Hi there, So let get started...Our ancestors started off with the barter system - something like "I will give you 2 goats in return for 5 shiny Perl". Soon they realised that the barter system had too many limitations - everyone didn't want goat, goats were neither divisible (not too many people would want 1/3rd goat or 0.35 goats) nor very portable (imagine having to carry a goat on your shoulders while going shopping).

So they moved on to more acceptable, divisible, homogeneous and portable forms of money - cowry shells, salt, gold, silver and lots more. The Chinese invention of paper eventually led to the birth of paper currency, which was initially backed by gold or other precious metals. Then the world moved on legal tender by government that is fiat money (currency declared as a legal tender but not backed by physical commodities like gold etc.)

Now, Have a look at a 100-rupee note. It carries a promise signed by the Governor of the Reserve Bank of India (RBI) – "*I promise to pay the bearer the sum of one hundred rupees*". If you were to take this note to the Governor of the RBI, he would give you coins
or one-rupee notes totalling 100 rupees. The RBI gets the power to issue currency notes by section 31 of the Reserve Bank of India Act, 1934. This section states that "*No person in India other than the Bank or, as expressly authorized by this Act, the Central Government shall draw, accept, make or issue any bill of exchange, hundi, promissory note or engagement for the payment of money payable to bearer on demand, or borrow, owe or take up any sum or sums of money on the bills, hundis or notes payable to bearer on demand of any such person…*"
What it essentially means is that a 100 rupee note or a fiat currency is just a legal tender and not backed by any physical assets or commodities like gold which it used to be in the fractional reserve system.

This brings us to an essential question – *what is money*? So let us understand what is money? Something to be called money it must function as a medium of exchange, a measure of value, a standard of deferred payment and a store of value.

The birth of computers and the Internet brought in many electronic payment systems including debit cards, stored value cards, gift cards, giro transfers, credit cards, net-banking, electronic bill payments, electronic cheques, paypal, paytm, mobile wallets, digital gold currencies, digital wallets, electronic funds transfer at point of sale, mobile banking, SMS banking, online banking, payment cards, real-time gross settlement systems, SWIFT, wire transfers and more.

And then came Satoshi Nakamoto's revolutionary whitepaper - *Bitcoin: A Peer-to-Peer Electronic Cash System* in October 2008. This brought the world its first
truly peer-to-peer electronic currency. Bitcoin earned a lot of bad name primarily because of its use by members of Silk Road which is now shut-down. A *Silk Road* – was an illegal online marketplace that facilitated the sale of hundreds of millions of dollars worth of drugs, guns, stolen financial information, counterfeit documents and more. All Silk Road transactions were conducted exclusively in bitcoin. That's why it got a bad reputation and various governments resisted it use. The underlying powerful technology behind Bitcoin is Blockchain.

Let us understand Blockchain Technology:

I am sure you would have used library at least once. When you borrow books from the Library the Librarian Notes down the date, time, book serial number (record number) and finally impound rubber stamp on the last page (a permanent record). Each page in the book is like a node in the Blockchain network. If you try to alter any one page or remove or replace a place the change is noticeable. Hence, you cannot tamper it. Same way Blockchain works.

> A Blockchain is essentially a distributed database of records or public ledger of all transactions or digital events that have been executed and shared among participating parties. Each transaction in the public ledger is verified by consensus of a majority of the participants in the system. And, once entered, information can never be erased. The Blockchain contains a certain and verifiable record of every single transaction ever made. To use a basic analogy, it is easy to steal a cookie from a cookie jar, kept in a secluded place than stealing the cookie from a cookie jar kept in a market place, being observed by thousands of people.

ADDRESS:

If somebody wants to post you a letter... obviously it will be posted to a particular address. This address includes the name and the destination to which the letter should reach.

Similarly, a combination of Alpha-Numeric (English Alphabets plus some random numbers) Characters looks something like this:

1PvBMSEYstVetqTFn5Au4m4GFg7xJaNVN2
Or
1PvPMSEYstVetqTFn5Au4m4GFg7xJaTRN2

- This address can vary from 26 characters to 32 characters or in be bit longer up to 33 characters or slightly shorter.
- It is always case sensitive
- It can be one time address or multiple use address
- This can even be generate offline
- Address is like a bank account number where you can send and receive payments
- Just by creating address it does not have any balance
- Bitcoin is separate from address.
- It not necessarily identifies the sender and receiver.

TIMESTAMP:

A timestamp indicates when the event occurred. The timestamp should be trusted and free from modification.

LEDGER:
It is a record of transactions. Ledger can be private ledger

(running on private nodes) or a public ledger (running on public nodes).

TRUST MACHINE = SHARED LEDGER
A shared ledger solves the problem of trusted third parties.

DISTRIBUTED LEDGER: multiple peers/systems/nodes stores the copies of Blockchain records. If changes are made – it needs to be approved by Blockchain Network.

PROOF OF WORK (PoW):

Competition is Healthy… Right? Proof-of-work is a competition among the miners (One who add transaction records to Bitcoin public ledger) to find the next correct hash by using enormous computation power.

PROOF-OF-STAKE (PoS):

Stake represents the ownership or share. The bigger investor will have a bigger stake and vice-versa. In Blockchain, miners which a larger consensus or stake have a greater opportunity to mine more blocks. Hence, they get bigger stake or reward.

Practical Byzantine Fault Tolerance (PBFT):

Practical Byzantine Fault Tolerance (PBFT) algorithm provides a solution to the **Byzantine Generals Problem (***Reliable computer systems must handle malfunctioning components that give conflicting information to different parts of the system. This situation can be expressed abstractly in terms of a group of generals of the Byzantine army camped with their troops around an enemy city. Communicating only by messenger, the generals must agree upon a common battle plan. However, one or more of them may be traitors who will try to confuse the others. The problem is to find an algorithm to ensure that the loyal generals will reach agreement. It is shown that, using only oral messages, this problem is solvable if and only if more than two-thirds of the generals are loyal; so a single traitor can confound two loyal generals.***)** that works in asynchronous environments like the Internet. PBFT involves a three-phase protocol and the conception of a primary (leader) node which acts as the block miner. The leader can be changed by the voting mechanism by the rest of the network if it exhibits arbitrary behavior. PBFT works on the assumption that less than one-third of the nodes are faulty (f), which say that it requires at least 3f +1 nodes.

A series of Cyber attacks has raised concerns on the security aspects of Blockchain.

People often tend to confuse Bitcoin or other Cryptocurrency with Blockchain.

Blockchain is the permanent / immutable distributed network : powered by miners and secured by strong cryptography.

CRYPTOGRAPHY:

It is an art to write secrete information. This can't be read by humans. Internet is un-trusted place – so to communicate (over a un-trusted or un-secured channel) we need cryptography.

Few notable points:

- Confidentiality: None of us can read the message sent. So it ensures confidentiality.

- Authentication: It authenticates via trusted identity.

- Integrity: It ensures that message sent is protected from any alternation.

- Non-repudiation: It is process to proof that the message is actually sent by sender and no one else.

Story of corporate lover to understand cryptography:

Heena is a notorious lover who's is planning to bung office for a few days to hand out with his lover but doesn't want the spying boss to know. She receives phone call from her lover "have you bought the pen for gift" meaning "let's go out tomorrow". When she says something about "3 pens" meaning "tomorrow at 3 pm"

What we just saw was cryptography (secrete means to communicate in real life).

"have you bought the pen for gift" – plain text

"let's go out tomorrow" – cipher text (something that heena's boss will never understand)

This process of converting plain text into hypertext is encryption and reverse is decryption.

It is believed that when Julius Caesar sent messages to his generals, he replaced every A in his messages with a D, every B with an E, and so on through the alphabet. Only someone who knew the "shift by 3" rule could decipher his messages.

For example, if we want to encode the word "SECRET" using Caesar's key value of 3, we offset the alphabet so that the 3rd letter down, begins the alphabet.

Using this scheme, the plaintext, "SECRET" encrypts as "VHFUHW". To allow someone else to read the cipher text, you tell him or her that the *key* is 3. This method is called *symmetric cryptography* and involves using the same key for encrypting as well as decrypting a message. This naturally poses a serious problem – what if an adversary gets hold of this key? At some point of time the sender and receiver need to exchange the key. That's when an adversary could get hold of the key. In modern cryptography, keys are really really large numbers.

The *secure-key-exchange* problem was solved with the birth of *asymmetric key cryptography* – in which two different but related keys are used - the public key to encrypt data and the corresponding private key to decrypt the data. If heena were to send an encrypted message to her lover, she would encrypt the message using his *public key* (which is available to the world). Once encrypted, the message can only be decrypted using lover's *private key* (which would only be available to his lover).

Let us understand few concepts before we drive into Blockchain security. **Hash functions**. A one-way *hash function* takes an input (e.g. a PDF file let say containing 5000 pages, a video of any size, an email, a string etc.) and produces a fixed-length output e.g. 160-bits.

The hash function ensures that if the information is changed in any way – even by just one bit – an entirely different output value is produced. The table below shows some sample output values using the sha1 (40) hash function.

http://www.sha1-online.com/

Prakash	17ca5e5eeed3eb94d9c8deac7751ce3606892a75
prakash	a0c130080eb83e6eeecb78320e4131e773a90852
Prakash (one space at the end)	f6dd5acd292c41480ed05915ef9cd6790e854d1e

What must be kept in mind is that irrespective of the size of the input, the hash output will always be of the same size.

Two things must be borne in mind with regard to one-way hash functions:

1. It is computationally infeasible to find two different input messages that will yield the same hash output.
2. It is computationally infeasible to reconstruct the original message from its hash output.

Having understood hash functions, let's have a look at another interesting concept called **proof-of-work**. This is a way to reduce spam and denial of service attacks by requiring a computer to spend some time and processing power to solve something.

The ender has to spent reasonable time and computational power to solve some puzzle (before sending an email), let us for now assume that the sender is not a spammer. The logic is that spamming would be economically infeasible if a spammer had to spend non-trivial time and computational power for every single email being sent because most spammers send bulk emails on a single click. Let's say it takes 10 seconds to solve this puzzle before sending an email. This may not take a genuine sender a lot of time and computational power but if a spammer were to make these calculations for millions of emails, it will take a huge amount of time (10 seconds x 10,000 emails = 1,00,000 seconds approx. 27 hours) and computational power.

At the receiver's end, it will simply take the header of the email hashed. If hash begins with 4 zeros not a spam and receiver will be able to view the email.

A very important application of public key cryptography is a **digital signature is explained via email spamming example we just discussed**. In this, the signer first calculates the hash of the message wants to digitally sign. Then using private key and the hash, creates a digital signature, using the relevant algorithm.

This digital signature is unique to the message.

The signer then sends the message and the digital signature to the receiver. The receiver re-computes the hash from the message. The receiver also computes another string using the digital signature and the signer's public key (using the relevant algorithm). If this string and the hash match, the digital signature is verified.

Now let us understand block, Genesis-block, Merkle Root/tree, Mining...

GENESIS –BLOCK:

The first block on Blockchain is genesis-block. Example of original genesis-block:

000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1 b60a8ce26f

Image a tree. The root from the seeds comes first then the plant grows and expands its branches. The plant gradually becomes tree.

Here, root of the plant is genesis-block and tree (also known as Merkle tree / Merkle root) is the collection of many blocks connected to root in a Blockchain network.

A block of one or more new transactions is collected into the transaction data part of a block. Copies of each transaction are hashed, and the hashes are then paired, hashed, paired again, and hashed again until a single hash remains, the **merkle root** of a **merkle tree.**

When a block is successfully mined, new bitcoins are generated in the block and paid to the miner. This mining bounty is large - currently 25 bitcoins per block. In addition, the miner gets any fees associated with the transactions in the block. Because of this, mining is very competitive with many people attempting to mine blocks. The difficulty and competitiveness of mining is a key part of Bitcoin security, since it ensures that nobody can flood the system with bad blocks.

There is no centralized Bitcoin server. Instead, Bitcoin runs on a peer-to-peer network. If you run a Bitcoin client, you become part of that network. The nodes on the network exchange transactions, blocks, and addresses of other peers with each other. When you first connect to the network, your client downloads the Blockchain from some random node or nodes. In turn, your client may provide data to other nodes. When you create a Bitcoin transaction, you send it to some peer, who sends it to other peers, and so on, until it reaches the entire network. Miners pick up your transaction, generate a A Bitcoin **address** is an identifier of 26 to 35 alphanumeric characters, beginning with the number 1 or 3, which represents a possible destination for a bitcoin payment.

A Bitcoin wallet does not store bitcoins. The wallet is a collection of public-private key-pairs.

As discussed, the Blockchain is a database of transaction information. It is constantly growing and is sent out to all nodes in the Bitcoin network. Every transaction is distributed to the network and all valid transactions are included in the next block, which is mined.

Imagine a real-world transaction where your salary is transferred to your bank account through an online transfer made by your employer. You then use your debit card to pay for dinner. This transfers some of the money to the restaurant's account. In these 2 transactions, did you see a single currency note? No. So we can say that in today's world most money exists as a history of transactions and balances.

Bitcoin, or for that matter most virtual currencies work the same way. They don't actually "exist" in the true sense of the word. They just are there!

Possibility is an "eclipse attack." Nodes on the Blockchain must remain in constant communication in order to compare data. An attacker who manages to take control of one node's communications and fool it into accepting false data that appears to come from the rest of the network can trick it into wasting resources or confirming fake transactions.

Blockchain systems connect with the real world—for example, in software clients and third-party applications.

https://www.quantstamp.com/
https://hosho.io

Blockchain Security:

While considering the security of the Blockchain following points must be kept in mind:
Access Control, hardening of Server, Network Security, Permission Management, Regulating Security Parameters, Secure backup of Keys etc
Now let us understand each of this in detail:

Access control is a security technique that can be used to regulate who or what can view or use resources in a computing environment. Too Technical right? ... Let me put it this way... Access control is a selective restriction of access to data. Stop access to critical or valuable resources.

There are two main types of access control: physical and logical. Physical access control limits access to campuses, buildings, rooms and physical IT assets. Logical access limits connections to computer networks, system files data and database.

who they claim to be = Authentication

who they say they are = Authorization (allowed or not)

An access control also includes:

Control channel configuration/creation (shared configuration is stored in one particular channel which contains configuration of each transaction collectively. It contains version information, Permission or associated policies, and root configuration),

Access to channel participation: A channel is like a virtual Blockchain network that sits on top of a physical Blockchain network with its own access rules. Channels employ their own transaction ordering mechanism and thus provide scalability, ultimately allowing for effective ordering and partition of huge amounts of data. -Hyperledger fabric.

Also, Proper administration, deployment, execution is necessary.

Suggested to Store; Access Control Policies in the Blockchain.

The easy target for any company or organization is its database. Most of the hackers get access to database of the company. Access control can control the access or restrict the access to the database. An Apt access control will not only restrict the APIs to access what is required but also restrict what isn't required.

Blockchain Node Interactions:
Node is a computer that is participating in the Blockchain network in p2p fashion. It allows the nodes to communicate with the other computers in the Blockchain network. Their main role is to replicate or propagate transactions and blocks in the Blockchain network. There are about 10k nodes that are accessible in the Blockchain network. Some are invisible or hidden. The node

that participates in the transaction is named a full node or a node that fully validate a valid the transaction. The node also keeps a full copy of the ledger of the Blockchain transactions may be on a storage device like hard disk or cloud storage. It connects random peers in the network and remembers it doesn't trust these peers. It monitors these peers for the transaction and verifies if it is a genuine or a malicious one. Also, it verifies that there is no double spend of the same transaction. If malicious node is detected it will stop talking to you like a rude girlfriend and blacklist your number here the IP address of the node. A miner gets the transactions and node validates it.

There is a User Interface then the app server running web apps or applications, then come the Blockchain Node. The node shouldn't in any circumstances be allowed to interact with the database. A compromised node will not only damage the functioning of your organisation but also will compromise the critical sensitive database or records.

Hardening of Server:
Harderding of server is usually the process of securing the server and reducing the attack surface.

Reducing available ways of attack typically includes changing default passwords, the removal of unnecessary software, unnecessary usernames or logins, and the disabling or removal of unnecessary services.

Now let us understand ways to secure server:

SSH (Shell Secure) - Secure way to administer remote server.
-Symmetric Encryption - Encrypt & Decrypt data same key. In SSH, it encrypts entire data let me explain: the server and client establishes secure session based key connection to guard against spoofing or man-in-middle attacks. Examples of symmetric encryption include: AES, 3DES, Blowfish, CAST128, and Arcfour etc.

-Asymmetric Encryption - Encrypt by public key and Decrypt by private key.
The client creates what is known as authorization key pair and then it uploads or sends the public key to access/connect to remote server. Now, inoder to grant access authentication is required and is supplied by private key. The Cryptographic hash is used to check data integrity and it is one-way function meaning no one can find or reverse the hashes to find the hidden data.

Note: SSH connection with keys is authorized to access/connect remote server whereas password based authentication is rejected.

Firewall: It allows or rejects network traffic based on the firewall rules. It can be software or hardware based. We should keep or expose only limited software/apps or ports required to run particular software or app; this will reduce the attack surface and vulnerabilities.

Block traffic on unused server IP addresses - helps reduce the risk to your server

Alert when end-user scripts sending excessive emails per hour - for identifying spamming scripts

Suspicious process reporting - reports potential exploits running on the server

Excessive user processes reporting

Excessive user process usage reporting and optional termination

Block traffic on a variety of Block Lists including DShield Block List (http://feeds.dshield.org/block.txt) and Spamhaus DROP List (http://www.spamhaus.org/drop/)

BOGON packet protection - A bogon is an bogus IP address from the bogon space.

Pre-configured settings for Low, Medium or High firewall security (cPanel servers only)

Allow Dynamic DNS IP addresses - always allow your IP address even if it changes whenever you connect to the internet

SYN Flood protection - A SYN flood is a form of denial-of-service attack in which an attacker sends a succession of SYN requests to a target's system in an attempt to consume enough server resources to make the system unresponsive to legitimate traffic.

Ping of death protection - On the Internet, ping of death is a denial of service (DoS) attack caused by an attacker deliberately sending an IP packet larger than the 65,536 bytes allowed by the IP protocol. In other words, A ping of death is a type of attack on a computer system that involves sending a malformed or otherwise malicious ping to a computer. A correctly-formed ping packet is typically 56 bytes in size, or 64 bytes when the Internet Protocol header is considered.

Port Scan tracking and blocking-a firewall alone won't completely protect you against port-scanning activity. The attacker will be able to detect ports that you've intentionally exposed to the Internet, and these can provide valuable reconnaissance information for a future attack.

The best line of defense against port scanning threats is a good intrusion prevention system (IPS).

Port Flooding Detection - Per IP, per Port connection flooding detection and mitigation to help block DOS attacks

Firewall Daemon, or simply Firewalld, is a program to control the firewall scripts rules in a TCP/IP network router.(http://www.pads.ufrj.br:443/Firewalld.old/Firewalld.html)

Review Processes and Remove Extra Software:
Know everything that runs on the server, why, and which users have access.
Disable any and all unused services.
Install anti-virus software and make sure it stays current, is running actively, and is generating logs

Control Access to Server:
Remove, disable or change passwords to default accounts.
All passwords should be strong passwords; they should also be unique, and changed periodically.
Disable "guest" accounts/access.
For data center-hosted servers, all administrative accounts must be SUNet ID accounts.
Review user accounts regularly and remove inactive accounts.
Keep a complete list of everyone who has access to the server, and make sure you know who has which read/write privileges.
No open file-sharing is allowed.
All remote access should be restricted to specific IP addresses, and encrypted from end to end (via VPN and Private Network): Only available to certain users or servers through virtual interface.

Root Access Restricted/Blocked: Block Root access, .htaccess is a configuration file used by apache web server, cPannel IP Address.
Disable Password Based Authentication – Replace with key based authenication
Disable access to Configuration Files.
Disable direct root login in SSH.
It's not advisable to ssh into your server as superuser(root). We should disable ssh as root user on the server, but before doing so, let's create a user with sudo powers so that you can ssh into the server and perform administrative tasks. Once you are logged into the server, you can always switch user to root, if needed.

File Permissions must be regulated:
Users can perform read (r), write (w) and execute (x) operations on files/directories. Here's how the three permission types breaks down when applied to directories and files:

Directories
read - ability to read contents of a directory
write - ability to rename or create a new file/directory within a directory (or delete a directory)
execute - ability to cd into a directory

Files
read - ability to read a file
write - ability to edit/write to a file (or delete a file)
execute - ability to execute a file (such as a bash command)

For any file and directory, we can define how users (u), groups (g) and others (o) can interact with a directory or file. Here's how that breaks down:

User - The permission for owners of a file or directory
Group - The permissions for users belonging to a group. A user can be part of one or more groups. Groups permissions are the primary means for how multiple users can read, write or execute the same sets of files
Other - The permissions for users who aren't the user or part of a group assigned to a file or directory
Note: That we can change a file or directory permissions with the chmod command.

Shadow Admin User: Shadow admin are privileged accounts that hide in the crowd of thousands of non-privileged accounts or less privileged admin accounts.

Security Updates and Security Patches:
Keep your version of the operating system up to date; you should be running the most recent stable version.
Install the all the latest security patches for your system and applications.

Disable unnecessary services: FTP, Webservers, outdated protocols, open ports, Disable Bluetooth, Wifi, Disable or remove applications (apps) and plug-ins that you don't actively use.
Know everything that runs on the server, why, and which users have access.
Disable any and all unused services.
Install anti-virus software and make sure it stays current, is running actively, and is generating logs.

Verifying OpenSSL version
Disabling IPv6:
Both IPv6 and its earlier version of IP is, IPv4, are used to transfer communications on the Internet. Most modern operating systems use IPv6 by default. If IPv6 is enabled on your device, but not supported by other systems/networks to which you are communicating, some OSes will attempt to pass IPv6 traffic in an IPv4 wrapper using tunnelling capabilities such as Teredo, 6to4, or ISATAP (IntraSite Automatic Tunnel Addressing Protocol). Because attackers could use these tunnels to create a hidden channel of communication to and from your system, you should disable tunneling mechanisms.

Disable /tmp, /var/tmp, and /dev/shm partitions (linux/Unix):
Since /tmp, /var/tmp and /dev/shm are world writable directories, if left unlocked anyone can read/write/execute anything from these directories and it becomes a major security concern.

With /etc/fstab you can limit what can be done in these partitions: if you see 'defaults' beside the /tmp line, remove it and replace it with 'noexec, nosuid'. This will stop any executables from being allowed to run.

Do the same for /dev/shm and make /var/tmp a shortcut (symbolic link) to /tmp.

Securing tmp partition :
Securing or hardening tmp involves a large role in securing your server from external attacks. All applications use /tmp directory to store the data temporarily. There is a chance to attack the server using Trojans if it's not secured properly. Temp hardening restricts all activities on /tmp. This prevents the attacker from executing code within the /tmp folder. The hacker tries to inject malicious scripts into /tmp folder through the web application exploit and they try to execute this file on the server bringing it down. When we harden the /tmp folder using nexus mode the user will not able to execute the script and it will prevent these types of attacks.

Securing server against bash vulnerability: Bash is a Unix Shell, installed by default with command-line-interface, provides an interface for the end user to execute commands. This vulnerability executes arbitrary command on Servers. This can be dangerous.

Setting up swap partition:
Swap space is the area on a hard disk which is part of the Virtual Memory of your machine, which is a combination of accessible physical memory (RAM) and the swap space. Swap space temporarily holds memory pages that are inactive. Swap space is used when your system decides that it needs physical memory for active processes and there is insufficient unused physical memory available. If the system happens to need more memory resources or space, inactive pages in physical memory are then moved to the swap space therefore freeing up that physical memory for other uses. Note that the access time for swap is slower therefore do not consider it to be a complete replacement for the physical memory. Swap space can be a dedicated swap partition (recommended), a swap file, or a combination of swap partitions and swap files.

Update your OS regularly – as in within hours of critical updates.
Update your control panel regularly.
Reduce information disclosure, e.g. changing ServerTokens in Apache.
Don't install software that is not used.
Don't store backups or old versions of software on the production system.
Make sure you log all admin level accesses with date, times and usernames and are working properly.
Remove default accounts in MySQL.
Disable unused services.
Maintain backups & Test backups.
Do not do development activities on production systems.
Stay updated with subscriptions to security notification services.
Monitor web traffic for unusual activity.
Conduct regular, remote security scans & local security scans.
Harden default service settings in Apache, SSH and other services.
Use root account only when required.
Use sudo to grant others root level access.
Enable SELinux if possible.
Use private networks for internal server traffic.
Do or Conduct password audits.

Secure Console Access:
You must protect Linux servers console access by disabling the booting from external devices such as DVDs / CDs / USB pen after BIOS setup. Also ,Set BIOS and grub boot loader password to protect these settings.

Disable Ctrl+Alt+Delete in Inittab:
Hitting Ctrl+Alt+Delete will take your server to rebooting process. So this is always advisable to disable this as someone can mistakenly reboot the system.

Network Security:
Network security is about computer systems and network access control, as well as detection and response to unwanted incursions. The risks from poor security are tremendous: theft, interruptions of service, physical damage, compromised system integrity and unauthorized disclosure of proprietary corporate information.

Guard against the compromised or malicious web servers is a common attack vector, Firewalls, intrusion detection and intrusion prevention systems, web application firewalls, anti-malware, Intrusion detection appliances, visibility monitoring, forensics tools, logging and alerting systems some important components of network security.

Let us understand the threat diagram...

Encrypt Your Sensitive Data:

Where practical, employ strong encryption to protect data at rest and in motion

Federal Information Processing Standard 140-2 provides recommendations on employing a wide variety of encryption schemes

Don’t implement what you monitor Encrypt Your Sensitive Data Using encryption can help prevent “man-in-the-middle attacks” and reduce the risk of data loss through lost or stolen equipment

Implement SSL/TLS to protect web sessions both internally and externally

Implement a rich network topology to create security enclaves within your environment

Understand which ports, protocols and services are used within your environment and restrict where it makes sense

Baseline Control in Network Security:

The baseline helps you control your enterprise by giving you a snapshot to refer back to as your systems change and grow, if there is divergence then there may be a need to either enforce the standard or to evaluate the need for a change to the baseline

Next-generation firewall: Small businesses can purchase a next generation firewall for under $1,000. These firewalls include the following additional security services like:

Filtering out web sites containing malicious content

Protection from Internet-based viruses: from other malware entering the network.

Threat prevention technology: that examines network traffic flows to detect and prevent Internet-based vulnerabilities from entering the network.

Segment the organization's internal network: to limit access by users to only those services that they require for business use.

Implement a Network Access Control: solution to prevent unknown computer systems from communicating with the organization's network.

Remove Unused Network-Facing Services*:* Most Linux distributions install with running network services which listen for incoming connections from the internet, the loopback interface, or a combination of both. Network-facing services which are not needed should be removed from the system to reduce the attack surface of both running processes and installed packages.

- Employ IDS to monitor network traffic
- Implement Firewall Security Controls
- Disable Ping
- White list only necessary ports

Permission Management:

Direct Control over the extent to which permission is granted.

Let us look at Multichain Permissions along with what they do...

- ❑ Various Privileges /permissions like SEND, RECEIVE, ADMIN, ACTIVATE, MINE needs to carefully Permissioned.
- ❑ Add / Remove Permissions by chain of multiple admin users. So that you can be sure that it can't be messed up.

Regulating Blockchain Parameters:

The strength of blockchain comes from it decentralised nature; it is able to streamline, protect, and order, operational processes and activities, then enable the anonymous exchange of digital assets. The 'operations on digital assets' happen using 'wallets'. These can be core wallets (requiring the full blockchain to be downloaded to your computer) or lightweight wallets (that store the blockchain on their servers and broadcast all transactions to the network). In either case, a wallet is capable of holding private keys that actually allow the digital asset exchange.

Importantly, there are several different **types of blockchain** implementation that differ in application and the access rights it has to manipulate the database or ledger (if the blockchain contains financial transactions).

Public blockchain: This is where anyone can read the blockchain, anyone can send transactions to it (and see them included if valid), and anyone can participate in the consensus process.
Consortium blockchain: This where the consensus process is controlled by a pre-selected set of nodes. For example, one might imagine a consortium of 3 entities, Node A, Node B and a Regulator Node, where all 3 nodes must sign every block in order to prove them valid. Read rights are either public for anyone to view or restricted to just the selected participants.
Private blockchain: This is where write rights are kept centralised to one organisation. Read rights may be made public or restricted to any arbitrary extent the organisation deems necessary.

Other Types of Blockchain
It is a given that blockchain represents a distributed digital ledger. However, there are other implementations that actually avoid chaining blocks together. For example -

Directed Acyclic Graph: This is where transactions are chained together instead of blocks
Permission-Based Ledger: This ledger works without chaining transactions or blocks! Ledger manipulations are permitted only by trusted authorities.
Payment Channels: Micropayments are settled off-chain in specific payment channels, once that are accumulated, they are then presented to the blockchain.

Formula for calculating Block Size:

Block Size (MB) = Header size (bytes) + Transaction Size (bytes) X Transactions in a Block

Formula for calculating Total Mining Power:

Total Mining Power (Hashes Per Second) =

Summation of nodes in the system (Hashes Per Second) X Mining Power of given Node

Formula for calculating Inter Node Time :

Inter Node Time (Seconds) =

Block Size (MB per second) / Bandwidth between two Nodes Where one node isn't Equal to the Second Node

Formula for calculating System Width :

System Width (Seconds) = Max {Inter Node Times in Seconds}

Formula for calculating Transaction Per Second:

Transaction Per Second =

Block Frequency (Blocks per Minute) x Transaction per Block

The target time interval between consecutive Block should be less: Let us analyse as to Why was the target block time chosen to be 10 minutes?

A 10 minute block is simply a compromise:

Shorter block time:

PRO - Faster 1 confirmation time (to protect from 0-confirm double spend)
PRO - Less payout variance for miners (less reliance on large pools)
CON - Requires increased bandwidth (inter node communication)
CON - More forks, longer forks, and longer re-org time
CON - A greater portion of the raw hashpower is wasted, resulting in lower effective security.

A longer block interval target of longer than 10 minutes:

Longer block time:
PRO - Requires increased bandwidth (inter node communication)
PRO - More forks, longer forks, and longer re-org time
PRO - A greater portion of the raw hashpower is wasted, resulting in lower effective security.
CON - Slower 1 confirmation time (Cannot protect from 0-confirm double spend)
CON - More payout variance for miners (More reliance on large pools)

Here we can see that with a longer block interval target of longer than 10 minutes, the pros and cons is reversed.

The major benefit of a shorter block time is the reduced 1 confirm time. While a quicker block's 1 confirm transaction has less strength than a longer block's 1 confirm transaction it is still better than any block's 0 confirm transaction.

The speed of 1st confirm may seem to be a huge benefit but in reality for most low value and time sensitive transactions such as the buying a cup of coffee, paying for a taxi, or using a vending machine, the risk of double spends is very low. Keep in mind that accepting credit cards is not without risk however merchants have long accepted they will face some losses however if those losses are minimal then it can just be seen as a cost of doing business. So many merchants could simply accept 0-confirm transactions without exposing themselves to more risk than they do from credit card fraud.

> The other factor that diminishes the real world potential of shorter target block intervals is that for many merchants, even "faster" confirmation times still isn't fast enough. For a Point of Sale transaction, an average confirmation time of 2 minutes is still significantly longer than what most merchants would consider to be workable. The average credit card transaction takes about 20 seconds (including delays by customer). The entire industry has spent significant resources to shave even a few seconds off. Changes like enabling customer to swipe card, swiping before all items have been rung up, and not requiring signatures on low value are all about shaving a couple seconds off an already quick process and the cost of those changes is considered acceptable in order to slightly improve the efficiency of a checkout.
>
> The other factor is that reducing the target interval only reduces the average confirmation time but half of them will be longer and the tail can be very long. Due to the random nature of blocks solutions about 15% of blocks will take longer than 2x the target, 3% longer than 3x the target and >7.5 minutes and about 0.5% will take longer than 4x the target. That uncertainty makes it difficult for a time sensitive business to as a matter of policy wait for confirmations. Having most transactions confirm in 30 seconds but some take minutes is going to lead to customer frustration at the point of sale.
>
> If the BTC economy grows large enough we could see expanded use of "green addresses" to fill the need for instant acceptance without confirmations. Such services could be provided by major corporations, and backed by insurance against fraud (for a small per transaction fee). This would be a more viable 0-confirm solution than a simple reduction of the block interval.
>
> That being said the 10 minute target was probably overly conservative and there are some advantages to a shorter block time.

On average, it takes about 10 minutes to find each block. The average block time can actually be slightly shorter or longer depending on if the total hash power of the Bitcoin network is growing or shrinking. Ignoring this detail though, this is why 6 confirmations take about 1 hour on average. In bitcoin, the expected block time is 10 minutes, while in ethereum it is between 10 to 19 seconds.

Now, two questions arise: Does mining diversity make it easier to tamper with the blockchain? If a node can predict that it is about to mine the next block, does it make it easier to tamper the blockchain?

The node which mines the next block can tamper in two ways:
Deliberately decide to censor a transaction, delaying its confirmation until a block generated by an honest miner.
Playing favorites when deciding which of two conflicting transactions should be confirmed.

Mining Diversity:
It is commonly used in multi-chain. Concept Illustrated:
In the Multi-Chain (mining diversity scheme), whitelisted miners add blocks to the chain in a round-robin manner, with some degree of leniency to allow for malfunctioning nodes.

A network parameter called "mining diversity" is used to calculate the number of blocks that a miner should wait for before attempting to mine again else the block will be rejected. A low value of the mining diversity parameter means that few miners need to collaborate in order to take over the network. If the number of collaborating miners is equal to or bigger than the number of blocks each miner should wait before attempting to mine again, then there is a probability that this will happen. Conversely, a higher value of the mining diversity parameter ensures that more permitted miners are included in the rotation, thus making the network take-over by a minority more difficult.

Mining Difficulty:
Difficulty is a measure of how difficult it is to find a hash below a given target. The Bitcoin network has a global block difficulty. Valid blocks must have a hash below this target. Mining pools also have a pool-specific share difficulty setting a lower limit for shares. In other words, Mining difficulty is How often the difficulty should be recomputed for the Blockchain (Mining Difficulty)

The Bitcoin network varies its difficulty levels after the discovery of every 2016 blocks to ensure a constant output. If the network hash rate is high and the time taken to discover a new block is less than 10 minutes, then the network will increase the difficulty level proportionately to increase the block discovery time. If the block discovery time is more than 10 minutes, then the same protocol will reduce the difficulty level.

In recent days, the Bitcoin mining difficulty levels have been constantly increasing, thanks to increasing network hash rate. Last week, the difficulty levels went up by over 7 percent. This is one of the highest hikes in the difficulty levels in the recent days, except for once in February when it increased by 20 percent. The network's hashing power currently stands at about 1279029147 GH/second (as on April 14, 2016). These changes are absolutely necessary to ensure the reliability and smooth functioning of the largest digital currency network.

Secure Backup of Keys:

Secure Backup of Each Nodes: Use RAID technology or auto-backup solutions such as cloud storage like Azure or Dropbox to resist disk failures.
Real time image of the disk as a backup is also suggested
Distributed Backups: In others words it's a peer-to-peer backup or is a step forward from common server-based network backup to a serverless distributed backup system that allocates backup data across the whole network

Logically, all computers within the network collaboratively create a virtual server that can be accessed by any client, but physically, there is no central data storage

Consensus Mechanism:

The consensus within a blockchain helps decide whether or not a transaction will be added to the blockchain. There are two ways that a blockchain can reach the necessary consensus required. Proof of work is the most strict approach and requires cryptographic computations. Another approach is more simple and vague, and uses proofs to achieve a higher number of transactions per second. These might rate from 6 to 100 000 depending on the consensus algorithm. This number is also determined by the size of the block. The bigger the block size, the higher the transaction output, however, this can lead to other issues such as network congestion, longer block propagations over the network, higher convergence-rate to a consensus or double spending.

Properties for Correct Consensus
• Termination: All correct processes eventually decide.
• Agreement: All correct processes select the same decision.
• Or…(stronger) all processes that do decide select the same decision, even if they later fail.
• Called uniform consensus: "Uniform consensus is harder then consensus."
• Integrity: All deciding processes select the "right" value.
– As specified for the variants of the consensus problem.

Properties of Distributed Algorithms
• Agreement is a safety property.
–Every possible state of the system has this property in all possible executions.
–I.e., either they have not agreed yet, or they all agreed on the same value.
• Termination is a liveness property.
–Some state of the system has this property in all possible executions.
–The property is stable: once some state of an execution has the property, all subsequent states also have it.

Fischer-Lynch-Patterson (1985)

•No consensus can be guaranteed in an asynchronous communication system in the presence of any failures.
•Intuition: a "failed" process may just be slow, and can rise from the dead at exactly the wrong time.
•Consensus may occur recognizably, rarely or often.
• e.g., if no inconveniently delayed messages
•FLP implies that no agreement can be guaranteed in an asynchronous system with Byzantine failures either.

Now, question is that How to detect that a member has failed? – It is through pings, timeouts, beacons, heartbeats, recovery notifications

Some of the important consensus Mechanisms is:
Byzantine Fault Tolerance – HyperLedger Fabric
Federated Byzantine Agreement – Ripple & Stellar
Proof of Work – Ethereum (Homestead)

Proof of Elapsed Time – Intel SawtoothLake
Proof-of-Stake Model (PoS) – Ethereum (Serenity)

Contingency policy, planning, testing, training & Backups:

Lets understand the meaning of Contingency:
a future event or circumstance which is possible but cannot be predicted with certainty.
a provision for a possible event or circumstance.

Contingency Plan:

The contingency plan protects resources, minimizes customer inconvenience and identifies key staff, assigning specific responsibilities in the context of the recovery.
Developing response in advance for various situation that might impact business

Contingency planning policy:

To develop, document, and disseminate-
-policy to relevant personnel,
-a contingency planning policy that addresses purpose, scope, roles, responsibilities, coordination and management commitment,
-compliance
-contingency plan & controls
-reviews and updates the current contingency plan policy preferably every 6 months.

Contingency plan Training, Testing and Backup:

The organization develops a contingency plan for the information system such that:
a. It Identifies essential missions and business functions and associated contingency requirements;
b. it Provides recovery objectives, restoration priorities, and metrics;
c. Addresses contingency roles, responsibilities, assigned individuals with their contact details;
d. Addresses maintaining essential missions and business functions despite an information system disruption, compromise, or failure;
e. Addresses eventual, full information system restoration without deterioration of the security safeguards originally planned and implemented; and
f. Is reviewed and approved as per organization policy

3. Top Down approach is adopted to reach the contingency plan and policies till the last person in the organization as per their role.

4. The organization coordinates contingency planning activities with incident handling activities.

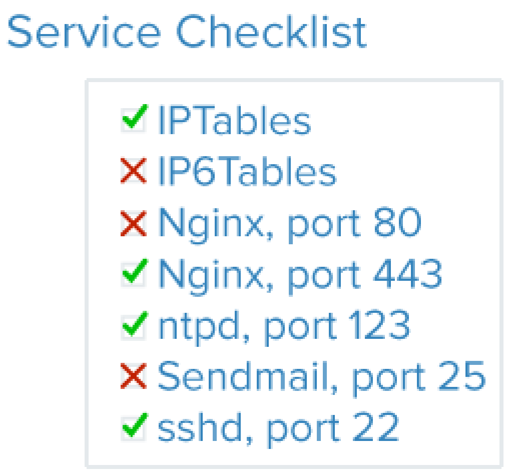
Note: the information system contingency plan must be reviewed every month.

5. The organization updates the contingency plan to address changes to the organization, blockchain, or environment of operation and problems encountered during contingency plan implementation, execution, or testing.

6. Any update or change in the contingency is conveyed to relevant personnel.

7 It should protect the contingency plan from unauthorized disclosure and modification.
8 There must be a proper backup in place to avoid business functions disruption in case of attack or contingency.
9 They must be trained properly to take apt care to protect the information system and crucial data breach.
10 There must be a monthly testing and mock drill must be conducted to check the preparedness of employees in a given organization.

Regular Service Auditing / Audits & Vulnerability scans:
Service auditing is a process of discovering what services are running on the servers in your infrastructure. Often, the default operating system is configured to run certain services at boot. Installing additional software can sometimes pull in dependencies that are also auto-started.


Service Checklist

✓ IPTables
✗ IP6Tables
✗ Nginx, port 80
✓ Nginx, port 443
✓ ntpd, port 123
✗ Sendmail, port 25
✓ sshd, port 22

Audit your servers and check the logs regularly: Auditing your server regularily is an important component of your IT infrastructure Management Lifecycle. This will help you to ensure that the minimum security requirements are always met and your users and administratora are compliant with your security policies. It will also enable you to identify any security issues that have to be fixed.

Scan your server for vulnerabilities: To identify vulnerabilities in your software and packages installed on your server(s), regular vulnerabiliy scans are important. Hackers are always scanning the internet to discover vulnerable servers and websites. Be proactive and fix any security issues before they are exploited by the bad guys.

Backup
Ensure your data is backed up regularly and securely: It is useful to keep regular backups in case your server has been compromised. Both WHM and Plesk have easy-to-use backup systems to create user data backups.

Secure code
Integrate the secure coding best practices to your development processes: The Open Web Application Security Project (OWASP) published a Quick Reference Guide which provides a comprehensive checklist that can be integrated into your development life cycle.

Incident Response Plan:

Before a breach happens, you should have identified a plan for
responding

Identify in writing an incident response team and set aside the resources necessary to appropriately deal with it, identifying the stakeholders and lines of communication

The law varies, so work closely with legal counsel to determine what your legal responsibilities are in terms of notifications and incident handling

Data Governance: Data governance is the capability that enables an organization to ensure that high data quality exists throughout the complete lifecycle of the data.

Data Security: Data security means protecting digital data, such as those in a database, from destructive forces and from the unwanted actions of unauthorized users, such as a cyberattack or a data breach

Data Governance & Data Security Checklist in the resources section

Blockchain Attacks:

Blockchain Attack Types:
The blockchain is designed to sustain against particular attacks. The most common types are -

Hard Forks attack: An attack when one party is able to hijack the main chain.
51% attack: An attack when a majority of votes is in the hand of one party only.
Double Spending attack: An attack when a transaction is duplicated and the token is spent twice.
Sybil attack: An attack when one node acquires several identities.

Denial of Service attack: An attack when nodes are unable to connect with each other.

Smart Contracts
"Smart contracts" installed on the blockchain are a computer protocol intended to digitally facilitate, verify, or enforce the negotiation or performance of an agreement. In addition, every contract is secured by cryptography to keep the parties intact. This technique permits the building of decentralised applications on top of the main blockchain and the use of their own tokens. To ensure the correct processing of smart contracts, the automated programs are deployed on top of the blockchain among participating nodes of the platform.

"A computerised transaction protocol that executes the terms of a contract based on pre-programmed logic or condition or rules."

Smart contracts are comprised of:

Subject of the contract: The automated program must have access to goods or services under the contract and be able to lock and unlock them automatically.
Digital signatures: All the participants initiate an agreement by signing the contract with their private keys.
Contract terms: Terms of a smart contract can take the form of an exact sequence of operations. All participants must sign these terms.
Decentralised platform: The smart contract is run on blockchain virtual machines and it can be programmed via multiple languages.

Smart Contact Security:

Smart Contract Vulnerabilities:

Blockchain
-Unpredictable state
-Generating Randomness
-Time Constraints

EVM bytecode - Ethereum

- Immutable bugs
- Ether lost in Transfer
- Stack size limit

Solidity

- Call to the unknown
- Exception disorders
- Gasless send
- Type casts
- Reentrancy

- Keeping Secrets

Unpredictable state: The state of a contract is determined by the value of its fields and balance. In general, when a user sends a transaction to the network in order to invoke some contract, he cannot be sure that the transaction will be run in the same state the contract was at the time of sending that transaction.
This may happen because, in the meanwhile, other transactions have changed the contract state. Even if the user was fast enough to be the first to send a transaction, it is not guaranteed that such transaction will be the first to be run. Indeed, when miners group transactions into blocks, they are not required to preserve any order; they could also choose not to include some transactions.

Generating randomness:
The execution of EVM bytecode is deterministic: in the absence of misbehaviour, all miners executing a transaction will have the
same results. Hence, to simulate non-deterministic choices, many contracts (e.g.lotteries, games, etc.) generate pseudo-random numbers, where the initialization seed is chosen uniquely for all miners.

A common choice is to take for this seed (or for the random number itself) the hash or the timestamp of some block that will appear in the blockchain at a given time in the future. Since all the miners have the same view of the blockchain, at run-time this value will be the same for everyone. Apparently, this is a secure way to generate random numbers, as the content of future blocks is unpredictable. However, since miners control which transactions are put in a block and in which order, a malicious miner could attempt to craft his block so to bias the outcome of the pseudo-random generator.

Time constraints: A wide range of applications use time constraints in order to determine which actions are permitted (or mandatory) in the current state. Typically, time constraints are implemented by using block timestamps, which are agreed upon by all the miners. Contracts can retrieve the timestamp in which the block was mined; all the transactions within a block share the same timestamp. This guarantees the coherence with the state of the contract after the execution, but it may also expose a contract to attacks, since the miner who creates the new block can choose the timestamp with a certain degree of arbitrariness.

Immutable bugs: Once a contract is published on the blockchain, it can no longer be altered. Hence, users can trust that if the contract implements their intended functionality, then its runtime behaviour will be the expected one as well, since this is ensured by the consensus protocol. The drawback is that if a contract contains a bug, there is no direct way to patch it. So, programmers have to anticipate ways to alter or terminate a contract in its implementation. The immutability of bugs has been exploited in various attacks, e.g. to steal ether, or to make it unredeemable by any user.

Ether lost in transfer: When sending ether, one has to specify the recipient address, which takes the form of a sequence of 160 bits. However, many of these addresses are orphan, i.e. they are not associated to any user or contract. If some ether is sent to an orphan address, it is lost forever (note that there is no way
to detect whether an address is orphan). Since lost ether cannot be recovered, programmers have to manually ensure the correctness of the recipient addresses.

Stack size limit: Each time a contract invokes another contract (or even itself via this.f()) the call stack associated with the transaction grows by one frame. The call stack is bounded to 1024 frames: when this limit is reached, a further invocation throws an exception. Until October 18th 2016, it was possible to exploit this fact to carry on an attack as follows. An adversary starts by generating an almost-full call stack (via a sequence of nested calls), and then he invokes the victim's function, which will fail upon a further invocation. If the exception is not properly handled by the victim's contract, the adversary could manage to succeed in his attack.

Call to the unknown: Few primitives used in Solidity to invoke functions and to transfer ether may have the side effect of invoking the fallback function of the recipient.

Exception disorder: In Solidity there are several situations where an exception may be raised and it isn't uniform in the way it handels exceptions, e.g. if
(i) the execution runs out of gas;
(ii) the call stack reaches its limit;
(iii) the command throw is executed.

Gasless send: When using the function send to transfer ether to a contract,it is possible to incur in an out-of-gas exception. This may be quite unexpected
by programmers, because transferring ether is not generally associated to executing code.

Type casts: The Solidity compiler can detects some type errors (e.g., assigning an integer value to a variable of type string). Types are also used in direct calls: the caller must declare the callee's / receipient interface, and cast to it the callee's / receipent address when performing the call.

Reentrancy: The atomicity and sequentiality of transactions may induce programmers to believe that, when a non-recursive function is invoked, it cannot be re-entered before its termination. However, this is not always the case, because
the fallback mechanism may allow an attacker to re-enter the caller function.

This may result in unexpected behaviours, and possibly also in loops of invocations

Keeping secrets: Fields in contracts can be public, i.e. directly readable by everyone, or private, i.e. not directly readable by other users/contracts. Still, declaring a field as private does not guarantee its secrecy. This is because, to set the value of a field, users must send a suitable transaction to miners, who will then publish it on the blockchain. Since the blockchain is public, everyone can inspect the contents of the transaction, and infer the new value of the field.

Many contracts, e.g. those implementing multi-player games, require that some fields are kept secret for a while: for instance, if a field stores the next move of a player, revealing it to the other players may advantage them in choosing their next move. In such cases, to ensure that a field remains secret until a certain event occurs, the contract has to exploit suitable cryptographic techniques, like e.g. timed commitments which eventually consume all the gas.

Brief Overview of ICO:

ICO is a way for start-ups to raise money by issuing a new cryptocoin, while users pay them bitcoin or ethereum. It's similar to crowdfunding but with digital money.

The first crypto-token sale (also known as an ICO) was held by Mastercoin in July 2013. Ethereum raised money with a token sale in 2014, raising 3,700 BTC in its first 12 hours, equal to approximately $2.3 million dollars. Another ICO was held by Karmacoin in April 2014 for its Karmashares project.

ICOs and token sales are now extremely popular. At the start of October 2017, ICO coin sales worth $2.3 billion had been conducted during the year, more than ten times as much as in all of 2016.

On September 4th 2017 China banned all ICOs and the Central Bank of China listed 60 exchanges involved in listing and trading ICO related tokens that would be inspected by the financial regulators. South Korea was the next country to severely restrict ICOs on September 29th 2017.

The first regulated market for ICOs was created in September 2017 by the Gibraltar Stock Exchange. This would be covered by Gibraltar DLT regulation. Utility tokens will be traded in the newly created Gibraltar Blockchain Exchange (GBX), while security or equity tokens will trade in the traditional stock market.

ICO Vulnerabilities, Risks and Security:
Security Issues Unmasked for ICO Projects

“A great deal of intelligence can be invested in ignorance when the need for illusion is deep.” S. Bellow

The new flavour of money is digital crypto-token currencies. Investors and crypto-currency enthusiasts place huge investments into ICOs. Some of these ICO projects are trustworthy while others are not. Why? Imagine someone cheating you with a faked token address. The chances of losing all investments are high.

Furthermore, some of the ICOs may even indicate in their offering document, that they have a team of developers spending a lot of hours working to build the project. But remember, the appearance may be deceptive. Mostly the ideas are good, but the implementation is a nightmare. Investors put huge faith in these projects with an anticipation of a stable, sustainable platform and crypto-token that can be safely transferred or delivered at the close of the ICO. The project must be built very precisely, and should be soundly tested by external security consultants or auditors.

Security is often a pain area for ICO investments as there is no blueprint available on testing the security of smart contracts. Accordingly, it´s all about expertise and expert views.

SECURITY RISKS:

Creation Time token adjustment and change (Modification): An attacker can possibly adjust the creation time before the token goes live.

Hardware Based Attacks:

PDoS (a Permanent denial of service attack) can cause a permanent hardware failure. The processing power of the powered node can be disrupted.

Replay Attacks:

Replay Attacks are possible due to APIs showing the number of requests remaining on the API account key, until it gets the desired random number key.

Key-based Attacks:

The risk of API keys being leaked out can lead to repeated requests. The Key must be hashed. The Response API Key Hash must be checked with the stamped response to eliminate key exhaustion.

Chain-based Attacks:

Any type of external query to the remaining API Key must be blocked to prevent cross-chain attacks or chain-based attacks.

Leaking Decryption Key:

An attacker can exploit ICO contracts (servers) and take over decryption keys. This has been proved to have a disastrous impact.

Social Engineering Attack:

Crypto users are not necessarily crypto-literate. A 30 second social engineering attack can trick them into giving up access to their crypto-token based smart contracts funded by the ICOs. Permissive Timestamp: Blockchain Blocks can take any timestamp; i.e.; after or before the block. This leaves room for exploitation.

Arbitrary Mining Attacks:

To prevent this attack, arbitrary mining must not be allowed once an ICO has started. Re-entrance attacks: In this attack, you do something, you are already in the process of getting it done, and you do it again. The risk of the user losing tokens in the contract is highly probable.

Based on these grounds, the Poorly Programmed smart contract is vulnerable to cyber-attacks. The security of crypto-tokens can be breached by malicious cyber criminals if the mentioned programming-bugs are still present. Therefore, the contract must be error-free and secure.

In addition, a proper due diligence is required to be conducted to ensure that every ICO is safe. Legal, compliance and security aspects must also be implemented to the account. One should have a basic knowledge of the ICO process and Crypto-currency functioning.

KIND ADVICE

Never send funds to an address posted on public networks or forums while participating in ICOs. A Bug Bounty program must be launched before the ICO is placed for subscription, to test the security aspects of the project.

A legit ICO is a treasure for the token holders as the potential upside can be very rewarding.