Some features dominate the dataset and this may result in uneven classification. Also some types of attacks are dominating the dataset such as denial_of_service attacks are more in numbers than user to root attacks. Applying feature selection and extraction methods is required to select the relevant features from the complete dataset. Also normalization of data is required as some features are having continuous values whereas some features may have discrete values. The dataset contains 4,94,000 records which may produce false alarms and high time complexity if applied machine learning algorithm applied directly. So preprocessed the training data, applied normalization to it and created five subsets having equal distribution of all types of attack instances and normal instances.

The other dataset which we used in our system is the data collected from real time packet capture through a network sensor. For the live packet capture, system constraints are checked against the received packets. The live packet capture stream is parsed for selecting the required attributes as all attributes of packets are not contributing to the intrusion detection. The data of the selected attributes is stored in the database for the intrusion detection analysis. Network traffic or data traffic is data in a network encapsulated in the packets. We collected network traffic dataset by setting up an environment that simulates the real time network environment.

- **System Constraint Check:** Preprocessing of data plays an important role in intrusion detection and prevention. As packets arrive at sensor with a very high speed, preprocessing process is also required to be quicker and efficient. To increase the speed the packets from the unknown or from known malicious source are not required to be processed. Also the traffic to the services which are vulnerable and blocked need not require to be processed. These constraints will make the preprocessing process fast as traffic is filtered based on these constraints. System constraints are the rules set by the system administrator for the incoming traffic. These rules are based on the activity log of various users as well as vulnerability study of various protocols and services. If any packet violates the rules, it is declared as invalid and discarded at the time of capture. The Constraints are related to the various features of protocols such as IP address, Packet Length, flag bits, protocol type etc. for example if ICMP traffic is restricted in the network by network administrator from a specific IP, then this constraint will discard such traffic at sensor.

- **Header Feature Extraction:** After filtering the network traffic, next stage in preprocessing is to extract the features from the packet header. As all the fields of packet header are not contributing to the intrusion detection process, we select only required fields from each header. As the incoming data is of different protocols we need to apply filtering process for selecting the protocol specific parameters. The configuration of each protocol is stored in special file which is used to parse the parameters. The header feature extraction process first creates two parts of the packets data and header. The header part is then further processed to collect the

required header variables. The process is then using JDBC connectivity to store the data in SQL database of packets. The process is shown in Figure 10.5

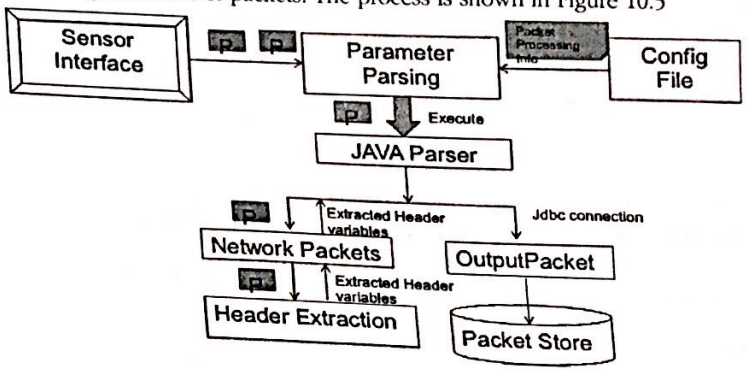


Figure 10.5 Header Feature Extraction

The process of header feature extraction is explained in Table 4.1

Table 10.1 Description of Classes

Class Name	Function
Config File	Configuration file stores the rules for feature extraction from the incoming traffic according to type of protocols. It stores the setting into different variables. Default settings will be applied if no match is found. Parse uses the configuration file to retrieve the setting for the packet to be preprocessed.
Parameter_ parsing	This class reads the Jpcap output line by line and processes it. Creates network packet for each line collects the packet from network interface. According to type of packet the retrieves the configuration file and forwards it to the parser class
Parser	It collects the network packet and configuration file. Determines the type of packet and read settings from config file. Call next class for header feature extraction and after receiving output provide the features to output packet class.
NetworkPacket	Read the input from the parser and create corresponding header as well as session information for data storage according to type of protocol. Provide a interface to extract header information even though packet protocol is different.

Header Extraction	Extract the header fields and options from hexadecimal packet payload and store them into variables. For TCP collect the flag fields, port information. For IP collect the source, destination, TTL, fragmentation information. For UDP port information, checksum and session records are managed. For ICMP type of message, size, source, destination is stored. Collect the superset of all the available header options for TCP/UDP.
OutputPacket	Read the extracted features of packets, create JDBC connection to the database and store the packet to the packetstore database for future analysis.

- **Feature Selection and Extraction System Constraint check:** The other dataset used in system is KDD dataset. Every record in the KDD 1999 data set symbolizes 41 features representing a variety of attacks such as the Probe, DoS, R2L and U2R. For intrusion analysis all the 41 features are not required. Some specific features are only contributing for a specific attack. This reduces the amount of work for intrusion detection and increases accuracy. However, using all the 41 features for detecting attacks belonging to all these classes severely affects the performance of the system and also generates unnecessary rules. Feature selection is performed to effectively detect different classes of attacks. Algorithm for feature selection is tested for each category of attack. For every category, all relevant attributes for that category are applied, calculated gain for them and generated small subset which contains most relevant attributes for that category. All the 41 features for attribute selection are considered in the first pass. Information gain is calculated for these attributes. The attribute with maximum information gain is selected as splitting attribute and the dataset is divided into subsets as per the value of attributes. Similarly find all the attributes whose information gain is above threshold. These attributes are selected and dataset is parsed for the extraction of values for the selected attributes. The newly created set is used for intrusion detection and analysis experimentation which is given in coming sections.

10.4.2 Intrusion Detection Engine

The second important module of IPS system is intrusion detection module. The preprocessing module parses the packets for system constraints and forwards the packet to intrusion detection module. As a gate-keeper, the intrusion detection module intercepts and filters all incoming packets according to a small set of precoded rules. These rules are attack signatures which are unique format of information that can be used to identify an attacker's attempt to exploit a known vulnerability. To make the intrusion detection faster the main process is subdivided into three modules according to type of protocol is shown in Figure 10.6

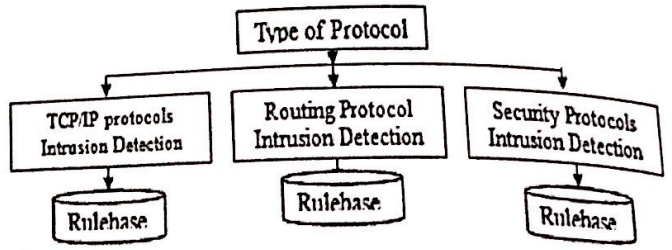


Figure 10.6 Intrusion Detection Engine

To make intrusion detection faster and efficient, rules applicable to respective protocol are only explored and applied. The rules are stored in the rulebase according to type of protocols. This technique helps in reducing the searching rules from rulebase. As the rule search process is protocol specific the intrusions can be detected in logarithmic time. The rulebases are indexed files which make the accessing faster. If intrusion is found, category of intrusion is delivered to the protection engine for performing actions. If no intrusion is found packet is declared normal but still processed and stored in databases according to their protocol field which makes the intrusion analysis productive.

The System is partitioned in three intrusion detection subsystems.

1. TCP/IP Intrusion Detection: this subsystem contains the intrusion signatures for TCP, IP, ICMP, SMTP and DNS protocols
2. Routing Protocols Intrusion Detection: this subsystem contains the intrusion signatures for routing protocols such as RIP, BGP and OSPF.
3. Security Protocols Intrusion Detection: This subsystem contains the intrusion signatures for Security protocols such as IPSec, SSL and DNSSec.

10.4.2.1 TCP/IP Intrusion Detection

After studying the vulnerabilities and attacks on the protocols of TCP/IP layers, here we propose a framework for detecting and preventing these attacks using data mining feature. System works at all layers of TCP/IP for detection as well as prevention. Figure 10.7 shows System architecture of TCP/IP Intrusion Detection Subsystem.

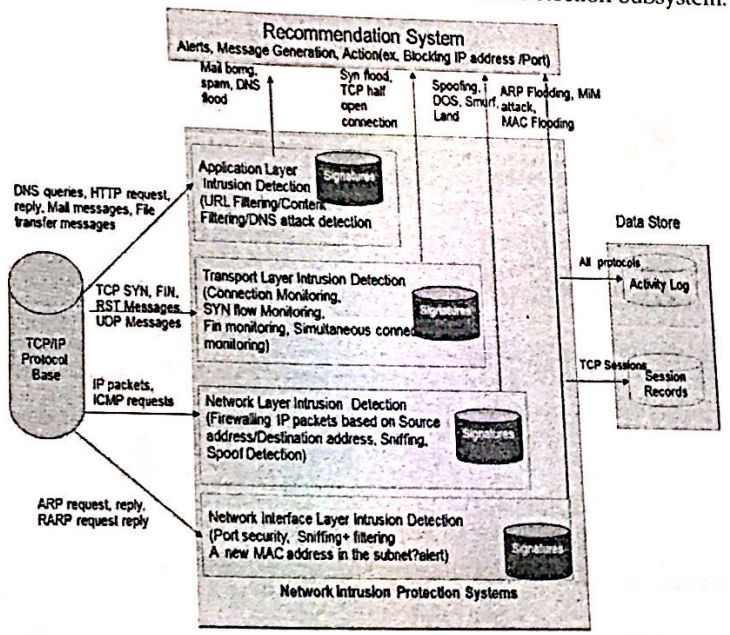


Figure 10.7 TCP/IP Intrusion Detection Subsystem

The Figure 10.7 shows the TCP/IP intrusion detection process. The Process applies the signature based detection to the collected network traffic for known attack detection. The above intrusion detection subsystem consists of four intrusion detection processes for each layer of TCP/IP model. These processes apply pattern matching to the packets for intrusion detection. Each process performs signature matching for the well-known attacks. Each intrusion detection process contains Rulebase of signatures for intrusion detection. For example Transport layer intrusion detection module contains signatures for Syn flood, Land, Half open connections. Network Layer Intrusion detection comprises of signatures for spoof detection, smurf, flooding, teardrop attacks. For each layer we analyzed the vulnerabilities and attacks. These attacks are then crafted as simple If_then_else rules for our system.

The TCP/IP intrusion detection subsystems consist of four modules: 1. Application Layer Intrusion Detection, 2. Transport Layer Intrusion Detection 3. Network Layer Intrusion Detection and 4. Network Interface Layer Intrusion Detection. The details are given as below.

1. **Application Layer Intrusion Detection:** Signatures for the well-known application layer attacks are stored in the signature database. The attack such as DNS Flooding, DNS amplification, DNS redirect, Denial of service of DNS protocol are hard coded in the system. Signatures for SMTP attack such as SMTP bounce, mail spoofing are also maintained in the system.
2. **Transport Layer Intrusion Detection:** Transport layer provides connection oriented service through TCP and Connectionless service through UDP protocol. The vulnerabilities and attacks are discussed in previous chapters. The intrusion detection at transport layer scans for TCP Syn attack, Land attack, TCP half open connections, TCP Syn Food attacks in the packet stream and matches the signatures of attacks with packets. For UDP protocol pattern matching is performed for UDP flood, Covert UDP and Smurf attacks.
3. **Network Layer Intrusion Detection:** Network layer is backbone of communication network as it holds important functions in communication such as addressing, routing and source to destination delivery of packets. Intrusion detection at network layer focuses on identifying spoof packets and flooding attacks which can cause Denial_of_service. This subsystem contains signatures for protocols IP and ICMP. The attacks which can be detected through the sub system are IP spoofing, Smurf, Ping of Death, ICMP redirect, Land and Flooding attacks. These attacks are already discussed in previous chapter on literature survey.
4. **Network Interface Layer Intrusion Detection:** Network Interface Layer is physical connection and data transfer between hosts to host. ARP and RARP play important role at this layer to deliver the content to correct host using MAC address or physical address of the host. The attacks at this layer are ARP flooding, cache poisoning, MAC flooding are detected using the signatures for the attacks.

Data stores used for TCP/IP Intrusion Detection

For TCP/IP Intrusion detection we have used three different data stores namely 1. Attack signatures 2. Activity logs and 3. Session records. Each is explained in detail below:

- **Attack Signatures**

  Signatures are divided into two logical sections, the header and contents. The signature header contains the protocol, source and destination IP addresses and the source and destination ports information. The contents contains the patterns to match, alert messages and options that define which parts of the packet should

be inspected to determine if the rule action should be taken. The signatures are written for every attack type defined above.

- **Activity Logs**

  Log events are the primary records of system and network activity. Log analysis can be used as the source of intrusion detection to detect computer misuse, malicious activity or security policy violations. We create activity logs in our system which stores the access information of various source addresses and resources. Each log entry consist of following fields

  L={Identification; Date; Timestamp; P; SIP, PN; DIP,PN, Seq; Ack; Window}

  Where P is protocol, SIP is source IP, DIP is Destination IP, PN is port number, Seq is sequence number and Ack is acknowledgement number.

  The log files are created for various protocols TCP, UDP and ICMP.

- ***Session Records***

  The connection records are essential in detection of many attacks. For example flooding attack, spoofing attacks requires the packet information to be stored for a particular time period. Over the time the activity pattern of user can be used to determine the malicious activity. Suppose source A is sending ping packet to the server. Only one packet cannot determine the malicious activity. But if you see over the time period of 10 sec or so, and number of ping packets from the source A to server is more than a threshold value, then we can determine it as flooding or ping of death attack. For session records information we maintain the records of timestamp, sourceIP, DestinationIP, source_port, destination_port, Protocol, counter, size etc.

10.4.2.2 Routing Intrusion Detection

TCP/IP model support various routing protocols such as RIP, BGP, OSPF and IGP. They are used for navigating traffic from source to destination in communication network. The existing vulnerabilities in these protocols make the routing communication insecure. Routing Intrusion detection module is designed to handle the well- known attacks on routing protocols. Routing was simulated using simulator and attacks were manually crafted. The attacks on routing protocol are already discussed in previous section. The Intrusion Prevention System is made stronger by integrating routing intrusion detection capabilities with standard Intrusion Detection Mechanisms. The subsystem involves detection of the routing exploits such as Smurf, Land, TCP SYN Flood, DRBDR Null, Periodic Injection, Hello Protocol attacks. For the system we have considered Routing Information Protocol (RIP), Border Gateway protocol (BGP) and Open Shortest path First (OSPF) algorithms for our system. Routing Intrusion Detection Subsystem is as shown in Figure 10.8

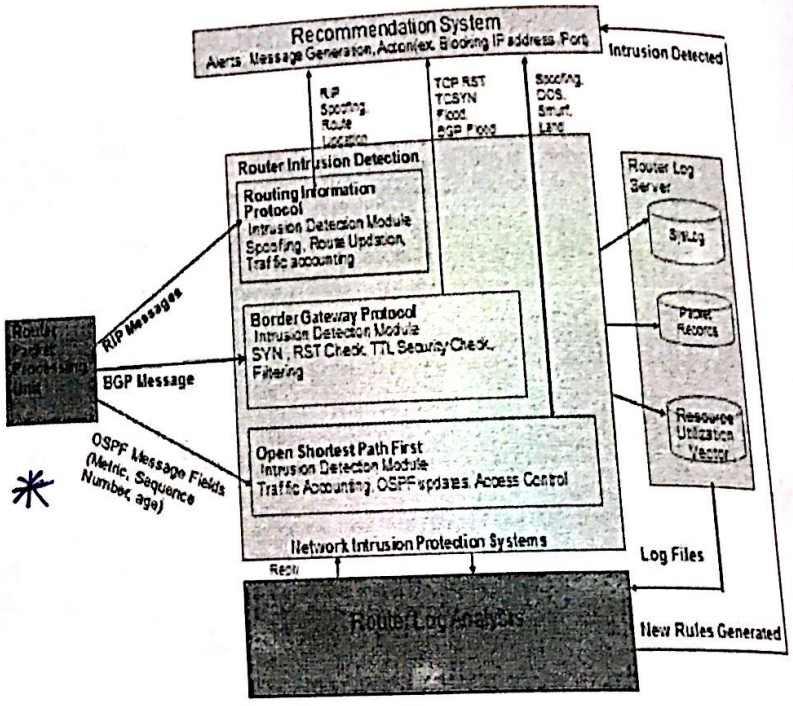


Figure 10.8 Routing Intrusion Detection Subsystem

Routing intrusion detection involves four major components:

1. Packet Preprocessing Unit,
2. Intrusion Detection Unit
3. Router Log Analysis
4. Data stores such as Signature database, activity log and session records.

Each module is explained in detail as below:

1. Packet Preprocessing Unit

A router's core component is the packet processing unit which inspects the various fields of RIP, BGP and OSPF messages. Nowadays a single router has more than one unit for packet processing. This facilitates faster processing. Packet processing unit is the software designed to capture packets and inspect for intrusions as shown in Figure 10.9.

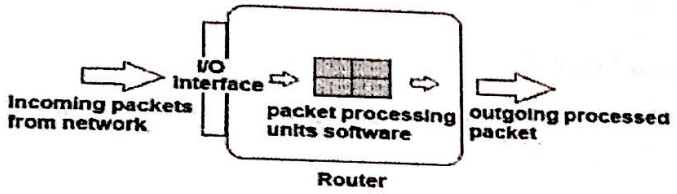


Figure 10.9 Preprocessing Unit

2. Intrusion Detection Unit

The intrusion detection module is signatures based system which detects intrusion by matching signatures of known attacks. System will be trained for normal functioning of router. If any attack is detected then it will drop the current packet and bring the processing unit back to fresh new state. The attack signatures are written manually for known attacks and stored in the files for access. Intrusion detection unit employs two main techniques for intrusion detection: Traffic filtering and Session monitoring

- **Traffic filtering:** The intrusion detection module inspects the different parameters for different protocols. For example for RIP protocol we inspect route updation messages, sessions from a source. For BGP routing we inspect SYN, FIN,RST flags, TTL security, session monitoring etc. for OSPF we check Hello packets, DR, BDR fields value, traffic filtering.
- **Session monitoring:** We maintain a few counters to find which session is going on between different hosts and which router is behaving abnormally in a network. Suppose A and B are the two routers participating in Session. Following three counters which will be maintained by router

1. A counter for packets which flow through both A, B
2. A counter for packets whose source is A but which flows through B
3. A counter for packets that flow "through" A but whose destination is B

3. Router Log Analysis

In routing intrusion detection system monitoring the log files of routing packets is applied as prevention approach. The major requirement for this approach is to log the routing information of packets. But performing all possible logging causes the router to become slow as many logs files will be generated. In order to use log for intrusion detection efficiently following can be done. As shown in Figure 4.10, the logs are directed to a separate server called as RouterLog Server. Log analysis and processing is performed on a separate machine. One more issue with log analysis is as follows. As huge amounts of logs will be generated every second and it will take a lot of router's physical memory, we dump them to the Log Server.

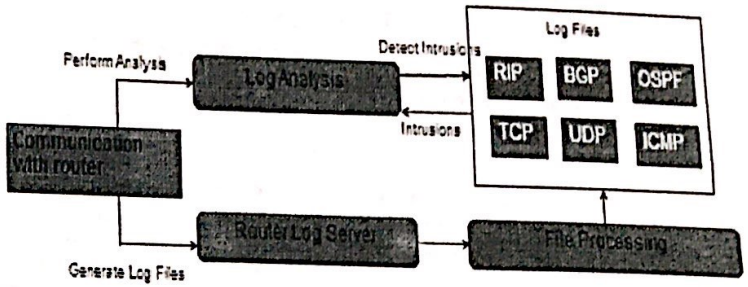


Figure 10.10 Router Log Analysis

The machine on which the Log server is running is connected to the core switch. In this way, all the routers can direct their logs to the Log server. This gives the network administrator control over monitoring all the routers. We used Syslog application for windows which is available on the internet. To limit the number of messages sent to the Router's Log Server, use the logging trap router configuration command. The logging trap command limits the logging messages sent to RouterLog Servers. To send logging messages to a RouterLog Server, its host address is specified with the logging command. The no logging trap command disables logging to RouterLog Server. Kiwi Server mentions the source of log entry. It will store the IP address of router from which the log entry has come. To perform intrusion detection from the logs regular expression matching mechanism is used. Java provides a lot of classes for regular expression. In our model, we have used class Pattern and class Matcher extensively.

The algorithm for implementing Log Server is given below

Algorithm 4.1 RouterLog Server

Input: Packet P

Output: Log Files

Step1: Oper UDPPort 514 on the system where IPAddress=ServerIPAddress

Step2: Receive Packets where port=514 && IPAddress=ServerIPAddress

Step3: Separate Header & Data from Packet

Step4: Write Data part in a file

Step 5: End

So now the logs are successfully stored on the Routerlog server. Log server will be continuously running for storing the log files and log analysis. Along with Log server, few other modules will be deployed on the system.

```
May 13 22:40:09.455: OSPF: Send hello to 224.0.0.5 area
0 on FastEthernet1/0 from 10.0.0.1
```

Figure 10.11 Router Log Entry

As one can see from the above image, every entry has month, date, time, protocol, log message. We will extract the protocol field through regular expression matching in java. Based on the protocol, the log entries will be written to different file as shown in Figure 4.11. Different files were created: TCP, UDP, ICMP, RIP, OSPF, BGP as we want to detect attacks protocol wise. For example, OSPF attacks will be different from TCP and so on.

Example of Log Analysis

Packets that flow through the network are routed based on the destination field. The source field is not verified to check whether it is a legitimate one or one which is forged. So an attacker can easily forge thousands of IP addresses and fire packets to a router destination. The routing table doesn't change extremely very often. So a packet with a certain source and a certain destination will flow through a certain path according to the routing table. The routers can learn such paths over time. So now if a source address is forged, then router will detect the unusual path that was taken for that source and destination. So a router can systematically learn IP address that it forwards. It can maintain this as the permitted list of IP addresses. Suddenly when the router sees huge traffic, it can stop learning the IP addresses and drop only those packets which are not there in the permitted list.

4. Data Structure used in Router Intrusion Detection

The data structures used for router intrusion detection are 1. Activity log and 2. Regular expressions as described below:

- **Activity Log for Routers**

People often forget of one important in-built intrusion detection system present in almost all network devices which is Log. In routers, many important messages about the functioning and configuration of routers are logged. But turning ON all possible logging causes the router to become slow because a lot of logs are generated. So this feature is often ignored because they don't want to compromise on the speed of the router. Moreover logs are so huge and their formats are also hard to understand. But if proper logging is done then one can utilize this LOG facility to **detect any misconfiguration** or **detect some attack**. Here system

monitors the separate log files of routing packets as prevention approach depends upon the routing protocols. The Log files are created according to type of protocols and store separately.

- **Regular Expressions**

Regular expressions, by definition, are string patterns that describe text. These descriptions can then be used in intrusion detection for string matching. The basic language constructs include character classes, quantifiers, and meta-characters. An important feature of regular expressions is the ability to group sections of a pattern, and provide alternate matches. These features of regular expressions are used in our system for router log analysis. Many attacks such as port scan, DOS, ICMP redirect can be detected using log analysis.

10.4.2.3 Security Protocols Intrusion Detection

The Internet is being used by many clients to access static and dynamic data residing on remote servers. The open architecture of the Internet and the use of open standards like TCP/IP Protocol suite constitute the provisioning of services vulnerable to known Internet attacks. The initial design of TCP/IP protocol suite does not take into consideration the security parameters and vulnerabilities of existing system. To overcome the flaws in present TCP/IP architecture addition of security protocols is done at each and every level of TCP/IP model such as IPSec at network layer, SSL/TLS at transport layer, and DNSSec at application layer. These protocols were successful in providing security to some of the important aspects of network but these protocols themselves suffer from vulnerabilities. Some of these vulnerabilities are the new source to potential attacks. Security protocols are concerned with properties such as integrity and privacy. The major role of security protocols is to tunnel the network traffic between communicating parties by providing authentication, confidentiality and privacy. The security protocols provide services such as cryptography, encapsulation to make the communication secure and confidential. The Intrusion detection subsystem for security protocols provides detection of DOS attacks, Replay attacks, and amplification attacks. The intrusion detection subsystem for security protocols is as shown in Figure 10.12

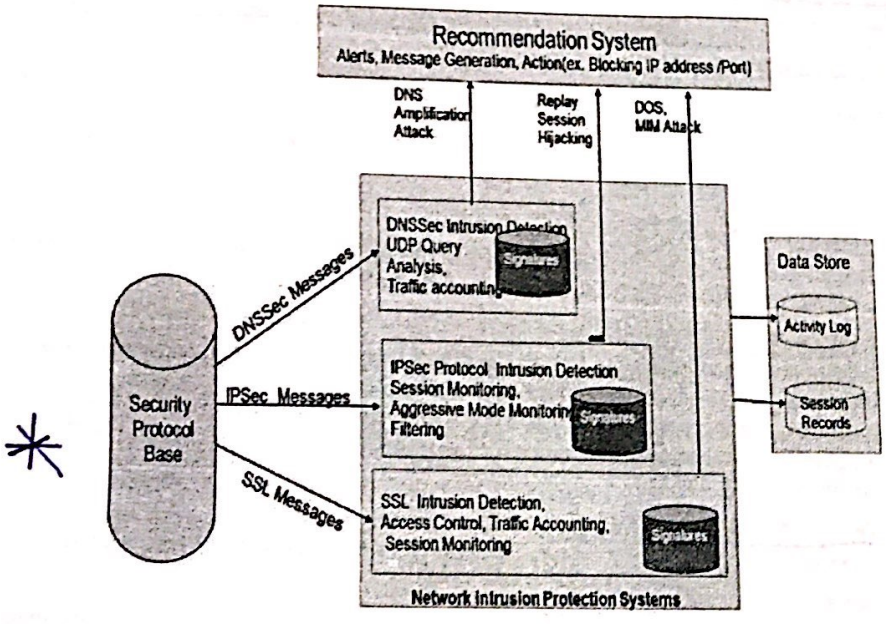


Figure 10.12 Security Protocol Intrusion Detection Subsystem

As shown in Figure 10.12, we concentrate on DNSSec messages, IPSec messages and SSL messages from the traffic. The packet capture and filtering provides filtered output from input packet stream. This decomposition will help to provide detection of protocol related intrusions faster and in an efficient way. We are considering IPSec UDP flooding attacks, SSL renegotiation attacks and DNSSec amplification attacks.

The subsystem for security protocol intrusion detection consists of four modules:

1. DNSSec Intrusion Detection
2. IPSec Intrusion Detection
3. SSL Intrusion Detection
4. Data Structures for Intrusion Detection

Each module is described in detail as below:

1. DNSSec Intrusion Detection

DNSsec packet inspection is to detect the size of the DNS reply is crossing 4000 bytes or not or monitoring DNS queries going to the DNS Server.

The intrusion detection methodology for DNSSec Amplification Attacks is as follows:

1. Source address verification is performed at routers for DNS Queries
2. Drop reply from DNS servers having more than 4000 bytes.

The system constraints are preset for the well-known attacks to increase time efficiency.

2. IPSec Intrusion Detection

IPSec intrusion detection provides monitoring of UDP packets and associated port numbers. This will stop flooding of UDP packets to the port 500 to the servers.

The intrusion detection methodology for IPSec UDP flooding works as follows

1. UDP protocol traffic is monitored and packets with port no 500 are inspected.
2. Counter is set for every known and outgoing entity.
3. If the sender is from known entities and counter is below threshold then packet is allowed. Otherwise packet will be discarded.

3. SSL Intrusion Detection

SSL attack detection check the renegotiation request in an ongoing session by keeping track of session IDs. The intrusion detection methodology for SSL renegotiation attacks is as follows

1. SSL renegotiation can be initiated both by the clients as well as the servers.
2. If initial request for renegotiation is from client, we will not permit the renegotiation.
3. Replay attack collects earlier messages between two peers and replays them so that transaction gets repeated. SSL protects against replay attacks by including an implicit sequence number in the MACed data. This mechanism also protects against delayed, re-ordered, or deleted data.

4. Data Structures for Intrusion Detection

In the case of security protocols only string matching is not sufficient to detect any attack. The packets which are satisfying the first round of check which is signature based intrusion detection will be forwarded to further analysis. The system provides packet filtering, content filtering and Session monitoring features. Also the configuration settings can be modified using the features provided in the system.

10.4.3 Analysis Engine

The main aim of intrusion detection or protection systems is to protect systems from the new threats that come with increasing network connectivity. IDSs and IPS's have

gained significant recognition as a necessary addition to every organization's security. The analysis engine is the core component of intrusion prevention system which analyzes the collected audit patterns to detect attacks. This is a critical component and one of the interesting research domains. As signature based intrusion detection is incapable of detecting novel and unknown attacks, machine learning based intrusion detection is integrated in the system. Approaches such as pattern matching, data mining and statistical techniques can be used in machine learning based intrusion detectors. The capability of the analysis engine to detect an unknown attack in less time often determines the strength of the overall system. For system two important machine learning functions are implemented: first is to detect anomalies from the collected set of audit data and second is to generate useful rules or signatures from the detected anomalies.

10.4.4 Recommendation Engine

Intrusion prevention systems will be classified as active or passive depending upon their response type. Passive intrusion detection systems will always be vulnerable to intrusions. This is due to the fact that passive IPS implements network monitoring which enables traffic ambiguity, making it difficult to provide a reliable intrusion detection mechanism. On the other hand active IPS identifies malicious traffic and it responds with some actions. IPS generates an alert or alarm that an attack has been detected. Administrator can configure signatures to generate event response. If signature is a serious threat, IPS can drop traffic. Besides denying traffic IPS enable Access Control List (ACL) to block suspicious network traffic. The last IPS response is to log network traffic. Logging enables security administrator and system to analyze network traffic and activity pattern of attacker. Our IPS provides these features through recommendation system as shown in Figure 10.13

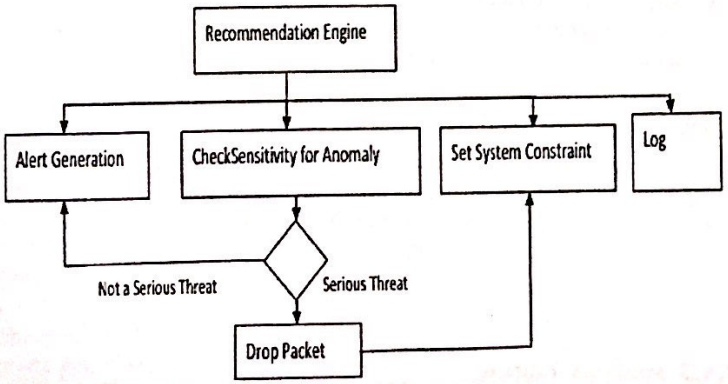


Figure 10.13 Recomendation Engine

Action recommendation is triggered by intrusion detection system upon detection of malicious activity. As shown in Figure the recommendation module provides four main features, alert generation, intrusion sensitivity check, setting system constraint and log generation.

Intrusion prevention system is incomplete without the provision of action for intrusion. The process of intrusion detection and prevention is completed by providing appropriate action for the detected intrusion. The recommendation system provides protection by following ways

1. Logging the captured packet information
   Logging of intrusions detected and action recommended can be used for further analysis of network traffic and malicious activity.
2. Calculate sensitivity of attack by the values of confidence and support.
   Frequent ruleset mining is used to generate signatures for the unknown attacks. Two parameters of ARM support and confidence can be used to measure sensitivity of attack. Support and confidence for the packet is calculated in the malicious network traffic records. Large value of support in the malicious traffic concludes as more severe attack.
3. Alert to system administrator.
   Alert generation display a system alert on the system administrator's screen.
4. Set system constraints for the future intrusions
   System constraints can be set by suggesting possible changes which can be made to the system settings to the administrator.

10.5 Advantages of Layered Approach for Intrusion Detection

Layered approach for Intrusion Detection System draws its motivation from Internet architecture, where a number of modification and checks are performed one after the other in a sequence. Similar to this model, the IPS represents a sequential layered approach to ensure confidentiality, availability and integrity of data over a network. Every layer in layered intrusion detection system framework is trained separately and then deployed sequentially. Our model consists of four sequential layers that correspond to the four TCP/IP. The layered architecture has several advantages over a single layered architecture.

- First the layered security check is applied at each layer of TCP/IP model based on the protocol. The idea is that if ever the attack type or category of any layer is misclassified then the next layer will identify that this record is suspicious.

- Second, it takes less preprocessing time and even workload is decreased in each layer. Preprocessing of packets is faster because each packet is checked for constraints of the system and then passed on for further security check. The system processes packets based on the protocol type and stores the information in the separate datasets.
- As each layer services are divided in separate modules we use only the related dataset for training and testing purpose. Also only the related attacks or training for each layer are scanned for detection. Each layer act as filters that classifies the attacks according to category which eliminate the need of further processing at subsequent layers.
- Third, we used a layered model to reduce the computation time required to detect anomalous events and intrusions. Every layer is trained separately to detect each attack category and then deployed sequentially. In order to make the layers independent, some features may be present in more than one layer.
- Storage space is highly reduced which improves Intrusion detection performance and speed during both the training and the testing phase of the system.
- We implement the Layered Approach to improve overall system performance as our layered intrusion protection system using machine learning gains high efficiency and improve the detection and classification rate accuracy while reducing false alarm rate.
- Fourth, layered approach makes the system adaptive and scalable as the training module can be reoriented at any point of time which makes its implementation adaptive to any new environment and/or any new attacks in the network.
- Attacks that are misclassified by the IDS as normal instances are analysed by analysis engine to check anomalous events and will be relabeled by the system. Our Model is inline model should be placed at the network server to monitor all passing data packets and determine suspicious connections. Therefore, it can alarm the system administrator with the malicious attack type. Moreover, the system is capable of detecting new attacks and generates signatures for them.

10.6 IPS Using Open Source Tools

Implementation of Intrusion Protection Systems using open source tools is described below:

Due to space problems, we have considered very few attacks and their defense mechanisms. The implementation of IPS is divided into following process: Attack Generation algorithms, Defense Against Attack (Attack Prevention algorithms) , Attack Detection Algorithms. Some of the sample attack detection and prevention rules are discussed below:

1 Attack Generation algorithms

Packet Capture: We used TCP dump and window dump to capture the incoming flow of information and analyzed this traffic by using the proposed IDS. Attack Generation Process can use different tools like NMAP, Nessus, hping3 and Scapy to generate different kinds of trailer made packet to do the attack.

For **Attack Generation we can use the** following tools

Scapy(**http://www.scapy.org**), Nmap(**http://www.nmap.org**), Hping3**http://www.hping.org**)

1 Land Attack Generation

#hping3 –a –spoof -flood <src_ip> <dst_ip>

where a:spoof source address

flood: sent packets as fast as possible. Don't show replies.

src_ip : source ip address which is spoofed

dst_ip : destination ip address

2 XMAS Attack Generation:

Using the Hping #hping3 –c 1 –V –p 80 –s 5050 –M 0 –UPF 192.16.0.103

Where c: count V: command line switch for addition information about the packet

p : port no, s: source port, M: set the sequence

3 SYN Flood Attack Generation

Using the command: hping3 –S –fast –a <src_ip> <dest_ip>

where S : SYN packets are generated

fast : 10 packets per second

a:for spoofing option

src_ip : is a Source ip

4 XMAS Attack Generation

Using Scapy

#hping3 –c 1 –V –p 80 –s 5050 –M 0 –UPF 192.16.0.103

Where:

src :source ip ,

dst :destination ip

flags : FPU-FIN,PUSH,URGENT

count : no of packet to generate.

2 Attack Detection Algorithms

Various Ids are found at the following links :

(www.snort.org),SPADE(www.silicondefence.com/Spice_JCS.pdf,www.silicondefense.org), NIDES(www.nides.org), HONEYPOT(www.Honeydpot.org), KESENSOR(www.keyfocus.net/kfsensor), HONEYD(www.Honeyd.org), TRIPWIRE(www.tripwire.org)

Attack detection task is carried out through Snort as below :

1 ICMP Attacks Detection

If protocol:ICMP and type: Request

check if state[ipaddress] : active

else if state[ipaddress] :active and returncheck if lastpacket.time < 1 [1in 1sec] count[ipaddress]++

else cout[ipaddress] : 0 if count[ipaddress] > 25 [70 in 1sec]

reset count[ipaddress]:0 and lastpacket.time :0

set alarm flag

2 Smurf attack Detection:

Alert icmp $External_net any : $home_net any (msg:"icmp smurf attack detected"; dsize:4; icmp_id:0; icmp_seq:0; itype:8; classtype: attempted – recon; sid:78787878;)

1 ICMP Flood/Ping Flood Prevention Rule

INPUT iptable rule:

iptables :A INPUT :p ICMP s <src_ip> d <dst_ip> -j DROP

where A : append rules, p : set policy for the chain for the given target, s : source specifications, d : destination specifications

2 Smurf attack Prevention rule

$iptables -A INPUT p ICMP icmp type echo request m pkttype pkttype broadcast j DROP

Where A : append rules, p: set policy for the chain for the given target, s : source specifications, j : specifies the action if match found, d : destination specifications m -> match option

3 SYN Flood Attack Prevention Rule:

INPUT iptable rule: iptables -A INPUT -p tcp -d <dst_ip> --tcp-flags ALL SYN -j DROP

where, A -> append, rules,p -> set policy for the chain for the given target, j -> specifies the target of the rule(what to do if find a match), d -> destination specifications

4 Land Attack Prevention rule:

iptables -A INPUT -s 172.18.61.50/32 -j DROP

iptables -I INPUT -s ${my_ip} -d ${my_ip} -j DROP

5 XMAS Attack Prevention rule:

iptables -A INPUT -p tcp –tcp-flags ALL FIN,PSH,URG -j DROP

Where ALL = all flags set, FIN = Finish flag, PSH =Push flag, URG = Urgent flag will be set

6 Fraggle Attack Prevention rule:

$iptables A INPUT p UDP m pkttype pkt-type broadcast j DROP

$iptables A INPUT p UDP-m limit –limit 3/s j ACCEPT

Where A = append rules, p = set policy for the chain for the given target

s = source specifications, j = specifies the action if match found

d = destination specifications, m = match option

Chapter 11

Network Design

The increasing importance of network infrastructure and services along with the high cost and difficulty of designing enforce efficient and correct design of network. For beginners it is difficult to plan and design a network based on the concepts of TCP/IP protocol suite. This chapter focuses on practical approach to plan and design your own network which is explained with the help of few case studies. This chapter presents different network devices used in building a network and provides the details of developing an effective network plan for given scenario. Lastly case studies are provided to understand the concepts of devices and network building. This chapter also discusses various router attacks and defense mechanism for the router protection

11.1 Nerwork Devices

Each layer of TCP/IP protocol suite contains different devices in the network. These devices play different role and are euipped with different features to carry out their designated task. Following are the devices which are used for building a network. Lets have a quick look through.

1. Multistation Access Units (MAUs)

A multistation access unit (MAC) is a central hub that links token ring nodes into a topology that physically resemble a star but in which packet are transfers in a logical ring pattern.

MAU technology has evolved into new devices, such as the controlled access unit (CAU), which contains option to coonect several units (stackable) to count as one MAU on a token ring network. CAUs also come with option to gather information used in network performance management.

The most basic MAU connect up to eight single-wire segment. Newer MAUs have 16 port to connect nodes. A MAU can be a passive or an active hub. A **passive hub** does nothing more than pass the signal from workstation to workstation. Some of the signal strength is absorbed each time it goes through the MAU, reducing the

maximum transmission capability of the network. For example, a network using passive hub and Type 3 cable (UTP) has an actual limit of 72 nodes. An active hub regenerates, retimes, and amplifies the signal each time it passes through to the next node. That ensures delivery of a stronger signal to outlying nodes and more than double the number of nodes supported.

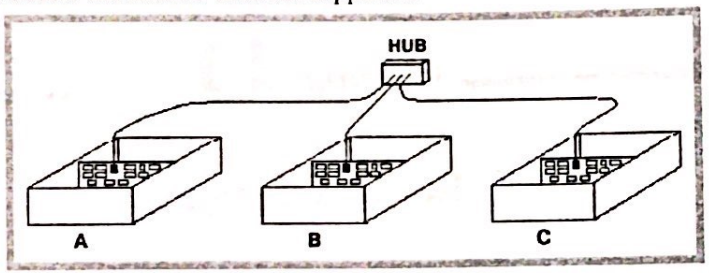


Figure 11.1 A Network Hub

- A **passive hub** connects nodes in a star topology, performing no signal enhancement as the packet moves from one nodes to the next through the hub. Each time the signal moves through the hub, it is weakened slightly because the hub absorbs some of the signal, reducing the total number of nodes that can be attached to a token ring network.
- A **active hub** connect nodes in a star topology, regenerating, retiming, and amplifying the data signal each time it passes through the hub. Using active hubs more than doubles the total number of nodes that can be connects to a token ring network.

2. Multiplexers or Switches

Multiplexers are network devices that can receive multiple input and transmit them to a shared network medium. Multiplexers simply are switches used in old and new technologies, such as the following:

- Telephone switching
- Switching telecommunications lines to create multiple channels on a single line (such as T1 lines)
- Serial communications to enable more than one terminal to communicate over a single line
- Fast Ethernet, X.25, ISDN, frame relay, ATM, and other networking technologies to create multiple communication channels over a single communications cable

Figure 11.2 Network Switch

A multiplexer is a switch that divides a communication medium into multiple channels so several nodes can communicate at the same time using ports. A signal that is multiplexed must be demultiplexed at the other end.

3. Repeaters

A **repeater** amplifies and retimes a packet-carrying signal so it can be sent along all cable segments. As used in this context, a segment of cable is one cable run within the IEEE specification, such as one run of 10BASE2 cable that is 185 meter log and that has 30 nodes or less.

4. Bridges

A bridge is a network device that connects one LAN segment to another. Bridges are used in the following circumstances:

- To extend a LAN when the maximum connection limit has been reached, such as the 30-node limit on an Ethernet segment
- To extend a LAN beyond the length limit, such as 185 meters for thin net Ethernet
- To segment LANs to reduce data traffic bottlenecks
- For security to prevent unauthorized access to a LAN

A **bridge** connect different LAN segment using the same access method, such as one Ethernet LAN to another Ethernet LAN or a token ring LAN to another token ring LAN.

5. Routers

A router performs functions similar to those of a bridge, such as learning, filtering, and forwarding. Unlike bridge, however, routers have built -in intelligence to direct frames to specific network, to study network traffic, and to adapt quickly to changes detected in the network. Router connects LANs at the network layer of the OSI model, which enables them to interpret more information from frame traffic than bridges. A router directing frame to a specific network rather than unnecessarily broadcasting that frame to all network connects to the router.

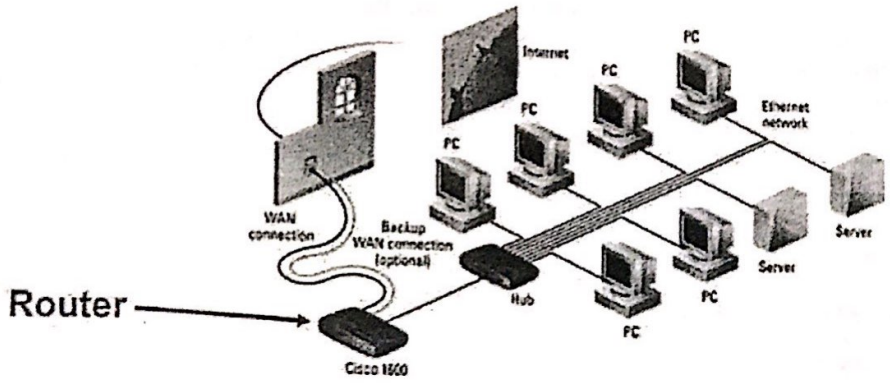


Figure 11.3 Router in a Network

In general, routers are used for the following:

- To efficiently direct packet packets from one network to another, reducing excessive traffic
- To join neighboring or distant networks.
- To prevent network bottlenecks by isolating portion of network.
- To connect dissimilar network.
- To secure portions of a network from intruders.

A router connects networks having the same or different access methods, such as Ethernet to token ring. It forwards packet to networks by using a decision making process based on routing table data, discovery of the most efficient routes, and preprogrammed information from the network administrator.

A **local router** is one that joins networks in the same building or between buildings in close proximity, for example, on the same business campus.

A **firewall** is software, hardware, or both employed to restrict who has access to a network, to specific network segment, or to certain network resources.

Brouter

A **brouter** is a network device that acts like a bridge in one circumstance and like a route in another. Brouter are used in the following situation:

- For efficient packet handling on the multi-protocol network with some protocols that can be routed and some that can not
- To isolate and direct network traffic to reduce congestion
- To join networks
- To secure a portion of a network, by controlling who can access it.

Organisation of Enterprise Network Architecture

An important part of documenting the current resource is to make a diagram of the existing network. If you are planning a new network, a network diagram is vital for visualizing the network and making improvements to the plan. Besides a diagram of the network, include the following information in your planning document:

- Number and kinds workstation
- Number and kinds of server and host computers
- Network topology
- Network communication media
- Types of network devices.
- Telecommunications services
- Current network performance statistics.

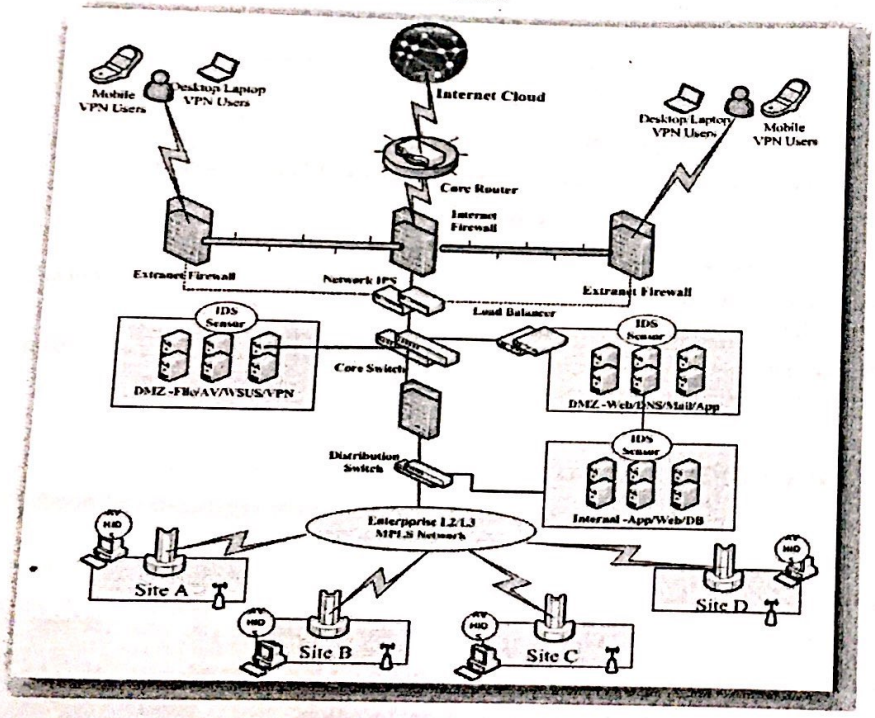


Figure 11.4 Organization of enterprise network and reference architecture

In order to understand the network and system security design and implementation challenges and task complexities from risk management standpoint, one has to look at the individual building blocks of an enterprise network, their configuration, present vulnerabilities and security threats as applicable. It would definitely help in designing better perimeter, communication as well as end point security defense mechanisms. In totality we shouldn't exclude the physical and environmental security controls. Organization network threat model varies from business to business hence the diverse data and information security requirements to be visited.

As depicted in the above figure 11.4, the network security hardware and software components for sample enterprise organization structure, along with defined adequate component level roles explicitly.

- Network Core routers
- Core and distribution switches
- Internet Firewall
- Extranet Firewall
- Internal Firewall
- Network IDPS
- Host IDPS
- IDS Sensors
- Load Balancers
- DMZ Servers (Web ,App,DNS,FTP,NTP,AD, Messaging Server etc.)
- Internal Servers
- Application servers
- End User Computing Infrastructure
- Remote Users
- VPN Concentrator

11.3 Case Study on Network Design

Planning and designing a network is explained with some real life case studies in this section.

11.3.1 Implementing A Network for an Engineering College Campus

Suppose you are given task of designing a campus network for an engineering college that have five core departments located in different buildings which is never networked before. Here we are connecting two networks public network and private

network. Public network is the outside network or Internet. Private network is within the oragnization network. The campus network connection architecture is shown in Figure 11.5.

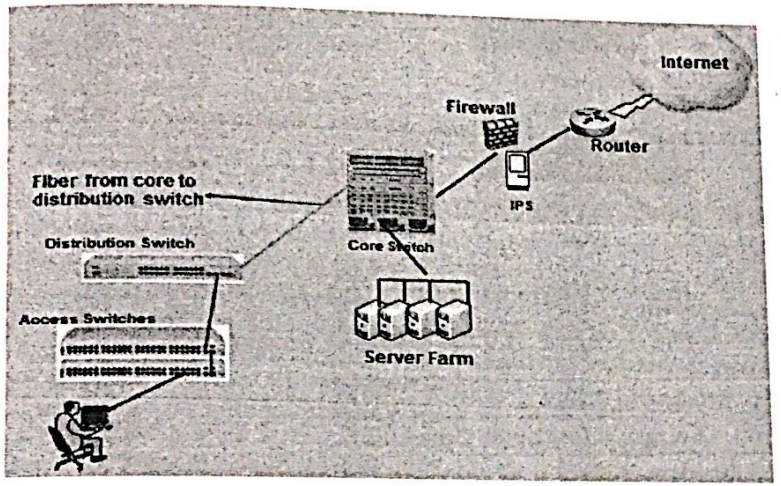


Figure 11.5 Network Devices

Router, IPS and firewall is required to connect outside network. Here three types of switches will play an important role namely: Core switch, Distribution switch and Access switch. Core switch is used to connect to the different servers of the architecture such as E-mail server, Application Server or Web Server. Distribution switch is connected to the core switch and divides the network in five departments of campus. As the number of departments increase the number of distribution switches will increase. Distribution switches are then connected to access or edge switches which provides floorwise connection of network. The Edge switches connects the computers of one department. The switch connections are shown in figure 11.6.

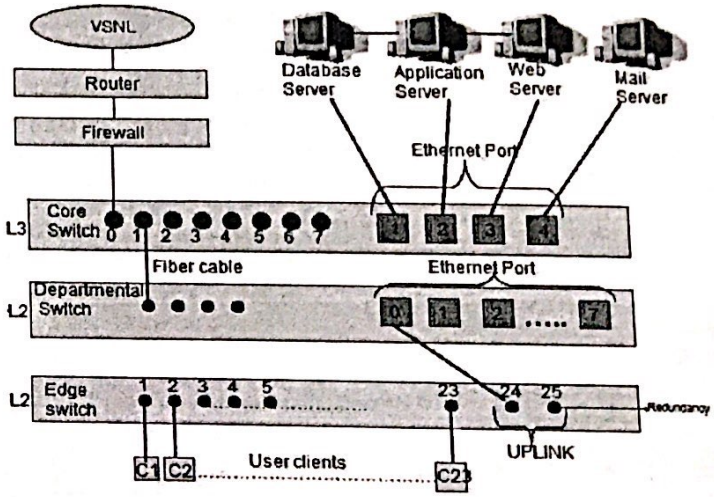


Figure 11.6 Network Design Diagram

As seen from Figure 11.6 servers such as application server, database server and web server are connected to core switch through ethernet ports. The departmental switches are connected to core switch. The edge or access switches are connected to department switch through ethernet ports. The clients or computers are connected through edge or access switches.

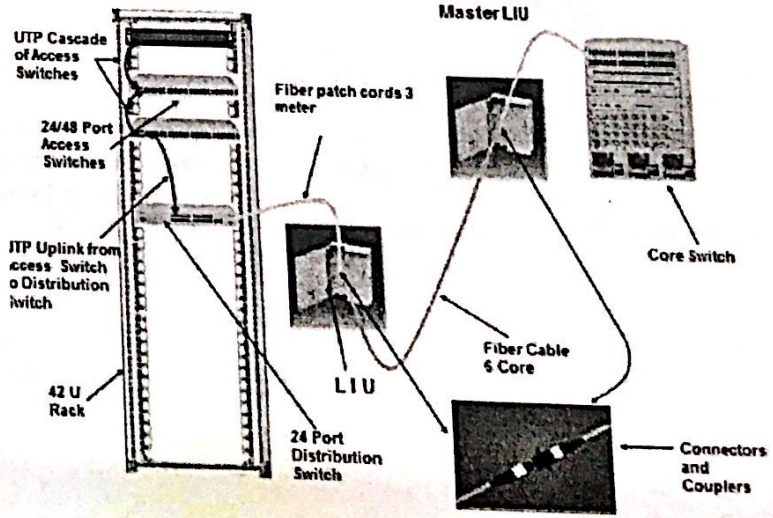


Figure 11.7 Building Plan

As seen from Figure 11.7 building plan consists of switch rack. In switch rack one distribution switch of department and access switches according to number of floors are fixed. If building have three floors then three access switches are required. Connector such as LIU are used. Fiber patch cords are used to connect two diffferent switches.

Access switches connects the computers in the Lab in network design. One access or edge switch can connect 24 computers and one patch panel. As seen from figure 11.8 patch panels are connected to access switch through RJ connectors and pacth cords. Cat6 cable and RJ connectors can be used to connect from patch pannel to the computer

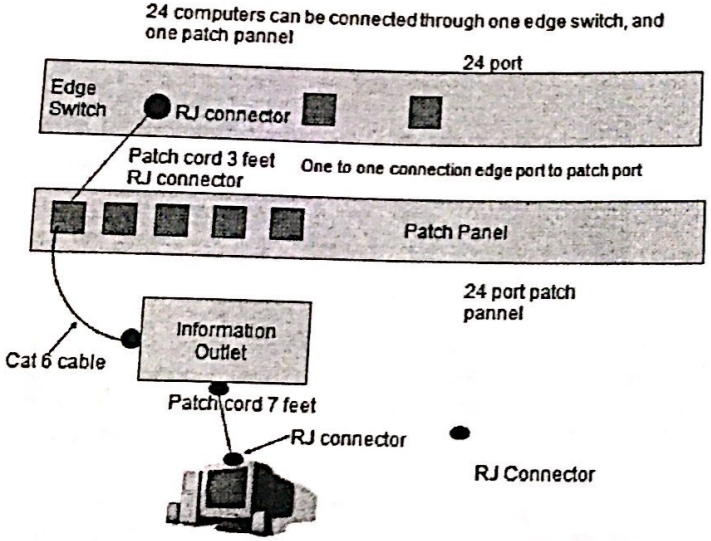


Figure 11.8 Floor Plan

The network design explained above shows the network plan for an engineering campus having different departments. Here simple network design with basic devices is the intention. However different devices and different connectors can be used to design the same network. The other case study is explained in next section.

User Connectivity from User to Access Switches is shown in figure 11.8

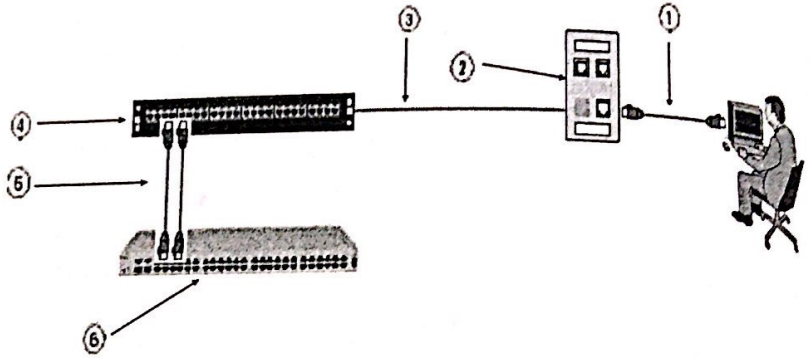


Figure 11.9 Connectivity from User to Access Switches

The Number shown in the figure 11.9 has following meaning:

1.	Cat 6, Patch cord 7 Feet
2.	Cat 6 UTP Single Information Outlet
3.	Cat 6 Ethernet Cable
4.	Cat 6, 24 Port Patch Panel
5.	Cat 6, Patch cord 3 Feet
6.	Access switch with 48-port 10/100

User Connectivity to Patch Panel is shown in figure 11.10

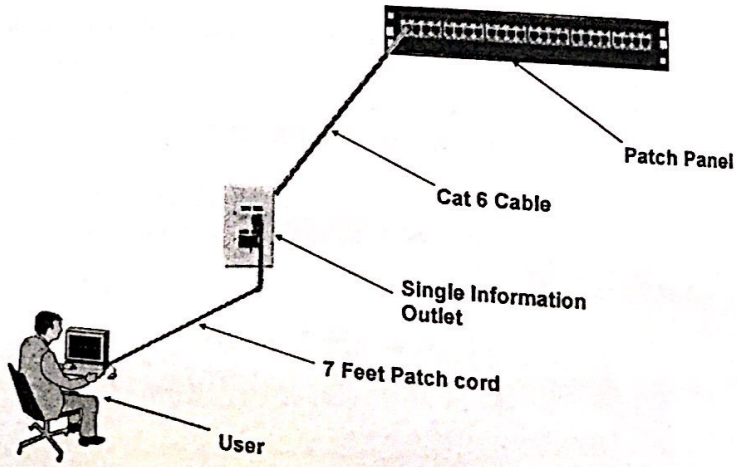


Figure 11.10 User Connectivity to Patch Panel

11.3.2 Implementing A Network on A Single Floor of A Building

Assume you are working with a insurance company that employs 50 people in a one story building. The building is 15 year old and has never been networked, although each employee has a desktop PC. There is no central computer; office member carry floppy disks to one another when they want to share files. Each insurance representative keeps individual client records on his or her own PC. Information for billing is carried by floppy disk to an office assistant, who compiles the information for all representatives and sends out bills. The company has decided to install a network and has hired you as a consultant.

In this situation a Windows NT file server would enable to manage network resource centrally from the server, such as files, user account, printer and security. Network traffic also can be monitored through the server. An Ethernet physical star, logical bus network would give option to grow while reducing the cost of network management by centralizing network communication. Problems with a node can be quickly identified and fixed through such a central design and network segments isolated so problems on one segment do not take down the entire networking

The physical connectivity can be accomplished by installing category 5 UTP cable connected to stackable hubs. Because the firm does not plan to transmit large files or graph ices at this time, a10BASE-T network with 10 Mbps hubs is likely to be sufficient. Also, combination 10 Mbps and 100 Mbps NICs are a good choice, providing easy conversion to a100BASE-T network in the future, if needed

11.4 Network Configurations

For the configuration requirements of the organisation you can use cisco devices or nortel devices based on your budget, how ever you can procure the latest devices available on the respective sites.

Sr. No.	Cisco Devices	Nortel Devices
1	Model: Cisco Catalyst 4500 Series Switch(4507R) 7-slot- core switch Chasis based with 24 port 1000 BaseT and 12 GBIC fiber port with 2 X1GB uplink port with passive backplane of 64 Gbps with 48 Mpps	Model: Nortel Routing S/w 8000 Series 10-slot-core Chasis based switch with 24 port 1000 BaseT and 16 GBIC fiber port with 2 X1GB uplink port with passive backplane of 320 Gbps with 96 Mpps.

2	Model: Cisco Catalyst 3560 Series Switch (3560G-24TS) Distribution switch with at least 24 port with POE-1000 BaseT port and 4 fiber port for Fiber Uplink.	Model: Nortel 5520 24T PWR Series Distribution switch with atleast 24 port with POE-1000 BaseT port and 4 fiber port for Fiber Uplink. 80 Gbps Stacking.
3	Model: Cisco Catalyst 2950 Series Switch (2950SX 24p) Access switch with 24-10/100 BaseT with 2 inbuilt SX port. 8.8 Gbps switching fabric	Model: Nortel 425 24T Series Access switch with 24-10/100 BaseT with 2 inbuilt SX port. 16 Gbps switching fabric
4	Model: Cisco Catalyst 2950 Series Switch (2950SX 48p) Access switch with 48-10/100 BaseT with 2 inbuilt SX port. 13.6 Gbps Switching Fabric	Model: Nortel 425 48T Series Access switch with 48-10/100 BaseT with 2 inbuilt SX port. 16 Gbps Switching Fabric.
5	Model: Cisco Catalyst 2960 Series Switch (2960 24TT) Access switch with 24-10/100 BaseT with 2 inbuilt 1000BaseT port	Model: Nortel 425 24T Series Access switch with 24-10/100 BaseT with 2 combo 10/100/1000BaseT port
6	Model: a. Cisco 1300 Series b. Cisco 1100 Series c. Cisco P&OD	Model: a. Nortel 2300 b. Nortel 2300 c. Nortel Make
7	Model: Cisco ASA 5500	Model: Nortel 5000 Series Firewall
8	Model: Cisco 2821 series router	Model: Nortel 5000 Series Router
9	Cisco LMS v2.5	Brand Nortel
10	Model: Hp Proliant NAS	Model: Tanberg

11.5 Router Attacks and Defence Mechanism

louter Attacks are as follows:

- Distributed Denial Of Service Attacks
- Man In The Middle Attack
- Attack On BGP
- Attacks On OSPF

11.5.1 Router Attacks

A. Distributed Denial of Service Attacks

Some of the DDoS attacks that can happen on a router are as follows:

1) Ping of Death

The typical Internet connection (dial-up, Ethernet, cable-modem, etc.) only supports packets of around a couple thousand bytes, but IP supports packets up to 64-kbytes. Thus, when sending a single packet that is too large for a link, it is broken up into smaller packet fragments.

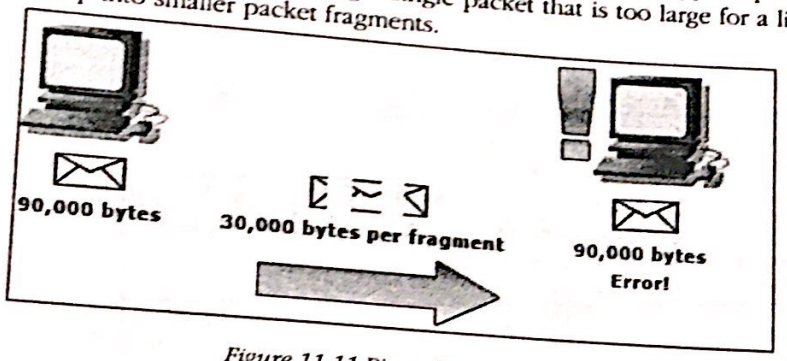


Figure 11.11 Ping of Death Attack

While a single packet cannot exceed 65536-bytes, the fragments themselves can add up to more than that. The "Ping of Death" technique does just that. Since this is a condition thought impossible, operating systems crash when they receive this data.

Generation of Ping of Death attack: Ping of death can actually be run from older versions of Windows.

1. Open command line
2. Type: **ping –l size victim's_ip_address**

example: **ping –l 90000 10.0.0.1**

where

-l 90000: Send buffer size.

A further bug in Windows is that it not only crashes when it receives the invalid data, but it can accidentally also generate it.

2) Smurf Attack

In a smurf attack a computer is used to send ping flood to the Victim which clogs the bandwidth of the victim and ultimately leads to DDOS. Smurf Attacks take the advantage of Aplifiers. When a Attacker send a single packet to the amplifier it responds back with hunderds of packets maybe thousands. Assume this is the IP range of a client of a ISP 1.1.1.1 - 1.1.1.255 and when you ping a single IP address like this

ping 1.1.1.50

This will result in a single machine replying, so you get a single packet back. but what if you ping like this (Suppose 1.1.1.255 is the Directed Broadcast System)

ping 1.1.1.255

then it reply's with many responses because it sends a single packet to every single host in the network, and which ultimately results in Amlifying.

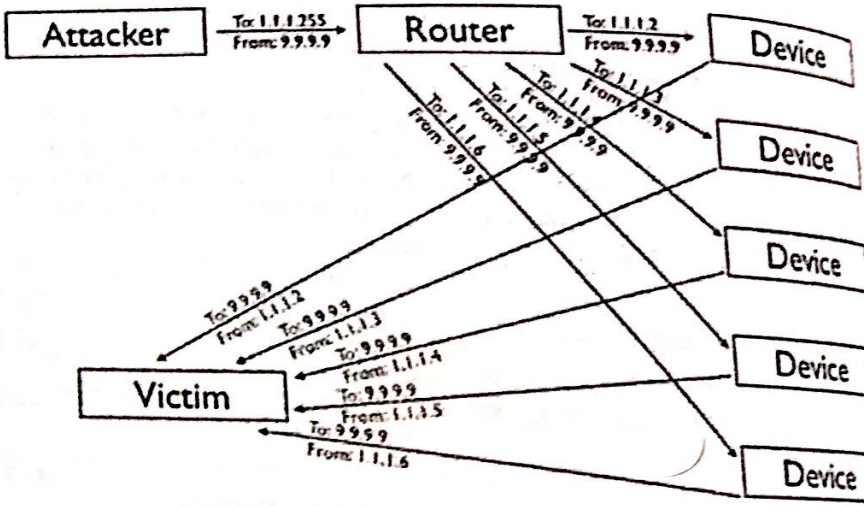


Figure 11.12 Smurf Attack

So this is how thousands of requests can be generated with just a single packe mechanism is known as Directed broadcast. So a single packet is broadca whole of the subnet and all the systems respond to it and generate a lot o Attacker spoofs the IP address of the victim, which leads to thousands of r replying back to the victim, but the victim has'nt pinged them but he is being So this is how smurf attack can cause lot of traffic.

Generation of Smurf attack

1. Open command line
2. Type: **hping3 -1 –flood -a VICTIM_IP BROADCAST_ADDRESS**

example: **hping3 -1 –flood -a 1.1.1.1 1.1.1.255**

where

-1: ICMP mode

--flood: send packets as fast as possible. Don't show replies.

-a: spoof spoof source address

3) Ping Flood Attack

A ping flood is a simple DoS attack where the attacker overwhelms the victim with ICMP Echo Request (ping) packets. It only succeeds if the attacker has more bandwidth than the victim (for instance an attacker with a DSL line and the victim on a dial-up modem). [14]

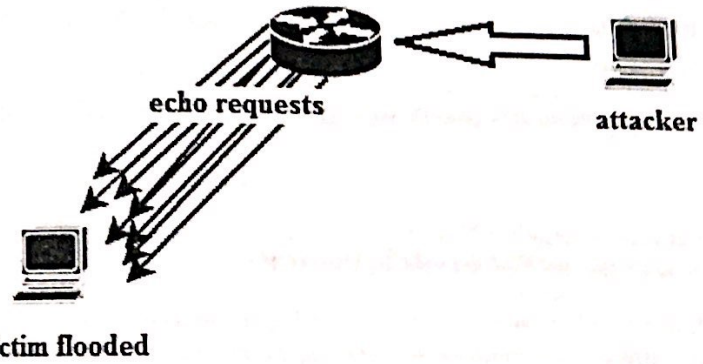


Figure 11.13 Ping Flood Attack

Generation of Ping Flood attack

1. Open command line
2. Type: **ping victim's IP address –l size –n count –w waiting time**

example: **ping 10.0.0.1 -l 65500 -n 10000000 -w 0.00001**

where

-l 65500: Send buffer size.

-n 10000000: Number of echo requests to send.

-w 0.00001: Timeout in milliseconds to wait for each reply.

4) ARP (Address Resolution Protocol) poisoning

Here the attacker continuously looks for ARP request packet in the network. Once it finds a request packet, it quickly forms a packet with a wrong MAC address and sends it as reply. So a wrong mapping will take place in ARP table which is referred to as ARP poisoning. So now the actual MAC will be denied of any further service.

Generation of ARP poisoning

1. The attacker should be in the network as the victim. The attacker will sniff the network to find the request packet.
2. Now the attacker can run the **code which sends the following ARP reply packet to the victim.**

Hardware type (layer 2)		Protocol type (layer 3)
Address length layer 2 (n)	Address length layer 3 (m)	Operation
Source address (layer 2) = Wrong MAC		
Source address (layer 3) = Spoofed address		
Destination address (layer 2) = Victim's MAC		
Destination address (layer 3) = Victim's IP		

Figure 11.14 ARP Poisoning Packet

- The source address(layer 3) of this packet is spoofed
- The source address(layer 2) of this packet is that wrong MAC filled in by the attacker
- The destination address(layer 3) of this packet is that of the victim
- The destination address(layer 2) of this packet is that of the victim

The attacker can run the following code which sends the above ARP reply packet the victim

```
byte[] a = new byte[14];
Arrays.fill(a, (byte) 0xff);
ByteBuffer b = ByteBuffer.wrap(a);
if (pcap.sendPacket(b) != Pcap.OK)
{
        System.err.println(pcap.getErr());
}
pcap.close();
```

B. Man in the Middle Attack

The attacker manages to intercept the data flowing from a source to a destination. The attacker can simply read the data or even modify the data.

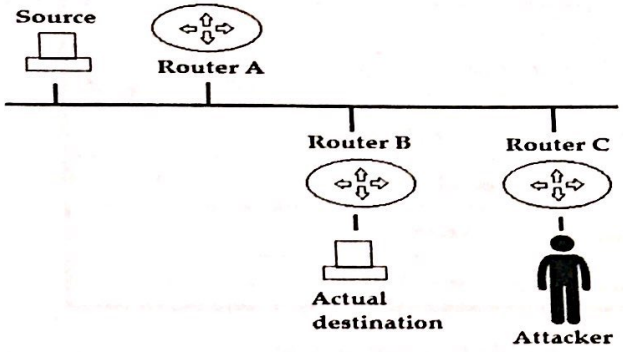


Figure 11.15 ICMP Redirect Attack

Such an attack can be carried out in many ways. One of them is using – ICMP redirect for a router. ICMP redirect packets are actually sent by routers to hosts if they find a better path to reach a destination. If a router receives a packet and forwards the packet to the same interface where it had received then an ICMP redirect can be sent. But an attacker can carry a man in middle attack using redirects. In Fig, the attacker is on the same subnet as the victim. The attacker will send an ICMP redirect which will create a new entry in source's routing table. This entry will have router C as next hop for reaching the actual destination. Thus the attacker successfully intercepted the connection.

Generation of ICMP redirect attack

1. Assume the following

Attacker IP Address: 172.16.235.99

Legitimate Router Gateway: 172.16.235.1

Victim IP Address: 172.16.235.100

We can use ICMP redirect host to insert a new route table entry for the 10.1.1.1 address as follows:

hping -I eth-dest -C 5 -K 1 -a 172.16.235.1 –icmp-ipdst 10.1.1.1 –icmp-gw 172.16.235.99 –icmp-ipsrc 172.16.235.100

where

-I eth-dest is the destination ethernet interface on the attacker to send the packets out of/from.

-a is the spoofed source address of the legit. router gateway

--icmp-ipdst is the new route table entry address you want to create

--icmp-gw is the new route destination address/gateway you want to create and must live within the same subnet as the victim.

--icmp-ipsrc must match the source address of the victim to pass sanity checking

C. Attack on BGP

TCP reset attack is the attack in which a TCP connection is terminated by the attacker using a spoofed packet with the RST (reset) bit in the TCP packet set.

To carry out this attack, the attacker simply sniffs the TCP connection to get the source IP address, source port number, destination IP address, destination port number and most importantly the ongoing sequence number. Now the attacker creates a fake TCP packet with proper source IP and port and destination IP and port. The sequence number is also filled appropriately. The RST bit in this packet is set. When this packet reaches destination, it sees that the RST bit is set and hence it terminates the connection. So the continuity is disrupted until an entire new TCP session is established. This is certainly not desirable.

The severity of this attack varies from application to application. For example, Cisco's BGP protocol is highly affected by this attack. BGP is the protocol that runs in the service provider architecture and has to manage huge routing tables. Whenever a route goes down, BGP does a lot of processing over many routers to fix the problem. In this

case if the TCP attack is carried out, BGP will do a lot of processing because of the terminated connection. If a connection is terminated frequently, BGP also gradually isolates that router from the network because it is held responsible for causing a lot of processing frequently over many routers. So a harmless router can be isolated because of such an attack.

Generation of TCP reset attack

1. The attacker should be in the network as the victim. The attacker will sniff the network to get the ongoing sequence number of the TCP connection
2. Now the attacker can run the following **code which sends a TCP reset packet to the victim.**

```
JPacket packet =
  new JMemoryPacket(JProtocol.ETHERNET_ID,
      " 001801bf 6adc0025 4bb7afec 08004500 "
    + " 0041a983 40004006 d69ac0a8 00342f8c "
    + " ca30c3ef 008f2e80 11f52ea8 4b578018 "
    + " ffffa6ea 00000101 080a152e ef03002a "
    + " 2c943538 322e3430 204e4f4f 500d0a");

Ip4 ip = packet.getHeader(new Ip4());
Tcp tcp = packet.getHeader(new Tcp());

tcp.destination(80);

ip.checksum(ip.calculateChecksum());
tcp.checksum(tcp.calculateChecksum());
packet.scan(Ethernet.ID);
```

Figure 11.16 Send TCP Packet

- The source address of this packet is spoofed.
- The destination address is the victim's address.
- The sequence number is in line with the ongoing sequence number
- the Reset bit is set

D. Attacks on OSPF

1) Hello packets dropped

OSPF neighbors are formed by exchanging hello packets. These hello are not acknowledged by the other end. When OSPF misses certain hello packets, the neighbor is considered as dead. This depends on dead timer of OSPF. An attacker can purposely delete some OSPF packets. This will cause the neighbor to be declared as dead.

Generation of Hello packets dropped attack

The attacker takes control of one of the routers between the victims and tells it to deliberately drop some packets.

Attacker configures following ACL on the victim router:

access-list 100 deny ospf any any

2) Max Sequence attack

OSPF sends LSAs (Link State Update) to exchange routing information with their neighbors. LSA contains a sequence number which helps the router determine as to which one is the freshest route. An attacker can send LSA containing the max sequence number which is 0x7FFFFFFF. Thus all routers will accept this as the freshest update. This update will stay in the LSDB (Link Sate Database) for one hour thus helping the attacker to harm the network within that period.

Generation of Max Sequence attack

1. The attacker should be in the network as the victim.
2. Now the attacker can run **code which sends the following OSPF LSA packet with sequence number 0x7FFFFFFF to the victim.**

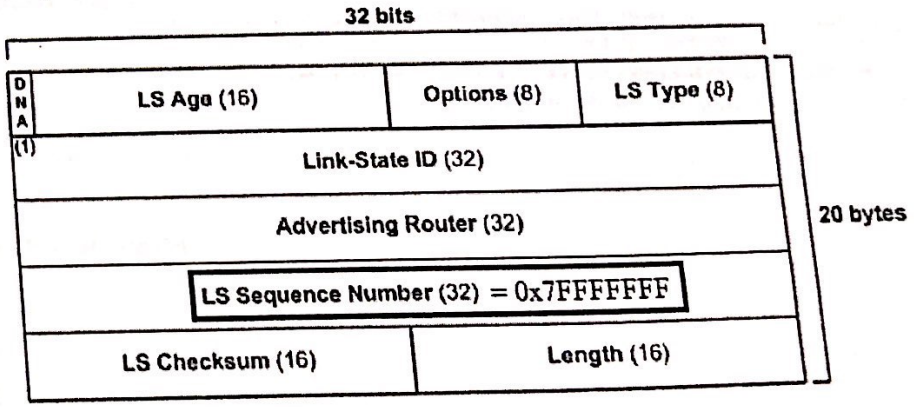


Figure 11.17 OSPF LSA Packet

3) Unknown Logins

Attackers who do not have direct physical access to a router can crack the routers telnet password and log into the router and reconfigure it which can make the router act maliciously.

Generation of unknown logins

1. Hack the victim router's password
2. Telnet into the router (port: 23)

This is illustrated in detail in the following section.

11.5.2 Router Attacks Detection and Protection Mechanism

This section will focus on various attacks and how these attacks will be detected successfully by router log analysis: Attacks detection. Some attacks can be detected by just analyzing one log entry such as BGP's session termination attack or ICMP redirect attack. On the other hand, some attacks require analyzing more than 1 line before actually declaring that an attack has happened.

Some of the methods /algorithms used for detection of router attacks are given below:

1) Port scan attack

An attacker can run a port scan on the router to see which ports are ON and which are not. This is mostly the first step for an attack. Loopholes can found after a port scan.

Log entry Generation of port scan attack: To generate log entry for each TCP packet sent or received, the

'debug IP TCP packet'

debugging command can be used. This will enable logging for TCP events.

Log entry format of port scan attack: When a port scan happens on a router, the log entries which are generated are shown in Figure 11.18


```
May 14 10:58:25.627: tcp0: I LISTEN 10.0.0.100:1495
10.0.0.1:1 seq 1473192529

May 14 10:58:25.791: tcp0: I LISTEN 10.0.0.100:1496
10.0.0.1:2 seq 4232257361

May 14 10:58:25.883: tcp0: I LISTEN 10.0.0.100:1497
10.0.0.1:3 seq 452016969

May 14 10:58:26.007: tcp0: I LISTEN 10.0.0.100:1498
10.0.0.1:4 seq 3164921931

May 14 10:58:26.087: tcp0: I LISTEN 10.0.0.100:1499
10.0.0.1:5 seq 2912730653
```

Figure 11.18 Log Entries During Port Scan Attack

Detection mechanism of port scan attack: The source and destination IP addres will be same everywhere. The destination ports will be different. A threshold ca be maintained by the algorithm which will tell how many packets to scan befo announcing a port scan attack. This threshold might be 10, 15 as stated by the netwo administrator. The algorithm for detecting port scan attack is as given below;

PORT_SCAN_DETECTION ALGORITHM:

Input:

line //syslog line

Output:

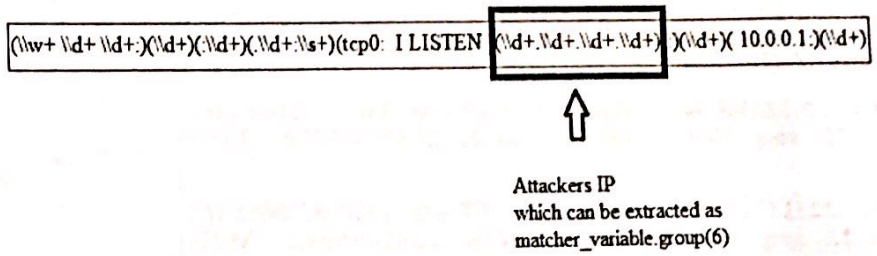
alert //port scan attack alert

1. Write the pattern for matching the TCP listen log entry. The source and destination ip will be hardcoded here. The destination port will be a variable.
2. If the pattern is matched
   a. Extract the time into *time* variable.
   b. if *flag* is equal to 0
      i. initialize *counter* to 0
      ii. copy the *time* into *timestamp*
      iii. set *flag* equal to 1

      else
      i. increment the counter
   c. if counter > *threshold* and *display* is equal to 0
      i. announce port scan attack
      ii. set *display* equal to 1
   d. if current time is *timestamp*+120 seconds
      i. set *flag* and *display* equal to 0

The display variable is used to prevent the announcement of attack more than once for the same attack. The flag variable is used to start a fresh new scan for the attack. Distributed Denial of Service (DDoS) attack on the router can also be detected in the similar fashion.

The regular expression pattern is as shown below to know the attackers IP:



Defense mechanism of port scan attack

Configure the following IP ACL

```
Router>en
Router#conf t
Router(config)#ip access-list standard port_scan
Router(config-std-nacl)#deny attackers_ip 0.0.0.0
Router(config-std-nacl)#exit
Router(config)#interface f1/0
Router(config-if)#ip access-group port_scan in
```

2) Unknown login attack

Some attacker can somehow get router's access via telnet over the network and try to do malicious activity inside the outer. Telnet attempts on the router can be detected by log analysis.

Log entry Generation of unknown login attack: To generate log entry for each TCP packet sent or received, the

'debug IP TCP packet'

debugging command can be used. This will enable logging for TCP events.

Log entry format of unknown login attack :The log entry after telnet attempt is shown in Figure 11.19.

```
May 13 22:15:15.915: tcp2: I ESTAB 10.0.0.100:2466
10.0.0.1:23 seq 3097509464 ACK 2052496558 WIN 17440

May 13 22:15:15.931: tcp2: O ESTAB 10.0.0.100:2466
10.0.0.1:23 seq 2052496577 DATA 31 ACK 3097509464
PSH WIN 4089

May 13 22:15:15.943: tcp2: I ESTAB 10.0.0.100:2466
10.0.0.1:23 seq 3097509464 ACK 2052496560 WIN
17438

May 13 22:15:15.951: tcp2: I ESTAB 10.0.0.100:2466
10.0.0.1:23 seq 3097509464 ACK 2052496562 WIN
17436
```

Figure 11.19 Log Entries After a Telnet Attempt into the Router

Detection mechanism of unknown login attack: Use the following regular expression to detect unknown telnet attempt

```
(\\w+ \\d+ \\d+:\\d+:\\d+)(.\\d+: \\w+)(\\d+)(: I SYNRCVD \\d+.\\d+.\\d+.\\d+(:\\d+ (10.0.0.1)(:23)
```

Attackers IP which will be extracted as matcher_variablegroup(5)

Defense mechanism of unknown login attack : Configure the following IP ACL

```
Router>en
Router#conf t
Router(config)#ip access-list standard unknwn_login
Router(config-std-nacl)#deny attacker's IP 0.0.0.0
Router(config-std-nacl)#exit
Router(config)#interface f1/0
Router(config-if)#ip access-group unknwn_login in
```


3) ICMP redirect attack

ICMP redirect packets are actually sent by routers to hosts if they find a better path to reach a destination. If a router receives a packet and forwards the packet to the same interface where it had received then an ICMP redirect can be sent. But an attacker can carry a man in middle attack using redirects.

Log entry generation of ICMP redirect attack: To generate log entry for ICMP packets, the

'debug IP ICMP'

debugging command can be used. This will enable logging for ICMP events.

Log entry format of ICMP redirect attack: The log entry after ICMP redirect attack is shown below

```
May 13 23:25:07.938: ICMP: redirect sent to 10.0.0.1 for
dest 172.16.1.111 use gw 172.21.80.23
```

Detection of ICMP redirect attack: Use the following regular expression to detect ICMP redirect attack

```
(ICMP: redirect sent to 10.0.0.1 for dest )(\\d+.\\d+.\\d+.\\d+)( use gw )(\\d+.\\d+.\\d+.\\d+)
```

new destination IP [gateway]

Defense mechanism of ICMP redirect attack; Remove the redirected route from routing table using following command

clear IP route destination IP [gateway]

The original (not redirected) route will now be learnt by the router through its running routing protocol.

4) BGP session termination attack

TCP reset attack is the attack in which a TCP connection is terminated by the attacker using a spoofed packet with the RST (reset) bit in the TCP packet set. This attack affects BGP protocol.

Log entry Generation of BGP session termination attack: To generate log entry for each BGP event, the

'debug IP bgp'

debugging command can be used. This will enable logging for BGP events.

Log entry format of BGP session termination attack :The log entry after BGP session termination attack is shown below

May 13 22:48:42.515: TCP: sent RST to 10.0.0.100:100 from 10.0.0.1:

Detection of BGP session termination attack: Use the following regular expression to detect BGP session termination attack

(TCP: sent RST to)(\\d+.\\d+.\\d+.\\d+)(:\\d+ from)(\\d+.\\d+.\\d+.\\d+)

Attacker's IP which can be extracted as matcher_variable.group(2)

Defense mechanism of BGP session termination attack ;Configure the following IP ACL

```
Router>en
Router#conf t
Router(config)#ip access-list standard unknwn_login
Router(config-std-nacl)#deny attacker's IP 0.0.0.0
Router(config-std-nacl)#exit
Router(config)#interface f1/0
Router(config-if)#ip access-group unknwn_login in
```

5) OSPF hello packet deletion attack

OSPF neighbors exchange hello packets every 10 seconds (Default hello timer is 10 seconds.) When 4 consecutive hello packets are missed by an OSPF process, then the OSPF process declares its neighbor as dead (Default dead timer is 40 seconds.) When a neighbor is dead, its entries will be flushed from the routing table. Attacker

can purposely delete OSPF hello packets. After 4 consecutive message deletion, the neighbor ship will break. This is an attack which has caused the OSPF neighbor ship to break resulting into flushing of its OSPF entries.

Log entry Generation of OSPF hello packet deletion attack: To generate log entry for each OSPF hello packet sent or received, the

'debug IP ospf events'

debugging command can be used. This will enable logging for OSPF events.

Log entry format of OSPF hello packet deletion attack: The log entry for the hello packet is as shown in Fig

May 13 22:40:09.455: OSPF: Send hello to 224.0.0.5 area 0 on FastEthernet1/0 from 10.0.0.1

Detection mechanism of OSPF hello packet deletion attack: The algorithm for detecting OSPF hello packet deletion attack is as given below:

HELLO_DELETION_DETECTION ALGORITHM

Input:

line //syslog line

Output:

alert //OSPF hello packet deletion attack alert

1. Write the pattern for matching the OSPF hello log entry.
2. Extract the seconds field of the time into *seconds_time*
3. When the first match occurs, copy *seconds_time* into *init_hello_time*.
4. Create an array *times* of size 6 of all possible seconds_time.
5. Match every new hello log entry *seconds_time* with *times[i]*. If the values are not equal then
   a. Calculate number of hello missed using modular arithmetic method within the *times* array.

6) OSPF DR BDR null attack

OSPF is a victim of DR, BDR null attack. OSPF elects DR, BDR on a multi access network. DR, BDR are elected based upon the priority and router ID (highest loopback address) sent in the hello message. After the election is done, the elected DR, BDR are sent in the hello message. An attacker can create a phantom router with highest priority and ID. Attacker will now set DR, BDR to null and then send that hello message. This will force reelection for DR, BDR and will elect the phantom router ad DR which will create undesirable effect.

Log entry generation of OSPF DR BDR null attack: To generate log entry for each OSPF event, the

'debug IP ospf events'

debugging command can be used. This will enable logging for OSPF events.

Log entry format of OSPF DR BDR null attack :The raw log after a DR, BDR null attack is shown in Figure 11.20.

```
May 13 20:02:36.447: OSPF: Interface FastEthernet1/0 going Up
May 13 20:02:36.447: OSPF: Send hello to 224.0.0.5 area 0 on FastEthernet1/0 from 10.0.0.1
May 13 20:02:46.451: OSPF: Send hello to 224.0.0.5 area 0 on FastEthernet1/0 from 10.0.0.1
May 13 20:02:54.699: %SYS-5-CONFIG_I: Configured from console by console
May 13 20:02:56.455: OSPF: Send hello to 224.0.0.5 area 0 on FastEthernet1/0 from 10.0.0.1
May 13 20:03:00.095: ICMP: echo reply rcvd, src 10.0.0.100, dst 10.0.0.1
May 13 20:03:00.163: ICMP: echo reply rcvd, src 10.0.0.100, dst 10.0.0.1
May 13 20:03:00.163: TCP: sent RST to 10.0.0.100:10230 from 10.0.0.1:23
May 13 22:49:44.134:      DR: none   BDR: none
May 13 22:49:44.122: OSPF: DR/BDR election on FastEthernet1/0
May 13 22:49:44.126: OSPF: Elect BDR 10.0.0.1
May 13 22:49:44.126: OSPF: Elect DR 200.0.0.1
May 13 22:49:44.134:      DR: 200.0.0.1 (Id)   BDR: 10.0.0.1
```

Figure 11.20 Raw Log After a DR, BDR Null Attack

Detection mechansim of OSPF DR BDR null attack: Use the following regular expression to detect DR BDR null attack

```
(\\w+ \\d+ \\d+:\\d+:\\d+)(.\\d+:\\s+)(DR: )(\\d+.\\d+.\\d+.\\d+|none)( )"+open_br+"(Id)"+close_br+"(
)(BDR: )(\\d+.\\d+.\\d+.\\d+|none)
```


where

```
open_br=Pattern.quote(open_br);
close_br=Pattern.quote(close_br);
```

Defense Mechanisms for Router Attacks

1 Defense In The Data Plane of Networks Using Hardware based monitoring
2 Detecting Disruptive Routers Using Counters
3 Locating Network Domain Entry And Exit Point/Path For DDOS Attack Traffic
4 Network Intrusion Prevention By Configuring ACLS On Routers Based On Snort IDS Alerts

1 Defense in the Data Plane of Networks Using Hardware Based Monitoring

Attackers most of the time attack the router's software and make the router to behave in a malicious way. Even if you try to detect the malicious activity by software, there may be fair chances that the monitoring software is also successfully compromised by the attacker. To overcome this problem, hardware based monitoring can be done. A router's main core software is the packet processing unit. Nowadays a single router has more than one unit for faster packet processing. This processing unit is complete software which can be compromised by an attacker. So an extra independent hardware is embedded into the router which can do monitoring to detect the software attacks. The hardware obviously can not be compromised by the attacks meant for the software. Now a typical router with packet processing units looks as shown Figure 11.21:

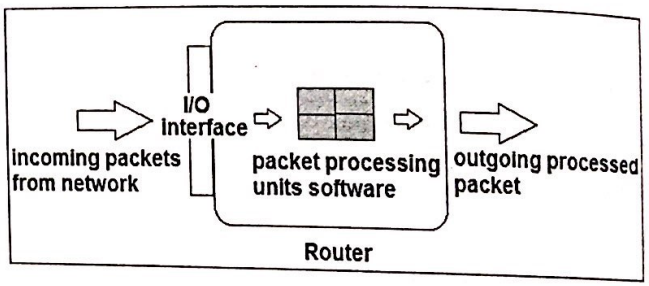


Figure 11.21 Router with Processing Units

The hardware modules which will do the monitoring will act upon the processing units.

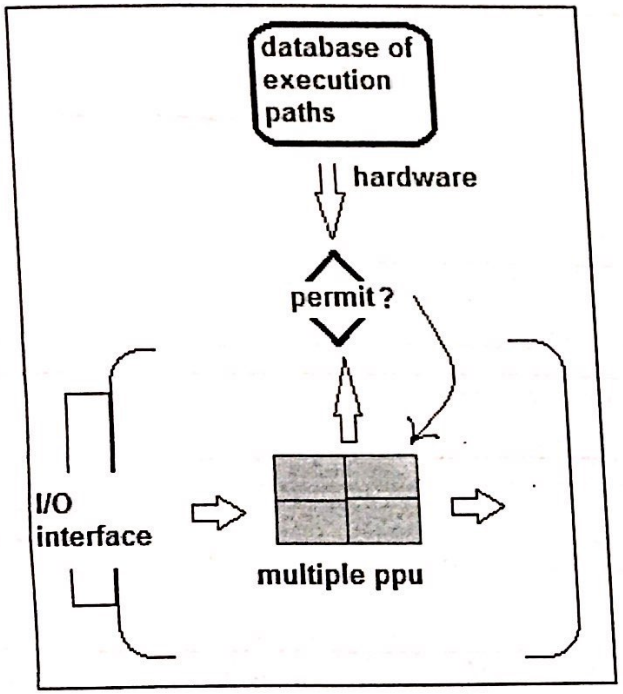


Figure 11.22 Hardware Monitoring Module

The hardware module will learn all the execution paths that can result out of a normal functioning router. If any anomaly is detected then it will detect an attack and drop the current packet and bring the processing unit back to fresh new state.

2 Detecting Disruptive Routers Using Counters

You can use counters to detect disruptive routers. Sometimes just by maintaining a few counters you can find which router is the one who is behaving abnormally in a network. Mostly only 6 counters are enough.

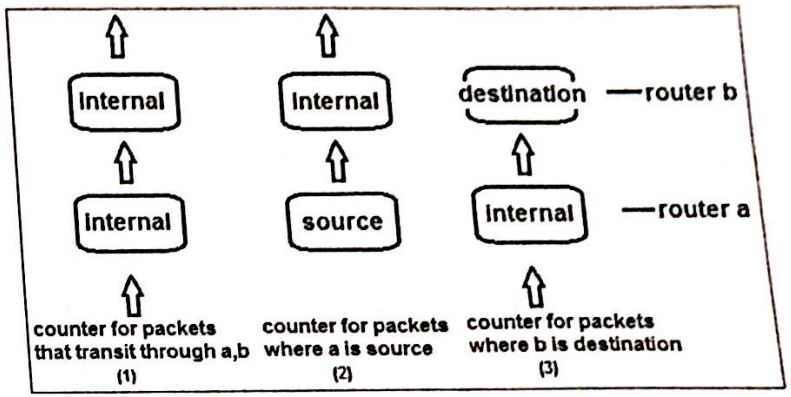


Figure 11.23 Counters Maintained by Router

Figure 11.23. shows three counters which will be maintained by router a

1. A counter for packets that flow "through" both a, b
2. A counter for packets whose source is a but which flows through b
3. A counter for packets that flow "through" a but whose destination is b

Similarly router a will also maintain three more counters for the opposite direction. Now by simply using a rule which says that the number of packets (consider size) that are outgoing from a should be equal to the number of packets that are incoming to b. This is the rule when both a, b are internal routers. Same rule can be transformed when some source or destination is involved.

3 Locating Network Domain Entry and Exit Point/Path for DDOS Attack Traffic

In this case, Packet with a certain source and a certain destination will flow through a certain path according to the routing table. Anomaly from this path is analyzed.

In networks, DDoS (Distributed Denial-of-Service) is the most popular attack. Moreover it is the attack where the router doesn't behave abnormally. It just receives too many packets for processing. So the router spends all its resources in processing the packets and drops the legitimate packets that really needed attention. Packets that flow through the network are routed based on the destination field. The source field is not verified to check whether it is a legitimate one or one which is forged. So an attacker can easily forge thousands of IP addresses and fire packets to a router destination. Now it was observed that the routing table doesn't change drastically very often. So a packet with a certain source and a certain destination will flow through certain path according to the routing table. The routers can learn such paths over tim

So now if a source address is forged, then router will detect the unusual path that was taken for that source and destination. So a router can systematically learn IP address that it forwards. It can maintain this as the permitted list of IP addresses. Suddenly when the router sees huge traffic, it can stop learning the IP addresses and drop only those packets which are not there in the permitted list till the attack is under control.

4 Network Intrusion Prevention by Configuring ACLS on Routers Based on Snort IDS Alerts

In this attack on router is prevented by the use of Snort to generate alerts for attacks and configure ACLs to defend.

Snort can also be used to detect any intrusion in router and also took measures to take action for that intrusion using ACLs(Access Control Lists). Snort is an open source tool which works as an IDS (Intrusion Detection System). Rules are written in Snort and they are matched against the packets. An example of a rule is as follows

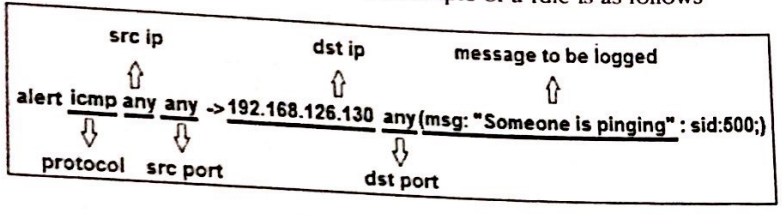


Figure 11.24 A Snort Rule

If a packet matches then messages are sent to the snort log. These logs now can be studied and appropriate access control lists can be generated for the router to curb the attack. Snort is an open source tool and is easily available. Access control lists is an in built feature in routers and very powerful when security is concerned. So the router is secured with very less cost overhead.

11.6 Hardening Linux Operating System for Network Security

We have provided the following guidelines for hardening the linux operating systems for network security

11.6.1 Workstation Security

Security begins with the first time you put that CD or DVD into your disk drive to install Red Hat Enterprise Linux. Configuring your system securely from the beginning makes it easier to implement additional security settings later.

Disk Partitions

Create separate partitions for /boot, /, /home, /tmp, and /var/tmp. The reasons for each are different and we will address each partition.

/boot –The boot loader and kernel images that are used to boot your system into RedHat Enterprise Linux are stored in this partition. This partition should not be encrypted. If this partition is included in / and that partition is encrypted or otherwise becomes unavailable then your system will not be able to boot.

/home - When user data (/home) is stored in / instead of in a separate partition, thepartition can fill up causing the operating system to become unstable. Also, when upgrading your system to the next version of Red Hat Enterprise Linux it is a lot easier when you can keep your data in the /home partition as it will not be overwritten during installation. If the root partition (/) becomes corrupt your data could be lost forever. By using a separate partition there is slightly more protection against data loss. You can also target this partition for frequent backups.

/tmp and /var/tmp - Both the /tmp and the /var/tmp directories are used to store datathat doesn't need to be stored for a long period of time. However if a lot of data floods one of these directories it can consume all of your storage space. If this happens and these directories are stored within / then your system could become unstable and crash. For this reason, moving these directories into their own partitions is a good idea.

To get mounted partition information use the following command. # df–lh, Where df: displays the amount disk space available on the file system, l: limit listening to local file system, h: print in human radable formate (i.e 22k) Utilize LUKS Partition Encryption: During the installation process an option to encrypt your partitions will be presented to the user. The user must supply a passphrase that will be the key to unlock the bulk encryption key that will be used to secure the partition's data [2].

Manually Encrypting Directories

1. Enter runlevel 1 by typing the following at a shell prompt as root: # init 1
2. Unmount your existing /home: # umount /sd7
3. If the command in the previous step fails, use fuser command to find proce hogging /home and kill them: # fuser -mvk /sd7

   Where m: mounted file system, v: verbose output, k: kill process access file
4. Verify /home is no longer mounted:

   # grep sd7 /proc/mounts

5. Fill your partition with random data:
   # dd if= /dev/urandom of= /dev/sda7
   This process can take many hours to complete.
6. Initialize your partition:
   [root@prem /] # cryptsetup --verbose –verify-passphrase luksformat /dev/sda7
7. Open the newly encrypted device:
   [root@prem /] # cryptsetup luksopen /dev/sda7 sd7
8. Make sure the device is present: [root@prem /1 # ls -1 /dev/mapper ! grep sd7
9. Create a file system:mkfs.ext3 /dev/mapper/home [root@prem /] # mkfs.ext4 dev/mapper/sd7
10. Mount the file system:
    [root@prem sd7] # mont /dev/mapper/sd7 /sd7/
11. Make sure the file system is visible: [root@prem sd]# df -hl
12. Add the following to the /etc/crypttab file:
    # home /dev/sd7 none
13. Edit the /etc/fstab file, removing the old entry for /home and adding the following line:
    # /dev/mapper/home /home ext3 defaults 1 2
14. Reboot the machine:
    # shutdown -r now
15. The entry in the /etc/crypttab makes your computer ask your luks passphrase on boot.
16. Log in as root and restore your backup.

Install Minimal Software

It is very critical to look at the default list of software packages and remove unneeded packages or packages that don't comply with your security policy, it is a good practice not to have development packages, desktop software packages (e.g. X Server) etc. installed on production servers. Other packages like FTP and Telnet daemons should not be installed as well unless there is a justified business reason for it (SSH/SCP/SFTP should be used instead). A good approach is to start with a minimum list of RPMs and then add packages as needed.To get a list of all installed RPMs you can use the

following command:#rpm–qa

Where rpm : RPM package manager,-q : query, -a : all

If you want to know more about a particular RPM, run the following command:

 # rpm -qi <package_name>

Where rpm : RPM package manager,-q : query, -i : information

To check for and report potential conflicts and dependencies for deleting a RPM, run following command

 # rpm -e --test <package_name>

Where rpm : RPM package manager, e : erase, test : don't actually uninstall anything

[root@prem]# rpm –e --test rpm

Install Signed Packages: Package signing uses public key technology to prove that thepackage that was published by the repository has not been changed since the signature was applied. Verifying Signed Packages: All Red Hat Enterprise Linux packages are signed with the Red Hat GPG key. Assuming the disc is mounted in / mnt/cdrom, use the following command as the root user to import it into the keyring (a database of trusted keys on the system):

[root@prem cdrom]# rpm—import RPM-GPG-KEY-redhat-beta

Now, the Red Hat GPG key is located in the /etc/pki/rpm -gpg/ directory.

To display a list of all keys installed for RPM verification, execute the following command:

[root@prem cdrom] # rpm -qa gpq-pubkey*

Where rpm : RPM package manager,-q : query , -a : all

To verify all the downloaded packages at once, issue the following command: [root@ prem packages] # rpm –k rpcbind-0.2.0-8.el6.i686.rpm

Where rpm : RPM package manager -K : check the package key

For each package, if the GPG key verifies successfully, the command returns gpg md5 OK. Packages that do not pass GPG verification should not be installed, as they may have been altered by a third party.

BIOS and Boot Loader Security

Password protection for the BIOS (or BIOS equivalent) and the boot loader can prevent unauthorized users who have physical access to systems from booting using removable media or obtaining root privileges through single user mode. BIOS Passwords: The two primary reasons for password protecting the BIOS of a computer are:

1. Preventing Changes to BIOS Settings
2. Preventing System Booting If you forget the BIOS password, it can either be resetwith jumpers on the motherboard or by disconnecting the CMOS battery. For this reason, it is good practice to lock the computer case if possible.

Preventing BIOS Password Circumvention: Since Linux distributions can be runfrom any form of removable media (CDs. VDs, floppy drives, and USB devices), disabling the ability to boot from any form of removable media is advisable and will keep out many of the lower-level, script-kiddie- attackers.

Boot Loader Passwords: The primary reasons for password protecting a Linux bootloader are as follows [2].

1. *Preventing Access to Single User Mode*
2. *Preventing Access to the GRUB Console*
3. *Preventing Access to Insecure Operating Systems*

Password Protecting GRUB: To do this, first choose a strong password, open a shell,log in as root, and then type the following command:

[root@prem log] # grub-md5-crypt

When prompted, type the GRUB password and press Enter. This returns an MD5 hash of the password. Next, edit the GRUB configuration file /boot/grub/grub.conf. Open the file and below the timeout line in the main section of the document, add the following line:

password --md5 *<password-hash>*

Replace <password-hash> with the value returned by /sbin/grub-m d5-crypt.

Whole Disk or Partition Encryption The best way to protect against data tampering or unintended disclosure is to implement one of the many whole disk or partition encryption methodologies available to Linux systems. This entails encrypting the entire contents of the hard drive, or partition, using a cryptographic encryption algorithm.

Password Security: For security purposes, the installation program configures the system to use Secure Hash Algorithm 512 (SHA512) and shadow passwords. The single most important thing a user can do to protect his account against a password cracking attack is create a strong password.

/etc/passwd file

[root@prem ~] #tail /etc/passwd /etc/shadow file

[root@prem ~] # tail /etc/shadow

1. Disallow Remote Root Login

Any actions requiring a direct log on to the system via _root' should be restricted to the local console. Edit ***/etc/securetty*** to reflect the following changes:

2. Disabling root access via any console device (tty)

To further limit access to the root account, administrators can disable root logins at the console by editing the /etc/securetty file.

To prevent the root user from logging in, remove the contents of this file by typing the following command at a shell prompt as root:

[root@prem]#echo > ***/etc/securetty***

11.6.2 Server Security

Disabling root SSH logins: To prevent root logins via the SSH protocol, edit the SSHdaemon's configuration file, /etc/ssh/sshd_config, and change the line that reads:#PermitRootLogin yes to read as follows:PermitRootLogin no

Disable CTRL-ALT-Delete: For those machines with poor or non-existent physical security, to disable the CTRL-ALT-Delete function that allows an attacker to shut down the machine. Edit /etc/inittab to comment out the following line.

 # ca::ctrlaltdel:/sbin/shutdown -t3 -r now

Save the change and to restart the service for it to take effect run the following command:

[root@hod BBM]# */sbin/init q*

Where q: tell init to re-exmine the /etc/inittab file

Thus We summarized the vulnerabilities, attacks and defense mechanisms.

When a system is used as a server on a public network, it becomes a target for attacks. Hardening the system and locking down services is therefore of paramount importance for the system administrator. Before delving into specific issues, review the following general tips for enhancing server security: Keep all services current, to protect against the latest threats., Use secure protocols whenever possible, - Serve only one type of network service per machine whenever possible, Monitor all servers carefully for suspicious activity.

Enhancing Security with TCP Wrappers

TCP Wrappers are capable of much more than denying access to services. This section illustrates how they can be used to send connection banners, warn of attacks from particular hosts, and enhance logging functionality. Refer to the hosts_options man page for information about the T CP Wrapper functionality and control language.

TCP Wrappers and Connection Banners: To implement a TCP Wrappers banner fora service, use the banner option. For this example, the file is called /etc/banners/ vsftpd and contains the following lines:

220-Hello, %c

220-All activity on ftp.exam ple.com is logged.

220-Inappropriate use will result in your access privileges being removed.

The %c token supplies a variety of client information, such as the username and hostname, or the username and IP address to make the .connection even more intimidating.

For this banner to be displayed to incoming connections, add the following line to the */etc/hosts.allow file:*

vsftpd : ALL : banners /etc/banners/

TCP Wrappers and Attack Warnings: If a particular host or network has beendetected attacking the server, TCP Wrappers can be used to warn the administrator of subsequent attacks from that host or network using the **spawn** directive.

Assume that a cracker from the 206.182.68.0/24 network has been detected attempting to attack the server. Place the following line in the */etc/hosts.deny* file to deny any connection attempts from that network, and to log the attempts to a special file:

ALL : 206.182.68.0 : spawn /bin/echo `date` %c %d >> /var/log/intruder_alert

The %d token supplies the name of the service that the attacker was trying to access. To allow the connection and log it, place the spawn directive in the */etc/hosts.allow* file.

TCP Wrappers and Enhanced Logging: If certain types of connections are of moreconcern than others, the log level can be elevated for that service using the severity option. For this example, assume that anyone attempting to connect to port 23 (the T elnet port) on an FT P server is a cracker. To denote this, place an emerg flag in the log files instead of the default flag, info, and deny the connection. To do this, place the following line in /etc/hosts.deny: in.telnetd : ALL : severity emerg This uses

the default authpriv logging facility, but elevates the priority from the default value of info to emerg, which posts log messages directly to the console.

116.3 Network Security

Protect portmap With TCP Wrappers: It is important to use TCP Wrappers to limitwhich networks or hosts have access to the portmap service since it has no built-in form of authentication. Further, use only IP addresses when limiting access to the service. Avoid using hostnames, as they can be forged by DNS poisoning and other methods.

Securing FTP

The File Transfer Protocol (FTP) is an older TCP protocol designed to transfer files over a network. Red Hat Enterprise Linux provides three FTP servers [2].

1. gssftpd -A Kerberos-aware

2. xinetd - based FTP daemon that does not transmit authentication information over the network.

3. Red Hat Content Accelerator (tux) - A kernel-space Web server with FTPcapabilities. vsftpd — A standalone, security oriented implementation of the FTP service. The following security guidelines are for setting up the vsftpd FTP service.

Configuring vsftpd for Anonymous FTP: The vsftpd.conf file controls the vsftpd daemon. The vsftpd binary has only one ommandline option, which allows you to specify the location of the vsftpd.conf configuration file.

 # vsftpd /etc/vsftpd.conf

Following shows a simple configuration file for a secure stand-alone anonymous FTP server that allows only downloads.

The /etc/vsftpd.conf file

 # General Configuration

listen=YES

background=YES

listen_address=192.168.0.1

nopriv_user=ftp_nopriv

xferlog_enable=YES

 # Mode and Access rights

anonymous_enable=YES

local_enable=NO

write_enable=NO

188

cmds_allowed=PASV,RETR,QUIT

 # Security

ftpd_banner=Puppy.YourDomain.Net FTP Server connect_from_port_20=YES

hide_ids=YES pasv_min_port=50000 pasv_max_port=60000

 # DoS

ls_recurse_enable=NO max_clients=200 max_per_ip=4

Securing SSH: Many network services like telnet, rlogin, and rsh are vulnerable toeavesdropping which is one of several reasons why SSH should be used instead. The Restricting System Access from Servers and Networks shows how direct logins can be disabled for shared and system accounts including root. For securing SSH Use the following parameters: no, UsePrivilegeSeparation yes, Allow Tcp Forwarding no, Strict Modes yes Ensure that all host-based authentications are disabled. These methods should be avoided as primary authentication. Ignore Rhosts yes, Host based Authentication no, Rhosts RSA Authentication no, Disable sftp if it's not needed: #Subsystem sftp /usr/lib/misc/sftp-server After changing any directives make sure to restart the sshd daemon run the following command: root@prem] # /etc/init.d/sshd restart

ssh passphrase: Passwords aren't very secure, Anyone who gains access to your drivehas gained access to every system you use that key with. This is also a Very Bad Thing. The solution is obvious, add a passphrase.

Step by step procedure to create passphrase for ssh as follow.

- Ssh server is running sshd so it will allow remote ssh login and which stores public key.
- ssh client has ssh client which is used to get ssh access to ssh server . Passphrase is created on ssh client and store private key.
- Create user ***prem*** on both server and client.
- **On client**

1) Generate an RSA key pair by typing the following command at a shell prompt:

 # ssh-keygen –d

Where: -d: dsa It will create .ssh directory in user home directory. Then it will ask for the passphrase, Enter the passphrase, the two files are created in .ssh directory,

1 id_dsa (private key) id_dsa.pub (public key)

2) Login on server from client using ssh **On sever**

1) Create .ssh directory in home directory (/home/hod) of user prem. Use the following command. # mkdir .ssh
2) To Go inside .ssh directory use follwing command. # cd .ssh
3) Use the following command to copy public key into server.

   # scp hod@client_ip:.ssh/id_dsa.pub authorized_keys

   Where : scp : secure copy command
4) Set owner permissions on /home/hod/.ssh directory and authorized_keys file.
5) To set permission use the following commands.

   # chmod 700 /home/prem/.ssh ,# chmod 600 authorized_keys

   Where ***: chmod :*** change the file or directory permissions

The Network Manager Daemon

The Network Manager daemon runs with root privileges and is usually configured to start up at boot time. You can determine whether the Network Manager daemon is running by entering this command ***as root:# service NetworkManager status:***

NetworkManager (pid 1527) is running. The service command will report Network Manager is stopped if the Network Manager service is not running. To start it for the current session: ***#service Network Manager start*** Run the chkconfig command to ensure that NetworkManager starts up every time the system boots: ***#chkconfig Network Manager on***

Security Enhanced Communication Tools

As the size and popularity of the Internet has grown, so has the threat of communication interception. Over the years, tools have been developed to encrypt communications as they are transferred over the network. Red Hat Enterprise Linux 6 ships with two basic tools that use high-level, public-key-cryptography-based encryption algorithms to protect information as it travels over the network.

OpenSSH - A free implementation of the SSH protocol for encrypting network communication. Gnu Privacy Guard **(GPG)** — A free implementation of the PGP (Pretty Good Privacy) encryption application for encrypting data. OpenSSH is a safer way to access a remote machine and replaces older, unencrypted services like telnet and rsh. OpenSSH includes a network service called sshd and three command line clientapplications: **ssh** — A secure remote console access client. **scp** — A secure remote copy command. **sftp** — A secure pseudo-ftp client that allows interactive file transfer sessions.

Enable TCP SYN Cookie Protection: A "SYN Attack" is a denial of service attackthat consumes all the resources on a machine. Any server that is connected to a network is potentially subject to this attack. To enable TCP SYN Cookie Protection, edit the */etc/sysctl.conf* file and add the following line:

net.ipv4.tcp_syncookies = 1

Disable IP Source Routing: Source Routing is used to specify a path or route throughthe network from source to destination. This feature can be used by network people for diagnosing problems. However, if an intruder was able to send a source routed packet into the network, then he could intercept the replies and your server might not know that it's not communicating with a trusted server. To enable Source Route Verification, edit the */etc/sysctl.conf* file and add the following line: ***net.ipv4.conf.all.accept_source_route = 0***

Disable ICMP Redirect Acceptance: ICMP redirects are used by routers to tell theserver that there is a better path to other networks than the one chosen by the server. However, an intruder could potentially use ICMP redirect packets to alter the hosts's routing table by causing traffic to use a path you didn't intend.

To disable ICMP Redirect Acceptance, edit the /etc/sysctl.conf file and add the following line: ***net.ipv4.conf.all.accept_redirects = 0***

Enable IP Spoofing Protection: IP spoofing is a technique where an intruder sends outpackets which claim to be from another host by manipulating the source address. IP spoofing is very often used for denial of service attacks. To enable IP Spoofing Protection, turn on Source Address Verification. Edit the /etc/sysctl.conf file and add the following line:

net.ipv4.conf.all.rp_filter = 1

Enable Ignoring to ICMP Requests: If you want or need Linux to ignore pingrequests, edit the /etc/sysctl.conf file and add the following line:

net.ipv4.icmp_echo_ignore_all = 1

This cannot be done in many environments.

Enable Ignoring Broadcasts Request: If you want or need Linux to ignore broadcast requests, edit the /etc/sysctl.conf file and add the following line:

net.ipv4.icmp_echo_ignore_broadcasts = 1

Enable Bad Error Message Protection: To alert you about bad error messages in thenetwork, edit the /etc/sysctl.conf file and add the following line:

net.ipv4.icmp_ignore_bogus_error_responses = 1

Enable Logging of Spoofed Packets, Source Routed Packets, Redirect Packets: Toturn on logging for Spoofed Packets, Source Routed Packets, and Redirect Packets, edit the /etc/sysctl.conf file and add the following line:

net.ipv4.conf.all.log_martians = 1

Case Studies: Network Design

1. Design the campus wide network for the following campus of the engineering department. The location of the core switch, distribution switch and access switch and departmental requirements for connectivity of the computers are shown in figure. Distribution switch is the 24 port switch, access switch has 24 port.
   1) Give the requirements of Active Components Passive Components, wire requirements -UTP, Fiber
   2) Also specify the server requirements and operating systems, requirements for setting the data centre.

Solution

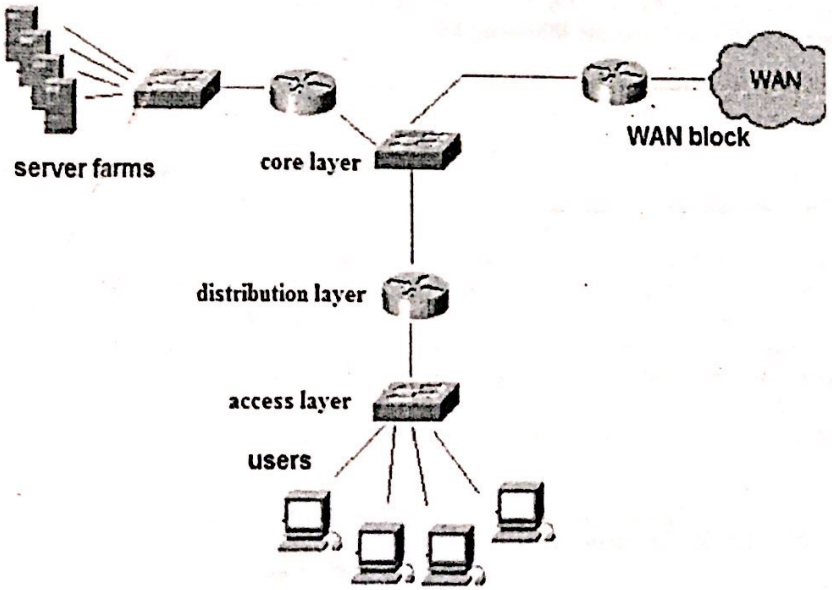


Figure 11.25 Location of the core distribution and access layer

The above locations of core switch at data centre, distribution layer at department level and access layer at floor of the department is considered in the following block diagram of the campus of the engineering

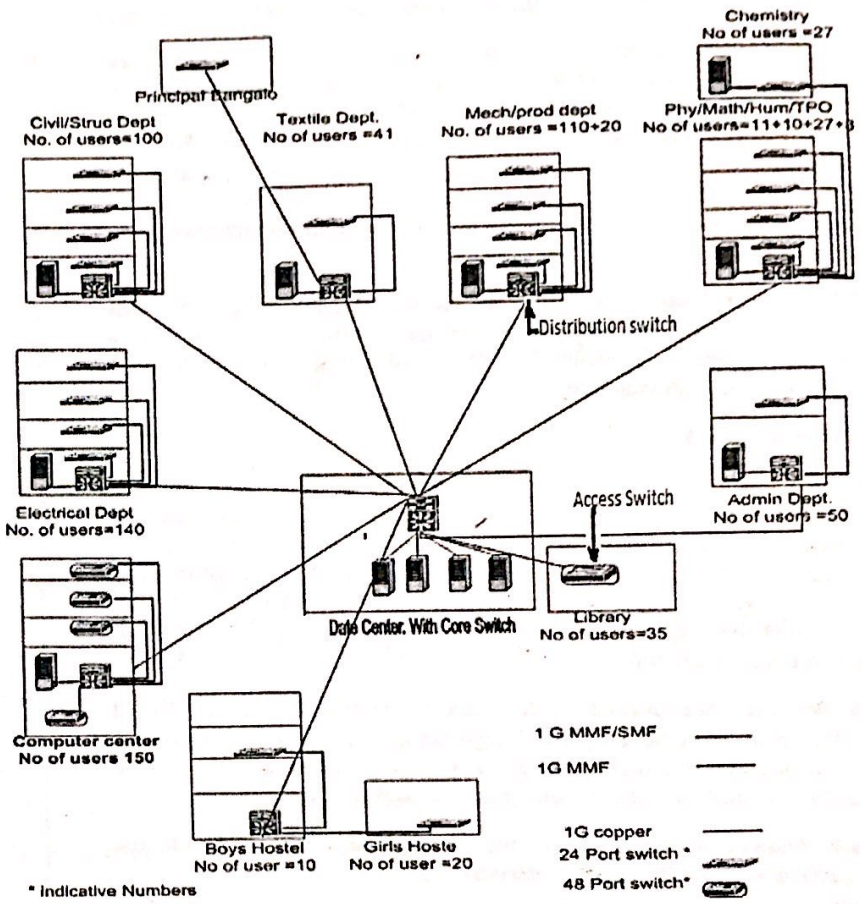


Figure 11.26 Design of Campus of Engineering College

You can define the cabling requiremnets as per the distances of the deparments from the core layer to distribution layer, distribution layer to access layer and from access layer to usres.