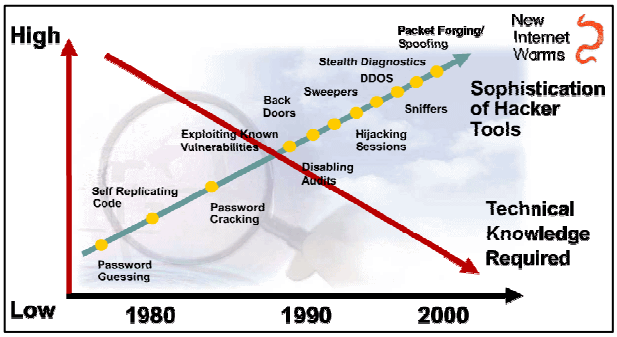

Figure 1: Relationship between technical knowledge and sophistication of hacker tools

1.2.3. Reconnaissance

Reconnaissance is the unauthorized discovery and mapping of systems, services, or vulnerabilities. It is also known as information gathering and, in most cases, it precedes an actual access or Denial of Service (DoS) attack. The malicious intruder typically ping sweeps the target network to determine which IP addresses are alive. After this, the intruder uses a port scanner to determine what network services or ports are active on the live IP addresses. From this information, the intruder queries the ports to determine the application type and version, as well as the type and version of operating system running on the target host. Based on this information, the intruder can determine if a possible vulnerability exists that can be exploited. Reconnaissance is somewhat analogous to a thief casing a neighborhood for vulnerable homes to break into, such as an unoccupied residence, easy-to-open doors, or open windows.

Using, for example, the nslookup and whois utilities, an attacker can easily determine the IP address space assigned to a given corporation or entity . The ping command tells the attacker what IP addresses are alive.

1.2.4. Eavesdropping

Network snooping and packet sniffing are common terms for eavesdropping. Eavesdropping is listening in to a conversation, spying, prying, or snooping. The information gathered by eavesdropping can be used to pose other attacks to the network.

An example of data susceptible to eavesdropping is SNMP version 1 community strings, which are sent in clear text. An intruder could eavesdrop on SNMP queries and gather valuable data on network equipment configuration. Another example is the capture of usernames and passwords as they cross a network.

Types of eavesdropping

A common method for eavesdropping on communications is to capture TCP/IP or other protocol packets and decode the contents using a protocol analyzer or similar utility. Two common uses of eavesdropping are as follows:

- **Information gathering** – network intruders can identify usernames, passwords, or information carried in the packet such as credit card numbers or sensitive personal information.
- **Information theft** – network eavesdropping can lead to information theft. The theft can occur as data is transmitted over the internal or external network. The network intruder can also steal data from networked computers by gaining unauthorized access. Examples include breaking into or eavesdropping on financial institutions and obtaining credit card numbers. Another example is using a computer to crack a password file.

Tools used to perform eavesdropping

The following tools are used for eavesdropping:

- Network or protocol analyzers.
- Packet capturing utilities on networked computers.

Methods to counteract attacks

Two of the most effective methods for counteracting eavesdropping are as follows:

- Implementing and enforcing a policy directive that forbids the use of protocols with known susceptibilities to eavesdropping.
- Using encryption that meets the data security needs of the organization without imposing an excessive burden on the system resources or the users.
- Using switched networks.

Encrypted data

Encryption provides protection for data susceptible to eavesdropping attacks, password crackers, or manipulation. Some benefits of data encryption are as follows:

- Almost every company has transactions, which, if viewed by an eavesdropper, could have negative consequences. Encryption ensures that when sensitive data passes over a medium susceptible to eavesdropping, it cannot be altered or observed.
- Decryption is necessary when the data reaches the router or other termination device on the far receiving LAN where the destination host resides.

By encrypting after the User Datagram Protocol (UDP) or Transmission Control Protocol (TCP) headers, so that only the IP payload data is encrypted, Cisco IOS network-layer encryption allows all intermediate routers and switches to forward the traffic as they would any other IP packets. Payload-only encryption allows flow switching and all access-list features to work with the encrypted traffic just as they would with plain text traffic, thereby preserving desired quality of service (QoS) for all data.

1.2.5. Access

System access is the ability for an unauthorized intruder to gain access to a device for which the intruder does not have an account or a password. Entering or accessing systems to which one does not have access usually involves running a hack, script, or tool that exploits a known vulnerability of the system or application being attacked. Some examples of methods used by hackers include the following:

Exploit easily guessed passwords

- brute force
- cracking tools
- dictionary attacks

Exploit misconfigured services

- IP services such as anonymous FTP, TFTP, and remote registry access
- trust relationships through spoofing and r-services
- file sharing services such as NFS and Windows File Sharing

Exploit application holes

- mishandled input data

Access outside application domain, buffer overflows, race conditions

- protocol weaknesses

Fragmentation, TCP session hijack

Trojan horses

- programs that introduce an inconspicuous backdoor into a host

Social engineering

- posing as a network administrator to gain information from users
- look for written usernames and passwords near computer or server terminals
- dumpster diving searching through trash cans to find access information

Unauthorized data retrieval is simply reading, writing, copying, or moving files that are not intended to be accessible to the intruder. Sometimes this is as easy as finding shared folders in Windows, Unix, or Macintosh file systems that have read or read and write access set for everyone.

1.2.6. Man-in-the-middle attack

A man-in-the-middle attack requires that the hacker have access to network packets that come across a network. An example could be someone who is working for an Internet service provider (ISP) and has access to all network packets transferred between the ISP network and any other network.

Such attacks are often implemented using network packet sniffers and routing and transport protocols. The possible uses of such attacks are theft of information, hijacking of an ongoing session to gain access to private network resources, traffic analysis to derive information about a network and its users, Denial of Service (DoS), corruption of transmitted data, and introduction of new information into network sessions.

1.2.7. Trust exploitation

Although it is more of a technique than a hack itself, trust exploitation refers to an attack in which an individual takes advantage of a trust relationship within a network. The classic example is a perimeter network connection from a corporation. These network segments often house Domain Name System (DNS), Simple Mail Transfer Protocol (SMTP), and Hypertext Transfer Protocol (HTTP) servers. Because all these servers reside on the same segment, the compromise of one system can lead to the compromise of other systems because these systems usually trust other systems attached to the same network.

Another example is a system on the outside of a firewall that has a trust relationship with a system on the inside of a firewall. When the outside system is compromised, it can take advantage of that trust relationship to attack the inside network. Another form of an access attack involves privilege escalation. Privilege escalation occurs when a user obtains privileges or rights to objects that were not assigned to the user by an administrator. Objects can be files, commands, or other components on a network device. The intent is to gain access to information or execute unauthorized procedures. This information will be used to gain administrative privileges to a system or device. They use these privileges to install sniffers, create backdoor accounts, or delete log files.

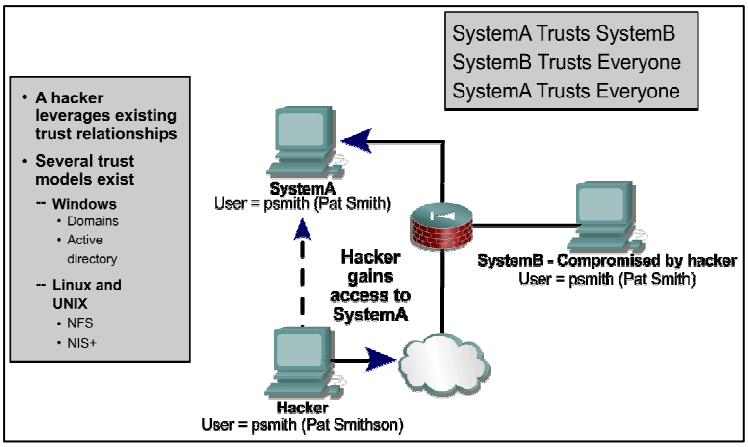

Figure 2: Trustworthiness between systems

1.2.8. Data manipulation

With data manipulation, the network intruder can capture, manipulate, and replay data sent over a communication channel.

Examples of specific attacks include the following:

- **Graffiti** – the intruder vandalizing a Web site by accessing the Web server and altering Web pages.
- **Manipulation of data on a networked computer** – the intruder alters files on the computer, such as password files, to enable further access to the network.

Some tools used to perform these attacks include the following:

- Protocol analyzers that record passwords as they pass over the wire.
- Password crackers, as shown in the figure, that contain algorithms to allow unauthorized persons to crack passwords, even ones that contain numeric and special characters.

1.2.9. Masquerade/IP spoofing

With a masquerade attack, the network intruder can manipulate TCP/IP packets by IP spoofing, falsifying the source IP address, there by appearing to be another user. The intruder assumes the identity of a valid user and gains that user's access privileges by IP spoofing. IP spoofing occurs when intruders create IP data packets with falsified source addresses.

During an IP spoofing attack, an attacker outside the network pretends to be a trusted computer. The attacker may either use an IP address that is within the range of IP addresses for the network or use an authorized external IP address that is trusted and provides access to specified resources on the network.

Normally, an IP spoofing attack is limited to the injection of data or commands into an existing stream of data passed between a client and server application or a peer-to-peer network connection. The attacker simply does not worry about receiving any response from the applications.

To enable bi-directional communication, the attacker must change all routing tables to point to the spoofed IP address. Another approach the attacker could take is to simply not worry about receiving any response from the applications.

If an attacker manages to change the routing tables they can receive all of the network packets that are addressed to the spoofed address, and reply just as any trusted user can. Like packet sniffers, IP spoofing is not restricted to people who are external to the network.

Some tools used to perform IP spoofing attacks are as follows:

- Protocol analyzers, also called password sniffers.
- Sequence number modification.
- Scanning tools that probe TCP ports for specific services, network/system architecture, and the OS.

1.2.10. Session replay

A sequence of packets or application commands can be captured, manipulated, and replayed to cause an unauthorized action.

Mercenary Messages are designed to use mobile code to penetrate e-mail systems in order to gain private and confidential information. Mobile technologies are easy to use and most traditional security solutions, such as firewalls or anti-virus software, do not detect these security violations. Some mechanisms used to perform these attacks are as follows:

- Cookies
- JavaScript or Active X scripts

1.2.11. Auto rooters

Auto rooters are programs that automate the entire hacking process. Computers are sequentially scanned, probed, and captured. The capture process includes installing a rootkit on the computer and using the newly captured system to automate the intrusion process. Automation allows an intruder to scan hundreds of thousands of systems in a short period of time.

1.2.12. Back doors

Back doors are paths into systems that can be created during an intrusion. The back door, unless detected, can be used again and again by an intruder to enter a computer or network. An intruder will often use the computer to gain access to other systems or to launch DoS attacks when they have no further use for the computer.

1.2.13. Social engineering

The easiest hack involves no computer skill at all. If an intruder can trick a member of an organization into giving over valuable information, such as locations of files, servers and passwords, then the process of hacking is made immeasurably easier.

1.3. Denial of service

Denial of service (DoS) implies that an attacker disables or corrupts networks, systems, or services with the intent to deny services to intended users. DoS attacks involve either crashing the system or slowing it down to the point that it is unusable. But DoS can also be as simple as deleting or corrupting information. In most cases, performing the attack simply involves running a hack or script. The attacker does not need prior access to the target because a way to access it is all that is usually required. For these reasons, DoS attacks are the most feared.

DoS attacks take many forms. Ultimately, they prevent authorized people from using a service by using up system resources. The following are some examples of common DoS threats:

- **ping of death** – this attack modifies the IP portion of the header, indicating that there is more data in the packet than there actually is, causing the receiving system to crash .
- **SYN flood attack** – this attack randomly opens up many TCP ports, tying up the network equipment or computer with so many bogus requests that sessions are thereby denied to others. This attack is accomplished with protocol analyzers or other programs .
- **packet fragmentation and reassembly** – this attack exploits a buffer–overrun bug in hosts or internetworking equipment.
- **e-mail bombs** – programs can send bulk e-mails to individuals, lists, or domains, monopolizing e-mail services.
- **CPU hogging** – these attacks constitute programs such as Trojan horses or viruses that tie up CPU cycles, memory, or other resources.
- **malicious applets** – these attacks are Java, JavaScript, or ActiveX programs that act as Trojan horses or viruses to cause destruction or tie up computer resources.
- **misconfiguring routers** – misconfiguring routers to reroute traffic disables web traffic.
- **the chargen attack** – this attack establishes a connection between UDP services, producing a high character output. The host chargen service is connected to the echo service on the same or different systems, causing congestion on the network with echoed chargen traffic.
- **out-of-band attacks such as WinNuke** – these attacks send out-of-band data to port 139 on Windows 95 or Windows NT machines. The attacker needs the victim's IP address to launch this attack.
- **denial of Service** - DoS can occur accidentally because of misconfigurations or misuse by legitimate users or system administrators.
- **land.c** – this program sends a TCP SYN packet that specifies the target host address as both source and destination. The program also uses the same port (such as 113 or 139) on the target host as both source and destination, causing the target system to stop functioning.

- **teardrop.c** – in this attack, the fragmentation process of the IP is implemented in such a way that reassembly problems can cause machines to crash.
- **targa.c** – this attack is a multi-platform DoS attack that integrates bonk, jolt, land, nestea, netear, syndrop, teardrop, and winnuke all into one exploit.

1.3.1. Examples of Distributed denial of service attacks

Distributed DoS (DDoS) attacks are designed to saturate network links with spurious data. This data can overwhelm an Internet link, causing legitimate traffic to be dropped. DDoS uses attack methods similar to standard DoS attacks but operates on a much larger scale. Typically hundreds or thousands of attack points attempt to overwhelm a target.

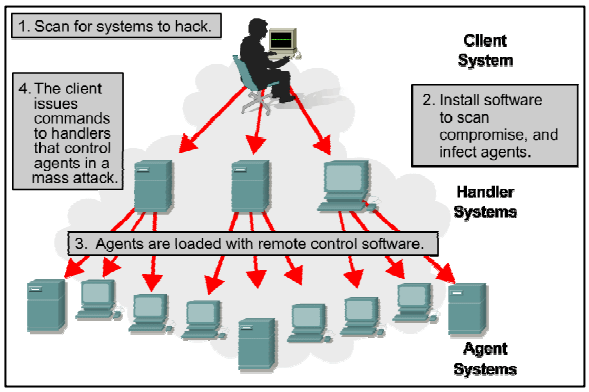

Figure 3: DDoS attack example

Examples of DDoS attacks include the following:

SMURF attack

The Smurf attack starts with a perpetrator sending a large number of spoofed ICMP echo, or ping, requests to broadcast addresses, hoping that these packets will be magnified and sent to the spoofed addresses . If the routing device delivering traffic to those broadcast addresses performs the Layer 3 broadcast-to-Layer 2 broadcast function, most hosts on that IP network will each reply to the ICMP echo request with an ICMP echo reply, multiplying the traffic by the number of hosts responding. On a multi-access broadcast network, there could potentially be hundreds of machines replying to each echo packet.

Example of Smurf attack: Assume the network has 100 hosts and that the attacker has a T1 link. The attacker sends a 768 s-kilobits-per-second (kbps) stream of ICMP echo, or ping packets, with a spoofed source address of the victim, to the broadcast address of the "bounce site". These ping packets hit the bounce site broadcast network of 100 hosts, and each of them takes the packet and responds to it, creating 100 outbound ping replies. A total of 76.8 megabits per second (Mbps) of bandwidth is used outbound from the bounce site after the traffic is multiplied. This is then sent to the victim, or the spoofed source of the originating packets. Turning off directed broadcast (interface command **`no ip directed-broadcast`**) capability in the network infrastructure prevents the network from being used as a bounce site.

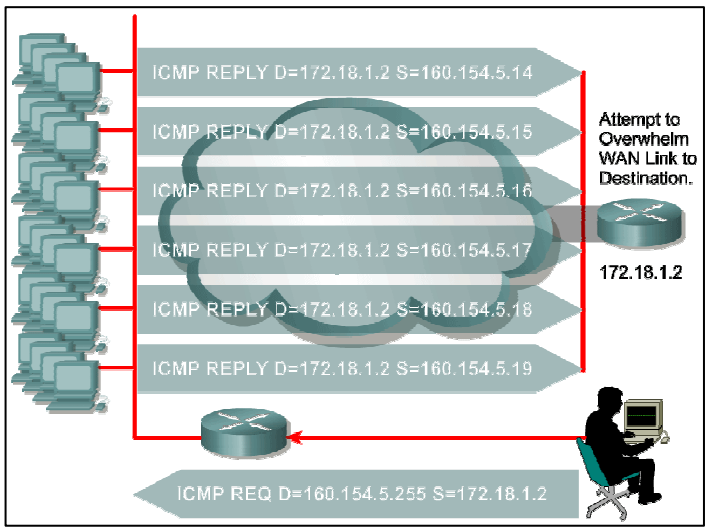

Figure 4: SMURF attack example

Tribe flood network (TFN)

Tribe Flood Network (TFN) and Tribe Flood Network 2000 (TFN2K) are distributed tools used to launch coordinated DoS attacks from many sources against one or more targets. A TFN attack has the capability to generate packets with spoofed source IP addresses. An intruder instructing a master to send attack instructions to a list of TFN servers or daemons carries out a DoS attack using a TFN network. The daemons then generate the specified type of DoS attack against one or more target IP addresses. Source IP addresses and source ports can be randomized, and packet sizes can be altered. Use of the TFN master requires an intruder-supplied list of IP addresses for the daemons.

Stacheldraht attack

Stacheldraht, German for "barbed wire", combines features of several DoS attacks, including Tribe Flood Network (TFN). It also adds features such as encryption of communication between the attacker and stacheldraht masters, and automated update of the agents. There is an initial mass-intrusion phase, in which automated tools are used to remotely root-compromise large numbers of systems to be used in the attack. This is followed by a DoS attack phase, in which these compromised systems are used to attack one or more site

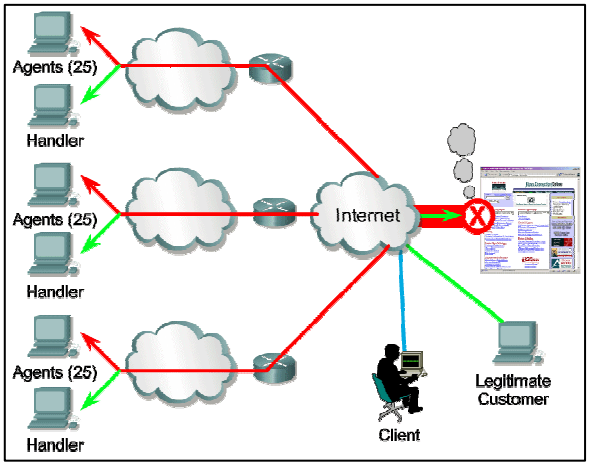

Figure 5: Example of stacheldrath attack

1.4. How to design network security

1.4.1. The security wheel

Most security incidents occur because system administrators do not implement available countermeasures, and hackers or disgruntled employees exploit the oversight. Therefore, the issue is not just one of confirming that a technical vulnerability exists and finding a countermeasure that works, it is also critical to verify that the countermeasure is in place and working properly.

This is where the Security Wheel, a continuous process, is an effective approach. The Security wheel promotes retesting and reapplying updated security measures on a continuous basis.

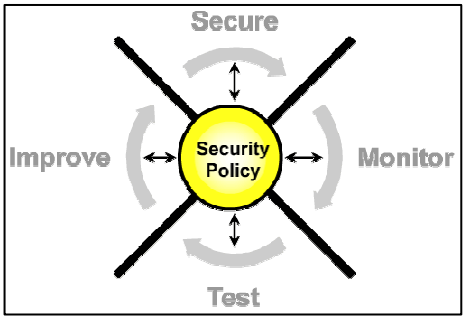

Figure 6: Security wheel

To begin the Security wheel process, first develop a security policy that enables the application of security measures. A security policy needs to accomplish the following tasks:

- Identify the security objectives of the organization.
- Document the resources to be protected.
- Identify the network infrastructure with current maps and inventories.
- Identify the critical resources that need to be protected, such as research and development, finance, and human resources.

After the security policy is developed, make it the hub upon which the next four steps of the Security Wheel; secure, monitor, test, and improve, are based.

Secure

Secure the network by applying the security policy and implementing the following security solutions:

- **Authentication** – give access to authorized users only. One example of this is using one-time passwords.
- **Firewalls** – filter network traffic to allow only valid traffic and services.
- **Virtual private networks (VPNs)** – hide traffic content to prevent unwanted disclosure to unauthorized or malicious individuals.
- **Vulnerability patching** – apply fixes or measures to stop the exploitation of known vulnerabilities. This includes turning off services that are not needed on every system. The fewer services that are enabled, the harder it is for hackers to gain access.

Monitor

Monitoring security involves both active and passive methods of detecting security violations. The most commonly used active method is to audit host-level log files. Most operating systems include auditing functionality. System administrators for every host on the network must turn these on and take the time to check and interpret the log file entries.

Passive methods include using intrusion detection or IDS devices to automatically detect intrusion. This method requires only a small number of network security administrators for monitoring. These systems can detect security violations in real time and can be configured to automatically respond before an intruder does any damage.

An added benefit of network monitoring is the verification that the security devices implemented in Step 1 of the Security Wheel have been configured and are working properly.

Test

In the testing phase of the Security Wheel, the security of the network is proactively tested. Specifically, the functionality of the security solutions implemented in Step 1 and the system auditing and intrusion detection methods implemented in Step 2 must be assured. Vulnerability scanning tools such as SATAN, Nessus, or NMAP are useful for periodically testing the network security measures.

Improve

The improvement phase of the Security Wheel involves analyzing the data collected during the monitoring and testing phases, and developing and implementing improvement mechanisms that feed into the security policy and the securing phase in Step 1. To keep a network as secure as possible, the cycle of the Security Wheel must be continually repeated, because new network vulnerabilities and risks are created every day.

With the information collected from the monitoring and testing phases, intrusion detection systems can be used to implement improvements to the security. The security policy should be adjusted as new security vulnerabilities and risks are discovered.

1.4.2. Network security case studies

Security policies can vary greatly in design. Three general types of security polices are open, restrictive and closed. Some important points are as follows:

- Security policy can be open or closed as a starting point.
- Choose the best end-to-end mix of security products and technology to implement the policy.
- Application-level security can include Secure Socket Layer (SSL) technology.

Like security policies, many devices can be classified as open, restrictive, or closed. For example, routers and switches are typically open devices, allowing high functionality and services by default. On the other hand, a firewall is typically a closed system that does not allow any services until they are switched on. Server operating systems can fall into any of the three categories, depending on the vendor. It is important to understand these principles when deploying these devices.

Open access

An open security policy is the easiest to implement. Very few security measures are implemented in this design. Administrators configure existing hardware and software basic security capabilities. Firewall, Virtual Private Networks (VPN), Intrusion Detection Systems (IDS) and other measures that incur additional costs are typically not implemented. Simple passwords and server security become the foundation of this model. If encryption is used, it is implemented by individual users or on servers.

This model assumes that the protected assets are minimal, users are trusted and threats are minimal. However, this does not exclude the need for data backup systems in most open security policy scenarios. LANs, which are not connected to the Internet or public WANs, are more likely to implement this type of policy.

This type of network design gives users free access to all areas. When security breaches occur, they are likely to result in great damage and loss. Network administrators are usually not held responsible for network breaches or abuse.

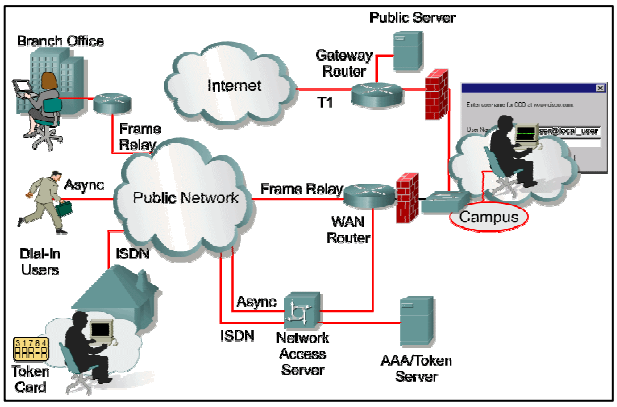

Figure 7: Network with open access

Restrictive access

A restrictive security policy is more difficult to implement. Many security measures are implemented in this design. Administrators configure existing hardware and software for security capabilities in addition to deploying more costly hardware and software solutions such as firewalls, VPN, IDS, and identity servers. Firewalls and identity servers become the foundation of this model.

This model assumes that the protected assets are substantial, some users are not trustworthy, and that threats are likely. LANs, which are connected to the Internet or public WANs, are more likely to implement this type of policy. Ease of use for users is diminished as security is tightened.

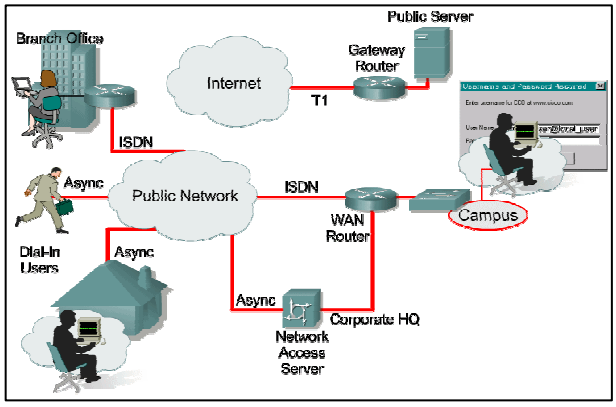

Figure 8: Network with restrictive access

Closed access

A closed security policy is most difficult to implement. All available security measures are implemented in this design. Administrators configure existing hardware and software for maximum-security capabilities in addition to deploying more costly hardware and software solutions such as firewalls, VPN, IDS and identity servers.

This model assumes that the protected assets are premium, all users are not trustworthy, and that threats are frequent. User access is very difficult and cumbersome. Network administrators require greater skills and more time to administer the network. Furthermore, companies require a higher number of network administrators to maintain this tight security.

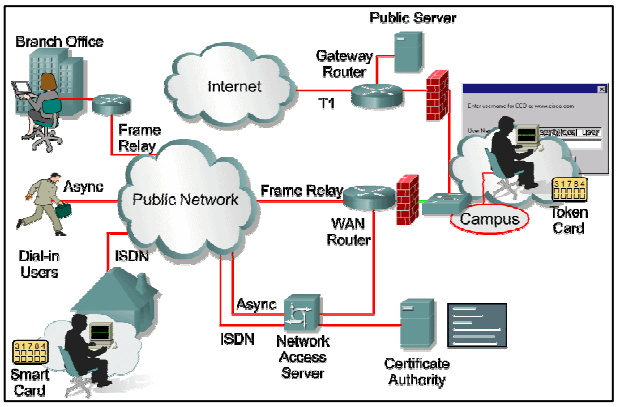

Figure 9: Top secured network

In many corporations and organizations, these administrators are likely to be very unpopular while implementing and maintaining security. Network security departments must clarify that they only implement the policy, which is designed, written, and approved by the corporation. Politics behind the closed security policy can be monumental. In the event of a security breach or network outage, network administrators may be held more accountable for problems.

1.5. Hardware and Software firewalls

1.5.1. Software-based firewalls

Software-based firewalls, also known as server-based, are software applications that are installed on an existing OS such as UNIX or Windows server platform. Advantages to a software firewall solution may include lower initial cost, at least for small networks, and the ability to combine the firewall with some other application such as a Web or FTP server. Software-based firewalls come in both small office/home office (SOHO) models and enterprise models. Examples of software-based firewalls include the following:

- Check Point Firewall-1
- Microsoft Internet Security and Acceleration (ISA) Server
- Novell BorderManager
- Linux iptables/ipchain

In addition to server-based firewalls, personal firewalls are available for the desktop PC. Typically these are used in a SOHO environment where there is no dedicated firewall. Some of the vendors include Zone Labs, McAfee, Norton, Tiny, and Internet Security Systems. Some of these are bundled with anti-virus software.

1.5.2. Hardware-based firewalls

Hardware-based firewalls, or dedicated firewalls, are devices that have the software pre-installed on a specialized hardware platform. The OS is often proprietary to the device, as is the case with the Cisco PIX Finesse OS. Examples of hardware-based firewall vendors include the following:

- Cisco
- NetScreen
- SonicWALL
- WatchGuard

In addition to the PIX Firewall, Cisco offers integrated firewall technologies in the IOS Firewall image for routers. Also, the Cisco Firewall Services Module (FWSM) provides an integrated chassis based firewall switch. Cisco is a company that provides a comprehensive set of security products and solutions.

There are many dedicated hardware appliance based firewalls available to secure a network. Cisco provides an integrated IOS Firewall and a dedicated Private Internet Exchange (PIX) Firewall. The IOS Firewall feature set can be installed and configured in perimeter routers. It adds features such as stateful, application-based filtering, dynamic per-user authentication and authorization, defense against network attacks, Java blocking, and real-time alerts. The PIX Firewall is a dedicated hardware/software security solution/appliance that provides packet filtering and proxy server technologies. Other dedicated firewall vendors include Netscreen, Nokia and Nortel Networks. Cisco provides a full lineup of firewall devices which are listen in following figure:

Security Connectivity	Extended Perimeter Security	Intrusion Protection	Identity Services	Security Management
Appliances Series VPN 3000 Concentrator PIX Firewall	**Appliances** PIX Firewalls	**Appliances** Cisco 4200 Series PIX Firewall Host Based	**Cisco Access Control Server 3.0 Software** **Identity Based Network Services (IBNS) 802.1X ext.**	**VPN Solutions Center** **CiscoWorks VPN/Security Management Solution** **CiscoWorks Hosting Solution Engine**
Integrated Switch VPN Module	**Integrated** Firewall Switch Module (FWSM)	**Integrated** Switch IDS Module (IDSM)		
Cisco IOS VPN	**Cisco IOS Firewall**	**Cisco IOS IDS**		
				
SOHO 90, 830, 1700, 2600, 3600, 3700, 7000 series				

Table 2: Cisco hardware-based security products.

VPN devices

VPNs can be created using many products including firewalls, routers, VPN concentrators, VPN software and hardware clients, intrusion detection devices, and management software. Figure illustrates some of the major devices available through Cisco Systems. Other VPN vendors include Netscreen, Check Point and Nortel.

Solution Breadth						
PIX Firewall		PIX 501	PIX 506E	PIX 515E	PIX 525	PIX 535
Switch module		Firewall Services Module (FWSM)				
IOS FW router		800	1700	2600	3xxx	7xxx
VPN Client		VPN client software built-in personal firewall				
Mgmt		Secure CLI	Web UI embedded mgr	Enterprise mgmt VMS		

Table 3: Cisco VPN solution

2. SECURING PHYSICAL, DATALINK AND NETWORK LAYER

2.1. Securing physical layer

Securing physical layer on a computer network is very important because if intruders have physical access to network equipment, they can run for example password recovery procedure (and get full access to the network equipment after one restart) or install packet sniffer. Very important is security in wireless LAN especially if LANs are using equipment in unlicensed band (for example 2.4 GHz or 10 GHz).

2.1.1. Securing the enterprise

The enterprise infrastructure is vulnerable to many different security threats (discussed earlier) from any number of intruders. The solution to the infrastructure security problem is to securely configure components of the network against vulnerabilities based on the network security policy. Physical and logical security includes the following:

Securing physical access to network equipment

It is very important that outsiders don't have physical access to network equipment. All network equipment must be places in special rooms – wiring closets, MDF (Main Distribution Facility) and IDF (Intermediate Distribution Facility). In these closets should have access network technicians only. All equipment should be installed in lockable rack. Access in MDF/IDF should be logged – for example access is granted only when using appropriate magnetic card. Physical access to a router or switch gives a sufficiently sophisticated user total control over that device. Nearly all switches and routers have "password-recovery techniques" or other back doors to access the device without a password. These techniques are publicly documented on the Internet. It makes no sense to install software security measures when access to the hardware is not controlled.

Physically secure network devices by doing the following:

- **Provide proper physical environment** – this includes lockable doors and backup power supplies.
- **Control direct access to the device** – this includes lockable racks and password protection to console and auxiliary ports. Ports that are not being used may also be disabled.

Securing console access

It's important to put the proper physical security mechanisms into place. If the proper physical security mechanisms are not in place, an intruder could potentially bypass all other logical security mechanisms and gain access to the device. If an intruder can gain access to the administrative interface of the router, he could view and change the device's configuration and gain access to other networking equipment. The first thing we should do to thwart intruders is to set a console password. If the intruder has already gained physical access to the device, he'll attempt to gain network access through the console port first. The console port supports many different methods for authenticating a user and allowing access, some of which are listed here:

- console password
- local user database
- TACACS+
- RADIUS

Example – setting console password:

```
SecureRouter#config t
Enter configuration commands, one per line. End with CNTL/Z.
SecureRouter(config)#line con 0
SecureRouter(config-line)#password Cisco
SecureRouter(config-line)#login
SecureRouter(config-line)#end
```

Example – setting console login with local user database:

```
username Tom privilege 15 password 0 Jerry
username Bilbo privilege 12 password 0 Hobbit
username Hlinik privilege 8 password 0 Humpolec
!
line con 0
login local
transport input none
```

When a user plugs into the console port of a router configured with local authentication, they are first prompted for their username; after successfully passing the correct username to the router, they are then prompted for the password that is associated with that username. The following example details these steps:

```
User Access Verification
Username: Fred
Password: Flintstone
SecureRouter#
```

When using local authentication and assigning privilege levels, you must be careful to associate the correct username with the correct privilege level. Anyone who logs in with a privilege level that is equal to 2 or above is logged directly into privileged mode!

Securing telnet access

Telnet is a protocol that allows a user to establish a remote connection to a device. After connected to the remote device, you are presented with a screen that is identical to the screen that would be displayed if you were directly connected to the console port. Telnet ports on a router are referred as virtual terminal ports. Telnet is really no different from a console connection, and as such, the proper logical security mechanisms should be put into place to ensure that only responsible personnel are allowed Telnet access. Virtual terminal ports support many different methods for authenticating a user and allowing access. Some of the methods are in the following list:

- vty password
- local user database
- TACACS+
- RADIUS

The steps involved in defining Telnet security are similar to the steps used to configure console security. An example of configuring the fourth requirement (after the first three have been met) can be seen here:

```
SecureRouter#config t
Enter configuration commands, one per line. End with CNTL/Z.
SecureRouter(config)#line vty 0 4
SecureRouter(config-line)#login
SecureRouter(config-line)#password Cisco
SecureRouter(config-line)#end
```

Routers and switches can also restrict telnet access to authorized users with the use of an access list. The access list is then applied to the virtual terminal ports of the router with the **`access-class`** command. This allows you to restrict telnet access from a particular IP address or a subnet of IP addresses.

Setting privilege levels

Privilege levels associate router commands with each security level configured on the router. This allows for a finer granularity of control when restricting user access. There are 16 privilege levels contained within the router operating system. Level 2 to level 14 are customizable and allow you to configure multiple privilege levels and multiple passwords to enable certain users to have access to specific commands.

Disabling password recovery

Setting passwords is the first line of defense against intruders. Sometimes passwords are forgotten and must be recovered. All Cisco password recovery procedures dictate that the user performs the password recovery process from the console port of the router or switch. There are, however, certain circumstances in which the widely available password recovery procedure should be disabled. One such circumstance is an emergency Add, Move, or Change (AMC), whereby a networking device needs to be in a location that does not have the proper mechanisms in place for physical security, thus allowing an intruder a greater chance of circumventing traditional security measures.

```
SecureRouter#config t
Enter configuration commands, one per line. End with CNTR/Z.
SecureRouter(config)#no service password-recovery
WARNING:
Executing this command will disable password recovery mechanism.
Do not execute this command without another plan for password recovery.
Are you sure you want to continue? [yes/no]: yes
```

Configuring password encryption

All Cisco console and Telnet passwords configured on the router are stored in plain text within the configuration of the router by default, thus making them easily readable. If someone issues the show running-config privileged mode command, the password is displayed. Another instance when the password can easily be read is if you store your configurations on a TFTP server, the intruder only needs to gain access into the TFTP machine, after which the intruder can read the configuration with a simple text editor. Password encryption stores passwords in an encrypted manner on the router. The encryption is applied to all configured passwords on the router.

It's relatively simple to configure password encryption on Cisco routers. When password encryption is configured, all passwords that are configured on the router are converted to an unsophisticated reversible cipher. Although the algorithm that is used to convert the passwords is somewhat unsophisticated, it still serves a very good purpose. Intruders cannot simply view the password in plain text and know what the password is. To enable the use of password encryption, use the command **`service password-encryption`**. For differences between configuration with encrypted passwords and configuration without encrypted password see Appendix 2.

Setting banner messages

You can use banner messages to issue statements to users, indicating who is and who is not allowed access into the router. Banner messages should indicate the seriousness of an attempt to gain unauthorized access into the device and should never reflect to the user that gaining unauthorized access is acceptable. If possible, recite certain civil and federal laws that are applicable to unauthorized access and let users know what the punishment would be for accessing the device without express written permission. If possible, have certified legal experts within the company review the banner message.

Configuring enable password

To configure enable mode access, you can use one of two commands: **`enable password`** or **`enable secret`**. Both commands accomplish the same thing, allowing access to enable mode. However, the **`enable secret`** command is considered to be more secure because it uses a one-way encryption scheme based on the

MD5 hashing function. Only use the **enable password** command with older IOS images and/or boot ROMs that have no knowledge of the newer **enable secret** command.

You configure an enable password by entering the **enable password *<password>*** command in global configuration mode:

```
SecureRouter#config t
Enter configuration commands, one per line. End with CNTL/Z.
SecureRouter(config)#enable password Omni-Pass01
SecureRouter(config)#enable secret Omni-Pass01
SecureRouter(config)#end
SecureRouter#
```

The preceding configuration sets the enable password to Omni-Pass01. The result of setting the enable password can be seen in the following output. From the user mode prompt, you must enter the enable command to gain access into privileged mode:

```
SecureRouter>enable
Password: Omni-Pass01
SecureRouter#
```

2.2. Securing datalink layer

2.2.1. VLANs

A VLAN is a group of hosts with a common set of requirements that communicate as if they were attached to the same wire, regardless of their physical location. A VLAN has the same attributes as a physical LAN, but it allows for end stations to be grouped together even if they are not located on the same LAN segment. Filtering inside a VLAN is possible using VLAN maps, which is the only way to control filtering within a VLAN. VLAN maps have no direction. To filter traffic in a specific direction using a VLAN map, an access list must be configured with a specific source or destination address. Unlike router ACLs, the default action for VLAN maps is to forward, and this action is taken if the packet does not match any of the entries within the map.

To create a VLAN map and apply it to one or more VLANs, perform these steps:

- Create the standard or extended IP ACLs or named MAC extended ACLs that are going to be applied to the VLAN.
- Enter the **vlan access-map** global configuration command to create a VLAN ACL map entry.
- In **access map** configuration mode, optionally enter an action, either **drop** or **forward**, and enter the **match** command to match the packet against one or more access lists.
- Use the **vlan filter** global configuration command to apply a VLAN map to one or more VLANs.

Remember that VLAN maps have an implicit forward feature at the end of the list; a packet is forwarded if it does not match any ACL within the VLAN map.

Follow these guidelines when configuring VLAN maps:

- If there is no router ACL configured to deny traffic on a routed VLAN interface, and there is no VLAN map configured, all traffic is permitted.
- As with access lists, the order of entries in a VLAN map is important.
- The system might take longer to boot if a large number of ACLs have been configured.

Examples of VLAN maps:

This example shows how to create an ACL and a VLAN map to deny a packet. In this example, any packets that match the ip1 ACL (TCP packets) would be dropped. First create the IP ACL to permit any TCP packet and no other packets; then set the action for packets that match the permit list to be dropped.

```
Switch(config)#ip access-list extended ip1
Switch(config-ext-nacl)#permit tcp any any
Switch(config-ext-nacl)#exit
Switch(config)#vlan access-map map_1 10
Switch(config-access-map)#match ip address ip1
Switch(config-access-map)#action drop
```

The example below shows how to create a VLAN map to permit a packet. In this example, ip2 permits UDP packets and any packets that match the ip2 ACL are forwarded.

```
Switch(config)#ip access-list extended ip2
Switch(config-ext-nacl)#permit udp any any
Switch(config-ext-nacl)#exit
Switch(config)#vlan access-map map_1 20
Switch(config-access-map)#match ip address ip2
Switch(config-access-map)#action forward
```

The following are examples of the show commands for access list and VLAN maps.

```
Switch#show access-list
Extended MAC access list mac1
deny any any decnet-iv
permit any any
Switch#show vlan access-map
Vlan access-map "map_1" 10
Match clauses:
ip address: ip1
Action:
drop
Vlan access-map "map_1" 20
Match clauses:
mac address: mac1
Action:
forward
Vlan access-map "map_1" 30
Match clauses:
Action:
drop
Switch#show vlan filter
VLAN Map map_1 is filtering VLANs:
20-22
```

2.2.2. Management VLAN

Virtual LANs can be used to control who can establish in-band communications with the switch. The switch will use VLAN 1 as the management VLAN unless it is explicitly configured to be on another VLAN. The management VLAN on a switch is the only VLAN from which in-band management sessions can be established. When shipped from the factory, all ports on the switch are configured as members of VLAN1. At factory default settings, the switch will accept management traffic from any port.

Network administrators should separate management traffic and user traffic. The recommended practice for managing network devices (router and switches) is to establish separate logical networks for user traffic and management traffic using multiple VLANs and subnets. The management VLAN should not carry any user traffic. It is also important that the management VLAN be included in the list of VLANs propagated by the VLAN Trunking Protocol (VTP) processes. Separating management and user traffic helps protect the network infrastructure against attacks from users of the network – in most cases it's suitable change management VLAN from default VLAN1 to another (for example VLAN6).

2.2.3. Spanning-Tree Protocol

Spanning-Tree Protocol (STP) is a Layer 2 protocol that utilizes a special-purpose algorithm to discover physical loops in a network and effect a logical loop-free topology. STP creates a loop-free tree structure consisting of leaves and branches that span the entire Layer 2 network. The greater discussion of STP concerns its behavior on trunk connections, which form the spanning tree. Loops can occur in a network for a variety of reasons. Usually, loops in a network are the result of a deliberate attempt to provide redundancy. However, loops can also result from configuration errors. Following three technologies are used with STP:

BackboneFast

BackboneFast is a Catalyst switch feature that is initiated when a Root Port or blocked port on a switch receives inferior BPDUs from its Designated Bridge. An inferior BPDU identifies one switch as both the Root Bridge and the Designated Bridge. When a switch receives an inferior BPDU, it means that a link to which the switch is not directly connected (an indirect link) has failed. Configuration of backbonefast feature is very simple:

```
Switch#configure terminal
Switch(config)#spanning-tree backbonefast
Switch(config)#end
Switch#
```

PortFast

Spanning tree PortFast is a Catalyst feature that causes a switch or trunk port to enter the spanning tree Forwarding state immediately, bypassing the Listening and Learning states. IOS-based switches only use PortFast on access ports connected to end stations. When a device is connected to a port, the port normally enters the spanning tree Listening state. When the Forward Delay timer expires, the port enters the Learning state. When the Forward Delay timer expires a second time, the port is transitioned to the Forwarding or Blocking state. When PortFast is enabled on a switch or trunk port, the port is immediately transitioned to the Forwarding state. As soon as the switch detects the link, the port is transitioned to the Forwarding state (less than 2 seconds after the cable is plugged in). If a loop is detected and PortFast is enabled, the port is transitioned to the Blocking state. Configuration of PortFast feature is following:

```
Switch#configure terminal
Switch(config)#interface fastethernet <interface number>
Switch(config-if)#spanning-tree portfast
Switch(config-if)#end
Switch#
```

UplinkFast

UplinkFast accelerates the choice of a new Root Port when a link or switch fails or when STP reconfigures itself. The Root Port transitions to the Forwarding state immediately without going through the Listening and Learning states, as it would with the usual STP process. UplinkFast is most useful in wiring-closet switches at the edge of the network. It is not appropriate for backbone devices. UplinkFast provides fast convergence after a direct link failure and achieves load balancing between redundant links using uplink groups. Configuration of backbonefast feature is very simple:

```
Switch#configure terminal
Switch(config)#spanning-tree uplinkfast
Switch(config)#end
```

2.3. Securing network layer

Routers and their routing decisions are made on network layer. Securing network layer is very important especially in network with dynamic routing protocols (for example RIP, EIGRP or OSPF). Most secure is static routing – on routers are static routes only and it is impossible to change routing table using spurious routes. Secure network use authentication in routing protocols, routes filtering and routes summarization.

2.3.1. Configuring RIP authentication

There are two versions of Routing Information Protocol (RIP): version 1 and version 2. RIP version 1 does not support authentication of routing updates; however, RIP version 2 supports both plaintext and MD5 authentication. Configuring authentication of RIP version 2 updates is fairly easy and very uniform. The basic configuration includes the following steps:

- Define the key chain using the command **`key-chain <name>`** in global configuration mode. This command transfers you to the key chain configuration mode.
- Specify the key number with the **`key <number>`** command in key chain configuration mode. You can configure multiple keys. For each key, identify the key string with the **`key-string <string>`** command.
- Configure the period for which the key can be sent and received. Use the following commands: **`accept-lifetime <starttime> {infinite|end-time|duration}`** and **`send-lifetime <starttime> {infinite|end-time|duration}`**
- Exit key chain configuration mode with the **`exit`** command.
- Under interface configuration mode, enable the authentication of RIP updates with this command: **`ip rip authentication key-chain <key chain name>`** - this command is all that is needed to use plain text authentication.

Optionally, under interface configuration mode enable MD5 authentication of RIP updates using the **`ip rip authentication mode md5`** command. For listing configuration with RIP authentication and debug example see Appendix 3.

2.3.2. Configuring EIGRP Authentication

EIGRP authentication of packets has been supported since IOS version 11.3. EIGRP route authentication is similar to RIP version 2, but EIGRP authentication supports only the MD5 version of packet encryption. EIGRP's authentication support may at first seem limited, but plain text authentication should be configured only when neighboring routers do not support MD5. Because EIGRP is a proprietary routing protocol developed by Cisco, it can be spoken only between two Cisco devices, so the issue of another neighboring router not supporting the MD5 cryptographic checksum of packets should never arise. The steps for configuring authentication of EIGRP updates are similar to the steps for configuring RIP version 2 authentication:

- Define the key chain using the command **`key-chain <name>`** in global configuration mode. This command transfers you to the key chain configuration mode.
- Specify the key number with the **`key <number>`** command in key chain configuration mode. You can configure multiple keys. For each key, identify the key string with the **`key-string <string>`** command.

- Optionally, you can configure the period for which the key can be sent and received. Use the following commands: **`accept-lifetime <starttime> {infinite|end-time|duration}`** and send**`-lifetime <starttime> {infinite|end-time|duration}`**.
- Exit key chain configuration mode with the **`exit`** command.
- Under interface configuration mode, enable the authentication of EIGRP updates with this command: **`ip authentication key-chain eigrp <autonomous system> <key chain name>`**

Optionally enable MD5 authentication of EIGRP updates using the following command: **`ip authentication mode eigrp <autonomous system> md5`**. For listing configuration with EIGRP authentication and debug example see Appendix 4.

2.3.3. Configuring OSPF Authentication

Open Shortest Path First (OSPF) supports two forms of authentication: plain text and MD5. Plaintext authentication should be used only when neighboring devices do not support the more secure MD5 authentication. To configure plaintext authentication of OSPF packets, follow these steps:

- In interface configuration mode, use the **`ip ospf authentication-key <key>`** command. The key that is specified is the plaintext password that will be used for authentication.
- Enter OSPF configuration mode using the **`router ospf <process id>`** command. Then use the **`area <area-id> authentication`** command to configure plain text authentication of OSPF packets for an area.

To configure MD5 authentication of OSPF packets, follow the steps outlined here:

- From interface configuration mode, enable the authentication of OSPF packets using MD5 with the following command: **`ip ospf message-digest-key <key-id> md5 <key>`** - the value of the key-id allows passwords to be changed without having to disable authentication.
- Enter OSPF configuration mode using the **`router ospf <process id>`** command. Then configure MD5 authentication of OSPF packets for an area using this command: **`area <area-id> authentication message-digest`**

This time, Routers A and B will be configured to authenticate packets across the backbone using the MD5 version of authentication. For listing configuration with OSPF authentication and debug example see Appendix 5.

2.4. Configuring route filters

Route filters work by regulating what networks a router will advertise out of an interface to another router or what networks a router will accept on an interface from another router. Route filtering can be used by administrators to manually assure that only certain routes are announced from a specific routing process or interface. This feature allows administrators to configure their routers to prevent malicious routing attempts by intruders. You can configure route filtering in one of two ways:

- **Inbound route filtering** – the router can be configured to permit or deny routes advertised by a neighbor from being installed to the routing process.
- **Outbound route filtering** – the route filter can be configured to permit or deny routes from being advertised from the local routing process, preventing neighboring routers from learning the routes.

2.4.1. Configuring Inbound Route Filters

The steps for configuring inbound route filters are as follows:

- Use the **access list** global configuration command to configure an access-list that permits or denies the specific routes that are being filtered.
- Under the routing protocol process, use the following command: **distribute-list <access-list-number> in [interface-name]**

In this example, an inbound route filter will be configured on Router B to deny routes from being installed into its routing process. Following configuration of routers A and B.

```
interface Loopback0
ip address 10.10.10.1 255.255.255.0
!
interface Loopback1
ip address 10.10.11.1 255.255.255.0
!
interface Ethernet0/0
ip address 10.10.12.1 255.255.255.0
!
interface Serial0/0
ip address 192.168.10.1 255.255.255.252
!
router rip
version 2
network 10.0.0.0
network 192.168.10.0
no auto-summary
```

Router B configuration:

```
interface Loopback0
ip address 10.10.13.1 255.255.255.0
!
interface Loopback1
ip address 10.10.14.1 255.255.255.0
!
interface FastEthernet0/0
ip address 10.10.15.1 255.255.255.0
!
interface Serial0/0
ip address 192.168.10.2 255.255.255.252
!
router rip
version 2
network 10.0.0.0
network 192.168.10.0
no auto-summary
```

Taking a look at the route table of Router B, notice that it has learned of three networks from Router A: 10.10.10.0, 10.10.11.0, and 10.10.12.0. Router B's route table:

```
Router-B#show ip route
C 10.10.13.0 is directly connected, Loopback0
C 10.10.14.0 is directly connected, Loopback1
C 10.10.15.0 is directly connected, FastEthernet0/0
R 10.10.10.0 [120/1] via 192.168.10.1, 00:00:16, Serial0/0
R 10.10.11.0 [120/1] via 192.168.10.1, 00:00:16, Serial0/0
R 10.10.12.0 [120/1] via 192.168.10.1, 00:00:16, Serial0/0
```

Now, a route filter will be configured on Router B to deny the 10.10.10.0 and 10.10.11.0 networks from being installed into the route table. This will allow only the 10.10.12.0 network to be installed into the route table from

Router A. Use the **access-list <number>** command to configure the router with a standard access list and use the **distribute-list <list number> in <interface>** command to apply the access list under the routing process. New configuration of router B:

```
interface Serial0/0
ip address 192.168.10.2 255.255.255.252
!
router rip
version 2
network 10.0.0.0
36
network 192.168.10.0
distribute-list 1 in Serial0/0
no auto-summary
!
access-list 1 permit 10.10.12.0
```

Looking back again at Router B's route table after applying the route filter, you can see that the 10.10.12.0 network is the only network that Router B is allowing to be installed into its route table:

```
Router–B#show ip route
C 10.10.13.0 is directly connected, Loopback0
C 10.10.14.0 is directly connected, Loopback1
C 10.10.15.0 is directly connected, FastEthernet0/0
R 10.10.12.0 [120/1] via 192.168.10.1, 00:00:16, Serial0/0
```

2.4.2. Suppressing Route Advertisements

To prevent other routers on a network from learning about routes dynamically, you can prevent routing update messages from being sent out a router interface. To accomplish this, use the **passive-interface <interface>** routing process configuration command. This command can be used on all IP-based routing protocols except for the Exterior Gateway Protocol (EGP) and Border Gateway Protocol (BGP). When an interface is configured to be in a passive state, the router disables the passing of routing protocol advertisements out of the interface; however, the interface still listens and accepts any route advertisement that is received into the interface. Configuring this on a router essentially makes the router a silent host over the interfaces that were specified. To configure an interface as passive, use the **passive-interface <interface>** command under routing protocol configuration mode; this command is all that is needed to make an interface no longer advertise networks. Here is an example of configuring an interface as passive:

```
interface FastEthernet0/0
ip address 10.10.15.1 255.255.255.0
!
interface Serial0/0
ip address 192.168.10.2 255.255.255.252
!
router eigrp 50
passive-interface FastEthernet0/0
passive-interface Serial0/0
```

3. Securing transport and application layer

3.1. Access lists

Securing transport layer is accomplished especially by access-list. Although we can use standard access-lists which control only layer 3 address, most used are extended access-lists. Extended access-lists can match more entries than standard access-lists and are used for filtering traffic based on protocols port numbers.

ACLs filter network traffic by controlling whether routed or switched packets are forwarded or blocked at the router or switch interface. The router or switch examines each packet to determine whether to forward or drop the packet, based on the criteria specified within the ACL. ACL criteria can be the source address of the traffic, the destination address of the traffic, or the upper-layer protocol. An ACL is constructed in two steps:

- Create an ACL.
- Apply the ACL to the proper interface and direction.

The IP ACL is a sequential collection of permit and deny conditions that apply to an IP address. The router or switch tests addresses against the conditions in the ACL one at a time. The first match determines whether the Cisco IOS software accepts or rejects the address. Because the Cisco IOS software stops testing conditions after the first match, it is important to order the conditions in the most efficient way. If no conditions match, the router rejects the address, due to an implicit deny any clause, which is fixed at the end of the list of conditions.

Editing an ACL requires special attention. For example, if the administrator were to delete a specific line from an existing numbered ACL, the entire ACL would be deleted. To edit numbered ACLs, copy the configuration of the router to a TFTP server or a text editor such as Notepad, make any changes, and copy the configuration back to the router. Beware that any deletions will be removed from the ACL and any additions will be made to the end of the ACL. In a production environment, changing any ACL could affect the security of the router during modifications.

Access-list are divided into categories which are listed in following table:

Access-list number	Description
1 – 99	standard IP access-list
100 – 199	extended IP access-list
200 – 299	protokol type-code access-list
300 – 399	DECnet access-list
400 – 499	standard XNS access-list
500 – 599	extended XNS access-list
600 – 699	AppleTalk access-list
700 – 799	access-list pro 48-bit MAC address
800 – 899	standard IPX access-list
900 – 999	extended IPX access-list
1000 – 1099	IPX SAP access-list
1100 – 1199	extended access-list for 48-bit MAC address
1200 – 1299	IPX summary access-list
1300 – 1999	standard IP access-list (expanded rang)
2000 - 2699	extended IP access-list (expanded rang))

Table 4: Access-list numbers.

3.1.1. Applying access-lists

ACLs can be defined without applying them. To make an ACL effective, the ACL must be applied to the interface of the router. Depending on where we place an ACL statement, we can reduce unnecessary traffic. Traffic that will be denied at a remote destination should not use network resources along the route to that destination. The rule is to put the extended ACLs as close as possible to the source of the traffic denied. Standard ACLs do not specify destination addresses, so you have to put the standard ACL as near the destination as possible.

Once it is determined what type of ACL should be defined, the next step is to determine where the ACL should be applied. No traffic is processed until an ACL is applied to an interface. When applying ACL to the interface we must specify the direction of the traffic. Here is example of applying extended access list 101 on FastEthernet interface:

```
Router#configure terminal
Router(config)#interface FastEthernet0/0
Router(config)#ip access-group 101 in
Router(config)#ip access-group 101 out
```

Directions of applied ACL are in following table:

ACL applied in outbound direction	traffic that has already been through the router and is leaving the interface the source would be where it has been or the other side of the router the destination where it is going
ACL applied in inbound direction	traffic that is arriving on the interface and will go through the router source would be where it has been destination is where it is going or the other side of the router

Table 5: Directions of access-lists

3.1.2. Standard access-lists

A number of different ACL types are available to filter traffic. Standard ACLs are the oldest type of ACL. Standard ACLs control traffic by comparing the source address of the IP packets to the addresses configured in the ACL. This is useful for example for defining range of address for NAT or for route filtering.

Example of standard access-list:

```
access-list 1 permit 10.10.0.0 0.0.3.255
access-list 1 permit 10.10.10.0 0.0.0.255
access-list 1 permit 10.10.11.0 0.0.0.255
```

3.1.3. Extended access-list

Extended ACLs control traffic by comparing both the source and destination addresses of the IP packets to the addresses configured in the ACL. For IP extended access lists, there are a number of well-known protocols that can be defined. The command syntax format of extended ACLs is following:

```
Router(config)#access-list access-list-number {permit | deny} protocol
source [source-mask destination destination-mask operator operand]
[established]
```

Example of extended access-list which deny FTP access from network 172.16.4.0/24 to the network 172.16.3.0/24:

```
Router(config)#access-list 101 deny 172.16.4.0 0.0.0.255 172.16.3.0
0.0.0.255 eq 21
Router(config)#access-list 101 permit ip 172.16.4.0 0.0.0.255 0.0.0.0
255.255.255.255
```

After applying this access-list to the interface Ethernet0 is configuration finished:

```
Router(config)#interface Ethernet0
Router(config-if)#ip access-group 101
```

3.1.4. Named access-list

Named ACLs allow standard and extended IP ACLs to be identified with an alphanumeric string (name) instead of the current numeric (1 to 199) representation. Named ACLs can be used to delete individual entries from a specific ACL. This enables you to modify your ACLs without deleting and then reconfiguring them. Use named ACLs when:

- We want to intuitively identify ACLs using an alphanumeric name.
- We have more than 99 simple and 100 extended ACLs to be configured in a router for a given protocol.

Consider the following before implementing named ACLs:

- Named ACLs are not compatible with Cisco IOS releases prior to Release 11.2.
- We cannot use the same name for multiple ACLs. In addition, ACLs of different types cannot have the same name. For example, it is illegal to specify a standard ACL named George and an extended ACL with the same name.

To name the ACL, use the following command:

```
Router(config)#ip access-list {standard | extended} name
```

In ACL configuration mode, specify one or more conditions permitted or denied. This determines whether the packet is passed or dropped:

```
Router(config {std- | ext-}nacl)#deny {source [source-wildcard] | any}
```

or

```
Router(config {std- | ext-}nacl)#permit {source [source-wildcard] | any}
```

In all access-lists type it's good to use remarks. As good programmers use remarks liberally in their programs to describe the function of certain blocks of code, comments make ACLs easier to understand and can be used for standard or extended IP ACLs. By describing access lists in simple terms, network engineers can quickly sum up the configuration of a router without having to sift through dozens of access list statements in an attempt to piece together its function. For description of ACL is used keyword **remark**.

After entering the **remark** keyword, the designer can include an alphanumeric string of up to 100 characters to describe the access list. A remark statement should be entered before configuring the permits and denies. This way, the description will appear as the first entry in the configuration file. In a long list, the network administrator may want to include multiple remark statements and enter them before each part of the list that requires description. Example of commented ACL:

```
access-list 101 remark --= Permit SMTP on 172.16.0.1 for host 10.0.0.1 =--
access-list 101 permit tcp host 10.0.01 host 172.16.0.1 eq 25
access-list 101 deny tcp any host host 172.16.0.1 eq 25
access-list 101 remark --= Deny SNMP to network 172.16.0.0/24 =--
access-list 101 deny udp any 172.16.0.0 0.0.0.255 eq 161
access-list 101 deny udp any 172.16.0.0 0.0.0.255 eq 162
access-list 101 deny tcp any 172.16.0.0 0.0.0.255 eq 161
access-list 101 deny tcp any 172.16.0.0 0.0.0.255 eq 162
access-list 101 remark --= Permit http/https for network 172.16.0.0/24 =--
access-list 101 permit tcp 172.16.0.0 0.0.0.255 any eq http
access-list 101 permit tcp 172.16.0.0 0.0.0.255 any eq https
```

3.1.5. Time-based access-list

Today's network security policies demand that access lists do more than use destination and source addresses to statically define whether a protocol is permitted. In some cases, an administrator may determine that certain traffic is permissible only during business hours, or that users have access to specific resources only at fixed times of day. This is possible using a time-based access list. Since IOS release 12.0.1(T), it is possible to implement time-based access lists based on the time of day and week by using the **time-range** command. There are many possible benefits of using time ranges, including the following:

- To provide more control over permitting or denying a user access to resources; these resources could be an application (identified by an IP address/mask pair and a port number), or an on-demand link (identified as interesting traffic to the dialer).
- To set time-based security policy, including the following:
- Perimeter security using the Cisco IOS Firewall feature set or access lists
- Data confidentiality with Cisco Encryption Technology or IP Security Protocol (IPSec)
- To provide enhanced policy-based routing and queuing functions
- To automatically reroute traffic cost effectively when provider access rates vary by time of day
- To support the quality of service (QoS) service-level agreements (SLAs) that are negotiated for certain times of day when service providers can dynamically change a committed access rate (CAR) configuration
- To control logging messages. Access list entries can log traffic at certain times of the day, but not constantly. Therefore, administrators can simply deny access without needing to analyze many logs generated during peak hours.

To implement a time-based extended access list, first define the name and times of the day and week of the time range, and then reference the time range by name in an access list. To apply restrictions to the access list, using the following steps:

1. Define a time range using a name:

```
router(config)#time-range time-range-name
```

2. In time-range configuration mode, use the **periodic** command, the **absolute** command, or some combination of them to define when the feature is in effect. Multiple **periodic** commands are allowed in a time range; only one **absolute** command is allowed. The **periodic** keyword specifies a recurring (weekly) start and end time for a time range. The **absolute** keyword specifies an absolute start and end time for a time range:

```
router(config-time-range)# periodic days-of-the-week hh:mm to[days-of-the-week] hh:mm
router(config-time-range)# absolute [start time date] [end time date]
```

The **periodic** command will take the following arguments: Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday. Other possible values are **daily** (Monday through Sunday), **weekdays** (Monday through Friday), and **weekend** (Saturday and Sunday).

3. Exit the time-range configuration mode:

```
router(config-time-range)#exit
```

Currently, IP and IPX named or numbered extended access lists are the only functions that can use time ranges. The time range allows the network administrator to define when the **permit** or **deny** statements in the access list are in effect. Prior to this feature, access list statements were always in effect once they were applied.

Example of time-based access-list:

```
Router#configure terminal
Router(config)#time-range NO-HTTP
Router(config-time-range)#periodic weekdays 8:00 to 18:00
Router(config-time-range)#exit
Router(config)#time-range UDP-YES
Router(config-time-range)#periodic weekend 12:00 to 20:00
Router(config-time-range)#exit
Router(config)#ip access-list extended STRICT
Router(config-ext-nacl)#deny tcp any any eq http time-range NO-HTTP
Router(config-ext-nacl)#permit udp any any time-range UDP-YES
Router(config-ext-nacl)#deny udp any any range netbios-ns netbios-ss
Router(config-ext-nacl)#permit ip any any
```

3.1.6. Lock and key access-list

Lock-and-key is a Cisco IOS feature that enables users to temporarily open a hole in a firewall without compromising other configured security restrictions. This feature is configured using a type of extended access list called a dynamic access list. In practice, lock-and-key users are typically power users or systems administrators because the user must Telnet to a Cisco router to create the hole in the firewall. However, some administrators may automate the procedure using a process such as scripts so that intermediate users can take advantage of this feature.

Dynamic access lists enable designated users to gain temporary access to protected resources from any IP address, or, from any specific addresses that you choose. When configured, lock-and-key modifies the existing IP access list of the interface so that it permits the IP addresses of designated users to reach specific destinations. After the user has disconnected, lock-and-key returns the access list back to its original state.

For lock-and-key to work, the user must first telnet to the router. When telneting, the user is provided an opportunity to tell the router who he or she is (by authenticating with a username and a password), and what IP address he or she is currently sending from. If the user successfully authenticates to the router, the user's IP address can be granted temporary access through the router. The dynamic access list configuration determines the extent of the access granted.

To configure lock-and-key, you start by defining a dynamic access list using the following syntax:

```
Router(config)#access-list access-list-number dynamic dynamic-name [timeout
minutes] [deny | permit] protocol source-address source-wildcard
destination-address destination-wildcard
```

Example of simple lock-and-key access-list:

```
RTA(config)#access-list 101 permit tcp any host 192.168.1.1 eq telnet
RTA(config)#access-list 101 dynamic UNLOCK timeout 120 permit ip any any
RTA(config)#int s0
RTA(config-if)#ip access-group 101 in
```

3.1.7. Reflexive access-list

Reflexive access-lists are very useful when configuring firewall on Cisco router. Reflexive access lists provide the capability to filter network traffic at a router, based on IP upper-layer protocol "session" information. Like the **`established`** argument, you can use reflexive access lists to permit IP traffic for sessions originating from within your network but to deny IP traffic for sessions originating from outside your network. Unlike the **`established`** argument, reflexive access lists can do this with all Internet protocols, not just TCP.

On other side – reflexive access lists unfortunately do not work with some applications that use port numbers that change during a session. For example, if the port numbers for a return packet are different from the originating packet, the return packet will be denied, even if the packet is actually part of the same session. FTP is an example of an application with changing port numbers. When you initiate an FTP session, that conversation typically uses TCP port 21 to send control information, including the three-way handshake and username/password negotiation.

Example of reflexive access-list configuration

The figure shows a reflexive access list for RTA. To create this list, first define and apply the outbound list that will be reflected.

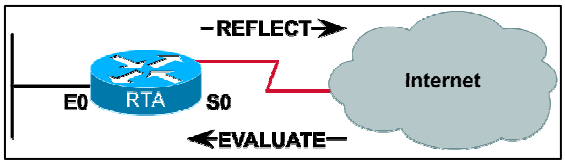

Figure 10: Topology with reflexive access-list

The reflection will generate reflexive access list entries:

```
RTA(config)#ip access-list extended OUTBOUND
RTA(config-ext-nacl)#permit ip any any reflect INVITED-TRAFFIC
RTA(config-ext-nacl)#exit
RTA(config)#interface serial0
RTA(config-if)#ip access-group OUTBOUND out
```

The commands shown in the above example create an extended named access list called OUTBOUND. This list includes an entry that creates the reflexive list, INVITED-TRAFFIC. Entries for INVITED-TRAFFIC will be generated dynamically based on a reflection of the outbound traffic flow.

Next, configure an inbound list that will match incoming traffic (traffic coming in from the Internet) to this reflexive list, as shown:

```
RTA(config)#ip access-list extended INBOUND
RTA(config-ext-nacl)#evaluate INVITED-TRAFFIC
RTA(config-ext-nacl)#exit
RTA(config)#interface serial0
RTA(config-if)#ip access-group INBOUND in
```

The commands in this example create an extended named access list called INBOUND. This list will be used to match traffic coming in from the Internet. Although you could include other entries, the only one shown here is the evaluate statement, which is a reflexive access list nested inside the list, INBOUND. This evaluate statement instructs the router to permit only traffic that matches the INVITED-TRAFFIC reflexive access list. If desired, you can set a global timeout to something other than the default, as shown:

```
RTA(config)#ip reflexive-list timeout 200
```

When configured with a reflexive access list, RTA presents a sophisticated firewall, but still a limited one. None of the access lists discussed so far in this chapter can go beyond Layer 4 to filter traffic based on application. However, the next generation of access lists, context-based access control, can do just that.

3.2. Context-based access-lists

CBAC provides users better protection from attacks. Table on the next page shows the protocols supported by CBAC, describes the added alert and audit trail features, and lists the CBAC configuration tasks. CBAC creates temporary openings in access lists at firewall interfaces. These openings occur when specified traffic exits the internal network through the firewall. CBAC allows the traffic back through the firewall only if it is part of the same session as the original traffic that triggered CBAC when exiting the firewall.

The following tasks are involved in the configuration of CBAC:

- Set audit trails and alerts.
- Set global timeouts and thresholds.
- Define Port to Application Mapping(PAM).
- Define inspection rules.
- Apply inspection rules and ACLs to interfaces.
- Test and verify.

Layer	Protocol
Application	VDOLiveRCP (Sun RPC, not DCE RPC) Microsoft RCP FTP TFTP UNIX R-commands SMTP Java SQL*Net RTSP (RealNetworks) H.323 (NetMeeting, ProShare, CU-SeeMe)
Presentation	all TCP and UDP sessions regardless of the application-layer
Session	
Transport	protocols: TCP, UDP, ICMP, GRE, IGRP, EIGRP ACL-filter on source and destination protocols/port numbers
Network	protocol: IP standard ACL-filters on source IP address extended-ACL filters on source and destination IP address
Datalink	MAC filtering standard ACL-filters on source MAC address extended-ACL filters on source and destination MAC address
Physical	

Table 6: Protocols supported by CBAC

A useful feature of CBAC is its ability to generate alerts and audit trails. This makes monitoring and tracking pre-defined security events much more efficient and effective. The alert and audit trail process works as follows:

- CBAC generates real-time alerts and audit trails based on events tracked by the firewall.
- Enhanced audit trail features use Syslog to track all network transactions, while recording time stamps, source host, destination host, ports used, and the total number of transmitted bytes for advanced, session-based reporting.
- Real-time alerts send Syslog error messages to central management consoles upon detecting suspicious activity.

Note that when using CBAC inspection rules, it is possible to configure alerts and audit trail information on a per-application protocol basis. For example, to generate audit trail information for HTTP traffic, simply specify that in the CBAC rule covering HTTP inspection.

Enabling alerts and audit trails

To disable CBAC alert messages, which are displayed on the console, use the ip inspect alert off command in global configuration mode. To enable CBAC alert messages, use the no form of this command.

```
ip inspect alert-off
no ip inspect alert-off
```

To turn on CBAC audit trail messages, which are displayed on the console after each CBAC session closes, use the **`ip inspect audit trail`** command in global configuration mode. Use the no form of this command to turn off CBAC audit trail messages.

Example of CBAC - configuring application layer protocol inspection

To configure CBAC inspection for an application layer protocol (except for RPC and Java), use the following command syntax:

```
Router(config)#ip inspect name inspection-name protocol [timeout seconds]
```

The protocol option can be any one of several possible arguments. Repeat this command for each desired protocol. Use the same inspection-name to create a single inspection rule, as shown:

```
RTA(config)#ip inspect name FIREWALL http
RTA(config)#ip inspect name FIREWALL ftp
RTA(config)#ip inspect name FIREWALL udp
RTA(config)#interface s0
RTA(config-if)#ip inspect FIREWALL out
```

These commands create a CBAC inspect list named FIREWALL that is applied to outbound traffic exiting interface S0. RTA will inspect outbound traffic and create dynamic access list entries to allow inbound traffic through the firewall, if it is part of the session started by an internal host.

The syntax for configuring a CBAC inspection for Java is as follows :

```
Router(config-if)#ip inspect name inspection-name http [java-list access-
list] [timeout seconds]
```

Java applets can represent a security risk because unaware users can download them into your network and then run malicious code behind your firewall. You can configure CBAC to filter Java applets at the firewall, which enables users to download only applets residing within the firewall and trusted applets from outside the firewall.

Java applet filtering distinguishes between trusted and untrusted applets by relying on a list of external sites that you designate as „friendly". If an applet is from a friendly site, the firewall allows the applet through. If the applet is not from a friendly site, the applet is blocked.

Following configuration blocks Java applets from sites known to be a risk, but to permit all others:

```
RTA(config)#access-list 24 deny 200.100.50.0 0.0.0.255
RTA(config)#access-list 24 deny 169.199.0.0 0.0.255.255
RTA(config)#access-list 24 permit any
RTA(config)#ip inspect name FIREWALL http java-list 24
RTA(config)#ip inspect name FIREWALL tcp
RTA(config)#interface s0
RTA(config-if)#ip inspect FIREWALL out
```

If RTA is configured accordingly, it will inspect traffic for Java and match it according to access list 24. Holes will not be opened in the firewall for Java traffic originating from the explicitly defined networks. Of course, the permit any statement makes this configuration extremely vulnerable to the thousands of other sites that may infect your network with malicious Java code. If you want to sacrifice functionality and end-user freedom, you can use an access list to explicitly permit Java code from friendly networks, and deny code from all others. The result will be a secure but highly restrictive configuration.

CBAC uses timeouts and thresholds to determine how long to manage state information for a session, and to determine when to drop sessions that do not become fully established. These timeouts and thresholds apply globally to all sessions. You can use the default timeout and threshold values, or you can change to values more suitable to your security requirements. You should make any changes to the timeout and threshold values before you continue configuring CBAC. For timeouts used by CBAC see Appendix 6.

4. DESIGNING FIREWALL

4.1. Why is a firewall needed?

The firewall exists to enforce the enterprise security. It enables a company to do business online, while providing the necessary security between the internal network of the enterprise and an external network. This external network can be an extranet linking the enterprise to its corporate partners or an Internet link through which the enterprise gains access to customers or remote users. This means that a perimeter firewall makes an ideal location for outward-facing resources such as Web and FTP servers. A firewall can be configured to allow Internet access to these systems while blocking or filtering admission to other protected resources.

Usually, IP traffic forwarding is disabled on the firewall to ensure that all traffic between the internal network and external networks passes through the firewall. This allows the firewall to inspect all network packets that traverse the network boundary. In addition to access control, the firewall also provides a natural focal point for the administration of other network security measures. Firewalls rely on one of three technologies:

- **Packet filtering** – limits information into a network based on static packet header information. Typical example of access lists.
- **Proxy server** – requests connections between a client on the inside of the firewall and the Internet.
- **Stateful packet filtering** – combines the best of packet filtering and proxy server technologies. Stateful packet filtering is the method used by the Cisco PIX Firewall.

4.2. Cisco PIX firewall overview

The PIX Firewall features the following technologies and benefits:

- **Finesse OS** – finesse is a non-UNIX, secure, real-time, embedded system. Unlike typical CPU-intensive proxy servers that perform extensive processing on each data packet at the application layer, the PIX Firewall uses a secure, real-time, embedded system, which enhances the security of the network.
- **Adaptive Security Algorithm (ASA)** – implements stateful connection control through the PIX Firewall.
- **Cut-through proxy** – a user-based authentication method of both inbound and outbound connections that provides improved performance in comparison to that of a proxy server.
- **Stateful packet filtering** – a secure method of analyzing data packets that places extensive information about a data packet into a table. For a session to be established, information about the connection must match the information in the table.

4.2.1. Cisco PIX family

The Cisco PIX Firewall 500 series scales to meet a range of requirements and network sizes. It currently consists of the following five models:

- **PIX Firewall 501** – has an integrated 10BaseT port and an integrated four-port 10/100 switch.
- **PIX Firewall 506E** – has dual integrated 10BaseT ports.
- **PIX Firewall 515E** – provides a modular chassis to support additional single-port or four-port 10/100 Ethernet cards.
- **PIX Firewall 525** – provides a modular chassis to support additional single-port or four-port 10/100 Fast Ethernet and Gigabit Ethernet.
- **PIX Firewall 535** – supports Fast Ethernet and Gigabit Ethernet. The PIX Firewall 515E, 525, and 535 models come with an integrated VPN Accelerator card.

For detail information about PIX Firewall see following table.

Model	501	506E	515E	525	535
					
Market	SOHO	ROBO	SMB	Enterprise	Enterprise +, SP
Licensed users	10 or 50 or unlimited	Unlimited	Unlimited	Unlimited	Unlimited
Max VPN peers	10	25	2,000	2,000	2,000
Size (RU)	<1	1	1	2	3
Proccesson (MHz)	133	300	433	600	1 GHz
Ram (MB)	16	32	64	256	1 GB
Max. interface	1 10BT +4 FE	2 10 BaseT	6	8	10
Failover	No	No	Yes	Yes	Yes
Clear text (Mbps)	60	100	188	360	1.7 Gbps
3Des/ AES (Mbps)	3/4.5	16/30	140/140	155/170	440/535
Max. Connections	20,000	50,000	175,000	650,000	1,000,000
* VAC+ v6.3					

Table 7: Comparing PIX Firewall family.

4.2.2. Administrative modes

The PIX Firewall contains a command set based on the Cisco IOS but not identical to Cisco IOS command set. The PIX Firewall provides five administrative access modes:

- **Unprivileged mode** – this mode is available when the user first accesses the PIX Firewall. The > prompt is displayed. This mode enables users to view a subset of all commands available.
- **Privileged mode** – this mode displays the # prompt and enables the user to change the current settings. Any unprivileged command also works in privileged mode.
- **Configuration mode** – this mode displays the (config)# prompt and enables the user to change system configurations. Unlike the IOS, all privileged and unprivileged commands work in this mode as well. Also, excluding a few cases, the PIX Firewall does not have configuration sub modes such as config-if, config-router, and so on.
- **Setup mode** – this mode allows configuration through interactive prompts. This mode is initiated when a PIX Firewall cannot find a configuration file during the boot process. The mode can also be initiated from configuration mode by typing in **`setup`** command. To exit the setup mode at any time type in ctrl-Z.
- **Monitor mode** – this mode displays the monitor> prompt after issuing a break command during the boot process. This special mode enables network administrators to update the image over the network. While in the monitor mode, administrators can enter commands specifying the location of the TFTP server and the binary image to download.

4.2.3. Basic PIX Firewall configuration commands

There are five basic PIX Firewall configuration commands to enable basic operation.

- The **`nameif`** command assigns a name to each perimeter interface on the PIX Firewall and specifies its security level, except for the inside and outside PIX Firewall interfaces, which are named by default.
- The **`interface`** command identifies the type of hardware, sets its hardware speed, and enables the interface.
- The **`ip address`** command is used to configure the IP address and netmask of each interface on the PIX Firewall.

- The **nat** command can specify translation of the internal, unregistered IP address for a single host or a range of hosts into registered, globally accepted IP address.
- If the **nat** command is used, the companion command, global, must be configured to define the pool of translated IP addresses.
- The **route** command is used to add static route statements to interface configurations. By default, the PIX Firewall will not know how to forward a packet with a destination address for a network that is not directly connected to it. The **route** command enables the PIX Firewall to handle such packets.

Other general PIX Firewall commands that are commonly used for configuration tasks include:

- The **name** command enables administrators to configure a list of name-to-IP address mappings on the PIX Firewall
- The **clock** set command sets the PIX Firewall clock
- The **ntp server** command synchronizes the PIX Firewall with the specified network time server

4.2.4. Examine PIX Firewall status

Now that some of the basic configuration steps have been examined, this section will discuss how to view the status of the PIX Firewall. The following are some basic troubleshooting and performance monitoring commands:

- The **show memory** command displays a summary of the maximum physical memory and current free memory available to the PIX Firewall operating system.
- The **show version** command can be used to display the PIX Firewall software version, operating time since the last reboot, processor type, Flash memory type, interface boards, serial number (BIOS identification), and activation key value.
- The **show ip address** command enables users to view which IP addresses are assigned to the network interfaces.
- The **show interface** command is one of the most common and useful troubleshooting commands available to the network administrator. This command enables the viewing of a significant amount of network interface information in a very compact space. It is one of the first commands that should be used when trying to establish connectivity.
- The **show cpu usage** command displays the CPU usage.
- The **ping** command for the PIX Firewall is identical to the **ping** command for Cisco routers and is used for the same purpose. The **ping** command determines if the PIX Firewall has connectivity, or if a host is available or visible to the PIX Firewall on the network.

4.3. PIX Firewall configuration

4.3.1. Network Address Translation

PIX Firewall supports NAT. It does this for two primary reasons:

- NAT can help conserve a limited number of IP addresses that a company may possess. It can do this because all of the hosts on the inside network, which use private IP addresses, will probably not need to reach an outside network simultaneously. It is, therefore, possible for an administrator to allocate fewer public addresses than there are hosts on the inside network.
- NAT provides an additional layer of security, because it does not allow hosts on the outside network to see the IP addressing scheme of the inside network. This makes it more difficult for attackers to locate specific devices that the attacker may wish to compromise.

4.3.2. Translation Types

The PIX Firewall supports the following four types of Network Address Translations (NATs):

Dynamic inside network address translation NAT

Translates host addresses on more secure interfaces to a range or pool of IP addresses on a less secure interface. This allows internal users to share registered IP addresses and hides internal addresses from view on the public Internet. Next figure show example of topology with dynamic inside network translation.

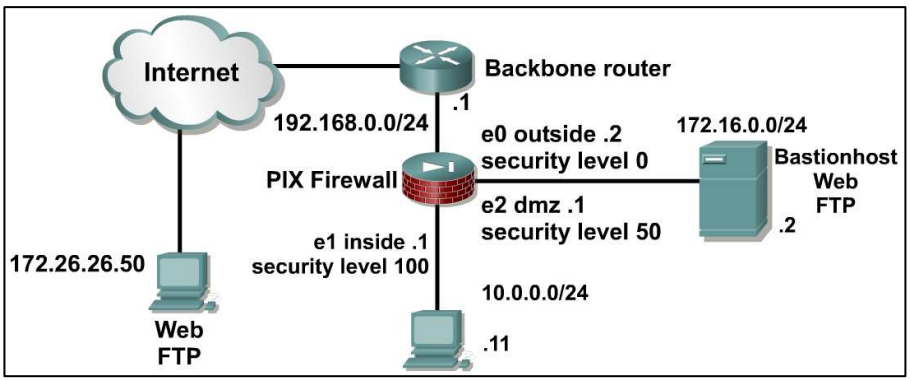

Figure 11: Example of dynamic NAT with three interfaces

Static inside NAT

Provides a permanent, one-to-one mapping between an IP address on a more secure interface and an IP address on a less secure interface. This allows hosts to access the inside host from the public Internet without exposing the actual IP address.

Dynamic outside NAT

Translates host addresses on less secure interfaces to a range or pool of IP addresses on a more secure interface. This is most useful for controlling the addresses that appear on inside interfaces of the PIX Firewall and for connecting private networks with overlapping addresses.

Static outside NAT

Provides a permanent, one-to-one mapping between an IP address on a less secure interface and an IP address on a more secure interface.

4.4. Attack guards

Attack guards are special feature of PIX Firewall. Guards can be put in place by the PIX Firewall in order to protect against various types of attacks. The PIX Firewall is able to provide attack guards for attacks from the following sources:

- e-mail
- DNS-based attacks
- fragmentation attacks
- access attacks
- SYN floods

4.4.1. Mail guard

Mail Guard provides a safe conduit for Simple Mail Transfer Protocol (SMTP) connections from the outside to an inside e-mail server. Mail Guard enables administrators to deploy a mail server within the internal network, without it being exposed to known security problems that exist within some mail server implementations.

Mail Guard works by restricting the SMTP commands that are allowed through the PIX. Only the SMTP commands specified in RFC 821 section 4.5.1 are allowed to a mail server. These commands are the most basic and secure. Other commands that could compromise the security of the server or the internal network are restricted. The following are the commands allowed for a mail server:

- HELO
- MAIL
- RCPT
- DATA
- RSET
- NOOP
- QUIT

By default, the Cisco Secure PIX Firewall inspects port 25 connections for SMTP traffic. If there are SMTP servers using ports other than port 25, these servers must use the **`fixup protocol smtp`** command so that the PIX Firewall will inspect these other ports for SMTP traffic.

4.4.2. DNS guard

In an attempt to resolve a name to an IP address, a host may query the same DNS server multiple times. The DNS Guard feature of the PIX Firewall recognizes an outbound DNS query and allows only the first answer from the server back through the PIX Firewall. All other replies from the same source are discarded. DNS Guard closes the UDP conduit that was opened by the DNS request after the first DNS reply and does not wait for the normal UDP timeout, which is 2 minutes by default. By closing out the UDP conduit as soon as it recognizes a response, the PIX Firewall can prevent attacks such as UDP session hijacking and certain types of Denial of Service (DoS) attacks.

In cases where a host queries several different DNS servers, the connection to each server is handled separately because each request is sent separately. For example, if the DNS resolver sends three identical queries to three different servers, the PIX Firewall creates three different connections. As the PIX Firewall receives a reply through each connection, it shuts down that single connection. It does not tear down all three connections because of the first reply. The DNS responses of all three servers queried are allowed through the Pix Firewall.

4.4.3. FragGuard and virtual reassembly

In an IP network, a fragment is a packet that has been broken down into smaller pieces in order to be accommodated on a network. There are many legitimate reasons why fragmentation of IP packets occurs. However, many hackers use IP packet fragments to propagate DoS attacks, such as the Teardrop.c attack. FragGuard and Virtual Reassembly help combat this problem, by providing the PIXFirewall a way to track fragment anomalies and reduce the strain these attacks place on the buffer.

FragGuard and Virtual Reassembly is a PIX Firewall feature that provides IP fragment protection. Virtual Reassembly is the process of gathering a set of IP fragments, verifying integrity and completeness, tagging each fragment in the set with the transport header, and not coalescing the fragments into a full IP packet. Virtual Reassembly, which is enabled by default, provides the benefits of full reassembly by verifying each fragment set for integrity and tagging it with the transport header. However, Virtual Reassembly does not require the buffer space that must be reserved for full packet reassembly. This is because full reassembly of packets requires that buffer space be reserved for collecting and coalescing the fragments. Virtual reassembly does not coalesce the fragments, and no pre-allocation of the buffer is needed.

FragGuard and Virtual Reassembly perform full reassembly of all Internet Control Message Protocol (ICMP) error messages and virtual reassembly of the remaining IP fragments that are routed through the PIX Firewall. It uses Syslog to log any fragment overlapping and small fragment offset anomalies, especially those caused by a Teardrop.c attack.

The fragment command provides management of packet fragmentation and improves PIX Firewall compatibility with the Network File System (NFS). NFS is a client/server application that enables computer users to view and optionally store and update files on a remote computer as though they were on their own computer. In general, the default values of the fragment command should be used. However, if a large percentage of the network traffic through the PIX Firewall is NFS, additional tuning may be necessary to avoid database overflow.

FragGuard and Virtual Reassembly are enabled by default on the PIX. There are also a number of options that can be configured using various fragment commands.

4.4.4. AAA flood guard

DoS attacks are based on the premise of utilizing the resources of a device so extensively that other legitimate traffic is crowded out. For example, when Authentication, Authorization, and Accounting (AAA) is being used in a network for authentication, a common DoS attack is to send many forged authentication requests to the PIX thus overwhelming AAA resources.

The **`floodguard`** command is designed to help the PIX Firewall combat this problem. This command enables the PIX to reclaim resources when the user authentication (uauth) subsystem runs out of resources. If an inbound or outbound uauth connection is being attacked or overused, the PIX Firewall actively reclaims TCP resources. When the resources are depleted, the PIX Firewall sends list messages about it being out of resources or out of TCP users. If the PIX Firewall uauth subsystem is depleted, TCP user resources are reclaimed depending on urgency in the following order:

- Timewait
- FinWait
- Embryonic
- Idle

4.4.5. SYN flood attack

SYN flood attacks, also known as TCP flood or half-open connections attacks are common DoS attacks perpetrated against IP servers. These attacks start with the attacker spoofing a nonexistent source IP address or IP addresses on the network of the target host. This floods the target host with SYN packets pretending to come from the spoofed host. Because SYN packets to a host are the first step in the three-way handshake of a TCP-type connection, the target responds to these spoofed hosts, as it would to any legitimate host, with SYN-ACK packets. However, because these SYN-ACK packets are sent to hosts that do not exist, the target sits and waits for the corresponding ACK packets that never show up. This causes the target to overflow its port buffer with embryonic or half-open connections and stop responding to legitimate requests.

Once the embryonic limit is set on a static translation, the PIX will keep track of all embryonic sessions on that translation (xlate) until the PIX tears them down or times them out.

For NAT, the embryonic connection limit on the PIX is set to a default value of zero, which means unlimited embryonic connections are allowed. The maximum depends on the connection license and the minimum is 1. A rule of thumb for the limit is the maximum number of connections to the connection license minus 30%; for example,

on a 100,000-session license, set it to at least 70,000. Set it lower for slower systems, higher for faster systems. Below are connection license values for the PIX Security Appliances running OS 6.3.

Maximum connections per model:

- **501** – 20.000
- **506** – 50.000
- **515E** – 175.000
- **525** – 650.000
- **535** – 1.000.000
- **FWSM** – 1.000.000

For static translations, set the embryonic limit to just under what the end host can handle for embryonic connections. It is suggested to set it to 120 because most servers can handle only 128 embryonic connections at a time.

The PIX Firewall uses the **`static`** and **`nat`** commands to protect hosts from SYN flood attacks. The **`static`** command is used to protect internal hosts against DoS attacks and the **`nat`** command is used to protect external hosts against these attacks. Both commands work by limiting the number of embryonic connections that are allowed to the server. The **`em_limit`** argument then limits the number of embryonic or half-open connections that the server or servers being protected can handle.

Example of configuration:

```
PIXFirewall#
PIXFirewall(config)#static (dmz,outside) 192.168.1.10 172.16.1.2 10000 1000
PIXFirewall(config)#nat (inside) 1 0.0.0.0 0.0.0.0 10000 1000
```

4.4.6. TCP intercept

In PIX Firewall software versions 5.2 and higher, the SYN Flood Guard feature of the static command offers an improved mechanism for protecting systems that can be reached via a static and TCP conduit from TCP SYN attacks. Previously, if an embryonic connection limit was configured in a static command statement, the PIX Firewall simply dropped any connection attempts once the embryonic threshold was reached. This meant that even a modest attack could stop Web traffic for an organization.

For **`static`** command statements without an embryonic connection limit, the PIX Firewall passes all traffic. If the target of an attack has no TCP SYN attack protection or insufficient protection, as with most operating systems, its embryonic connection table overloads and all traffic stops.

With the new TCP intercept feature in versions 5.2 and higher, once the optional embryonic connection limit is reached, and until the embryonic connection count falls below this threshold, every SYN bound for the affected server is intercepted. For each SYN, the PIX Firewall responds on behalf of the server with an empty SYN-ACK segment. The PIX Firewall retains pertinent state information, drops the packet, and waits for acknowledgement from the client. If the ACK is received, a copy of the SYN segment from the client is sent to the server and the TCP three-way handshake is performed between the PIX Firewall and the server. The three-way handshake must be completed for the connection to resume as normal.

4.5. Configurations

In this chapter are security (firewalls, access-lists etc.) configurations which I designed for Contactel's customers. Following configurations are based on Cisco IOS firewall features and PIX Firewalls.

4.5.1. PIX Firewall – deny http, permit IP

This customer has policies that do not allow their employees to access the Internet. Customer has implemented PIX Firewall for VPN connection, so I used following configuration:

```
nameif ethernet0 outside sec0
nameif ethernet0 inside sec100
access-list acl_out deny tcp any any eq www
access-list acl_out permit ip any any
access-group acl_out in interface inside
nat (inside) 1 10.0.0.0 255.255.255.0
global (outside) 1 192.168.0.20-192.168.0.254 netmask 255.255.255.0
```

4.5.2. PIX Firewall – permitting web access to DMZ

Customer has own web server with e-commerce application. Hosts on the outside network must be able to access this web server but rest of the internal network must be maximally secure:

```
nameif ethernet0 outside sec0
nameif ethernet0 inside sec100
nameif ethernet0 dmz sec50
ip address outside 192.168.0.2 255.255.255.0
ip address inside 10.0.0.1 255.255.255.0
ip address dmz 172.16.0.1 255.255.255.0
static (dmz,outside) 192.168.0.11 172.16.0.2
access-list acl_out_dmz permit tcp any host 192.168.0.11 eq www
access-list acl_out_dmz deny ip any any
access-group acl_out_dmz in interface outside
```

4.5.3. PIX Firewall - URL filtering

Customer needs web traffic filtering from security and work efficiency reasons. It is often desirable for a company that want monitor and control which sites network users are allowed to access. The PIX Firewall does not support selective URL filtering on its own. Instead, it relies on specialized URL filtering applications that are configured on separate servers. The PIX Firewall is capable of working with three different URL filtering applications. They are Websense, N2H2, and SmartFilter. Configuration is easy:

```
PIXFirewall(config)# url-server (dmz) host 172.16.0.3 timeout 10 protocol
TCP version 4
PIXFirewall(config)#filter url http 0 0 0 0 allow
```

4.5.4. Cisco router – IPSec VPN + IOS-based protocol inspection

Customer has the headquarter in Prague. This headquarter is connected using leased line. Next – customer have ten branches that are connected using ADSL. Computers in branches use Cisco VPN Client software which is responsible for secure connection between given branch and headquarter. Next used IOS feature is protocol inspection. Configuration on headquarter router is following:

IPSec VPN

```
crypto isakmp policy 1
 hash md5
 authentication pre-share
 lifetime 3600
```

```
crypto isakmp key qwerty123456 address 194.108.136.37[3]
!
!
crypto ipsec transform-set cm-transformset-1 esp-des esp-sha-hmac
!
crypto map cm-cryptomap local-address Serial0/0.500
crypto map cm-cryptomap 1 ipsec-isakmp
 set peer 194.108.136.37
 set transform-set cm-transformset-1
 set pfs group1
 match address 100
!
interface Tunnel2
 description Tunnel to Hradec Kralove
 ip address 10.10.0.1 255.255.255.252
 ip mtu 1500
 tunnel source 212.65.243.125
 tunnel destination 194.108.136.37
!
access-list 100 remark --=== ADSL branch ===--
access-list 100 permit ip 192.168.1.0 0.0.0.255 192.168.11.0 0.0.0.255
access-list 100 permit ip 192.168.1.0 0.0.0.255 192.168.12.0 0.0.0.255
```

IOS protocol inspection

```
ip inspect name FW1 cuseeme timeout 3600
ip inspect name FW1 ftp timeout 3600
ip inspect name FW1 realaudio timeout 3600
ip inspect name FW1 smtp timeout 3600
ip inspect name FW1 tftp timeout 30
ip inspect name FW1 udp timeout 15
ip inspect name FW1 tcp timeout 3600
ip inspect name FW1 h323 timeout 15
ip inspect name FW1 http timeout 3600
interface FastEthernet0/0
 description LAN
 ip address 192.168.1.1 255.255.255.0
 ip nat inside
 ip inspect FW1 in
```

4.5.5. Cisco IOS – PAT and packet filter

Customer has some remote access applications which require access to server on local network. Customer didn't demand public IP address and all access to local network is realized using port address translation.

```
ip nat translation tcp-timeout 3600
ip nat translation udp-timeout 250
ip nat translation port-timeout udp 69 15
ip nat inside source list 1 interface Loopback0 overload
ip nat inside source static udp 192.168.1.149 12000 194.108.235.252 12009
extendable
…
ip nat inside source static udp 192.168.1.149 12000 194.108.235.252 12000
extendable
ip nat inside source static tcp 192.168.1.149 12000 194.108.235.252 12000
extendable
…
ip nat inside source static tcp 192.168.1.149 12000 194.108.235.252 12009
extendable
ip nat inside source static tcp 192.168.1.1 22 194.108.235.252 22
extendable
```

[3] IP address of ADSL router

```
ip nat inside source static tcp 192.168.1.1 23 194.108.235.252 23
extendable
ip nat inside source static tcp 192.168.1.1 32779 194.108.235.252 32779
extendable
ip nat inside source static tcp 192.168.1.1 21 194.108.235.252 21
extendable
ip nat inside source static tcp 192.168.1.1 20 194.108.235.252 20
extendable
ip nat inside source static tcp 192.168.1.149 1433 194.108.235.252 1433
extendable
ip nat inside source static tcp 192.168.1.149 5900 194.108.235.252 5900
extendable
ip nat inside source static tcp 192.168.1.149 5800 194.108.235.252 5800
extendable
ip nat inside source static tcp 192.168.1.149 80 194.108.235.252 80
extendable
ip nat inside source static tcp 192.168.1.99 80 194.108.235.252 8080
extendable
ip nat inside source static tcp 192.168.1.5 3389 194.108.235.252 3389
extendable
ip nat inside source static tcp 192.168.1.5 3390 194.108.235.252 3390
extendable
ip nat inside source static 192.168.1.5 212.65.213.145
ip nat inside source static 192.168.1.6 212.65.213.146
ip nat inside source static tcp 192.168.1.149 2107 194.108.235.252 2107
extendable
ip nat inside source static tcp 192.168.1.149 2106 194.108.235.252 2106
extendable
!
ip access-list extended From_internet
 remark ========== Access from everywhere =================
 permit ip any host 192.168.1.5
 permit ip any host 192.168.1.6
 permit ip host 160.218.176.232 any
 permit tcp any 192.168.1.0 0.0.0.255 gt 1023 established
 permit udp any 192.168.1.0 0.0.0.255 gt 1023
 permit icmp any any echo-reply
 permit icmp any any unreachable
 permit ip 83.208.3.0 0.0.0.255 host 192.168.1.25
 remark ========= VPN access ==================
 permit ip 192.168.0.0 0.0.31.255 192.168.1.0 0.0.0.255
 remark =========== Access to 192.168.1.1 =============
 permit ip 193.165.208.152 0.0.0.7 host 192.168.1.1
 permit ip 193.165.208.136 0.0.0.7 host 192.168.1.1
 permit ip host 62.24.69.212 host 192.168.1.1
 permit ip 83.208.3.0 0.0.0.255 host 192.168.1.1
 permit tcp host 83.208.201.217 host 192.168.1.1 eq ftp-data
 permit tcp host 83.208.201.217 host 192.168.1.1 eq ftp
 permit tcp host 83.208.201.217 host 192.168.1.1 eq 22
 permit tcp host 83.208.201.217 host 192.168.1.1 eq telnet
 remark ============== Access to IP 192.168.1.149 and given port ==========
 permit tcp any host 192.168.1.149 eq 2106
 permit tcp any host 192.168.1.149 eq 2107
 permit tcp host 62.245.102.236 host 192.168.1.149 eq www
 permit tcp host 62.245.102.236 host 192.168.1.149 eq 1433
 permit tcp host 62.245.102.236 host 192.168.1.149 eq 5800
 permit tcp host 62.245.102.236 host 192.168.1.149 eq 5900
 permit tcp host 62.245.102.236 host 192.168.1.149 range 12000 12009
 deny   ip any any
!
```

```
interface Ethernet0
 description LAN
 ip address 192.168.1.254 255.255.255.0
 ip access-group From_internet out
 ip nat inside
 service-policy input LM-MARK-DATA
 no ip route-cache cef
 no ip route-cache
 no ip mroute-cache
 full-duplex
 no cdp enable
```

5. Encryption and VPN

5.1. VPN overview

A Virtual Private Network (VPN) provides the same network connectivity for remote users over a public infrastructure, as they would have over a private network. VPN services for network connectivity include authentication, data integrity, and confidentiality. There are two basic VPN types:

- **LAN-to-LAN VPNs** – there are two common types of LAN-to-LAN VPNs, also known as site-to-site VPNs:
  - **intranet VPNs** connect corporate headquarters, remote offices, and branch offices over a public infrastructure.
  - **extranet VPNs** link customers, suppliers, partners, or communities of interest to a corporate Intranet over a public infrastructure.
- **Remote access VPNs** – which securely connect remote users, such as mobile users and telecommuters, to the enterprise.

5.1.1. VPN technology options

Figure shows the methods of protection implemented on different layers. With implementation of encryption on one layer, this layer and all layers above it are automatically protected. Network layer protection offers one of the most flexible solutions. It is media independent as well as application independent.

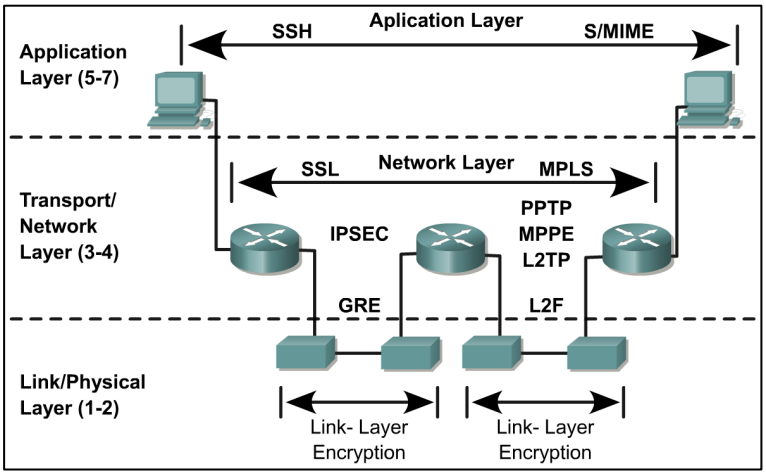

Figure 12: VPN technology options

Providing privacy and other cryptographic services at the application layer was also very popular in the past. In some situations, it is still heavily used today. However, application layer security is application specific and protection methods need be reimplemented in every application.

Some standardization has been successful at Layer 4 of the OSI model with protocols such as Secure Socket Layer (SSL) providing privacy, authenticity, and integrity to TCP-based applications. SSL is used heavily in modern e-commerce sites. However, SSL fails to address the issues of flexibility, ease of implementation, and application independence. One of the latest technologies available, Transport Layer Security (TLS), will address many of the limitations of SSL.

Protection at lower levels of the OSI stack, especially the Data Link layer, was also used in communication systems of the past. This provided protocol independent protection on specific untrusted links. However, Data Link layer protection is expensive to deploy on a large scale because there is a need to protect every single link separately.

Data Link layer protection allows for man-in-the-middle attacks on intermediate stations, or routers, and is usually proprietary.

Because of the above limitations, Layer 3 has become the most popular level to apply cryptographic protection to network traffic.

5.1.2. Tunnel interfaces

Tunnel interfaces provide a point-to-point connection between two routers via a virtual software interface. They also appear as one direct link between routers that are connected via a large IP network, such as the Internet. However, tunnel interfaces should not to be confused with IPSec or L2TP tunnels, which can act as tunnels but not as true Cisco IOS interfaces.

Further tunnel interface configuration information that may prove important is as follows:

- Unnumbered Layer 3 addresses are supported but not allowed for by IPSec.
- Access-lists can be applied to the tunnel interface.
- QoS supports traffic requiring consistent service such as voice over IP.
- Committed Access Rate (CAR), Weighted Fair-Queue (WFQ), and Weighted Random Early Detection (WRED) are not supported on tunnel interfaces at this time.

5.2. IPSec

IPSec is a framework of security protocols and algorithms used to secure data at the network layer . Prior to the IPSec standard, Cisco implemented its proprietary Cisco Encryption Technology (CET) to provide protection at the packet level. RFC 2401 describes the general framework for this architecture. Like all security mechanisms, RFC 2401 helps to enforce a security policy. The policy defines the need for security on various connections, which will be IP sessions. The framework provides data integrity, authentication, and confidentiality, as well as security association and key management.

IPSec consists of two protocols. The first protocol is Encapsulating Security Payload (ESP). It encapsulates the data, but does not provide protection to the outer headers. ESP encrypts the payload for data confidentiality. The second protocol is Authentication Header (AH). The AH protocol provides protection to the entire datagram by embedding the header in the data. The AH verifies the integrity of the IP datagram. AH and ESP use symmetric secret key algorithms, although public key algorithms are feasible.

This feature is supported across the Cisco IOS-based 1600, 2x00, 36x0, 4x00, 5x00, 7x00 platforms using IOS release 12.0(x) or higher, PIX Firewalls, and VPN Client and Concentrators. The multiple feature sets allow the multiple deployment scenarios.

5.2.1. Site-to-Site IPSec VPN Using Pre-shared Keys

There are following steps when configuring site-to-site IPSec VPN:

Step 1

Determine IKE phase one policy. Determine the IKE policies between IPSec peers based on the number and location of the peers. Some planning steps include the following:

- Determine the key distribution method.
- Determine the authentication method.
- Identify IPSec peer IP addresses and host names.
- Determine ISAKMP policies for peers.

Configuring IKE consists of four steps:

- Enable or disable IKE with the **`crypto isakmp enable`** command.
- Create IKE policies with the **`crypto isakmp policy`** commands.
- Configure pre-shared keys with the **`crypto isakmp key`** and **`associated`** commands.
- Verify the IKE configuration with the **`show crypto isakmp policy`** command.

Step 2

Determine IKE phase two policy. Identify IPSec peer details such as IP addresses, IPSec transform sets, and IPSec modes. Crypto maps will be used to gather all IPSec policy details together during the configuration phase. In this step is following configured:

- Configure transform set suites with the **`crypto IPSec transform-set`** command.
- Configure global IPSec security association lifetimes with the **`crypto IPSec security-association lifetime`** command.
- Configure crypto ACLs with the **`access-list`** command.
- Configure crypto maps with the **`crypto map`** command.
- Apply the crypto maps to the terminating/originating interface with the **`interface`** and **`crypto map`** commands.

Step 3

Check the current configuration. Use the **`show running-configuration`**, **`show isakmp [policy]`**, and **`show crypto map`** commands. Other show commands can be used to check the current configuration of the router.

Step 4

Ensure that the network works without encryption. This step should not be avoided. Ensure that basic connectivity has been achieved between IPSec peers using the desired IP services before configuring IPSec. Use the **`ping`** command to check basic connectivity. Cisco IOS software contains a number of **`show`**, **`clear`**, and **`debug`** commands useful for testing and verifying IPSec and ISAKMP. Following commands are useful for testing configuration of IPSec:

- To display configured IKE policies, use the **`show crypto isakmp`** policy command.
- To display configured transform sets, use the **`show crypto IPSec transform set`** command.
- To display the current state of IPSec SAs, use the **`show crypto IPSec sa`** command.
- To view configured crypto maps, use the **`show crypto map`** command.
- To debug IKE and IPSec traffic through the Cisco IOS, use the **`debug crypto IPSec`** and **`debug crypto isakmp`** commands.

Step 5

Ensure that the ACLs on perimeter devices are compatible with IPSec. Ensure that perimeter routers and the IPSec peer router interfaces permit IPSec traffic. Use the **`show access-lists`** command for this step.

Example of site-to-site VPN with pre-shared keys

Assume to have topology on following figure:

Figure 13: Example topology for site-to-site VPN

Configuration of both routers is following:

```
RouterA#show running-config
crypto ipsec transform-set mine esp-des
!
crypto map mymap 10 ipsec-isakmp
set peer 172.30.2.2
set transform-set mine
match address 110
!
interface ethernet0/1
ip address 172.30.1.2 255.255.255.0
no ip directed-broadcast
crypto map mymap
!
access-list 110 permit tcp 10.0.1.0 0.0.0.255 10.0.2.0 0.0.0.255


RouterB#show running-config
crypto ipsec transform-set mine esp-des
!
crypto map mymap 10 ipsec-isakmp
set peer 172.30.1.2
set transform-set mine
match address 101
!
interface ethernet0/1
ip address 172.30.2.2 255.255.255.0
no ip directed-broadcast
crypto map mymap
!
access-list 101 permit tcp 10.0.2.0 0.0.0.255 10.0.1.0 0.0.0.255
```

5.2.2. Site-to-Site IPSec VPN using Digital Certificates

The configuration process for RSA signatures consists of five major tasks. The following tasks and steps are identical to pre-shared keys:

Task 1 – prepare for IKE and IPSec

Preparing for IPSec involves determining the following detailed encryption policy:

- Identifying the hosts and networks to be protected
- Determining IPSec peer details
- Determining the IPSec features that are needed
- Ensuring existing access lists are compatible with IPSec.

Task 2 – configure CA Support

Configuring CA Support involves setting the router hostname and domain name, generating generate the keys, declaring a CA, and authenticating and requesting network-own certificates. This task assume following steps:

- **Step 1 – manage the non-volatile RAM (NVRAM) memory usage.** This step is optional. In some cases, storing certificates and Certificate Revocation Lists (CRLs) locally does not present a problem.

However, in other cases, memory might become an issue. This is especially true if the CA supports an RA and a large number of CRLs are stored on the router.

- **Step 2 – set the router time and date.** The router must have an accurate time and date to enroll with a CA server. Commands **`clock timezone`** and **`clock set`**.
- **Step 3 – configure the router hostname and domain name**. The hostname is used in prompts and default configuration filenames. The domain name is used to define a default domain name that the Cisco IOS software uses to complete unqualified hostnames. Commands **`hostname <name>`** and **`ip domain-name <name>`**.
- Step 4 – generate an RSA key pair. RSA keys are used to identify the remote VPN peer. Administrators can generate one general-purpose key or two special purpose keys. Command **`crypto key generate rsa usage keys`**.
- **Step 5 – declare a CA.** To declare the CA, use the **`crypto ca trustpoint`** global configuration command. Use the no form of this command to delete all identity information and certificates associated with the CA. Command **`crypto ca trustpoint <name>`**.
- **Step 6 – authenticate the CA.** The router needs to authenticate the CA. It does this by obtaining the CA's self-signed certificate that contains the CA public key. Command **`crypto ca authenticate <name>`**.
- **Step 7 – request a certificate.** Complete this step to obtain a router identity certificate from the CA. Command **`crypto ca enroll name`**.
- **Step 8 – save the configuration.** After configuring the router for CA support, the configuration should be saved. Commands **`write memory`** or **`copy running-config startup-config`**.
- **Step 9 – monitor and maintain CA interoperability.** Optional. Command **`crypto ca trustpoint <name>`**. The following steps are optional, depending on network requirements:
- Request a CRL.
- Delete the router RSA keys.
- Delete both public and private certificates from the configuration.
- Delete the peer public keys.
- Step 10 – verify the CA support configuration. The commands allow and verify any configured CA certificates. Commands **`show crypto ca certificates`** and **`show crypto key mypubkey | pubkey-chain`**.

Task 3 – configure IKE for IPSec

Configuring IKE involves enabling IKE, creating the IKE policies, and validating the configuration.

Task 4 – configure IPSec

IPSec configuration includes defining the transform sets, creating crypto access lists, creating crypto map entries, and applying crypto map sets to interfaces.

Task 5 – test and verify IPSec

Use **`show`**, **`debug`**, and related commands to test and verify that IPSec encryption works, and to troubleshoot problems. For troubleshooting are useful following commands:

- Display the configured IKE policies using the **`show crypto isakmp policy`** command.
- Display the configured transform sets using the **`show crypto IPSec transform set`** command.

- Display the current state of the IPSec SAs with the **`show crypto IPSec sa`** command.
- View the configured crypto maps with the **`show crypto map`** command.
- Debug IKE and IPSec traffic through the Cisco IOS with the **`debug crypto IPSec`** and **`debug crypto isakmp`** commands.
- Debug CA events through the Cisco IOS using the **`debug crypto key-exchange`** and **`debug crypto pki commands`**.

5.2.3. Remote access VPN

A Remote Access VPN secures connections for remote users, such as mobile users or telecommuters, to corporate LANs over shared service provider networks. There two types of Remote Access VPNs:

- **Client-Initiated** – remote users use a VPN client or web browser to establish a secure tunnel across a public network to the enterprise.
- **NAS-Initiated** – remote users dial in to an ISP Network Access Server (NAS). The NAS establishes a secure tunnel to the enterprise private network that might support multiple remote user-initiated sessions.

Remote access is targeted to mobile users and home telecommuters. In the past, corporations supported remote users via dial-in networks. This typically necessitated a call to access the corporation. With the advent of VPN a mobile user can connect to any ISP via dial, cable, or DSL, and connect to the Internet to access the corporation.

For remote access VPN Cisco offers complete solution called Cisco Easy VPN. The Easy VPN consist of two main components:

- **Cisco Easy VPN Server** – the Cisco Easy VPN Server enables Cisco IOS routers, PIX Firewalls, and Cisco VPN 3000 Concentrators to act as VPN devices in site-to-site or remote access VPNs where the remote office devices are using the Cisco Easy VPN Remote feature. Using this feature, security policies defined at the central office are pushed to the remote VPN device ensuring that those connections have up-to-date policies in place before the connection is established. In addition, an Easy VPN Server-enabled device can terminate VPN tunnels initiated by mobile remote workers running VPN Client software on PCs. This flexibility makes it possible for mobile and remote workers to access their headquarters where critical data and applications exist.
- **Cisco Easy VPN Remote** – the Cisco Easy VPN Remote feature enables Cisco IOS routers, PIX Firewalls, Cisco VPN 3002 hardware, or software clients to act as remote VPN Clients. As clients, these devices can receive security policies from an Easy VPN Server, minimizing VPN configuration requirements at the remote location.

How Easy VPN works

When a Cisco VPN Client initiates a connection with a Cisco Easy VPN Server gateway, the conversation that occurs between the peers generally consists of following major steps:

- Device authentication via Internet Key Exchange (IKE).
- User authentication using IKE Extended Authentication (Xauth).
- VPN policy push using Mode Configuration.
- IPSec security association (SA) creation.
- Easy VPN Server accepts the SA proposal.
- Easy VPN Server initiates a username/password challenge.
- The mode configuration process is initiated.

- The Reverse Route Injection (RRI) process is initiated.
- IKE quick mode completes the connection.

5.2.4. Intrusion detection

Intrusion detection is the ability to detect attacks against a network. The network can be made up of network devices such as routers, printers, firewalls, and servers. Intrusion protection should provide the following active defense mechanisms:

- **Detection** – identifies malicious attacks on network and host resources.
- **Prevention** – stops the detected attack from executing.
- **Reaction** – immunizes the system from future attacks from a malicious source.

Host-based intrusion detection system (HIDS)

A host-based intrusion detection system (HIDS) audits host log files and host file systems and resources. An advantage of HIDS is that it can monitor operating system processes and protect critical system resources, including files that may exist only on that specific host. This means it can notify network managers when some external process tries to modify a system file in a way that may include a hidden back door program.

A simple form of host-based intrusion detection is enabling system logging on the host. This is called passive detection. However, it can require intensive manpower to recover and analyze these logs. Current host-based intrusion detection software requires agent software to be installed on each host to monitor activity performed on and against the host. The Agent software performs the intrusion detection analysis and protects the host.

HIDS can support both passive and active detection. Active detection can be set to shut down the network connection or to stop the impacted services. This has the benefit of being able to quickly analyze an event and take corrective action. Cisco provides HIDS using the Entercept and Okena products. Some other vendors of HIDS include Symantec, Internet Security Systems (ISS), and Enterasys.

Network-based intrusion detection system (NIDS)

A network-based intrusion detection system (NIDS) involves the deployment of probing devices, or sensors throughout the network, which capture and analyze the traffic as it traverses the network. The Sensors detect malicious and unauthorized activity in real time, and can take action when required. Sensors can be deployed at designated network points that enable security managers to monitor network activity while it is occurring, regardless of the location of the target of the attack. NIDS Sensors are typically tuned for intrusion detection analysis. The underlying operating system is stripped of unnecessary network services and essential services are secured.

Just like host-based IDS, a network intrusion detection system can be based on active or passive detection. Sensors are deployed at network entry points that protect critical network segments. The network segments have both internal and external corporate resources. The Sensors report to a central Director server located inside the corporate firewall.

6. Case study: Contactel

Backbone of Contactel's network has physical topology of extended star. Centre of this network is located in Prague. Local network in branches is designed as star or extended star.

Goal of this case study is to create new design of Contactel's network to provide secure communication between branches and enable IP telephony in branches. Current connection of branches is unsecured and every branch has another type of last mile, CPE and configuration. After applying this case study to network topology should all branches have the same hardware, software version and configuration. Last mile to the branches will be ready to care VoIP and all traffic between branches and headquarter will be encrypted. From these reasons all network equipment (routers, switches etc.) used in this network will be from Cisco.

Contactel has branches in Plzeň, Liberec, Brno, Ostrava and Olomouc. Small branches (sales representatives) are in Písek, Karlovy Vary and Ústí nad Labem. There are some differences between connection branches and sales representatives – these differences are described later.

6.1. Branches connection

Branches in Plzeň, Brno, Ostrava and Olomouc are located in a building where is POP of Contactel's backbone network. In these cases is router of local area network connected using ethernet cable directly to backbone router. Branch in Liberec is located outside backbone POP – this branch is connected via 2 Mbps wireless line from Coprosys company.

Network services and IP telephony is provided by following Cisco routers:

- **Cisco 1721** – for data communication. This router will have two FastEthernet ports, VPN module and operating system with IPSec support.
- **Cisco 1751V** – for voice communication. This router will have two FastEthernet ports a two analog interfaces (FXS) for connecting fax to PSTN. Operating system will have VoIP support.

Branch in Liberec has different topology – last mile is realized by Coprosys company and their radio equipment (MicroLAN – working in public band 10 GHz). This wireless circuit is terminated in serial interface of router Cisco1751V. All routers have maximum available RAM memory (96 MB) and flash memory (32 MB). Hardware configuration, software version and some next detail are in following table.

Router	Memory RAM/flash	Version of operating system IOS	Expanded cards
Branches Plzeň, Brno, Ostrava, Olomouc			
Cisco 1721	96 MB/32 MB	c1700-k9o3sy7-mz.123-13.bin	WIC-1ENET
Cisco 1751V	96 MB/32 MB	c1700-ipvoice-mz.123-13.bin	WIC-1ENET VIC2-2FXS
Branch Liberec			
Cisco 1721	96 MB/32 MB	c1700-k9o3sy7-mz.123-13.bin	WIC-1ENET
Cisco 1751V	96 MB/32 MB	c1700-ipvoice-mz.123-13.bin	WIC-1T WIC-1ENET VIC2-2FXS

Table 8: Configuration of routers

6.1.1. Connecting routers

As I noticed early connectivity to branches routers is realized using Ethernet cable or using wireless serial line (in Liberec). For physical topology see following figure.

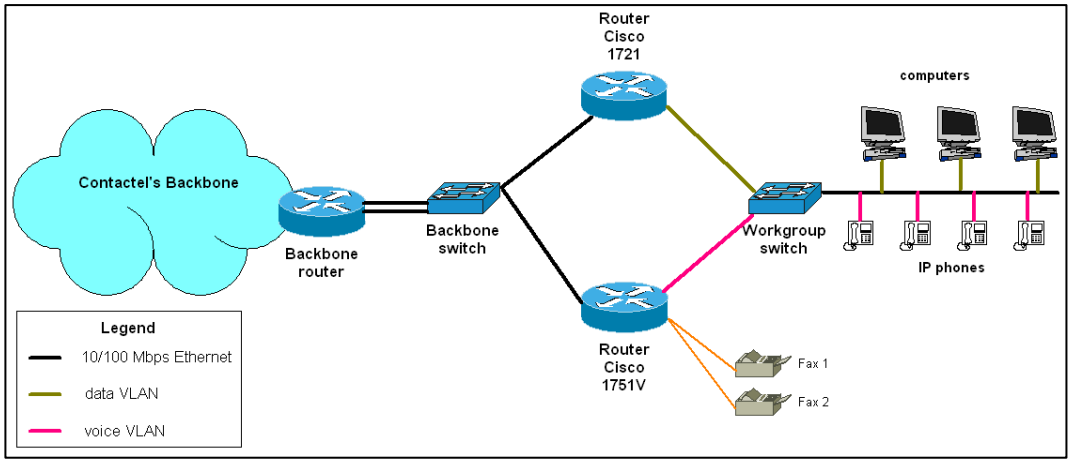

Figure 14: Physical topology in Plzeň, Brno, Ostrava and Olomouc

There are two Ethernet cables (100 Mbps) between backbone switch and workgroup routers. In workgroup router are these lines terminated in built-in interface FastEthernet0. In the expanding slot 1 of workgroup routers are installed cards WIC 1ENET which make a link to the workgroup switch.

Workgroup switches are made by Cisco too and we will use models Catalyst 2950-12 or Catalyst 2950-24 (difference between these switches is in number of FastEthernet ports). Two ports on workgroup switch will work in trunking mode – will form trunk between workgroup switch and workgroup router. There are two VLANs – one for voice traffic and on for data traffic – in effect we'll have two independent local area networks. Ports on workgroup switch are divided in following three groups:

- **port 1** – trunk to workgroup router Cisco 1721
- **port 2** – trunk to workgroup router Cisco 1751V
- switch Catalyst 2950-12:
  - ports 3-7 – VLAN 10 for workstations
  - ports 8-12 – VLAN 11 for IP phones
- switch Catalyst 2950-24:
  - ports 3-13 – VLAN 10 for workstations
  - ports 14-24 – VLAN 11 for IP phones

Topology in Liberec is a bit different from other branches. Branch in Liberec is connected using wireless circuit from Coprosys company. This line has bandwidth of 2.048 kbps and use wireless technology MicroLAN. This technology works in public radio band 10 Ghz. Line is terminated by interface X.21 and this interface is connected to the workgroup router. Router Cisco 1751V in Liberec has expanded card WIC1-T in slot 0.

This router acts as a perimeter router for Liberec branch. This router has two analog ports for faxes and two ethernet port. Ethernet ports have following job:

- **Built-in interface FastEthernet0** – configured with VLAN 11 for IP phones and is connected directly to the workgroup switch Catalyst 2950-12/2950-24.
- **Interface Ethernet1/0** – expanding card installed in slot 1 – form point-to-point line between router Cisco 1751V and router Cisco1721.

For Liberec topology see figure on next page.

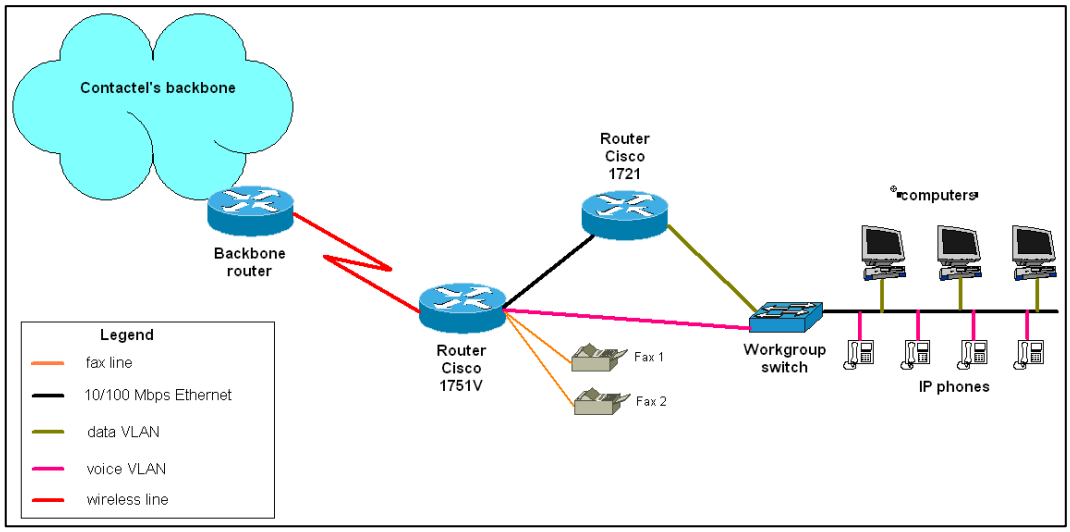

Figure 15: Topology of Liberec network

On routers Cisco 1721 are terminated tunnels for encrypted communication between branches and headquarter. Routers Cisco 1751V and Cisco 1721 are connected using straight-through UTP cable. Ethernet ports on Cisco 1721 have following purpose:

- **Built-in interface FastEthernet0** – connected to the workgroup switch Catalyst 2950-12/2950-24 and configured with VLAN 10 (VLAN for workstations).
- **Interface Ethernet1/0** – installed in expanding slot 1 – point-to-point line between router Cisco 1751V and Cisco1721.

Cables in all branches are realized using common UTP Cat. 5 cables. All branches are located in rented offices where cables are installed already. It's possible to use power over Ethernet feature for IP phones. Unfortunately switches Catalyst 2950-12/2950-24 don't support this feature – all phones are delivered with appropriate power supply.

All network equipment in branches is placed in lockable rack. In this rack is placed uninterrupted power supply for active network equipment. There is placed Coprosys's radio unit in rack in Liberec – this radio unit is connected to antena on the roof using optical cable.

Local area network in Prague is larger and more complicated comparing with other branches. Headquarters is located in eight-floor building. „The heart" of whole network (MDF – Main Distribution Facility) is located on the fifth floor. There are small network centres (IDF – Intermediate Distribution Facility) on every floor where horizontal cabling end. Horizontal cabling uses common UTP Cat. 5 cables. Connection between MDF a IDF is realised using optical cables. Simple topology of local area network in Prague's headquarters is on following figure.

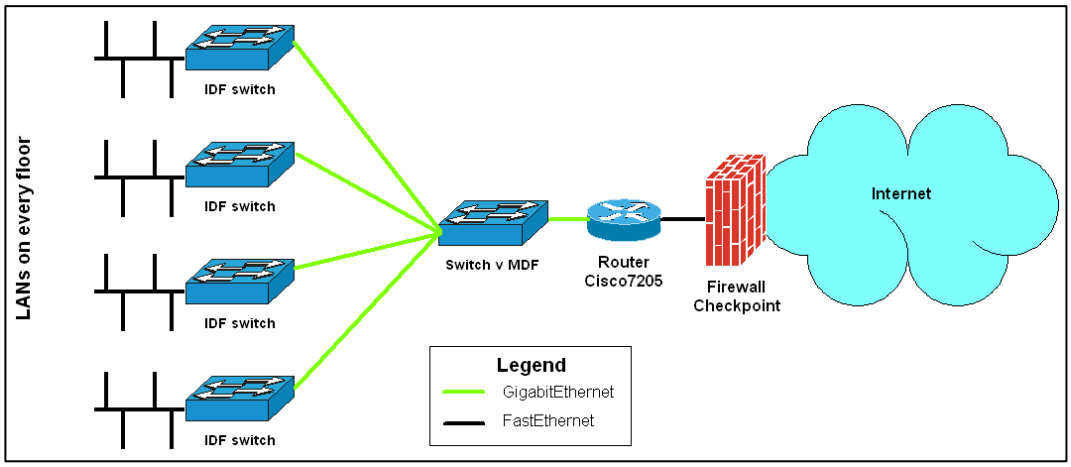

Figure 16: LAN topology in Prague.

There are two switches Catalyst 2950-48 placed in every IDF. These switches are used for connecting workstations on the floor to the LAN. These workgroup switches are connected to the enterprise switch placed in MDF. Enterprise switch is model Catalyst 3550-48. This switch will be replaced before branches connection realization. New switch will be Catalyst 4560 and this switch has direct connection to the enterprise router Cisco 7205. Enterprise router Cisco 7205 has following job:

- Communication between VLAN in headquarter.
- Routing private networks inside Contactel's intranet.
- Basic traffic filtering (securing intranet server etc.).
- Tunnels from branches are terminated at this router.
- Perform network address translation.
- DHCP server.

Intranet together with router Cisco7205 is protected with Checkpoint firewall. This firewall protects intranet before unauthorized access from Internet, perform network address translation and defines demilitarize zone. In DMZ are placed various servers – for example email, web server and servers for special purposes.

IP phones aren't used in Contactel's headquarters by all employees. Reason is money – IP phone installation for all employees will cost about three millions CZK. From this reason are IP phones installed in technical department and sales department only.

6.2. IP addressing

Current network uses few inconsistent IP address ranges. It's necessary to consolidate addressing for all Contactel. For addressing workstations and inside network equipment we will use class A private addresses – 10.0.0.0/8. This address range is divided into concrete subnets. Using of these subnets and appropriate VLANs are described in following table:

Network	VLAN	Purpose
10.0.1.0/24	10	workstation addresses – financial department and marketing
10.0.2.0/24	20	workstation addresses – IT department
10.0.3.0/24	30	workstation addresses – customer care department
10.0.4.0/24	40	workstation addresses – technical department
10.0.5.0/24	50	workstation addresses – branch Plzeň
10.0.6.0/24	60	workstation addresses – branch Liberec
10.0.7.0/24	70	workstation addresses – branch Brno
10.0.8.0/24	80	workstation addresses – branch Olomouc
10.0.9.0/24	90	workstation addresses – branch Ostrava
10.0.10.0/24	100	testing addresses – IT department
10.0.11.0/24	110	testing addresses – technical department
10.0.12.0/24	120	testing addresses – NOC
10.50.0.0/24	500	addresses for IP phones – headquarter
10.51.0.0/28	510	addresses for IP phones – branch Plzeň
10.51.0.16/28	511	addresses for IP phones – branch Liberec
10.51.0.32/28	512	addresses for IP phones – branch Brno
10.51.0.48/28	513	addresses for IP phones – branch Olomouc
10.51.0.64/28	514	addresses for IP phones – branch Ostrava
10.52.0.0/29		IP addresses for sales representative in Písku
10.52.0.8/29		IP addresses for sales representative in Karlovy Vary
10.52.0.16/29		IP addresses for sales representative in Ústí nad Labem
10.100.0.0/24	2	management VLAN for network equipment

Table 9: IP addressing

Next address ranges are free for future use. Router Cisco 7205 is administered by specialist from technical department. Some address ranges are not documented and use for testing purpose – every new address range must be consulted with technical department specialist.

IP addresses for workstation are assigned by DHCP server. On every workgroup router Cisco 1721 is DHCP server running. Lease time is set to 48 hours for workstations and 24 hours for IP phones. Addresses in every network are divided into following groups:

IP address	Purpose
10.x.x.0	network address
10.x.x.1	default gateway for given segment
10.x.x.2 – 10.x.x.49	static IP addresses (for example network printers etc.)
10.x.x.50 – 10.x.x.254	dynamic IP addresses
10.x.x.255	broadcast

Table 10: IP address for given network

All IP addresses from ranges for workstations (e.g. obtained from DHCP – range 10.x.x.50 – 10.x.x.254) have permitted access to Internet via network address translation on Checkpoint firewall. On router Cisco 7205 is applied packet filters which allow communication between workstations and servers (for example employees from technical department don't need access to billing servers etc.).

6.3. Sales representatives connection

Next to branches Contactel has sales representatives in Písek, Karlovy Vary and Ústí nad Labem. These small branches don't have direct connection to backbone router in POP. Connection to Contactel's intranet is realized through secure tunnel – these tunnels are terminated in router Cisco 837.

Connection to sales representatives office is realized by ADSL provided by Czech Telecom network. In future will be connection realized by Telenor company. This connection has higher quality parameters (no aggregation, guaranteed bandwith etc.) and it is possible to run IP telephony on this connection. Now (February/March 2005) is Telenor ADSL in testing in Ustí nad Labem.

Router Cisco 837 used in sales representative offices has built-in four-port switch – it is enough pro given purpose (there is only one man in every sales representative office). Router Cisco 837 has following parameters:

- **RAM:** 80 MB
- **flash memory:** 24 MB
- **version of IOS:** c837-k9o3sy6-mz.123-14.T.bin

6.4. Configurations

All configurations on all routers are the same, differences are between IP addresses and SNMP variables. All routers and switches support remote access using telnet. Configurations can be divided into a few groups:

- **Basic configuration which is same for all routers** – AAA configuration, remote access, time and secure access to the router.
- **IPSec configuration** – for data encryption.
- **VoIP configuration** – configuration for IP phones and faxes.

6.4.1. Basic configuration

```
version 12.3
no service pad
service tcp-keepalives-in
service tcp-keepalives-out
service timestamps debug datetime localtime
service timestamps log datetime localtime
service password-encryption
no service finger
no service pad
no service udp-small-servers
no service tcp-small-servers
no service dhcp
no cdp run
no ip bootp server
no ip http server
no ip finger
no ip source-route
no ip gratuitous-arps
no ip identd!
hostname název_routeru
!
boot-start-marker
boot system flash: filename depend on used platform
boot-end-marker
!
!
enable secret 5 $1$JoiG$qJQt#NBW7QN%6Zze343tWGV32R.
!
username manager password 7 0203FRWD104g8d2524fdsr95A7562632D
clock timezone CET 1
clock summer-time CEST recurring last Sun Mar 2:00 last Sun Oct 3:00
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180
aaa new-model
!
!
aaa authentication login default group tacacs+ local
aaa authentication enable default group tacacs+ enable
aaa authorization exec default group tacacs+ local
```

```
aaa authorization commands 0 default group tacacs+ local
aaa authorization commands 1 default group tacacs+ local
aaa authorization commands 2 default group tacacs+ local
aaa authorization commands 3 default group tacacs+ local
aaa authorization commands 4 default group tacacs+ local
aaa authorization commands 5 default group tacacs+ local
aaa authorization commands 6 default group tacacs+ local
aaa authorization commands 7 default group tacacs+ local
aaa authorization commands 8 default group tacacs+ local
aaa authorization commands 9 default group tacacs+ local
aaa authorization commands 10 default group tacacs+ local
aaa authorization commands 11 default group tacacs+ local
aaa authorization commands 12 default group tacacs+ local
aaa authorization commands 13 default group tacacs+ local
aaa authorization commands 14 default group tacacs+ local
aaa authorization commands 15 default group tacacs+ local
aaa accounting commands 0 default start-stop group tacacs+
aaa accounting commands 1 default start-stop group tacacs+
aaa accounting commands 2 default start-stop group tacacs+
aaa accounting commands 3 default start-stop group tacacs+
aaa accounting commands 4 default start-stop group tacacs+
aaa accounting commands 5 default start-stop group tacacs+
aaa accounting commands 6 default start-stop group tacacs+
aaa accounting commands 7 default start-stop group tacacs+
aaa accounting commands 8 default start-stop group tacacs+
aaa accounting commands 9 default start-stop group tacacs+
aaa accounting commands 10 default start-stop group tacacs+
aaa accounting commands 11 default start-stop group tacacs+
aaa accounting commands 12 default start-stop group tacacs+
aaa accounting commands 13 default start-stop group tacacs+
aaa accounting commands 14 default start-stop group tacacs+
aaa accounting commands 15 default start-stop group tacacs+
aaa session-id common
ip subnet-zero
no ip source-route
no ip gratuitous-arps
!
!
ip domain name ctt.cz
ip name-server 212.65.193.157
ip name-server 212.65.242.210
!
no ip bootp server
ip cef
ip audit po max-events 100
no ftp-server write-enable
!
interface Loopback0
 ip address {public IP addess with subnet mask 255.255.255.255}
!
interface Null0
 no ip unreachables
!
```

```
interface Ethernet0
 description Local Area Network
 ip address 10.x.x.1 255.255.255.0
 full-duplex
 no ip redirects
 no ip proxy-arp
 no ip unreachables
 no ip directed-broadcast
 no ip mask-reply
 no cdp enable
!
interface FastEthernet0
 description Line to backbone router
 ip address {public PtP address}
 load-interval 30
 no ip redirects
 no ip proxy-arp
 no ip unreachables
 no ip directed-broadcast
 no ip mask-reply
 speed 100
 full-duplex
 no cdp enable
!
ip classless
no ip forward-protocol udp netbios-ns
no ip forward-protocol udp netbios-dgm
ip tacacs source-interface Loopback0
no ip http server
no ip http secure-server
access-list 2 remark ---=== Enable telnet access ===---
access-list 2 permit 10.0.11.0 0.0.0.255
access-list 7 remark ---=== Enable SNMP access ===---
access-list 7 permit 10.0.11.35
access-list 7 permit 10.0.11.135
!
tacacs-server host 10.0.11.146
tacacs-server host 10.0.11.254
tacacs-server timeout 2
no tacacs-server directed-request
tacacs-server key 7 HD39084JDFWer34$fsjifkfaskj3478d
snmp-server community La3Foshiy348XrTn RO 7
snmp-server enable traps tty
!
line con 0
 logging synchronous
 stopbits 1
line aux 0
line vty 0 4
 access-class 2
 exec-timeout 30 0
 history size 256
 transport preferred none
 transport input telnet
 transport output telnet
!
ntp clock-period 17179878
ntp server 194.108.219.2
ntp server 194.108.219.3
```

6.4.2. IPSec configuration

```
class-map match-all SET-VPN
  description VPN traffic
  match access-group name vpn
!
!
policy-map LM-MARK-DATA
  class SET-VPN
   set ip precedence 2
  class class-default
   set ip precedence 1
!
crypto isakmp policy 1
 encr 3des
 authentication pre-share
 group 5
 lifetime 28800
crypto isakmp key verysecurekey address 212.65.242.182
!
crypto ipsec transform-set vpn-prague2 ah-sha-hmac esp-3des esp-sha-hmac
!
crypto map eth0 local-address Ethernet0/1
crypto map eth0 3 ipsec-isakmp
 description VPN - Praha vpn-prague-2
 set peer 212.65.242.182
 set transform-set vpn-prague2
 match address 103
!
interface Tunnel0
 description Line to VPN - Praha
 ip unnumbered Ethernet0/1
 tunnel source Loopback0
 tunnel destination 212.65.242.182
 crypto map eth0
!
interface Ethernet0/1
 description LAN segment for PC
 encapsulation dot1Q 1 native
 ip address 10.x.x.1 255.255.0.0
 ip helper-address 10.0.11.155
 no ip proxy-arp
 service-policy input LM-MARK-DATA
 no cdp enable
 crypto map eth0
!
ip route 0.0.0.0 0.0.0.0 Tunnel0
ip route 212.65.242.182 255.255.255.255 (PtP address on backbone router)
!
ip access-list extended vpn
 permit ip any 192.168.0.0 0.0.255.255
 permit ip any 10.0.0.0 0.255.255.255
 permit ip any 172.16.0.0 0.128.255.255
access-list 103 permit gre host {loopback 0 address} host 212.65.242.182
```

6.4.3. VoIP configuration

```
ip dhcp pool dhcp1
   description DHCP for IP phones
   network 10.x.x.x 255.255.255.0
   default-router 10.x.x.x
   option 150 ip 194.212.247.101
   option 66 ascii "194.212.247.101"
!
voice call send-alert
voice call carrier capacity active
!
voice class codec 1
 codec preference 1 g729r8 bytes 40
!
class-map match-all VOICE
   description Low latency - voice traffic
  match ip precedence 5
!
policy-map LM-MARK-VOIP
  class class-default
   set precedence 5
!
policy-map LM-QOS
  class VOICE
   priority xxx   (in kbps depending on number of IP phones - 26 kbps/call)
  class class-default
   set precedence 1
!
gw-accounting aaa
 acct-template callhistory-detail
 suppress rotary
!
interface Loopback0
 description Loopback for FXS ports
 ip address {public IP address with subnet mask 255.255.255.255}
 h323-gateway voip interface
 h323-gateway voip bind srcaddr {loopback IP address}
!
interface Loopback1
 description Loopback pro IP telefony
 ip address {public IP address with subnet mask 255.255.255.255}
!
interface Ethernet0/1
 no ip address
 no ip redirects
 no ip unreachables
 no ip proxy-arp
!
interface Ethernet0/1.1
 description IP phones
 encapsulation dot1Q 1
 ip address 10.x.x.1 255.255.255.0
 ip nat inside
 service-policy input LM-MARK-VOIP
!
ip nat inside source list 50 interface Loopback1 overload
!
ip access-list extended voice
 permit ip any any precedence critical
 permit tcp any range 1718 1720 any
access-list 50 remark ---=== ACL for IP phones ===---
```

```
access-list 50 permit 10.x.x.x 0.0.0.255
!
ip radius source-interface Loopback1
!
radius-server host 212.65.216.81 auth-port 1812 acct-port 1813
radius-server host 212.65.216.80 auth-port 1812 acct-port 1813
radius-server deadtime 30
radius-server key 7 38947IDFHfer#W3#8SI7&3KF887233KAH833
radius-server vsa send accounting
!
voice-port 2/0
 echo-cancel coverage 32
 no comfort-noise
 cptone CZ
 timeouts initial 30
 timeouts interdigit 4
!
voice-port 2/1
 echo-cancel coverage 32
 no comfort-noise
 cptone CZ
 timeouts initial 30
 timeouts interdigit 4
!
dial-peer voice 1 pots
 huntstop
 destination-pattern {phone number for fax 1}
 port 2/0
!
dial-peer voice 2 pots
 huntstop
destination-pattern { phone number for fax 1}
 port 2/1
!
dial-peer voice 1001 voip
 description International calls
destination-pattern 00T
 voice-class codec 1
 session target ipv4:194.108.2.3
 ip qos dscp cs5 media
 ip qos dscp cs5 signaling
 no vad
!
dial-peer voice 1002 voip
 description National calls
destination-pattern [23456789].......
 voice-class codec 1
 session target ipv4:194.108.2.3
 ip qos dscp cs5 media
 ip qos dscp cs5 signaling
 no vad
!
dial-peer voice 1500 voip
 description Emergency calls
destination-pattern 1T
 translate-outgoing called 10
 voice-class codec 1
 session target ipv4:194.108.2.3
 ip qos dscp cs5 media
 ip qos dscp cs5 signaling
 no vad
```

```
!
dial-peer voice 2000 voip
 description VOICE VPN
 destination-pattern {another branch phone number}
 translate-outgoing called 40
 voice-class codec 1
 session target ipv4:194.108.2.3
 ip qos dscp cs5 media
 ip qos dscp cs5 signaling
 no vad
```

6.4.4. Cisco 837 configuration

```
vc-class atm ADSL512
  ubr 602
  oam-pvc manage
  encapsulation aal5snap
!
interface Loopback0
 description Loopback pro VPN
 ip address {public IP address with subnet mask 255.255.255.255}
!
interface ATM0/0
 no ip address
 no atm ilmi-keepalive
 dsl operating-mode auto
!
interface ATM0/0.1 point-to-point
 description ADSL 512/256
 ip address {public PtP address}
 ip nat outside
 pvc x/y    {x/y is VCI/VPI specified by Czech Telecom/Telenor}
```

6.4.5. Cisco 7205 configuration

Configuration of central router is the same as the configuration above. On this router are terminated tunnels from branches router – only the difference is in source and destination addresses of tunnel – addresses are swaped. Configuration for IPSec, AAA and VoIP is identical. The all network is based on static routing; no dynamic routing is used for security purposes.

DHCP server on all routers use following template:

```
ip dhcp excluded-address 10.x.x.1 10.x.x.49
ip dhcp pool Branch_name_of_brach
 network 10.x.x.0 255.255.255.0
 domain-name in.contactel.cz
 dns-server 10.0.11.2 10.0.11.3 212.65.193.157
 default-router 10.x.x.1
 lease 48
```

DHCP server must be started using command **`service dhcp`**.

Conclusion

In my bachelor work I tried to explain the basics of network security. As I declared in the Introduction of my bachelor work, it is impossible to describe all possible security threads, security holes and defense against these security problems. My bachelor work flows from knowledge which I gained in Contactel. This knowledge reflects my working experiences and procedures applied in Contactel.

The first chapter of my bachelor focuses on possible network security threads and vulnerabilities. I choose common security problems which are today widely used. One subchapter deals with widely used Internet attacks – Denial of Service attacks. I describe in this chapter the basics of designing network security and possible firewall solutions.

Second chapter called "Securing physical, datalink and network layer" describe securing on the first three layers of OSI model. Very important is securing the physical access to network equipment because many active network elements allow various password recovery procedures. If the potential intruder has physical access to network equipment than the intruder can take over full access to the system and change configuration or install malicious programs. When physical security is assured, datalink and network layers are responsible for the correct functioning of the rest of the network. Securing datalink layer using VLANs is widely used and is recommended. Remote access to network equipment (using telnet, ssh or management protocols) is protected on network layer – filtering base on source IP addresses. Very important is securing routing updates.

"Security transport and application layer" is the title of the third chapter. This chapter describes packet filters (called ACL – access control list) used on Cisco routers and firewalls. Various types of access list and usage of this access lists is described here. In subchapter "Context-based access lists" is described technology of stateful packet filtering. This configuration is often used on CPEs connected to Contactel's network.

Hardware firewalls are widely used in enterprise networks. Cisco has a special series of hardware firewalls – PIX Firewall and the fourth chapter is focused on this network equipment. PIX is powerful network equipment which allow VPN connections, stateful packet filtering and protect enterprise network. I described some configurations which I prepared for Contactel's customers.

For secured remote access and secured communication for example between company branches and headquarters or for secured communication over Internet are very important encryption and virtual private networks. Fifth chapter deals with VPN and encryption. Again I described some configurations of customer's solutions which I prepared and configured on CPE and backbone routers.

The last chapter represents the scheme of future Contactel's private network. The goal of this chapter is to design a private network between Contactel's headquarter and branches in Plzeň, Liberec, Brno, Ostrava and Olomouc. Final network will be able to carry user data and voice packets from IP telephones. Network topology and configurations will be first tested in a lab – after successful tests the final design will be implemented.

Bibliography

Czech literature

WENSTROM, M.. Zabezpečení sítí Cisco. 1. vyd. Brno : Computer Press, 2003. 784 s. ISBN 80-7226-952-6.

TEARE, D.. Návrh a realizace sítí Cisco. 1. vyd. Brno : Computer Press, 2003. 784 s. ISBN 80-251-0022-7.

HUCABY, D.. Konfigurace směrovačů Cisco. 1. vyd. Brno : Computer Press, 2004. 632 s. ISBN 80-722-6951-8.

CHAPMAN, D.. Zabezpečení sítí Cisco pomocí PIX Firewall. 1. vyd. Brno : Computer Press, 2004. 368 s. ISBN 80-722-6963-1.

KABELOVÁ, A. a DOSTÁLEK, L.. Velký průvodce protokoly TCP/IP a systémem DNS,. 3. roz. vyd. Brno : Computer Press, 2002. 552 s. ISBN 80-7226-675-6.

DOSTÁLEK, L. a kol.. Velký průvodce protokoly TCP/IP: Bezpečnost. 2. akt. vyd. Brno : Computer Press, 2003. 592 s. ISBN 80-7226-849-6.

SHINDER, D.. Počítačové sítě. 1. vyd. Praha : Softpress, 2003. 752 s. ISBN 80-8649-755-0.

MCCLURE, S. a SHAH, S.. Web hacking – útoky a obrana. 1. vyd. Praha : Softpress, 2003. 448 s. ISBN 80-8649-753-4.

HORÁK, J.. Bezpečnost malých počítačových sítí. 1. vyd. Praha : Grada, 2003. 254 s. ISBN 80-247-0663-6.

COLEMAN, P. a DYSON, P.. Intranet – plánování, výstavba, provoz – podrobný průvodce. 1. vyd. Praha : Grada, 1998. 352 s. ISBN 80-7169-670-6.

KÁLLAY, F. a PENIAK, P.. Počítačové sítě a jejich aplikace. 1. vyd. Praha : Grada, 2003. 356 s. ISBN 80-247-0545-1.

Foreign literature

GOUGH, C.. CCNP BSCI – Exam Certification Guide. 1. vyd. Indianapolis : CiscoPress, 2003. 918 s. ISBN 1-58720-078-3.

MORGAN, B.. CCNP BCRAN – Exam Certification Guide. 2. vyd. Indianapolis : CiscoPress, 2004. 520 s. ISBN 1-58720-084-8.

RANJBAR, A.. CCNP CIT – Exam Certification Guide. 2. vyd. Indianapolis : CiscoPress, 2004. 315 s. ISBN 1-58720-081-3.

BOLLAPRAGADA, V. a KHALID, M.. IPSec VPN Design. 1. vyd. Indianapolis : CiscoPress, 2005. 384 s. ISBN 1-58705-111-7

DEAL, R.. Cisco Router Firewall Security. 1. vyd. Indianapolis : CiscoPress, 2004. 912 s. ISBN 1-58705-175-3.

CONVERY, S.. Network Security Architectures. 1. vyd. Indianapolis : CiscoPress, 2004. 792 s. ISBN 1-58705-115-X.

Appendix 1

Network security weaknesses

Configuration weaknesses	
Unsecured user accounts	User account information may be transmitted insecurely across the network, exposing usernames and passwords to snoopers.
System accounts with easily guessed password	This common problem is the result of poorly selected and easily guesses user passwords.
Misconfigured Internet services	A common problem is to turn on JavaScript in Web browsers, enabling attacks via hostile JavaScript when accessing untrusted sites. IIS, FTP and Terminal Services also pose problems.
Unsecured default settings within products	Many products have default settings that enable security holes (for expample default community strings in SNMP).
Misconfigured network equipment	Misconfiguration of the equipment itself can cause significant security problems. For example, misconfigured access lists, routing protocols or SNMP community strings can open large security holes.
Security policy weaknesses	
Lack of written security policy	An unwrite policy cannot be consistently applied or enforced.
Politics	Political battles and turf wars can make it difficult to implement a consistency od security politics.
Logical access controls not applied	Poorly choosen, easily cracked or default passwords can aloow unauthorized access to the network. Inadequate monitoring and auditing allow attacks and unauthorized use to continue, wasting company resources. This could result in legal action or termination against IT technicians, IT management or even company leadership that allows these unsafe conditions to persist.
Software and hardware installation and changes do not follow policy	Unauthorized changes to the network topology or installation of unapproved applications create security holes.
Disaster recovery plan is nonexistent	The lack of a disaster recovery plan allows chaos, panic and confusion to occur whne someone attacks the enterprise.

Appendix 2

Encrypted/unencrypted passwords

The following example shows a router configuration prior to enabling password encryption. An enable password, a console password, and a Telnet password is configured:

```
SecureRouter#show running-config
!
enable password Cisco
!
line con 0
password Networking
!
line vty 0 4
password Security
!
```

The following example shows the command you would use to enable password encryption on the router:

```
SecureRouter#config t
Enter configuration commands, one per line. End with CNTL/Z.
SecureRouter(config)#service password-encryption
SecureRouter(config)#end
SecureRouter#
```

The results of enabling password encryption can be seen in the following example. Notice that each password is now represented by a string of letters and numbers, which represents the encrypted format of the password:

```
SecureRouter#show running-config
!
enable password 7 05280F1C2243
!
line con 0
password 7 04750E12182E5E45001702
!
line vty 0 4
password 7 122A00140719051033
```

Appendix 3

RIP authentication

Router A configuration with MD5 authentication.

```
key chain systems
key 1
key-string router
!
interface Loopback0
ip address 10.10.10.1 255.255.255.0
!
interface Ethernet0/0
ip address 10.10.11.1 255.255.255.0
!
interface Serial0/0
ip address 192.168.10.1 255.255.255.252
ip rip authentication mode md5
ip rip authentication key-chain systems
!
router rip
version 2
network 10.0.0.0
network 192.168.10.0
```

Router B configuration with MD5 authentication.

```
key chain cisco
key 1
key-string router
!
interface Loopback0
ip address 10.10.12.1 255.255.255.0
!
interface FastEthernet0/0
ip address 10.10.13.1 255.255.255.0
!
interface Serial0/0
ip address 192.168.10.2 255.255.255.252
ip rip authentication mode md5
ip rip authentication key-chain cisco
!
router rip
version 2
network 10.0.0.0
network 192.168.10.0
no auto-summary
```

The output of the command **`debug ip rip`** displays how Router A receives RIP routing updates from Router B.

```
Router-A#debug ip rip
RIP protocol debugging is on
Router-A#
RIP: received packet with MD5 authentication
RIP: received v2 update from 192.168.10.2 on Serial0/0
10.10.12.0/24 -> 0.0.0.0 in 1 hops
10.10.13.0/24 -> 0.0.0.0 in 1 hops
```

Appendix 4

EIGRP authentication

Router A configuration with MD5 authentication.

```
key chain router-a
key 1
key-string eigrp
!
interface Loopback0
ip address 10.10.10.1 255.255.255.0
!
interface Ethernet0/0
ip address 10.10.11.1 255.255.255.0
!
interface Serial0/0
ip address 192.168.10.1 255.255.255.252
ip authentication mode eigrp 2 md5
ip authentication key-chain eigrp 2 router-a
!
router eigrp 2
network 10.0.0.0
network 192.168.10.0
no auto-summary
eigrp log-neighbor-changes
```

Router B configuration with MD5 authentication.

```
key chain router-b
key 1
key-string eigrp
!
interface Loopback0
ip address 10.10.12.1 255.255.255.0
!
interface Ethernet0/0
ip address 10.10.13.1 255.255.255.0
!
interface Serial0/0
ip address 192.168.10.2 255.255.255.252
ip authentication mode eigrp 2 md5
ip authentication key-chain eigrp 2 router-b
clockrate 64000
!
router eigrp 2
network 10.0.0.0
network 192.168.10.0
no auto-summary
```

The **debug eigrp packet** command informs you if the router has received a packet with the correct key value and key string. The output of issuing this command can be seen here:

```
Router-A#debug eigrp packet
EIGRP Packets debugging is on
(UPDATE, REQUEST, QUERY, REPLY, HELLO, IPXSAP, PROBE, ACK)
Router-A#
EIGRP: received packet with MD5 authentication
EIGRP: received packet with MD5 authentication
```

Appendix 5

OSPF authentication

Router A configured for MD5 authentication:

```
interface Loopback0
ip address 10.10.10.1 255.255.255.0
!
interface Ethernet0/0
ip address 10.10.11.1 255.255.255.0
!
interface Serial0/0
ip address 192.168.10.1 255.255.255.252
ip ospf message-digest-key 15 md5 miller
clockrate 64000
!
router ospf 60
area 0 authentication message-digest
network 10.10.10.0 0.0.0.255 area 10
network 10.10.11.0 0.0.0.255 area 11
network 192.168.10.0 0.0.0.255 area 0
```

Router B configured for MD5 authentication:

```
interface Loopback0
ip address 10.10.12.1 255.255.255.0
!
interface Ethernet0/0
ip address 10.10.13.1 255.255.255.0
!
interface Serial0/0
ip address 192.168.10.2 255.255.255.252
ip ospf message-digest-key 15 md5 miller
!
router ospf 50
area 0 authentication message-digest
network 10.10.12.0 0.0.0.255 area 12
network 10.10.13.0 0.0.0.255 area 13
network 192.168.10.0 0.0.0.255 area 0
```

Router A configured with multiple keys and passwords.

```
interface Loopback0
ip address 10.10.10.1 255.255.255.0
!
interface Ethernet0/0
ip address 10.10.11.1 255.255.255.0
!
interface Serial0/0
ip address 192.168.10.1 255.255.255.252
ip ospf message-digest-key 15 md5 miller
ip ospf message-digest-key 20 md5 ampaq
!
router ospf 60
area 0 authentication message-digest
network 10.10.10.0 0.0.0.255 area 10
network 10.10.11.0 0.0.0.255 area 11
network 192.168.10.0 0.0.0.255 area 0
```

APPENDIX 6

CBAC TIMEOUTS

Timeout or threshold value to change	Command	Default
The length of time that the software waits for a TCP session to reach the established state before dropping the session.	**ip inspect tcp synwait-time *second***	30 second
The length of time that a TCP session will still be managed after the firewall detects a FIN-exchange.	**ip inspect tcp finwait-time *second***	5 second
The length of time that a TCP session will still be managed after no activity (TCP idle timeout).	**ip inspect tcp idle-time *second***	1 hour
The length of time that a UDP session will still be managed after no activity (UDP idle timeout).	**ip inspect udp idle-time *second***	30 second
The length of time that a DNS name lookup session will still be managed after no activity.	**ip inspect dns-timeout *second***	5 second
The number of existing half-open sessions that will cause the software to start deleting half-open sessions.	**ip inspect max-incomplete high *number***	500 half-open sessions
The number of existing half-open sessions that will cause the software to stop deleting half-open sessions.	**ip inspect max-incomplete low *number***	400 half-open sessions
The rate of new unestablished sessions that will cause the software to start deleting half-open sessions.	**ip inspect one-minute high *number***	500 half-open sessions per minute
The rate of new unestablished sessions that will cause the software to stop deleting half-open sessions.	**ip inspect one-minute low *number***	400 half-open sessions per minute
The number of existing half-open TCP sessions with the same destination host address that will cause the software to strat dropping half-open sessions to the same destination host address.	**ip inspect tcp max-incomplete host *number* block-time *minutes***	50 existing half-open TCP session 0 minutes