U.S. DEPARTMENT OF ENERGY'S
CYBERFORCE®
PROGRAM

CyberForce® 101

Typical Services

October 2023

cyberforcecompetition@anl.gov

Typical Services

Pre-Requisites

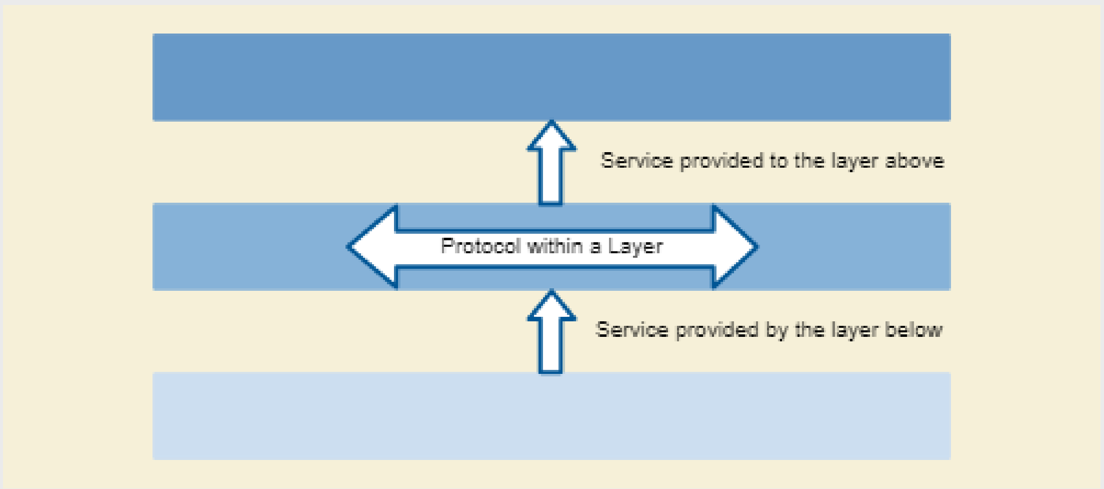
- **Intro to Linux**
- **Intro to Networking**
- **Useful Protocols**

Overview

This guide provides information on common services that we usually see running on servers and also suspicious services that shouldn't be running. You want to aim for as few as services as possible to avoid outside connections that can be vulnerable to attacks. The idea is that a port that isn't opened can't be broken into, at least to outside connections.

Protocols vs Services

A **service** is a set of capabilities or operations that a particular layer provides to the entities of its higher layer. A **protocol**, however, refers to the set of rules and conventions governing the format and interpretation of different components of frames, packets, or messages by peer entities within a layer. Basically, a protocol specifies a horizontal dialogue between two computing systems across a network, while a service describes a vertical relationship within a system.


Service provided to the layer above
Protocol within a Layer
Service provided by the layer below

Relationship of Services to Protocol (From G Overview)

System Audit

In order to see the services running on our system, we need to perform a system audit. `netstat` is a very useful utility for viewing the current state of your network status. We can see what servers are listening for incoming connections, what interfaces they listen on, who's connected to us, who we connect to, etc.

If we want to see both TCP and UDP connections, we can use the flag `-tua` after the command. By adding the `-n` flag we can see the names, and consequently the port numbers. Local address is `0.0.0.0` meaning all interfaces are available, and the local port is 515. Also, important to note that the `netstat` output does not reflect whether there may be a firewall in place that may be filtering incoming connections.

Open a terminal on your machine and use the command `netstat -tap | grep LISTEN`. Then you should see a list of all currently running servers (indicated by the keyword LISTEN) along with the PID and Program Name that started each particular service. On a Windows machine, the `netstat` command can be run to view the services. `grep` is specific to Linux.

Some services that we usually want to keep include services like `sshd` (Secure SHell Daemon) that's used to access the system and `smbd` (Samba) that's used for file sharing. SMTP is a common email server while Apache is a common web server.

If there are services that don't look familiar to you, you might get a brief explanation in your `/etc/services` file.

It is worth noting that `telnet` and `ftp` daemons can be found as servers, aka "listeners." That means these accept incoming connections to you, but yo do not need (or want) these just to use the `ftp` or `telnet` clients. You can download files from a FTP site with just the `ftp` client without running an ftp server on your end. These can cause serious security implications.

You should also look up the versions of your services that are running to determine if your system is running an unsafe, insecure version. Look to see if there have been recent updates and security patches for the services running on your system.

Services that Should NOT Run Over the Internet

You should either disable these, uninstall them, or ensure they are current, patched versions and effectively firewalled (only if you really need them running). These services are potentially insecure by their very nature and are often seen as targets. This is not a complete list but common services that are sometimes started on default Linux installations.

- NFS (Network File System) and related services, including `nfsd`, `lockd`, `mountd`, `statd`, `portmapper`, etc. NFS is used for sharing file systems across a network but is dangerous over the Internet

- `rpc. *` services (Remote Procedure Call *), typically NFS and NIS related
- Printer services (`lpd`)
- `r*` (remote, i.e. Remote SHell) services, including `rsh` , `rlogin` , `rexec` , `rcp` , etc. These are unnecessary, insecure, and potentially dangerous. `ssh` will do everything these commands do and much more securely. These will probably show in `netstat` output without the "r" (`rlogin` is just `login` , etc.)
- `telnet` . Use `sshd` instead (Refer to Useful Protocols telnet section for more detail)
- `ftp` server. Much better, safer alternatives for most systems to exchange files like `scp` or `http` . Only use this if running a dedicated `ftp` server.
- BIND (`named`), DNS server package. Mostly not necessary and requires special handling. Only really needed if you are an authoritative name server for a domain.
- Mail Transport Agent, aka "MTA" (`sendmail` , `exim` , `postfix` , `qmail`). Not really needed. If receiving mail directly from other hosts on your LAN, you may need this. Safer to initially disable this.

Common Ports and the Associated Risk

Port Number	Name	Description and Risk
1 - 19	Assorted protocols	Many not needed and can be left off
20	FTP-DATA	Low risk; used for data to come through
21	FTP server	Very high risk
22	SSH (Secure Shell)	Low to moderate risk
23	Telnet	Use ssh instead; moderate risk
25	SMTP	Moderate risk; bad history of exploits
37	Time service	Built-in inetd time service; low risk but for LAN use only
53	DNS	High risk; name servers listen and answer queries for resolving host names to IP addresses
67 (UDP)	BOOTP or DHCP server port	Low risk
68 (UDP)	BOOTP or DHCP client port	Low risk
69	tftp (Trivial File Transfer Protocol)	Very insecure

Port Number	Name	Description and Risk
79	Finger	Used to provide information about the system and logged in users; low risk but gives out too much info and shouldn't be used
80	HTTP standard	Low risk
98	Linuxconf web access administrative port	LAN only, if needed
110	POP3	Low risk
111	sunrpc (Sun Remote Procedure Call) or portmapper	High risk; used by NFS, NIS, etc.
113	identd or auth	Mostly not needed; low risk but can give too much information
119	nntp or news server port	Low risk
123	Network Time Protocol	Low risk but not required for most users
137-139	NetBios (SMB) services	Windows mostly; common port attempt
143	IMAP	Low to moderate risk
161	SNMP	Not needed for most of us; low risk
177	XDMCP	Low risk
443	HTTPS	Low risk
465	SMTP over SSL	Low risk
512 (TCP)	exec (rexec)	High risk
512 (UDP)	biff	Mail notification protocol; low risk
513	login (rlogin)	High risk
514 (TCP)	shell (rsh)	Very insecure; high risk
514 (UDP)	syslog daemon	Low risk; not needed usually
515	lp	Print server; high risk
587	MSA (submission)	Low risk
631	CUPS (print daemon)	Web management port; low risk
635	mountd	Part of NFS

Port Number	Name	Description and Risk
901	SWAT	Samba web admin tool; LAN only
993	IMAP over SSL	Very low risk
995	POP over SSL	Very low risk
1024	First unprivileged port	Dynamically assigned by the kernel; can be almost anything
1080	Sock Proxy	Target for attacks
1243	SubSeven Trojan	Windows only problem
1433	MS SQL server port	Windows only; sometimes a target
2049	nfsd (Network File Service Daemon)	High risk; recommended only LAN use
3128	Squid proxy server	Low risk; use only for LAN
3306	MySQL server	Low risk; should be LAN only
5432	PostgreSQL server	LAN only; low risk
5631 (TCP and UDP)	PCAnywhere ports	Windows only; can be "noisy" and broadcast wide address ranges
31337	Back Orifice	Commonly probed for Windows Trojan

Stopping Services

Linux:

Init services are typically started automatically during the boot process. There is a naming scheme that uses symlinks to determine which services are to be started, or stopped, at any given runlevel. The scripts should be in the `/etc/init.d` (or `/etc/rc.d/init.d`). You can get a listing of these scripts with the command `ls -l /etc/init.d/ | less`.

To stop a running service, use the following command: `sudo /etc/init.d/SERVICE_NAME stop`. That should work for most systems, but if not, use the command `sudo /etc/rc.d/init.d/SERVICE_NAME stop`. This only stops the service now, restarting on the next reboot. Depending on your distribution, you either need to use `update-rc.d` or `chkconfig`. The commands can be seen below.

```
update-rc.d -f SERVICE_NAME remove
# or
```

```
chkconfig SERVICE_NAME off
```

`inetd` is called a "super-daemon" as it spawns sub-daemons. `inetd` generally is started via init scripts and "listens" on the various ports as determined by which services are enabled in its configuration file, `/etc/inetd.conf`. To stop services under the control of `inetd`, you have to adjust the `inetd` configuration. To disable these services, you have to open the `inetd.conf` file with a text editor and comment out the services, save the file, and then restart `inetd` as a root user.

`xinetd` is an `inetd` replacement, serving the same purpose as `inetd` but with a different configuration. The configuration can be found in `/etc/xinetd.conf` or in individual files in the directory `/etc/xinetd.d`. Turning off these services can be done by either deleting the corresponding configuration section or file or by using a text editor and setting `disable = yes` for the appropriate service. Then you need to restart `xinetd`.

If you can't figure out the "right" way to stop a service or if a service is being started and you're not sure where, you can "kill" the process. You need to know the PID, which can be found with `ps`, `top`, `fuser`, or other utilities. For instance if the PID is 1113, you can use the command `sudo kill 1113`. Verify it's gone and if not, use the command `sudo kill -KILL 1113`.

In Windows:

Open the `Services` app on your Windows machine. Right-click on the service for more options, and you can stop it. This doesn't delete or uninstall the program, however.

You can uninstall a program with the Windows Registry (`regedit.exe`). Move to the `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services` key. Select the key of the service you want to delete. From the `Edit` menu, select `Delete`. You can also use a utility supplied by the NT resource kit called `INSTSRV.EXE` that can be used to install and remove services like this `instsrv SERVICE_NAME remove`.

Sources

1. Security Quick-Start HOWTO for Linux (which services do we really need?)
2. G Overview of the OSI Layer and Services Concepts
3. Relationship between Network Services and Protocols
4. Security Quick-Start HOWTO for Linux (netstat tutorial)
5. How do I delete a Service?
6. Linux and Windows Interoperability: Planning and Deploying a Network