**UNIT-II**

**FUNDAMENTALS OF IoT:**

**What is IoT (Internet of Things)?**

* Internet of Things (IoT) is a network of physical objects or people called “things” that are embedded with software, electronics, network, and sensors that allows these objects to collect and exchange data.
* The goal of IoT is to extend to internet connectivity from standard devices like computer, mobile, tablet to relatively dumb devices like a toaster.
* IoT makes virtually everything “smart,” by improving aspects of our life with the power of data collection, AI algorithm, and networks.
* The thing in IoT can also be a person with a diabetes monitor implant, an animal with tracking devices, etc.

**History of IoT**

* 1970- The actual idea of connected devices was proposed
* 1990- John Romkey created a toaster which could be turned on/off over the Internet
* 1995- Siemens introduced the first cellular module built for M2M
* 1999- The term “Internet of Things” was used by Kevin Ashton during his work at P&G which became widely accepted
* 2004 – The term was mentioned in famous publications like the Guardian, Boston Globe, and Scientific American
* 2005-UN’s International Telecommunications Union (ITU) published its first report on this topic.
* 2008- The Internet of Things was born
* 2011- Gartner, the market research company, include “The Internet of Things” technology in their research

## How does Internet of Thing (IoT) Work?

The working of IoT is different for different IoT echo system (architecture). However, the key concept of there working are similar. The entire working process of IoT starts with the device themselves, such as smartphones, digital watches, electronic appliances, which securely communicate with the IoT platform. The platforms collect and analyze the data from all multiple devices and platforms and transfer the most valuable data with applications to devices.



**The four fundamental components of an IoT system:**

**1) Sensors/Devices:**Sensors or devices are a key component that helps you to collect live data from the surrounding environment. All this data may have various levels of complexities. It could be a simple temperature monitoring sensor, or it may be in the form of the video feed.

* A device may have various types of sensors which performs multiple tasks **apart** from sensing. Example, A mobile phone is a device which has multiple sensors like GPS, camera but your smartphone is not able to sense these things.

**2) Connectivity:** All the collected data is sent to a cloud infrastructure. The sensors should be connected to the cloud using various mediums of communications. These communication mediums include mobile or satellite networks, Bluetooth, WI-FI, WAN, etc.

**3) Data Processing:** Once that data is collected, and it gets to the cloud, the software performs processing on the gathered data. This process can be just checking the temperature, reading on devices like AC or heaters. However, it can sometimes also be very complex like identifying objects, using computer vision on video.

**4)User Interface:** The information needs to be available to the end-user in some way which can be achieved by triggering alarms on their phones or sending them notification through email or text message. The user sometimes might need an interface which actively checks their IoT system. For example, the user has a camera installed in his home. He wants to access video recording and all the feeds with the help of a web server.

**IoT Applications:**



IoT solutions are widely used in numerous companies across industries. Some most common IoT applications are given below:

|  |  |
| --- | --- |
| **Application type** | **Description** |
| Smart Thermostats | Helps you to save resource on heating bills by knowing your usage patterns. |
| Connected Cars | IoT helps automobile companies handle billing, parking, insurance, and other related stuff automatically |
| Smart home | Smart home encapsulates the connectivity inside your homes. It includes smoke detectors, home appliances, light bulbs, windows, door locks, etc. |
| Smart City | Smart city offers all types of use cases which include traffic management to water distribution, waste management, etc. |
| Parking Sensors | IoT technology helps users to identify the real-time availability of parking spaces on their phone. |
| Connect Health | The concept of a connected health care system facilitates real-time health monitoring and patient care. It helps in improved medical decision-making based on patient data. |

**Challenges of Internet of Things (IoT)**

**At present IoT is faced with many challenges, such as:**

* Insufficient testing and updating
* Concern regarding data security and privacy
* Software complexity
* Data volumes and interpretation
* Integration with AI and automation
* Devices require a constant power supply which is difficult
* Interaction and short-range communication

## Advantages of IoT

* **Efficient resource utilization:** If we know the functionality and the way that how each device work we definitely increase the efficient resource utilization as well as monitor natural resources.
* **Minimize human effort:** As the devices of IoT interact and communicate with each other and do lot of task for us, then they minimize the human effort.
* **Save time:** As it reduces the human effort then it definitely saves out time. Time is the primary factor which can save through IoT platform.
* **Enhance Data Collection:**
* **Improve security:** Now, if we have a system that all these things are interconnected then we can make the system more secure and efficient.

## Disadvantages IoT

* **Security**− The data is travelling all over the Internet. So maintaining its privacy is still a Big Challenge. End-to-end Encryption is a must in IoT.
* **Compatibility**− There is no International Standard for the monitoring of the equipment.
* **Complexity**− Most of the devices still contain some software bugs. Each device must be able to seamlessly interact with other devices in the network.
* **Safety**− Suppose a patient is left unattended by a doctor. And some notorious guy changes the prescription or Health monitoring devices malfunctioned. Then it can result in the death of the patient.
* **Policies**− Government authorities must take some steps to make policies and standards related to IoT to stop the Black marketing of IoT devices.

 **ARCHITECTURE OF IoTs:**

* Internet of Things (IoT) technology has a wide variety of applications and use of Internet of Things is growing so faster.
* Depending upon different application areas of Internet of Things, it works accordingly as per it has been designed/developed. But it has not a standard defined architecture of working which is strictly followed universally.
* The architecture of IoT depends upon its functionality and implementation in different sectors. Still, there is a basic process flow based on which IoT is built.



The above image it is clear that there is 4 layers are present that can be divided as follows:

* **Sensing Layer**
* **Network Layer**
* **Data processing Layer**
* **Application Layer.**

**1. Sensing Layer –**
Sensors, actuators, devices are present in this Sensing layer. These Sensors or Actuators accepts data(physical/environmental parameters), processes data and emits data over network.

**2. Network Layer –**
Internet/Network gateways, Data Acquisition System (DAS) are present in this layer. DAS performs data aggregation and conversion function (Collecting data and aggregating data then converting analog data of sensors to digital data etc). Advanced gateways which mainly opens up connection between Sensor networks and Internet also performs many basic gateway functionalities like malware protection, and filtering also some times decision making based on inputted data and data management services, etc.

**3. Data processing Layer –**
This is processing unit of IoT ecosystem. Here data is analyzed and pre-processed before sending it to data center from where data is accessed by software applications often termed as business applications where data is monitored and managed and further actions are also prepared. So here Edge IT or edge analytics comes into picture.

**4. Application Layer –**
This is last layer of 4 stages of IoT architecture. Data centers or cloud is management stage of data where data is managed and is used by end-user applications like agriculture, health care, aerospace, farming, defense, etc.

**IoT SECURITY REQUIREMENTS:**



The most frequently mentioned information security requirements are confidentiality, integrity, and availability – the CIA triad, as they are often called. In “traditional” IT systems, this order also represents the importance of these requirements.

**1. Confidentiality**

It means the protection of information from illegitimate read access. Not all sorts of information needs confidentiality, but there are sensitive data that must definitely be kept secret.
In IoT systems, data may not have very strict confidentiality requirements, although this can depend on the application domain as well. In general, the data generated by IoT systems are sensor readings, which are usually not secret; anyone could actually measure the same parameters and obtain the data himself. Yet, in the domains of security and surveillance, healthcare, retail, and even in home automation some data may need to be kept confidential. Images of surveillance cameras in security applications and inventory data in retail applications can easily be imagined to contain sensitive information, while data collected in homes by home automation applications and data collected from patients by healthcare applications may need confidentiality due to privacy reasons. When data requires confidentiality, it must be provided for both storage and transmission of that data. The latter is especially important in case of wireless communications, which is notoriously easy to eavesdrop. In addition, IoT systems also use access credentials, such as passwords, and cryptographic keys, which definitely need confidentiality. In security application, transport systems, and industrial environments, configuration data and control programs may also be kept secret, as they may contain intellectual property.

**2. Integrity** means protection against illegitimate modification of data, and it is one of the most important information security requirements in IoT systems. Sensor data generated by IoT systems are used to keep track of and control physical processes, so they need to be accurate. If sensor data can be changed by attackers, then tracking becomes inaccurate and control may receive wrong input. In addition, if control commands can be changed by an attacker, then control is definitely corrupted. The consequence of both can range from simple failures to fatal accidents. Similarly, parameters and software updates that are sent to IoT devices should not be modified by attackers, as such modifications can have similar effects as modifications of sensor data or control commands. Integrity is important in all IoT application domains, but perhaps transportation, industry automation and process control, and healthcare are outstanding here, as in these applications, violation of information integrity may really have fatal consequences.

**3. Availability** is the second most important information security requirement in IoT systems. It means that information is always available to entities who need it, and this is something we need to ensure in case of data used in control type of applications, such as transportation and industrial process control, and in certain healthcare applications as well. Availability of information is also very important in security and surveillance applications. In other applications, availability is desired, but it may not lead to serious negative outcomes if data is not always available in a timely manner; unavailability of information may lead to financial damage in retail applications and to inconveniences in home and office automation.

**4. Authenticity** means that the origin of the data can be verified by the intended receiver of the data. As the intended receiver typically acts upon the data, it is very important to make sure that the data originates from a trusted source. In fact, data origin authentication is as important as data integrity in IoT applications: if it was not provided, attackers could spoof fake data in the system making them appear to come from legitimate and trusted sources, and the consequences of that would be the same as being able to modify the data illegitimately. Authenticity is of paramount importance in case of control commands, configuration parameters, and software updates received by IoT devices.

**5. Non-repudiation** is similar to authenticity, but in this case, not only the intended receiver of the data can verify its source, but the origin of the data can be proven to any third parties. This means that the source of the data cannot deny or repudiate that the data originates from him or her, hence the name non-repudiation. Non-repudiation is not always required even in “traditional” IT systems, because in many cases messages that are exchanged are not kept long enough to make them available for verification by third parties. Regarding IoT applications, non-repudiation may be required in transport and healthcare applications, where there could be multiple entities involved in interactions and logs are kept for later audits in case of fatal accidents.

**6. Access control and authorization** can be important in all IoT application domains, in particular if the underlying IoT system and the services it provides are not meant to be publicly available or available for free. Clearly, one would not like to give free access to his home automation system to a stranger. Similarly, access to security and surveillance services should be restricted to the owner or the operator of the building or location being physically protected by those services. Industrial automation and process control systems are not supposed to be accessible for entities that are not part of the industrial facility or its contractors, and healthcare IoT systems must also be restricted to the patient and authorized medical staff. Transport and retail IoT systems are meant to be available to the public, but still certain parts of even those systems should be restricted to their operators. In the retail case, consumers should not have access to stock and order information of the retailer, whereas in the transport case, certain services should be authorized only to distinguished entities (e.g., to emergency vehicles).

**7. Trustworthy computing** is a general requirement and it means that one must be assured that the system and its services work as expected by the user at any time and in all conceivable situations. It is also a fundamental requirement, as if it cannot be satisfied, then other requirements, such as proper access control and authorization, could not be satisfied either, or at least, one would never be sufficiently assured that they are enforced properly. This is because system and service level security requirements are typically satisfied by implementing security mechanisms on the underlying computing platforms, and those implementations should be trustworthy. More specifically, the trustworthy computing requirement means that the computing environment on which the system is running and services are provided is difficult to corrupt by attackers. This covers the requirement to protect IoT devices from being hacked or infected by malware. In addition, trustworthiness requires assurance, which means that not only it is difficult to compromise system components, but it is possible to verify that the system is still in a healthy state, if it is, and if not, it can be brought back to a healthy state.

**8. Denial-of-Service** protection makes it difficult for attackers to render the services provided by the IoT system unavailable or substantially under-performing. This is a requirement similar to availability of information, but here we require availability of services rather than just information. Also, this requirement complements the trustworthy computing requirement in the sense that making it difficult for attackers to corrupt services (i.e., modify them) may not be sufficient to ensure that those services are available at all. In other words, system components may be intact, and hence, the service semantics may be unchanged, and yet the service may not be available. As an example, consider a typical DoS attack: the service is flooded with requests and overloaded, such that it cannot serve legitimate requests anymore. The service is still the same semantically, system components remained uncorrupted, yet the service became unavailable. Hence, the DoS protection requirement is needed besides the trustworthy computation requirement to fully express what we expect from a secure IoT system.

Protection against DoS attacks is important in all IoT application domains for two reasons. First, the very purpose of any IoT system is to provide some service; if this service can be rendered unavailable easily, then the entire IoT system becomes useless. Second, it is indeed rather easy to mount DoS type attacks against any kind of IT system, because such attacks do not need access to the system and corruption of the system components, as we explained above. As IoT systems consists of IoT devices that are often resource constraints, and they often operate in environments with special conditions, DoS attacks seems to be easier to carry out against IoT systems than against “traditional” IT systems.

**9. Privacy** means that human users can control how private information about them is stored, processed, and used in the IoT system and beyond. This requirement is relevant in application domains where the IoT system handles user related private information. This is not the case in industrial automation and process control applications, but privacy issues must be considered in all other IoT application domains. In home and office automation applications, generated data may reveal the users’ behavioral habits, in security and surveillance applications, as well as in transportation applications, data may reveal the users’ location, in retail applications, data may reveal user preferences towards certain types of products and also their shopping behavior, and finally, in healthcare applications, data may reveal the users’ illnesses or other types of medical problems. Essentially, as IoT is meant to convert our everyday life easier by using embedded intelligence in everyday objects, it seems to be unavoidable that IoT systems observe humans and collect data about them. Even if this data collection is not explicitly targeted to collect private information, whatever data such systems collect, it may contain a large amount of extractable private information. This is not a problem in itself, if privacy preserving mechanisms are also provided to the users that help them to keep the control on how their private information is used by other entities. Hence, the privacy requirement does not postulate the complete elimination of obtaining private information from data generated by IoT systems, but rather it postulates the need for appropriate privacy preserving mechanisms, and a privacy-by-design approach whenever possible (e.g., in case of new systems just about to be created).

**IoT Privacy Preservation issues:**

**IoT Preservation Issues:**

**1. Public Perception:** If the IoT is ever going to truly take off, this needs to be the first problem that manufacturers address. The 2015 Icontrol State of the Smart Home study found that 44% of all Americans were “very concerned” about the possibility of their information getting stolen from their smart home, and 27% were “somewhat concerned.” With that level of worry, consumers would hesitate to purchase connected devices.

**2. Vulnerability to Hacking:** Researchers have been able to hack into real, on-the-market devices with enough time and energy, which means hackers would likely be able to replicate their efforts. For example, a team of researchers at Microsoft and the University of Michigan found a plethora of holes in the security of Samsung’s SmartThings smart home platform, and the methods were far from complex.

**3. True Security:** Jason Porter, AT&T’s VP of security solutions, told Insider Intelligence that securing IoT devices means more than simply securing the actual devices themselves. Companies also need to build security into software applications and network connections that link to those devices.

**IoT Privacy Issues**

**1. Too Much Data:** The sheer amount of data that IoT devices can generate is staggering. A Federal Trade Commission report entitled “Internet of Things: Privacy & Security in a Connected World” found that fewer than 10,000 households can generate 150 million discrete data points every day. This creates more entry points for hackers and leaves sensitive information vulnerable.

**2. Unwanted Public Profile:** You’ve undoubtedly agreed to terms of service at some point, but have you ever actually read through an entire document? The aforementioned FTC report found that companies could use collected data that consumers willingly offer to make employment decisions. For example, an insurance company might gather information from you about your driving habits through a connected car when calculating your insurance rate. The same could occur for health or life insurance thanks to fitness trackers.

**3. Eavesdropping:** Manufacturers or hackers could actually use a connected device to virtually invade a person’s home. German researchers accomplished this by intercepting unencrypted data from a smart meter device to determine what television show someone was watching at that moment.

**4. Consumer Confidence:** Each of these problems could put a dent in consumers’ desire to purchase connected products, which would prevent the IoT from fulfilling its true potential.

**ATTACK MODELS:**

**Ciphertext-only attack (COA) -** in this type of attack it is assumed that the cryptanalyst has access only to the ciphertext, and has no access to the plaintext. This type of attack is the most likely case encountered in real life cryptanalysis, but is the weakest attack because of the cryptanalyst's lack of information. Modern ciphers are required to be very resistant to this type of attack. In fact, a successful cryptanalysis in the COA model usually requires that the cryptanalyst must have some information on the plaintext, such as its distribution, the language in which the plaintexts are written in, standard protocol data or framing which is part of the plaintext, etc

**Brute force attack or exhaustive key search -** in this attack every possible key is tried until the correct one is found. Every cipher except the unbreakable Information-theoretically secure methods like the one time pad is vulnerable to this method, and as its difficulty does not depend on the cipher but only on the key length - it's not considered a real cryptanalysis of the cipher. If the key has N bits, there are 2N possible keys to try, so a brute-force attack can recover the cipher in a worst-case time proportional to 2N and an average time of 2N-1. This is often used as a standard of comparison for other attacks. Brute-force can be applied in ciphertext-only settings, but the cryptanalyst must have enough information about the plaintext (at least N bits) to allow the identification of the correct key once it is tried.

**Known-plaintext attack (KPA) -** in this type of attack it is assumed that the cryptanalyst has access to at least a limited number of pairs of plaintext and the corresponding enciphered text. An interesting example dates back to World War II, during which the Allies used known-plaintexts in their successful cryptanalysis of the Enigma machine cipher. The plaintext samples are called "cribs"; the term originated at Bletchley Park, the British World War II decryption operation.[3][4] Very early on cribs were produced from stolen plaintext and intercepted ciphertext, and as such qualify for their classification as a known-plaintext attack. However, as knowledge and experience increased, the known-plaintexts were actually generated mostly through a series of intelligent guesses based on gained experience and logic, and not through a channel providing direct access to these plaintext. Technically the latter attacks are classified as the harder-to-execute ciphertext-only attacks.

**Chosen-plaintext attack (CPA) -** in this attack the cryptanalyst is able to choose a number of plaintexts to be enciphered and have access to the resulting ciphertext. This allows the analyst to explore whatever areas of the plaintext state space they wish and may allow them to exploit vulnerabilities and nonrandom behavior which appear only with certain plaintexts. In the widely used public-key cryptosystems, the key used to encrypt the plaintext is publicly distributed and anyone may use it, allowing the cryptanalyst to create ciphertext of any plaintext they want. So public-key algorithms must be resistant to all chosen-plaintext attacks.

**Adaptive chosen-plaintext attack (CPA2) -** in this attack the analyst can choose a sequence of plaintexts to be encrypted and have access to the ciphertexts. At each step they have the opportunity to analyze the previous results before choosing the next plaintext. This allows them to have more information when choosing plaintexts than if they were required to choose all the plaintexts beforehand as required in the chosen-plaintext attack.

**Chosen-ciphertext attack (CCA) -** in this attack the analyst can choose arbitrary ciphertext and have access to plaintext decrypted from it. In an actual real life case this would require the analyst to have access to the communication channel and the recipient end.

**Lunchtime attack or midnight attack -** In this variant it is assumed the cryptanalyst can only have access to the system for a limited time or a limited number of plaintext-ciphertext pairs, after which he must show progress. The name comes from the common security vulnerability in which an employee signs into their encrypted computer and then leaves it unattended while they go for lunch, allowing an attacker a limited-time access to the system.

**Adaptive chosen-ciphertext attack (CCA2) -** in this attack the analyst can choose a series of ciphertexts and see the resulting plaintexts, with the opportunity at each step to analyze the previous ciphertext-plaintext pairs before choosing the next ciphertext.

**Open key model attacks -** where the attacker has some knowledge about the key for the cipher being attacked.

**Related-key attack -** in this attack the cryptanalyst has access to ciphertext encrypted from the same plaintext using other (unknown) keys which are related to the target key in some mathematically defined way. For example, the analyst might know that the last N bits of the keys are identical. This is relevant because modern computer encryption protocols generate keys automatically, leading to the possibility of relations between them. The Wired Equivalent Privacy (WEP) privacy protocol which was used to protect WiFi internet devices was found to be vulnerable to a related-key attack due to a weakness in RC4.

**Side-channel attack -** This is not strictly speaking a cryptanalytic attack, and does not depend on the strength of the cipher. It refers to using other data about the encryption or decryption process to gain information about the message, such as electronic noise produced by encryption machines, sound produced by keystrokes as the plaintext is typed, or measuring how much time various computations take to perform.

**ATTACKS TO SENSOR IN IoTs:**

**1. Bad Mouthing Attack:** this attack occurs when an intruder tries to distort the innocent nodes’ reputation by sending negative reputation values about these nodes. For example, a malicious node (A) announces negative reputation about an innocent node (B). Such case will let other sensor nodes avoid sending any data to the node (B), while node (B) is not an attacker node. If the network has such an attack, after a while, the number of isolated nodes will increase because attackers will repeat such behavior with all its neighbors. The purpose of this attack is to isolate as much as possible network nodes.

**2. Good Mouthing attack:** in this attack, intruders try to deceive the base station or the cluster heads by sending positive reputation values about bad nodes. This attack is the contrary of bad mouthing attack. It has the following form, malicious node (A) announces positive reputation of another malicious node (B). The purpose of this attack is to dominate the network traffic, to break down the entire network.

**3. Whitewashing Attack:** this attack occurs when a malicious node tries to re-enter the network with a new identifier and a new reputation. This attack occurs when the system successfully detects a malicious node and isolates it from the network; then this malicious node tries to re-join the network with a new identifier to delude the system and have a new trust value.

**4. Energy Drain Attack:** in this attack, a malicious node asks the neighbor nodes to respond to useless traffic. Usually, the malicious node has unlimited power and high communication range to be able to send a lot of useless messages to its neighbors, such as control message or corrupted data. The purpose of this attack is to break down the entire network quickly.

**5. Exhaustion Attack:** this attack has a form where the attacker asks neighbor nodes to retransmit messages even there is no collision. The exhaustion attack is similar to the energy drain attack, where malicious nodes aim to destroy the network by discharging nodes’ batteries. However, in this attack, the malicious node pretends that data transmission is failed and asks for retransmission multiple times.

**6. Homing Attack:** in this attack, intruders investigate the network traffic to understand the geographical area of cluster heads or base station. When intruders know the network structure, they will be able to determine the most critical nodes and then attack these nodes to destroy the entire network quickly.

**7. Node Replication Attack:** this attack occurs when there exists a duplication of the node’s unique identifier. In this attack, the malicious node appears with an identifier that is assigned to another node; this case leads to inaccurate data aggregation. Also, the location estimation techniques that depend on the nodes’ identifiers will not be accurate.

**8. Sniffing Attack:** the sniffer node imposes into the network to capture the network’s valuable data. The malicious node, in this case, does not affect the network performance nor its lifetime instead, it eavesdrops the sent packet looking for any valuable information. The effect of this attack can be horrible in sensitive data applications such as military field services. Usually, the malicious sniffing node has unlimited power and high communication range, as illustrated in figure 4 below, where the intruder eavesdrops any communication channel between two nodes, or between a sensor node and the base station.

**ATTACKS TO RFIDs IN IoTs**

RFID systems, like most electronics and networks, are susceptible to both physical and electronic attacks. As the technology matures and becomes more widespread, so do hackers who aim to gain private information, entrance to secure areas, or take a system down for personal gain. Below are 7 known security attacks hackers can perform on an RFID system.

**1. Reverse Engineering**

Like most products, RFID tags and readers can be reverse engineered; however, it would take a lot of knowledge about the protocols and features to be successful. Hackers would take apart the chip in order to find out how it works in order to receive the data from the IC.

**Purpose: Steal Information and/or Gain Access**

**2. Power Analysis**

This attack requires nothing more than the brain of a hacker and a cell phone. According to leading experts1, power analysis attacks can be mounted on RFID systems by monitoring the power consumption levels of RFID tags. Researchers stumbled upon this hacking technique when studying the power emission levels in smart cards, especially in the difference in power levels between a correct passcode and an incorrect passcode

**Purpose: Steal Information and/or Gain Access**

**3. Eavesdropping & Replay**

Eavesdropping, like it sounds, occurs when an unauthorized RFID reader listens to conversations between a tag and reader then obtains important data. It is still necessary for the hacker to know the specific protocols and tag and reader information for this technique to work.

Replay attacks builds on eavesdropping and specifically occur when one part of communication in an RFID system is recorded and then ‘replayed’ at a later time to the receiving device in order to steal information or gain access.

**Purpose: Steal Information and/or Gain Access**

**4. Man-in-the-Middle Attack or Sniffing**

A man-in-the-middle attack happens during the transmission of a signal. Like eavesdropping, the hacker listens for communication between a tag and reader and then intercepts and manipulates the information. The hacker diverts the original signal and then sends false data while pretending to be a normal component in the RFID system.

**Purpose: Take Down System**

**5. Denial of Service**

A Denial of Service attack is the broad concept of an RFID system failure that is associated with an attack. These attacks are usually physical attacks like jamming the system with noise interference, blocking radio signals, or even removing or disabling RFID tags.

**Purpose: Take Down System**

**6. Cloning & Spoofing**

Technically two specific events, cloning and spoofing are usually done back to back. Cloning is duplicating data from a pre-existing tag, and spoofing is then using the cloned tag to gain access to a secured area or item. Because the hacker has to know the data on the tag to clone it, this type of attack is mainly seen in access or asset management operations.

**Purpose: Gain Access**

**7. Viruses**

According to some sources1, RFID tags currently do not have enough memory capacity to store a virus; but in the future, viruses could be a serious threat to an RFID system. A virus programmed on an RFID tag by an unknown source could cripple an RFID system when the tagged item is read at a facility. When read, the virus would transfer from tag to reader and then to a company’s network and software – bringing down connected computers, RFID components, and networks.

**Purpose: Take Down System**

**ATTACKS TO NETWORK FUNCTIONS IN IoTs:**

**What Is a Network Attack?**

A network attack is an attempt to gain unauthorized access to an organization’s network, with the objective of stealing data or perform other malicious activity. There are two main types of network attacks:

* **Passive:** Attackers gain access to a network and can monitor or steal sensitive information, but without making any change to the data, leaving it intact.
* **Active:** Attackers not only gain unauthorized access but also modify data, either deleting, encrypting or otherwise harming it.

**We distinguish network attacks from several other types of attacks:**

* **Endpoint attacks—**gaining unauthorized access to user devices, servers or other endpoints, typically compromising them by infecting them with malware.
* **Malware attacks—**infecting IT resources with malware, allowing attackers to compromise systems, steal data and do damage. These also include ransomware attacks.
* **Vulnerabilities, exploits and attacks—**exploiting vulnerabilities in software used in the organization, to gain unauthorized access, compromise or sabotage systems.
* **Advanced persistent threats—**these are complex multilayered threats, which include network attacks but also other attack types.

In a network attack, attackers are focused on penetrating the corporate network perimeter and gaining access to internal systems. Very often, once inside attackers will combine other types of attacks, for example compromising an endpoint, spreading malware or exploiting a vulnerability in a system within the network.

**What are the Common Types of Network Attacks?**

**Following are common threat vectors attackers can use to penetrate your network.**

**1. Unauthorized access**

Unauthorized access refers to attackers accessing a network without receiving permission. Among the causes of unauthorized access attacks are weak passwords, lacking protection against social engineering, previously compromised accounts, and insider threats.

**2. Distributed Denial of Service (DDoS) attacks**

Attackers build botnets, large fleets of compromised devices, and use them to direct false traffic at your network or servers. DDoS can occur at the network level, for example by sending huge volumes of SYN/ACC packets which can overwhelm a server, or at the application level, for example by performing complex SQL queries that bring a database to its knees.

**3. Man in the middle attacks**

A man in the middle attack involves attackers intercepting traffic, either between your network and external sites or within your network. If communication protocols are not secured or attackers find a way to circumvent that security, they can steal data that is being transmitted, obtain user credentials and hijack their sessions.

**4. Code and SQL injection attacks**

Many websites accept user inputs and fail to validate and sanitize those inputs. Attackers can then fill out a form or make an API call, passing malicious code instead of the expected data values. The code is executed on the server and allows attackers to compromise it.

**5. Privilege escalation**

Once attackers penetrate your network, they can use privilege escalation to expand their reach. Horizontal privilege escalation involves attackers gaining access to additional, adjacent systems, and vertical escalation means attackers gain a higher level of privileges for the same systems.

**6. Insider threats**

A network is especially vulnerable to malicious insiders, who already have privileged access to organizational systems. Insider threats can be difficult to detect and protect against, because insiders do not need to penetrate the network in order to do harm. New technologies like User and Even Behavioral Analytics (UEBA) can help identify suspicious or anomalous behavior by internal users, which can help identify insider attacks.

**ATTACKS TO BACK-END SYSTEMS:**

****

**1. Data injection risks**

Just as injection attacks can affect your web application's frontend, it's possible to perform injection attacks against your application's backend too. Attacks can craft queries to your web application's backend, and if there aren't any checks in place to verify the origin of the query, attackers can run commands directly on your backend—which in normal circumstances would have been filtered and blocked by the frontend. Securing your backend from accepting inputs from non-authorized sources is an effective way to prevent data injection attacks.

**2. Lack of authentication security**

Web application backends consist of multiple services with authentication requirements, databases as well as console/OS level access have logins, and all of these services run directly on the operating system layer. Therefore, maintaining authentication security is crucial—otherwise, any vulnerability entering the system can lead to the whole operating system being compromised. For example, when it comes to the web server, restricting logins to certain users or IP addresses, using HTTP authentication on development areas, or using automated brute force detection systems (that automatically ban offending IP addresses) helps a lot.

**3. Access control-related misconfigurations**

A frequently overlooked aspect of web applications is access control levels, commonly known as ACLs. ACLs define what parts of a backend a team member or customer can access. Misconfigurations in this area can lead to team members or customers gaining access to sensitive parts of your web application. We've seen some big data exposure issues with the cloud that began with misconfigured bucket ACLs, as seen in our article "5 AWS Misconfigurations That May Be Increasing Your Attack Surface". In many other cases, ACL misconfigurations could also lead to targeted attacks with team members' systems getting compromised, and their systems in turn being used to access sensitive areas of your web application. Ensuring that your team members and customers have just the right amount of access is important when managing your web application's security.

**4. Software misconfigurations**

Your website's frontend relies on its backend to do the heavy lifting and processing of user inputs. Consequently, any misconfigurations in the backend can result in sensitive information being presented to the user. For example, error messages, and error handling in general, are frequently overlooked when configuring software. At times, these error messages contain sensitive information, including data paths and variable names, which can lead attackers to these files and variables directly. Configuring the software that runs and handles your backend—including your webserver, coding language or run-time handler such as PHP—to not output sensitive error messages to the end user is highly recommended. Another misconfiguration danger is the threat of DoS attacks. These attacks take advantage of software misconfigured with a lack of resource limitations, and attempt to exhaust system resources. Once the system resources are exhausted, the web application's behaviour can be unpredictable, with the base operating system killing any process it sees fit. This could cause your database to crash while keeping the web server active, or crash the web server while keeping the database active. Either scenario would result in a poor experience for your end users or customers alike.

**5. Outdated/end-of-life software components**

Multiple software components are used to make any web application work, with web servers, databases and other software helping to improve performance. With all these bits of software in use, the security of each individual application has to be considered. For example, if your web server is vulnerable, it can cause your entire web application to be vulnerable—by accepting inputs from users which can expose sensitive areas of your web application. As we shared in our 'Why software gets hacked' blog post, having any outdated or end-of-life software component on the public internet is a huge risk, especially with tools like Shodan available, allowing users to search for servers running specific versions of software. This can make you and your web application an easy target when running outdated and end-of-life software.

**6. Lack of vulnerability scanning**

Scanning for vulnerabilities is another neglected safeguard regarding web applications, from frontend to backend. Only when you scan will you know what is and isn't vulnerable. Scanning is often thought of as a difficult and time consuming task, but modern tools have made it possible to scan automatically and with a low amount of effort. Using online vulnerability scanners and other tools like Nikto or OpenVAS helps you stay on top of your web application's safety by automatically scanning and generating reports for you to review.

**7. Sensitive data exposure**

Applications often cache or hold data in temporary locations for customers to access. This data can be used to improve performance or simply allow users to download their files, but if data isn't removed in time—or isn't restricted to the specific customer—it can allow attackers to find and download this sensitive information. For example, if a user uploads his avatar onto a web application, the web application stores it in a publicly accessible folder for the image to render. However, if the folder isn't well secured, an attacker can locate the folder and download all the user's avatars. Securing folders and other publicly accessible information is a must. Also, performing self-scans by using Google Dorks enables you to quickly find public information crawled by search engines.

**8. Lack of encryption between frontend and backend**

Communication between your web application's frontend and backend is what drives your web application. And this communication often goes over the internet unencrypted, as the software in use is often built without encryption in mind. Man-in-the-middle-attacks apply not only between clients and web applications, but can also poison or steal requests from your frontend or backend, and transmit them to the attacker. While this is a more sophisticated type of attack, it's still quite possible for someone who wishes to target your web application. Encrypting requests between the frontend and backend is a simple yet critical solution for preventing these attacks.

**9. SSL misconfigurations and lack of monitoring**

Using SSL certificates has become a norm for the modern internet. Most browsers and search engines give priority to websites with SSL certificates, and display various forms of alerts and errors when browsing websites with insecure, misconfigured or expired SSL certificates. Be sure to monitor your SSL certificate and configure it correctly. Ensuring your certificates are valid is the first step, noting that they're set up with strong ciphers and that secure and modern TLS versions are in use on the server-side. This will go far in maintaining that your web application's backend is communicating securely with its frontend.

**10. Lack of centralized log management**

With multiple backends, services and distributed servers (sometimes even across different public/private clouds and access levels), modern web applications call for a centralized logging system, to which all services can write their log files. This allows for further monitoring and logging of errors, unauthorized entry attempts (such as failed login alerts), and other information which can be used to improve performance as well. A lack of centralized logging often leads to unnecessary privileges being made available to team members, or simply due to the scale of things, log files getting overlooked. This can be particularly dangerous if the log files indicate attacks against your web application's backend, and their neglect means an attack isn't effectively blocked in time.

**SECURITY IN FRONT-END SENSOR AND EQUIPMENT:**

Front-end security, also known as client-side security, refers to securing websites and web applications on the customers’ side of a digital transaction. Consumers today rely on digital devices to manage banking, healthcare, shopping, and communications. Customers expect to be able to do business safely and securely online, without their PII or payment information being stolen.

**Why is front-end security important?**

Front-end vulnerabilities are increasingly common, primarily due to the rise in the number of businesses and end users that use web applications to share sensitive and personally identifiable information (PII). Because so many websites and web applications are written in JavaScript (~98% of all global websites), which also happens to be an inherently insecure programming language, threat actors are simply exploiting JavaScript, embedding malicious scripts into existing code. Front-end security issues also arise due to the abundance of third- and fourth-party scripts used to compile websites.

One of the most important actions ​​any business can take is protecting their customers from front-end security threats. Unfortunately, because of the sophisticated and subtle nature of these attacks, they can be hard to detect until it’s too late.​​

To ensure that businesses are offering a safe and secure digital experience, they must be diligent about securing their website or web application from dangerous front-end attacks.



**FRONT-END SECURITY**

**FRONT-END SECURITY SYSTEM**



**What is Security Equipment?**

Security System is considered most essential these days considering the increase in rate of crimes. Be it your home or your business premises you need security equipment to protect yourself and your family or your business assets and employees. It is, therefore, advisable to make provision for latest security equipment as cost of action to buy such equipment is nominal but result of in-action to go in for these is some time most disastrous.

**Security Equipment List**

* X-Ray Baggage scanners,
* Door Frame Metal Detectors,
* Hand Held Metal Detectors,
* Letter Bomb Detectors,Explosives Detector,
* Vehicle inspection detector,
* CCTV cameras,
* Electronic Article Surveillance System tags,barriers.

**1. X-RAY BAGGAGE SCANNERS**

To ensure high level of security at public places where football is quite heavy, one of the important security gadget that is selected is X-ray Baggage Scanner which is used for detection of in-wanted objects or substances hidden inside carry Bags etc These machines are manufactured in different tunnel sizes to suit the requirement.

For example if the machine is to be used at the entrance of a shopping mall the size of carry bags that need scanning would include ladies hand bags or laptop bags of gents.For this purpose generally tunnel size of 50x30cm is preferred. For larger sizes of carry bags machines having tunnel size of 65x50 cm is preferred or for air ports where large size of suitcases are to be scanned tunnel size of 100x100cm is preferred The process of scanning is simple and quick.You can immediately view on the monitor Non-flickering digital image for which there is provision for archiving. There are many other features and software that can be provided on these machines To look after radiation hazard attributed to these machines Govt of India has entrusted the job to Atomic Energy Regulatory Board (AERB )who not only examine and approve the manufacturing facilities but also grant approval for different models.

**2. DOOR FRAME METAL DETECTORS**

A Door Frame Metal Detector(DFMD) which is also called Walk Through Metal Detector Is a door which is designed and fabricated to detect any metal hidden on the body of a person passing through it. DFMDs are manufactured in single zones or in multi zones. Difference between the two is that in single zone display is provided at the top at one place where LED according to sensitivity levels are fitted. These LEDs glow according to the size of metal being carried through it.

For example if a person is hiding a pistol in his pocket all the LEDs would glow. However in a Multi Zone DFMD LEDs are provided in the side panels and if a person carrying a pistol passes through this DFMD and he has hidden the pistol in left pocket of his pant LEDs closer to that portion would glow which would immediately help to search the exact body part a part from multi-zone DFMDs are also manufactured with IP address for monitoring the working from a remote place Recently,DFMDs with thermal imaging features have been also added to the different types of DFMDs being manufactured these days.

**3. HAND HELD METAL DETECTOR (HHMD)**

Hand Held Metal Detector (HHMD) is most common security gadget used for General Frisking or Pilferage Check. For general frisking models with rechargeable batteries and charger are preferred For pilferage check HHMDs are manufactured to detect pilferage of small metals like nuts or bolts The most sensitive model manufactured have sensitivity of staple pin that is metal as small as staple pin can be detected so far as general frisking is concerned it is done by almost touching the body of a person to find out whether any metal has been hidden.

**4. VEHICLE INSPECTION DETECTOR**

This security gadget is manufactured to inspect undercarriage of any vehicle it is mounted on a trolley and a light is provided on the extendable handle to facilitate inspection. As the vehicle shows up at the designated spot and rolls over the imaging unit, the cameras catch pictures of the underside and send them to the control unit which shows them on a screen. The control unit and screen can be situated outside in closeness to the designated spot or inside the guardhouse.

**5. CCTV CAMERAS**

Closed Circuit Television (CCTV) is a system in which all elements from the cameras to the recording apparatuses are directly connected in order to keep the motion viewing record from being sent far and wide over public air-waves and on a shut way taken by electric current ( for this reason the name ). CCTV was first used by the germans to make observations of the get started of the goes over-quick . Since then, it has become into the in great detail, wide range safety camera technology we have knowledge of and use it as today. From public buildings to private offices, and from private houses to country clubs, with an overall view placed, safety cameras are used to view events as they take place, as well in connection with take part for looking at a later time. Some common uses for CCTV technology cover:

**6. ELECTRONIC ARTICLE SURVEILLANCE SYSTEM**

Electronic Article Surveillance) A security system for putting a stop to taking of property without right to do trade with general public stores that uses and put to side names on computer records and so on loose ends or reusable hard loose ends made joined to the goods to be traded . A danger sign is put into motion when walking through a small stage at the store way out if a use and put to side tag was not deactivated or a reusable tag was not taken away at the check-out bit for recording points. Also named "single bit RFID tags" because the tag at is either on or off, The first EAS technologies are radio number of times every unit time (RF), acousto-magnetic (AM) and electromagnetic (EM).

**PREVENT UNAUTHORIZED ACCESS TO SENSOR DATA:**

**1. Strong Password Policy**

Enforce best practices for user passwords—force users to select long passwords including letters, numbers and special characters, and change passwords frequently. Educate users to avoid using terms that can be guessed in a brute force attack, inform them about routine password updating, and to tell them to avoid sharing passwords across systems.

Just setting a password policy may not be enough. Consider using tools—such as enterprise password management or Identity and Access Management (IAM)—to centrally manage user credentials and ensure they conform to security best practices.

**2. Two Factor Authentication (2FA) and Multifactor Authentication**

Credentials based on user names, passwords, answers to security questions, etc. are known more generally as knowledge-based security factors. Knowledge-based factors are an important authentication method, but they are inherently weak and easy to compromise.

One of the best ways to prevent unauthorized access in your organization is to supplement knowledge-based factors with additional authentication methods:

**Possession factors —** authentication via objects possessed by the user. For example, a mobile phone, a security token or a physical card.

**Inherence factors —** authentication via something the user is or has. This includes biometric authentication using fingerprints, iris scans or voice recognition.

**3. Physical Security Practices**

As important as cybersecurity is, don’t neglect physical security. Train users to always lock devices when walking away from their desks, and to avoid writing down passwords or leaving sensitive documents in the open. Have a clear policy about locking office doors and ensure only authorized parties can enter sensitive areas of your physical facility.

**4. Monitoring User Activity**

It is crucial to monitor what is happening with user accounts, to detect anomalous activity such as multiple login attempts, login at unusual hours, or login by users to systems or data they don’t usually access. There are several strategies for monitoring users and accounts:

**Log analysis —** security analysts can gain visibility into logs of sensitive enterprise systems and uncover suspicious activity

**Rule-based alerts** — security tools can alert security staff to suspicious activity patterns, such as multiple login attempts or incorrect login to sensitive systems

**Behavioral analytics** — User and Event Behavioral Analytics (UEBA) monitors users and systems, establishes a baseline of normal activity, and detects any behavior that represents an anomaly and may be malicious.

**5. Endpoint Security**

Historically, most security breaches were a result of penetrating the network perimeter. Today, many attacks circumvent network defences by directly targeting endpoints, such as employee workstations, servers, cloud instances. Installing antivirus on every endpoint is the most basic security measure.