Server Hardening

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.

How does WhatsApp end-to-end encryption work ?

The term 'end-to-end encryption' (E2EE) has entered the common lexical use and is no more restricted to the geeks, thanks to WhatsApp which popularised it and brought it to over a billion users globally. It has become the part of our daily digital life as it is the definitive security mechanism that protects our personal data (messages etc.) such that it can only be read on by the sender, and by the recipient on the other end. No one else, including the hackers or the government, can snoop and read the encrypted data.

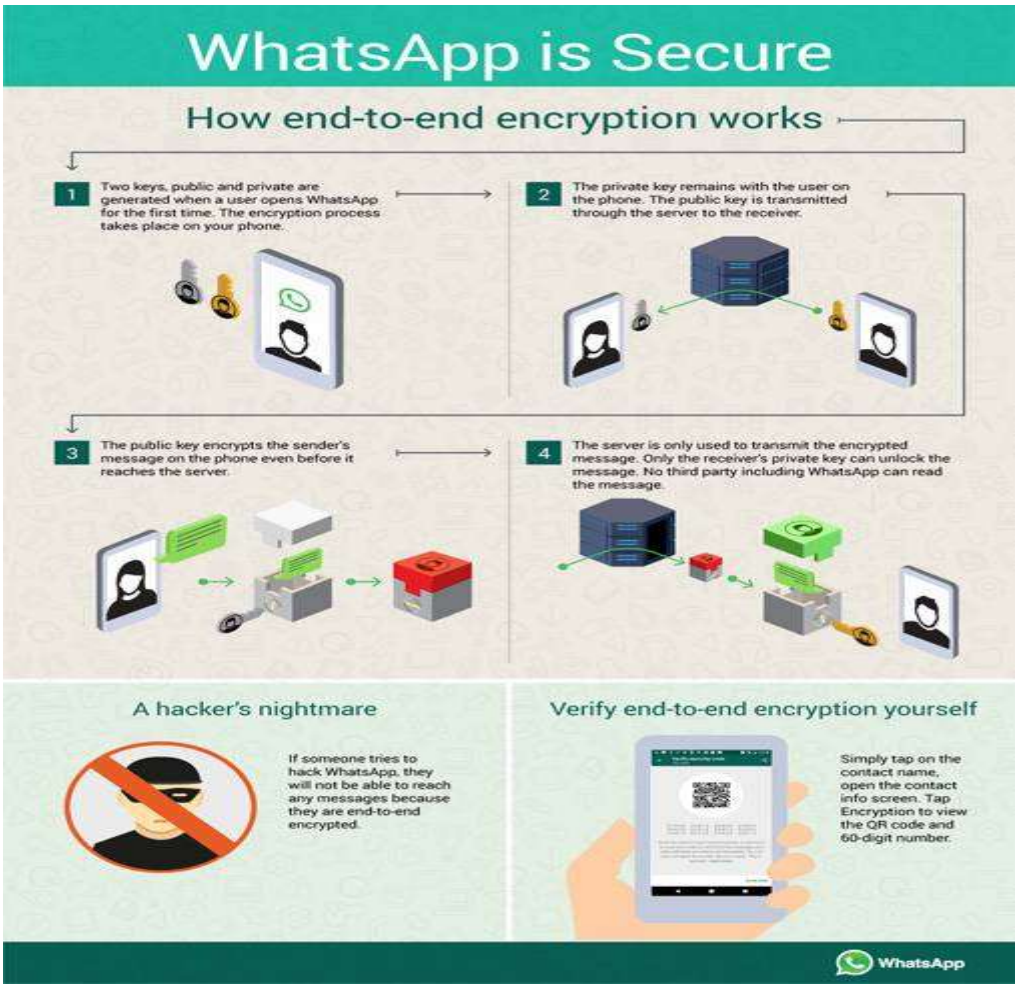
WhatsApp's end-to-end encryption ensures that only you and the person you're communicating with can read what's sent. Nobody in between, not even WhatsApp, can read the messages. The messages are secured with locks, and only the recipient has the special key to unlock and read the messages. WhatsApp uses Signal Protocol developed by Open Whisper Systems. The following steps describes the working of E2EE when two people communicate on WhatsApp.

1. When the user first opens the WhatsApp, two different keys (public & private) are generated. The encryption process takes place on the phone itself.
2. The private key must remain with the user whereas the public key is transferred to the receiver via the centralised WhatsApp server.
3. The public key encrypts the senders message on the phone even before it reaches the centralised server.
4. The server is only used to transmit the encrypted message. The message can only be unlocked by the private key of the receiver. No third part, including WhatsApp can intercept and read the message.

5. If a hacker tries to hack and read the messages, they would fail because of the encryption.

How do I verify that WhatsApp is using end-to-end encryption?

To manually verify the encryption between the sender and the receiver, simply tap on the contacts name on WhatsApp to open the info screen. Now tap on 'Encryption' to view the QR code and 60-digit number. You can scan your contacts' QR code or visually compare the 60-digit number. If you scan the QR code, and if they match, then your chats are encrypted and no one is intercepting your messages or calls.



What is BIOS?

Stands for "Basic Input/Output System." Most people don't need to ever mess with the BIOS on a computer, but it can be helpful to know what it is. The BIOS is a program pre-installed on Windows-based computers (not on Macs) that the computer uses to start up. The CPU accesses the BIOS even before the operating system is loaded. The BIOS then checks all your hardware connections and locates all your devices. If everything is OK, the BIOS loads the operating system into the computer's memory and finishes the boot-up process.

Since the BIOS manages the hard drives, it can't reside on one, and since it is available before the computer boots up, it can't live in the RAM. So where can this amazing, yet elusive BIOS be found? It is actually located in the ROM (Read-Only Memory) of the computer. More specifically, it resides in an eraseable programmable read-only memory (EPROM) chip. So, as soon as you turn your computer on, the CPU accesses the EPROM and gives control to the BIOS.

The BIOS also is used after the computer has booted up. It acts as an intermediary between the CPU and the I/O (input/output) devices. Because of the BIOS, your programs and your operating system don't have to know exact details (like hardware addresses) about the I/O devices attached to your PC. When device details change, only the BIOS needs to be updated. You can make these changes by entering the BIOS when your system starts up. To access the BIOS, hold down the DELETE or F2 key as soon as your computer begins to start up.

What is the purpose of BIOS?

BIOS is an important part of the computer framework. It primary purpose is to initialize and test the system hardware components, as well as to load a boot loader or an operating system from memory.

BIOS is usually pre-loaded on to the motherboard. It is typically loaded onto an erasable programmable read-only memory (EPROM) chip. When the computer it turned on, the BIOS is the first thing to boot up. Firstly, it determines whether all of the attachments are in place and are operational. It then loads the operating system from the hard disk to the random access memory (RAM), after which the RAM takes over.

Boot Process

When you push the power button, power is sent to a small **bootloader program**, which loads the computer's operating system. The bootloader is located in the cache memory. The **cache memory** is a portion of your RAM that is directly attached to the central processing unit (CPU), which is the brains of your computer.

Once the bootloader program gets power, it starts the process of activating the operating system. If you were to see this happening, it would show a black screen with the text of the boot up processes.

During the boot process, the first thing that happens is the **POST** or Power on Self Test. When the POST is running, you will typically see lights flashing and hear a series of beeps. Basically the computer is performing a test to make sure all the attached hardware is communicating clearly with the CPU.

Once the POST is complete, the **BIOS**, or Basic Input/Output System, is activated. The BIOS is actually stored in read only memory (ROM). So, the bootloader program opens or wakes up the BIOS, which then finds the complete loading instructions on a bootable device, typically the hard disk.

Linux Boot Process

Basically "**Boot Process**" A **CPU** gets its **instructions from memory.** The CPU **reads instruction** from the **BIOS** and searches for the **hard disks, CD drives** and **other hardware. The BIOS program looks at the first sector for boot code.**

The boot process for Linux goes through several stages. Different systems follows different stage here in Linux have 6 stages of Boot.

- **BIOS** (Basic Input/Output System)
- **MBR** (Master boot Record)
- **GRUB** (Grand Unified Boot Loader)
- **Kernel**
- **init**
- **run level** (init 0—init 6)

BIOS : The term **BIOS** stands for "**Basic I/O System**", It looks for boot loader in CD Rom, hdd, or other boot-able media. Once the boot loader is detected and loaded into memory, BIOS gives the control to it.

So in simple terms we can say that "**BIOS loads and executes the MBR boot loader** "

MBR : The term **MBR** stands for **"Master boot records"** it is **located in the first sector** of the boo-table disks like /dev/hda or /dev/sda , And it has **three component** and MBR is less than or equal to 512 Bytes in size

1. **Primary boot loader info in 1st 446 bytes**

2. **Partition table info in next 64 bytes.**

3. **MBR validation check in last 2 bytes.**

so, we can say that, MBR loads(Boot info) and execute the GRUB boot loader.

GRUB : As mention above the term GRUB stands for "**Grand unified boot loader**" , if we have multiple images installed on our system we can choose one to be execute, **GRUB display a splash screen, waits for few seconds**, if you don't enter or press anything, it loads the default Kernel image as specified in the **GRUB Configuration file,**

Configuration files:
/boot/grub/grub.conf "it is link with" **/etc/grub.conf** **-** in Red Hat Linux
/boot/grub/grub.cfg "it is link with" /etc/default/grub - in ubuntuB loads.

So, GRUB loads, it configuration file **(** for BIOS system **/boot/grub/grub.conf,** for UEFI System **/boot/efi/EFI/redhat/grub)** and Kernel into System memory. As we will in configuration file, **it contains Kernel & initrd image**. So in simple terms GRUB just loads and execute Kernel & initrd images.

Kernel : The term **Kernel** is the core of an operating system which provides access to services and hardware. So the boot loader loads one or multiple "**initramfs images"** into system memory. [**initramfrs: initial RAM Disk**], The kernel use "initramfs"to read drivers and needed modules for booting the system.

And it mount the “root file system” as specified in the ‘root=(GRUB configuration file).

So we can say that, After the **kernel & initramfs** image are loaded, control of the boot process goes through kernel. The **kernel** now **initializes and configure memory** and contacted **hardware** and **kernel execute** the the**/sbin/init** program ## since ‘**init**’ was the 1st program to be executed by linux kernel. It has the process id (PID) of 1 , we can check through command “ **ps -ef | grep init** “.

init : During this process, the system locates the root partition and file system. Both are checked and mounted then the system start the **init process.** Which runs the **initialization scripts** invoking different scripts in the **“/etc/rc*.d “**directory. so we can say that init execute run level program.

so init execute the Run level program (init 0- init 6) depend on the default init level settings, system will execute. Look at the /etc/init tab following are the available **run level** “**/etc/init.d/rc.*d”**

- init 0 “Halt” /etc/rc.d/rc0.d
- init 1 “Single user mode” /etc/rc.d/rc1.d
- init 2 “Multi user mode without NFS” /etc/rc.d/rc2.d
- init 3 “Multi user mode with NFS” **/etc/rc.d/rc3.d**
- init 4 “unused” **/etc/rc.d/rc4.d**
- init 5 “X11- GUI mode” **/etc/rc.d/rc5.d**
- init 6 “Reboot” **/etc/rc.d/rc6.d**

Typically we would set the default run level to either **init 3 or init 5**

Run level : When the Linux System booting up, Those are the run level programs execute from the run level directory or defined by your run level directory.

Under the /etc/rc.d/rc*.d / directories, you would see program that start with**S & K.**

Program start with S are used during startup S for startup, K for kill.

Windows Boot Process

To begin the boot process, turn on the computer. This is called a cold boot. When the computer is powered on, it performs a Power On Self Test (POST). Because the video adapter has not yet been initialized, errors that occur at this point in the boot process are reported by a series of audible tones, called beep codes.

After POST, the BIOS locates and reads the configuration settings that are stored in the CMOS memory. The boot device priority, is the order in which devices are checked to locate the operating system. The boot device priority is set in the BIOS and can be arranged in any order. The BIOS boots the computer using the first drive that contains an operating system.

Hard drives, network drives, USB drives, and even removable magnetic media, such as CompactFlash or Secure Digital (SD) cards can be used in the boot order, depending on the capabilities of the motherboard. Some BIOS also have a boot device priority menu that is accessed with a special key combination while the computer is starting but before the boot sequence begins. You can use this menu to select the device to boot, which is useful if multiple drives can boot the computer.

Windows Boot Loader and Windows Boot Manager in Windows 7 and Windows Vista

When the drive storing the OS is located, the BIOS finds the Master Boot Record (MBR). At this point, Windows Boot Manager (BOOTMGR) controls several installation steps. For instance, if more than one OS is present on the disk, BOOTMGR gives the user a chance to select which one to use. If there are no other operating systems, or if the user does not make a selection before the timer expires, the following process occurs:

1. WinLoad uses the path specified in BOOTMGR to find the boot partition.

2. WinLoad loads two files that make up the core of Windows 7: NTOSKRNL.EXE and HAL.DLL.

3. WinLoad reads the Registry files, chooses a hardware profile, and loads the device drivers.

NOTE: If another OS version is on the disk that is Windows Vista or later, BOOTMGR repeats the process. If another OS version is on the disk that is Windows XP or earlier, BOOTMGR invokes the Windows XP boot loader (NTLDR).

NTLDR and the Windows Boot Menu in Windows XP

When the drive with the OS is located on a computer running Windows XP, the BIOS locates the MBR. The MBR locates the OS boot loader NTLDR. At this point, NTLDR controls several installation steps. For instance, if more than one OS is present on the disk, BOOT.INI gives the user a chance to select which one to use. If there are no other operating systems, or if the user does not make a selection before the timer expires, the following process occurs:

1. NTLDR runs to get information about the installed hardware.

2. NTLDR uses the path specified in the BOOT.INI to find the boot partition.

3. NTLDR loads two files that make up the core of XP: NTOSKRNL.EXE and HAL.DLL.

4. NTLDR reads the Registry files, chooses a hardware profile, and loads the device drivers.

NT Kernel

At this point, the NT kernel takes over. The NT kernel is the heart of all Windows operating systems. The name of this file is NTOSKRNL.EXE. It starts the login file called WINLOGON.EXE and displays the Windows Welcome screen.

What is the difference between LVM and RAID?

- **RAID:**
  - RAID is used for redundancy.
  - 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.
  - RAID is a way to create a redundant or striped block device with redundancy using other physical block devices.
  - RAID is either a software or a hardware technique to create data storage redundancy across multiple block devices based on required RAID levels.
  - 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:**

- LVM is a way in which you partition the hard disk logically and it contains its own advantages.
- LVM is a logical layer that that can be anipulated in order to create and, or expand a logical presentation of a disk device to an Operating System.
- 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
- 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.
- LVM is a disk management approach that allows us to create, extend, reduce, delete or resize the volume groups or logical volumes.