Welcome!

RH253

Red Hat Network Services
and Security Administration

Objectives

- Understanding and Managing System Monitoring
  - Understanding Monitoring
  - Monitoring Techniques
  - Using System Logs and Files
- Security Concerns and Policy
  - Understanding Security
- Securing Networks Using Firewall (IPTables)
  - Understanding Firewall and IPTables
  - Applying Firewall and Securing Network
  - IP Forwarding and Routing

Understanding and Managing System Monitoring

What is Monitoring?

- An important part of maintaining a secure system is keeping track of the activities that take place on the system. If you know what usually happens, such as understanding when users log into your system, you can use log files to spot unusual activity

Monitoring Techniques

- Learn to identify files statistics
- Ensure filesystem integrity
- Understanding system log configuration
- Learn Log file analysis
- Understand Process Monitoring

Using LOG Files

- Monitoring Log files will help detect:
  - **Equipment problems such as hard disk crashes or any other devices**
  - **Users problems such as repeated login failures**
  - **Security breaches from outside the system**

Using syslogs

- Red Hat Enterprise Linux 4 comes with several utilities you can use to monitor activity on a system. These utilities can help you identify the culprit if there is a problem.
- RHEL 4 comes with two logging daemons. The kernel log daemon service, **klogd**, logs kernel messages and events. The syslog daemon, **syslogd**, logs all other process activity. You can use the log files that syslogd generates to track activities on your system. If you are managing multiple Red Hat Enterprise Linux systems, you can configure the syslogd daemon on each system to log messages to a central host system.
- Both **syslogd** and **klogd** are configured in ***/etc/syslog.conf*** file

syslog.conf file

- Location
- **/etc/syslog.conf**
- The format is straightforward. The first entry specifies a semi-colon delimited list of **facility.priority** declarations. The second filed specifies the **log location**, which is usually a file.
- Syntax:
- **facility.priority   log_location**

Facilities and Priorities:

- Facilities are like services and Priorities are like type of log want to generate like info, error and alert etc…
- Examples:

- **Facilities**
  - **cron**
  - **mail**
  - **lpr**
- **Priorities**
  - **info**
  - **err**
  - **alert**

Syntax of syslog.conf file

- Syntax:

```
facility.priority        log_location
```

Example:

```
kern.info        /var/log/kernel
```

Security Concerns and Policy

Understanding Security

- A network is only as secure as the most open system in that network. Although no system can be 100 percent secure, you can follow certain basic host measures to enhance the security on any given system and, consequently, your network. When devising security measures, you have to plan for two types of security violations: user accidents and break-ins.
- Accidents happen because users lack adequate training or are unwilling to follow procedures. If security is too burdensome, productivity may suffer, and your users will try to get around your rules. Password security falls into this category.
- When a cracker breaks into your system, some crackers may be looking for secrets such as credit card information. Others may just want to bring down your system.

Understanding Security

Types of Security

- **Network (External)**
- **Local (Internal)**
- **Physical**

Hacker versus Cracker

- A hacker is someone who programs creatively and usually for the pure enjoyment of it (most programmers who work on Linux are hackers in this sense). The correct term for someone who breaks into computer systems is a cracker.
- There are many types of crackers, ranging from professional computer criminals to the hobbyist types that break into computers for the thrill. The growth of the cracker problem has kept pace with the growth of the Internet. A new, younger generation of cracker is emerging. These teenage pseudo-crackers do not have all the knowledge and skill of their true cracker counterparts, but they have access to a growing number of cracker tools that automate the breaking of a system's security.

Understanding Attack Techniques

- Attacks on computing systems take on different forms, depending on the goal and resources of the attacker. Some attackers desire to be disruptive, while others desire to infiltrate your machines and utilize the resources for their own nefarious purposes. Still others are targeting your data for financial gain or blackmail. Here are three major categories of attacks:
- **Denial of Service (DOS)**
- **Distributed Denial of Service (DDOS)**
- **Intrusion attacks**

Denial of Service (DOS)

- The easiest attacks to perpetrate are Denial of Service attacks. The primary purpose of these attacks is to disrupt the activities of a remote site by overloading it with irrelevant data. DOS attacks can be as simple as sending thousands of page requests per second at a Web site. These types of attacks are easy to perpetrate and easy to protect against. Once you have a handle on where the attack is coming from, a simple phone call to the perpetrator's ISP will get the problem solved.

Distributed Denial of Service (DDOS)

- More advanced DOS attacks are called Distributed Denial of Service attacks. DDOS attacks are much harder to perpetrate and nearly impossible to stop. In this form of attack, an attacker takes control of hundreds or even thousands of weakly secured Internet connected computers. The attacker then directs them in unison to send a stream of irrelevant data to a single Internet host. The result is that the power of one attacker is magnified thousands of times. Instead of an attack coming from one direction, as is the case in a normal DOS, it comes from thousands of directions at once. The best defense against DDOS attack is to contact your ISP to see if it can filter traffic at its border routers.

Intrusion attacks

- To remotely use the resources of a target machine, attackers must first look for an opening to exploit. In the absence of inside information such as passwords or encryption keys, they must scan the target machine to see what services are offered. Perhaps one of the services is weakly secured and the attacker can use some known exploit to finagle his way in.

Diagnostic Utilities

- **Port Scanners**
  - Show what services are available on a system
  - **nmap**
- **Packet Sniffers**
  - Stores and analyzes all network traffic
  - **tcpdump**
  - **ethereal**

Securing Networks Using Firewall (IPTables)

What is Firewall?

- Information security is commonly thought of as a process and not a product. However, standard security implementations usually employ some form of dedicated mechanism to control access privileges and restrict network resources to users who are authorized, identifiable, and traceable. Red Hat Enterprise Linux includes several powerful tools to assist administrators and security engineers with network-level access control issues

What is Firewall?

- Firewalls are one of the core components of a network security implementation. Several vendors market firewall solutions catering to all levels of the marketplace: from home users protecting one PC to data center solutions safeguarding vital enterprise information. Firewalls can be standalone hardware solutions, such as firewall appliances by Cisco, Nokia, and Sonicwall. There are also proprietary software firewall solutions developed for home and business markets by vendors such as Checkpoint, McAfee, and Symantec.
- Apart from the differences between hardware and software firewalls, there are also differences in the way firewalls function that separate one solution from another

What is Firewall?

- three common types of firewalls and how they function:
- NAT
- Packet Filtering
- Proxy

NAT

- *Network Address Translation* (NAT) places private IP subnetworks behind one or a small pool of public IP addresses, masquerading all requests to one source rather than several.

Packet Filtering

- A packet filtering firewall reads each data packet that passes within and outside of a LAN. It can read and process packets by header information and filters the packet based on sets of programmable rules implemented by the firewall administrator. The Linux kernel has built-in packet filtering functionality through the Netfilter kernel subsystem.

Proxy

- Proxy firewalls filter all requests of a certain protocol or type from LAN clients to a proxy machine, which then makes those requests to the Internet on behalf of the local client. A proxy machine acts as a buffer between malicious remote users and the internal network client machines.

Netfilter and iptables

- The Linux kernel features a powerful networking subsystem called *Netfilter*. The Netfilter subsystem provides stateful or stateless packet filtering as well as NAT and IP masquerading services. Netfilter also has the ability to *mangle* IP header information for advanced routing and connection state management. Netfilter is controlled through the iptables utility.

Firewall Policies

- Firewall sits between your internal network and the outsides network
- Filters information on a packet by packet basis
- Info in packets : Source address, types of data, destination address

Linux Firewall commands

- Ipfwadm for linux kernel 2.0
- Ipchains for linux kerkel 2.0
- IPTables for linux kernel 2.4 and now for 2.6 also

IPTables

- **IPTables** is really and front-ent (user-space) tool to manage Netfilter

  (integrated within the Linux Kernel)

- **IPTables** functions primarily at OSI Layers 3 (Network (IP)) & 4 (Transport (TCP,UDP))

  Layer 3 focuses on Source Address & Destination Address

  IP Addresses are based on 32-bit ranges (4 billions address)

  Layer 4 focuses on Protocols:Ports TCP:80, UDP:69

  TCP/UDP ports use a 16-bit range (0- 65535)

- **IPTables** can manage ICMP

  ICMP uses types : echo-request, echo-reply

IPTables Command

- Iptables **–t** table (Action / Direction) **(Packet Pattern) –j (fate)**
- **Tables : filter (default) , nat , mangle**
- **Actions : -A append, -D delete, -L list, -F flush**
- **Direction : - INPUT, OUTPUT, FORWARD**
- **Packet Pattern: -s Source IP-Address –d Destination IP-Address**
- **Fate: DROP, ACCEPT, REJECT**

IPTables Commands

- Examples :
- iptables -A INPUT -s 192.168.1.0/24 -j REJECT
- iptables -A INPUT -s 192.168.0.20 -p icmp -j DROP
- iptables -A INPUT -m mac --mac-source 12:34:56:89:90:ab -j ACCEPT
- iptables -A OUTPUT -d www.yahoo.com -j REJECT

?

Questions