System and Network Security

Saurabh Barjatiya

2012-03-20 Tue

Contents

1 Introduction

Objective of this lecture is to introduce very basic concepts of system and network security so that students can secure their system from malware / network attacks. Hence in this lecture first we will study few tools that help in determining current state of the system, with respect to running process, open files, open network sockets, etc. This information will help in determining processes running on current system. Once users have this information they can determine whether a particular process should be accessing some file or whether it should be listening on given port or not. Malicious processes can be killed and unwanted services can be stopped as first step towards securing the system.

In second part of this lecture we will look at iptables firewall that is provided with most Linux distributions and its basic configuration. With help of iptables students will be able to protect their system from network attacks, by limiting number of open ports (allowed connections) and there by limiting surface area of attack.

2 System security

In this section we will have a look at various tools that help in gathering information about current system. We will also mention few tips in maintaining secure system.

2.1 Tools

2.1.1 `top`

If users face performance and response time problems with a system then they can use `top` to get list of processes running on system arranged in descending order of their CPU usage. In `top` output we can look at top three four most CPU consuming processes and kill / stop them if they are not necessary.

Sometimes system maintenance processes like `makewhatis`, `prelink`, `updatedb` etc. which are run periodically to maintain system in good health and update various databases consume lot of CPU. We should avoid killing this processes to improve response time as these processes help in proper functioning of the system.

If users are not sure what top CPU using processes are for and their purpose, they should first search using Internet search engines to determine the purpose of these processes before deciding to terminate them.

- Memory usage

  If users are facing memory usage problems then they can press 'F' followed by 'n' to sort the processes in decreasing order of memory usage and determine which processes are using most memory and restart / stop those processes.

  In order to see whether lack of memory is causing problems for system performance one can use `free -m` command and look at usage of swap usage. If more than 100 MB of swap is being used on a desktop system which has not been running for more than a week then system is most likely suffering from memory related issues.

- Hard-disk usage

  If some process uses considerable hard-disk then it will slow down complete system. To determine whether hard-disk usage is causing

system to run slow, one can look at `wa` percentage shown by top in third line. Whenever this percentage is very high (more than 30%) continuously for long duration then system is suffering from slow disk I/O problem.

- Too many processes

  One of the parameters in measuring system performance is load average. `top` shows load average in its first line. We can also find load average with help of `uptime` and `w` commands. Load average in approximate sense indicates how much each process has to wait before it gets its turn to execute on one of the CPUs. If this value is close to 1.00 or worse greater than 1.00 then system is facing problem of having too many processes.

  Three values of load average are maintained by system: 1 min, 5 min and 15 min. Hence if there is some persistent load / process problem on a system then its load average of last 15 min would be near 1.00 indicating serious problem.

  If command line based `top` is not easy to use then one can also try its GUI counterpart `gnome=system=monitor` which also provides similar information.

2.1.2 `lsof`

`lsof` command has to executed with superuser / root privileges. It gives list of all open file descriptors by all processes running on system. This information is useful in checking which files are being used by a given process. We can check files opened by suspected processes and stop them in case they seem to be opening files which they are not supposed to.

2.1.3 `netstat`

`netstat` gives two very important set of information related to network and current system:

- List of ports on which some program is listening for incoming connections along with name of program for each particular port
- List of established connections for each program including source IP, source port, destination IP and destination port information.

To get detailed information on which process is listening on given port and gets its PID we via netstat, we need to run netstat with superuser / root privileges. We can run netstat without root privileges but in that case we will only get information of open ports and established connections but not about which process is responsible for given connection / port.

Once we have network status information, we can determine whether a particular process should be listening on given port or not. We can also verify whether a given application requires a given network connection to some remote host or not and act accordingly.

2.1.4 `w`

It is important to see if there are other users logged on our system and what they are doing. `w` commands help in listing other users logged on our system, username with which they have logged in and their IP addresses. It also tells when they logged in and since how long they have been idle.

2.1.5 `last`, `lastb` and `lastlog`

`last` command lists last successful login on current system for this month and duration of each login. We can find details of older login with older versions of `wtmp` file (`wtmp.1`, `wtmp.2`, etc.) present in `/var/log` folder.

`lastb` command lists last unsuccessful login attempts for some user from remote machine or using GUI / console on same machine. This information is useful in determining if someone has tried to login on our system with various different passwords (brute-force / dictionary attack).

`lastlog` command lists time of last login for each user on system. This helps in finding out inactive users who have not logged in since many months, so that their accounts can be locked, deactivated or even deleted.

2.1.6 `logwatch`

Fedora / RHEL / Cent-OS based systems come with tool called `logwatch` installed which monitors logs generated in last one day and emails the report to root@localhost. Administrators can login as root and use `mail` command to read mails sent to root user. Logwatch emails contain good overview of events that happened on system in last one day and about any potential problems.

Advanced users can configure `/etc/aliases` and `sendmail` so that these emails go directly to administrators normal email address, so that adminis-

trators do not have to manually login as root user on each system to reach logwatch report.

2.1.7 `service` and `chkconfig`

Many servers and processes are provided in form of services so that they automatically start at system boot and continue to run in background without getting affected by user login / logout events. The startup / stop commands for each of these scripts are mentioned in executable shell scripts kept in `/etc/init.d` folder. To configure which services will start and which will stop in given runlevel, symbolic links of these script files are created in `/etc/rc.d/rc<n>.d` folders.

To help with easy configuration of these services to commands: `service` and `chkconfig` are provided on many Linux distributions. With help of `service` command we can start, stop, restart, etc. any service that is available on system. With help of `chkconfig` command we can enable / disable automatic starting of particular service on given runlevel.

To keep system safe and to make it boot faster, it is best to disable all unwanted services from system. One can use following command as root use to find services that will automatically start in runlevels 3 to 5 on current system

```
chkconfig --list | grep '[3-5]:on'
```

One can then find out details of each service using search engines or description mentioned in `/etc/init.d/<service>` files and choose to stop them from running on start-up using

```
chkconfig <service> off
```

To stop/start any service just for current boot we can use: service <service-name> {start | stop | restart}

2.2 Tips

Some tips that can be used to maintain system in healthy and secure state are:

- Prefer installations done via package managers like apt-get, yum, synaptic, etc. over installation done by source. Packages have list of dependencies and package managers will not allow installation of software for

which required libraries or required version of libraries is not present. Package managers also provide option of easy un-installation which is not possible when software are installed via source

- Keep system updated with all security updates applied. Most distributions will provide package managers to keep system updated with all security patches. We should keep our system as up to date as possible to avoid security problems from known vulnerabilities.

- Do not keep desktop systems on for long time. Rebooting system will kill all unwanted processes and close all unused file descriptors. During reboot system also checks partitions for consistency. Hence it is important for desktop systems to be rebooted regularly.

- Take regular backups (even if system is on raid). Backups are very important and provide a base line for recovery even if system crashes or gets completely compromised.

- Read logwatch reports regularly. They have important information on which commands were run via sudo user, which packets were logged by firewall, errors encountered by servers like web server, users who logged in remotely via SSH, etc. Logwatch emails also contain pre-fail warning about bad sectors on hard-disk which can be used as signal to take immediate backup and change hard-disks to avoid data loss from disk failures.

- Do not give execute permissions (`chmod +x`) to files downloaded from Internet / LAN unless you trust the source and you know exactly what you are doing.

3 Network Security

3.1 Introduction to iptables

A very basic introduction to iptables is hosted at http://www.sbarjatiya.in/website/tutorials/iptables/ip
Students are advised to go through the tutorial to understand basics of iptables firewall.

3.2 Sample iptables configuration for desktops

Simple iptables configuration for dekstops which will only allow remote SSH connections and web connections and deny everything else is:

```
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -i lo -j ACCEPT
-A INPUT -p icmp --icmp-type any -j ACCEPT
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 80 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 443 -j ACCEPT
-A INPUT -p udp -j DROP
-A INPUT -m state --state NEW -m limit --limit 2/min -j LOG --log-prefix "denied_conne
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT
#
```

Above rules will also log denied connection attempts on other ports. This information can be read following day in logwatch reports to investigate intrusion attempts made on current system, if any.

4 More tools

There are many more tools and techniques that help in securing system but were not discussed in the lecture. Interested students can try and learn following tools / topics to secure their systems even more:

- Auditing daemon and auditing configuration
- Syslog configuration and logging
- Tripwire / AIDE file based Intrusion Detection Systems (IDS)
- Snort network Intrusion Prevention System (IPS)
- Denyhosts and secure SSH server configuration
- iptables port knocking