# SERVER HARDENING FOR WEB APPLICATION

Server Hardening is the process of enhancing server security through a variety of means which results in a much more secure server operating environment. This is due to the advanced security measures that are put in place during the server hardening process. The Web Server is a crucial part of web-based applications. Having misconfigured and keeping default configuration can expose sensitive information, and that's risk. As a website owner or administrator, you should regularly perform security scan against your website to find for online threats so you can take action before a hacker does.

**Server Hardening**, probably one of the most important tasks to be handled on your servers, becomes more understandable when you realize all the risks involved. The default config of most operating systems are not designed with security as the primary focus. Instead, default setups focus more on usability, communications and functionality. To protect your servers you must establish solid and sophisticated server hardening policies for all servers in your organization. Developing a server hardening checklist would likely be a great first step in increasing your server and network security. Make sure that your checklist includes minimum security practices that you expect of your staff. If you go with a consultant you can provide them with your server hardening checklist to use as a baseline.

## TIPS AND TRICK FOR SERVER HARDENING

Every server security conscious organization will have their own methods for maintaining adequate system and network security. Often you will find that server hardening consultants can bring your security efforts up a notch with their specialized expertise.

**Some common server hardening tips & tricks include**:

- Use Data Encryption for your Communications
- Avoid using insecure protocols that send your information or passwords in plain text.
- Minimize unnecessary software on your servers.
- Disable Unwanted SUID and SGID Binaries

- Keep your operating system up to date, especially security patches.
- Using security extensions is a plus.
- When using Linux, SELinux should be considered. Linux server hardening is a primary focus for the web hosting industry, however in web hosting SELinux is probably not a good option as it often causes issues when the server is used for web hosting purposes.
- User Accounts should have very strong passwords
- Change passwords on a regular basis and do not reuse them
- Lock accounts after too many login failures. Often these login failures are illegitimate attempts to gain access to your system.
- Do not permit empty passwords.
- SSH Hardening
  --- Change the port from default to a non standard one
  --- Disable direct root logins. Switch to root from a lower level account only when necessary.
- Unnecessary services should be disabled. Disable all instances of IRC - BitchX, bnc, eggdrop, generic-sniffers, guardservices, ircd, psyBNC, ptlink.
- Securing /tmp /var/tmp /dev/shm
- Hide BIND DNS Sever Version and Apache version
- Hardening sysctl.conf
- Server hardening by installing Root Kit Hunter and ChrootKit hunter.
- Minimize open network ports to be only what is needed for your specific circumstances.
- Configure the system firewall (Iptables) or get a software installed like CSF or APF. Proper setup of a firewall itself can prevent many attacks.
- Consider also using a hardware firewall
- Separate partitions in ways that make your system more secure.
- Disable unwanted binaries
- Maintain server logs; mirror logs to a separate log server
- Install Logwatch and review logwatch emails daily. Investigate any suspicious activity on your server.
- Use brute force and intrusion detection systems
- Install Linux Socket Monitor - Detects/alerts when new sockets are created on your system, often revealing hacker activity
- Install Mod_security as Webserver Hardening

- Hardening the Php installation
- Limit user accounts to accessing only what they need. Increased access should only be on an as-needed basis.
- Maintain proper backups
- Don't forget about physical server security

## WHAT IS BIOS

BIOS (basic input/output system) is the program a personal computer's microprocessor uses to get the computer system started after you turn it on. It also manages data flow between the computer's operating system and attached devices such as the hard disk, video adapter, keyboard, mouse and printer. The BIOS is built-in software that contains generic code required to control the keyboard, display screens, disk drives and other functions. The primary purpose of the BIOS is to set up hardware and further load and start an operating system. BIOS is placed in a non volatile ROM chip inside the computer, ensuring the availability of BIOS at all times and preventing accidental disk failure. The BIOS checks every hardware connection and locates the devices, after which the operating system is loaded into computer memory.

### POST

The first job of the BIOS after you switch your computer on is to perform the Power On Self Test. During the POST, the BIOS checks the computer's hardware in order to ensure that it is able to complete the startup process. If the POST is completed successfully, the system usually emits a beep. If the test fails, however, the system generally emits a series of beeps. You can use the number, duration and pattern of these beeps to identify the cause of the test failure.

### Startup

With the POST completed, the BIOS then attempts to load the operating system through a program known as a bootstrap loader, which is designed to locate any available operating systems; if a legitimate OS is found, it is loaded into memory. BIOS drivers are also loaded at this point. These are programs designed to give the computer basic control over hardware devices such as mice, keyboards, network hardware and storage devices.

**Security**

The BIOS can also play a role in computer security. Most BIOS software versions have the option to password-protect the boot process, which means that you must enter a password before any BIOS activity can take place. With the BIOS performing virtually all of its functions during startup, this effectively password-protects the operation of the whole computer. However, resetting a lost BIOS password can be time-consuming and involve working on some of the computer's most sensitive components.

**Hardware**

The BIOS software itself generally resides on a Read-Only Memory, or ROM, or a flash memory chip attached to your computer's motherboard. The location of the BIOS software on the chip is important, as it is the first software to take control of your computer when you turn it on. If the BIOS was not always located in the same place on the same chip, your computer's microprocessor would not know where to locate it, and the boot process could not take place.

## THE BOOTING PROCESS

Booting (also known as booting up) is the initial set of operations that a computer system performs when electrical power is switched on. The process begins when a computer that has been turned off is re-energized, and ends when the computer is ready to perform its normal operations. On modern general purpose computers, this can take tens of seconds and typically involves performing power-on self-test, locating and initializing peripheral devices, and then finding, loading and starting an operating system. Many computer systems also allow these operations to be initiated by a software command without cycling power, in what is known as a soft reboot, though some of the initial operations might be skipped on a soft reboot. A boot loader is a computer program that loads the main operating system or runtime environment for the computer after completion of self-tests.

**The order of booting –**

In order for a computer to successfully boot, its BIOS, operating system and hardware components must all be working properly; failure of any one of these three elements will likely result in a failed boot sequence.

When the computer’s power is first turned on, the CPU initializes itself, which is triggered by a series of clock ticks generated by the system clock. Part of the

CPU's initialization is to look to the system's ROM BIOS for its first instruction in the startup program. The ROM BIOS stores the first instruction, which is the instruction to run the power-on self test (POST), in a predetermined memory address. POST begins by checking the BIOS chip and then tests CMOS RAM. If the POST does not detect a battery failure, it then continues to initialize the CPU, checking the inventoried hardware devices (such as the video card), secondary storage devices, such as hard drives and floppy drives, ports and other hardware devices, such as the keyboard and mouse, to ensure they are functioning properly.

Once the POST has determined that all components are functioning properly and the CPU has successfully initialized, the BIOS looks for an OS to load.

The BIOS typically looks to the CMOS chip to tell it where to find the OS, and in most PCs, the OS loads from the C drive on the hard drive even though the BIOS has the capability to load the OS from a floppy disk, CD or ZIP drive. The order of drives that the CMOS looks to in order to locate the OS is called the boot sequence, which can be changed by altering the CMOS setup. Looking to the appropriate boot drive, the BIOS will first encounter the boot record, which tells it where to find the beginning of the OS and the subsequent program file that will initialize the OS.

Once the OS initializes, the BIOS copies its files into memory and the OS basically takes over control of the boot process. Now in control, the OS performs another inventory of the system's memory and memory availability (which the BIOS already checked) and loads the device drivers that it needs to control the peripheral devices, such as a printer, scanner, optical drive, mouse and keyboard. This is the final stage in the boot process, after which the user can access the system's applications to perform tasks.

## UNIFIED EXTENSIBLE FIRMWARE INTERFACE

The **Unified Extensible Firmware Interface** (**UEFI**) is a specification that defines a software interface between an operating system and platform firmware. UEFI replaces the Basic Input/Output System (BIOS) firmware interface originally present in all IBM PC-compatible personal computers,[1][2] with most UEFI firmware implementations providing legacy support for BIOS services. UEFI can support remote diagnostics and repair of computers, even with no operating system installed.[3]

Intel developed the original **Extensible Firmware Interface** (**EFI**) specification. Some of the EFI's practices and data formats mirror those

from Microsoft Windows.[4][5] In 2005, UEFI deprecated EFI 1.10 (the final release of EFI). The Unified EFI Forum is the industry body that manages the UEFI specification.

**ADVANTAGE**

The interface defined by the EFI specification includes data tables that contain platform information, and boot and runtime services that are available to the OS loader and OS. UEFI firmware provides several technical advantages over a traditional BIOS system:[16]

- Ability to use large disks (over 2 TB) with a GUID Partition Table (GPT)[17][a]
- CPU-independent architecture[a]
- CPU-independent drivers[a]
- Flexible pre-OS environment, including network capability
- Modular design
- Backward and forward compatibility

# DIFFERENCE BETWEEN LVM AND RAID

| S.No. | RAID | LVM |
|---|---|---|
| 1. | RAID is used for redundancy. | LVM is a way in which you partition the hard disk logically and it contains its own advantages. |
| 2. | A RAID device is a physical grouping of disk devices in order to create a logical presentation of one device to an Operating System for redundancy or performance or a combination of the two. | LVM is a logical layer that that can be manipulated in order to create and, or expand a logical presentation of a disk device to an Operating System. |
| 3. | RAID is a way to create a redundant or striped block device with redundancy using other physical block devices. | LVM usually sits on top of RAID blocks or even standard block devices to accomplish the same result as a partitioning, however it is much more flexible than partitions. You can create multiple volumes crossing multiple physical devices, remove physical devices without loosing data, resize the volumes, create snapshots, etc |
| 4. | RAID is either a software or a hardware technique to create data storage redundancy across multiple block devices based on required RAID levels. | LVM is a software tool to manage large pool of storage devices making them appear as a single manageable pool of storage resource. LVM can be used to manage a large pool of what we call Just-a-bunch-of-Disk (JBOD) presenting them as a single logical volume and thereby create various partitions for software RAID. |
| 5. | RAID is NOT any kind of Data backup solution. Its a solution to prevent one of the SPOFs (Single Point of Failure) i.e. DISK failure. By configuring RAID you are just providing an emergency substitute for the Primary disk. It NEVER means that you have configured DATA backup. | LVM is a disk management approach that allows us to create, extend, reduce, delete or resize the volume groups or logical volumes. |