The Technical Gang

INTERNET OF THINGS

Subject Code: 18CS81

Assignment-3

Module-4 & 5

By : Rakshit Rangarajan

technicalgang.in

Scan to view document

Answer the Following:

1. Explain the challenges in IOT Security.

Ans:

Common Challenges in OT Security: The security challenges faced in IoT are by no means new and are not limited to specific industrial environments. The following sections discuss some of the common challenges faced in IoT.

- **Erosion of Network Architecture:** There is a wide variety in secured network designs within and across different industries. For example, power utilities have a strong history of leveraging modern technologies for operational activities, and in North America there are regulatory requirements in place from regulatory authorities, such as North American Electric Reliability Corporation's (NERC's) Critical Infrastructure Protection (CIP)
- **Pervasive Legacy Systems**: Due to the static nature and long lifecycles of equipment in industrial environments, many operational systems may be deemed legacy systems. For example, in a power utility environment, it is not uncommon to have racks of old mechanical equipment still operating alongside modern intelligent electronic devices (IEDs). In many cases, legacy components are not restricted to isolated network segments but have now been consolidated into the IT operational environment. From a security perspective, this is potentially dangerous as many devices may have historical vulnerabilities or weaknesses that have not been patched and updated, or it may be that patches are not even available due to the age of the equipment.
- **Insecure Operational Protocols:** The structure and operation of most of these protocols is often publicly available. While they may have been originated by a private firm, for the sake of interoperability, they are typically published for others to implement. Thus, it becomes a relatively simple matter to compromise the protocols themselves and introduce malicious actors that may use them to compromise control systems for either reconnaissance or attack purposes that could lead to undesirable impacts in normal system operation.
- **Device Insecurity:** Beyond the communications protocols that are used and the installation base of legacy systems, control and communication elements themselves have a history of vulnerabilities.

To understand the nature of the device insecurity, it is important to review the history of what vulnerabilities were discovered and what types of devices were affected. A review of the time period 2000 to 2010 reveals that the bulk of discoveries were at the higher levels of the operational network, including control systems trusted to operate plants, transmission systems, oil pipelines, or whatever critical function is in use.

---

2. Discuss OCTAVE and FAIR formal risk analysis.

Ans:

Formal Risk Analysis Structures: OCTAVE and FAIR

The key for any industrial environment is that it needs to address security holistically and not just focus on technology. It must include people and processes, and it should include all the vendor ecosystem components that make up a control system.

OCTAVE

OCTAVE (Operationally Critical Threat, Asset and Vulnerability Evaluation) has undergone multiple iterations. The version this section focuses on is OCTAVE Allegro, which is intended to be a lightweight and less burdensome process to implement. Allegro assumes that a robust security team is not on standby or immediately at the ready to initiate a comprehensive security review. This approach and the assumptions it makes, are quite appropriate, given that many operational technology areas are similarly lacking in security-focused human assets. Figure 8-5 illustrates the OCTAVE Allegro steps and phases.

OCTAVE is a balanced information-focused process. What it offers in terms of discipline and largely unconstrained breadth, however, is offset by its lack of security specificity. There is an assumption that beyond these steps are seemingly means of identifying specific mitigations that can be mapped to the threats and risks exposed during the analysis process.

FAIR

FAIR (Factor Analysis of Information Risk) is a technical standard for risk definition from The Open Group. While information security is the focus, much as it is for OCTAVE, FAIR has clear applications within operational technology. Like OCTAVE, it also allows for non-malicious actors as a potential cause for harm, but it goes to greater lengths to emphasize the point. For many operational groups, it is a welcome acknowledgement of existing contingency planning. Unlike with OCTAVE, there is a significant emphasis on naming, with risk taxonomy definition as a very specific target.

FAIR places emphasis on both unambiguous definitions and the idea that risk and associated attributes are measurable. Measurable, quantifiable metrics are a key area of emphasis, which should lend itself well to an operational world with a richness of operational data. At its base, FAIR has a definition of risk as the probable frequency and probable magnitude of loss. With this definition, a clear hierarchy of sub-elements emerges, with one side of the taxonomy focused on frequency and the other on magnitude.

Loss even frequency is the result of a threat agent acting on an asset with a resulting loss to the organization. This happens with a given frequency called the threat event frequency (TEF), in which a specified time window becomes a probability. There are multiple sub-attributes that define frequency of events, all of which can be understood with some form of measurable metric. Threat event frequencies are applied to a vulnerability. *Vulnerability* here is not necessarily some compute asset weakness, but is more broadly defined as the probability that the targeted asset will fail as a result of the actions applied. There are further sub-attributes here as well.

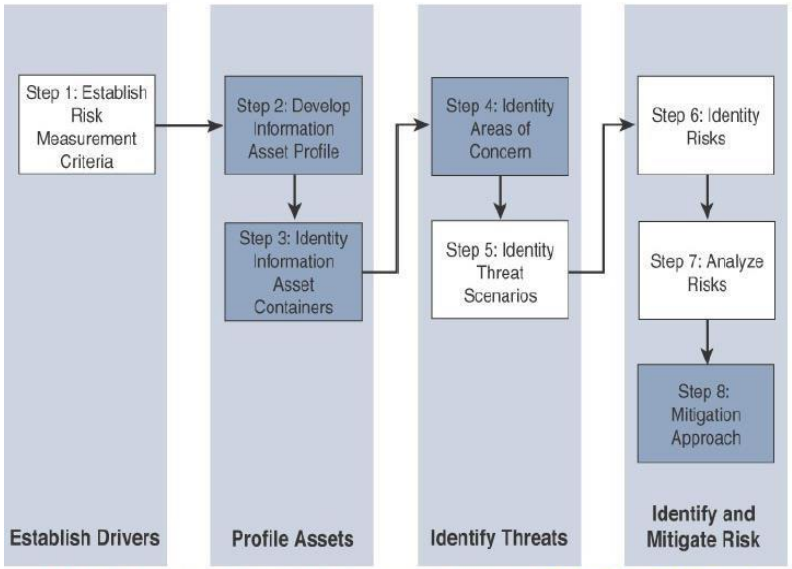

Figure 8-5 *OCTAVE Allegro Steps and Phases (see https://blog.compass-*

3. Illustrate Lambda architecture.

Ans:

Lambda Architecture

Ultimately the key elements of a data infrastructure to support many IoT use cases involves the collection, processing, and storage of data using multiple technologies. Querying both data in motion (streaming) and data at rest (batch processing) requires a combination of the Hadoop ecosystem projects discussed.

One architecture that is currently being leveraged for this functionality is the Lambda Architecture. Lambda is a data management system that consists of two layers for ingesting data (Batch and Stream) and one layer for providing the combined data (Serving). These layers allow for the packages discussed previously, like Spark and MapReduce, to operate on the data independently, focusing on the key attributes for which they are designed and optimized. Data is taken from a message broker, commonly Kafka, and processed by each layer in parallel, and the resulting data is delivered to a data store where additional processing or queries can be run. Figure 7-11 shows this parallel data flow through the Lambda Architecture.

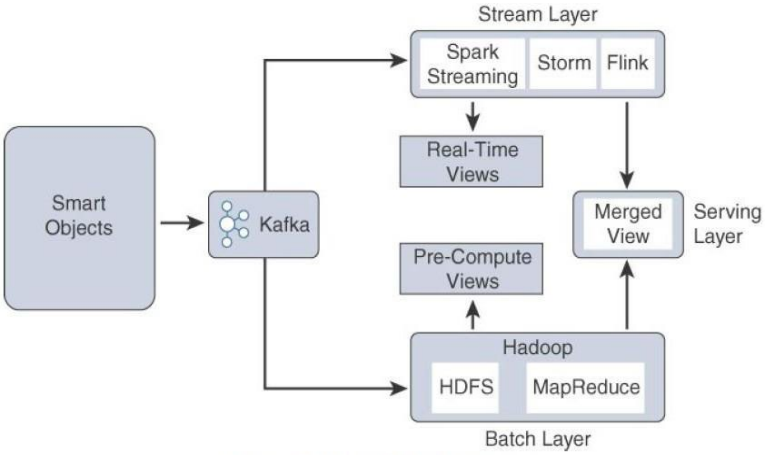

Figure 7-11 *Lambda Architecture*

The Lambda Architecture is not limited to the packages in the Hadoop ecosystem, but due to its breadth and flexibility, many of the packages in the ecosystem fill the requirements of each layer nicely:

- **Stream layer:** This layer is responsible for near-real-time processing of events. Technologies such as Spark Streaming, Storm, or Flink are used to quickly ingest, process, and analyse data on this layer. Alerting and automated actions can be triggered on events that require rapid response or could result in catastrophic outcomes if not handled immediately.
- **Batch layer:** The Batch layer consists of a batch-processing engine and data store. If an organization is using other parts of the Hadoop ecosystem for the other layers, MapReduce and HDFS can easily fit the bill. Other database technologies, such as MPPs, NoSQL, or data warehouses, can also provide what is needed by this layer.
- **Serving layer:** The Serving layer is a data store and mediator that decides which of the ingest layers to query based on the expected result or view into the data. If an aggregate or historical view is requested, it may invoke the Batch layer. If real-time analytics is needed, it may invoke the Stream layer. The Serving layer is often used by the data consumers to access both layers simultaneously.

---

4. Explain in detail how IT and OT security practices and systems vary in real time.

Ans:

How IT and OT Security Practices and Systems Vary

The differences between an enterprise IT environment and an industrial-focused OT deployment are important to understand because they have a direct impact on the security practice applied to them.

The Purdue Model for Control Hierarchy

Regardless of where a security threat arises, it must be consistently and unequivocally treated. IT information is typically used to make business decisions, such as those in process optimization, whereas OT information is instead characteristically leveraged to make physical decisions, such as closing a value, increasing pressure, and so on. Thus, the operational domain must also address physical safety and environmental factors as part of its security strategy —and this is not normally associated with the IT domain. Organizationally, IT and OT teams and tools have been historically separate, but this has begun to change, and they have started to converge, leading to more traditionally IT centric solutions being introduced to support operational activities. For example, systems such as firewalls and intrusion prevention systems (IPS) are being used in IoT networks.

Figure 8-3 *The Logical Framework Based on the Purdue Model for Control Hierarchy*

This model identifies levels of operations and defines each level. The enterprise and operational domains are separated into different zones and kept in strict isolation via an industrial demilitarized zone (DMZ):

- **Enterprise zone**
  - **Level 5: Enterprise network:** Corporate-level applications such as Enterprise Resource Planning (ERP), Customer Relationship Management (CRM), document management, and services such as Internet access and VPN entry from the outside world exist at this level.
  - **Level 4: Business planning and logistics network:** The IT services exist at this level and may include scheduling systems, material flow applications, optimization and planning systems, and local IT services such as phone, email, printing, and security monitoring.
- **Industrial demilitarized zone**
  - **DMZ:** The DMZ provides a buffer zone where services and data can be shared between the operational and enterprise zones. It also allows for easy segmentation of organizational control. By default, no traffic should traverse the DMZ; everything should originate from or terminate on this area.
- **Operational zone**
  - **Level 3: Operations and control:** This level includes the functions involved in managing the workflows to produce the desired end products and for monitoring and controlling the entire operational system. This could include production scheduling, reliability assurance, systemwide control optimization, security management, network management, and potentially other required IT services, such as DHCP, DNS, and timing.
  - **Level 2: Supervisory control:** This level includes zone control rooms, controller status, control system network/application administration, and other control-related applications, such as human-machine interface (HMI) and historian.
  - **Level 1: Basic control:** At this level, controllers and IEDs, dedicated HMIs, and other applications may talk to each other to run part or all of the control function.
  - **Level 0: Process:** This is where devices such as sensors and actuators and machines such as drives, motors, and robots communicate with controllers or IEDs.
- Safety zone
  - **Safety-critical:** This level includes devices, sensors, and other equipment used to manage the safety functions of the control system.

---

5. Explain different components of flexible Net Flow Architecture (FNF).

Ans:

Netflow Components

The basic concept with Flexible Netflow (and the Original Netflow) is to categorize and track different traffic flows.

Records

These flows are defined by a number of different pieces of traffic information; the information used when using Flexible Netflow can be defined by user records or within standard records. With the original Netflow, a flow was defined by seven different pieces of information that is used to categorize traffic; this information includes the following fields:

- Source IP address
- Destination IP address
- Source port number
- Destination port number
- Layer 3 protocol type
- Type of service (ToS)
- Input logical interface

Traffic with the same values for these seven fields was defined as a flow and individually tracked. Flexible Netflow provides the ability to either use this original flow definition ("Record") or to create a new, more specific flow definition. When creating a user-defined flow definition, the fields that are going to be tracked are selected and then defined as either a key field or as a nonkey field; these key fields are then used by Flexible Netflow to define traffic flows; the fields that are defined as nonkey are captured with the flow but are not used to define specific flows.

Flow Monitor

The Netflow flow monitor component is used to provide the actual traffic monitoring on a configured interface. When a flow monitor is applied to an interface, a flow monitor cache is created that is used to collect the traffic based on the key and nonkey fields in the configured record. There are three different modes of flow monitor cache that can be used with each flow monitor:

- **Layer 3**—When in the normal mode, cache entries are aged out according to timeout parameters, based on the activity of a flow. This is the default mode.
- **Immediate**—When in the immediate mode, cache entries are aged out as soon as created. When in this mode, each flow contains only one packet; this is used when traffic information is required immediately at the flow export destination (see next section).
- **Permanent**—When in the permanent mode, cache entries that are newer are aged out. This is useful when long term statistics on a device are required and the number of flows is expected to be low.

Flow Exporter

A flow exporter is used to transfer the contents of the Netflow cache from the device to a remote system. The Netflow Data Export Format Version 9 is used with Flexible Netflow (as opposed to Version 5) in order to provide additional flexibility. Multiple flow exporters can be

configured and assigned to a variety of different flow monitors if there is a need to export to multiple locations.

Flow Sampler

A flow sampler is used when there is a high volume of traffic to analyze that could potentially affect the performance of the monitored device. In this situation, a flow sampler can be used to limit the number of packets that will be analyzed by the flow monitor. For example, 1 out of every 2 packets could be captured and analyzed.

---

6. Briefly Explain Edge streaming analytics in IoT.

Ans:

Edge Streaming Analytics

One industry where data analytics is used extensively is the world of automobile racing. For example, in Formula One racing, each car has between 150 to 200 sensors that, combined, generate more than 1000 data points per second, resulting in hundreds of gigabytes of raw data per race. The sensor data is transmitted from the car and picked up by track-side wireless sensors. During a race, weather conditions may vary, tire conditions change, and accidents or other racing incidents almost always require an adaptable and flexible racing strategy. As the race develops, decisions such as when to pit, what tires to use, when to pass, and when to slow down all need to be made in seconds. Teams have found that enormous insights leading to better race results can be gained by analysing data on the fly—and the data may come from many different sources, including trackside sensors, car telemetry, and weather reports.

To summarize, the key values of edge streaming analytics include the following:

- **Reducing data at the edge:** The aggregate data generated by IoT devices is generally in proportion to the number of devices. The scale of these devices is likely to be huge, and so is the quantity of data they generate. Passing all this data to the cloud is inefficient and is unnecessarily expensive in terms of bandwidth and network infrastructure.
- **Analysis and response at the edge:** Some data is useful only at the edge (such as a factory control feedback system). In cases such as this, the data is best analysed and acted upon where it is generated.
- **Time sensitivity:** When timely response to data is required, passing data to the cloud for future processing results in unacceptable latency. Edge analytics allows immediate responses to changing conditions.

---

7. Explain Raspberry Pi learning board.

Ans:

- **Processor** – The Broadcom BCM2835 SoC (System on Chip) used in the first-generation Raspberry pi is somewhat equivalent to the chip used in first generation smart phones,

which includes a 700MHz ARM1176JZF-S Processor, Video Core IV graphics processing unit (GPU) and RAM.

• This has a level 1 (L1) cache of 16 KB and a level 2 (L2) cache of 128 KB.

• The level 2 cache is used primarily by the GPU.

• The Raspberrypi2 uses a Broadcom BCM2836 SoC with a 900 MHz 32-bit quad-core ARM cortex A7 processor with 256KB shared L2 cache.

• The Raspberrypi2 uses a Broadcom BCM2837 SoC with a 1.2 GHz 64-bit quad-core ARM cortex A53 processor, with a 512 KB shared L2 cache.

- **Power Source**: - The recommended and easiest way to power the Raspberry pi is via the Micro USB port on the side of the unit.
  - The recommended input voltage is 5V, and the recommended input current is 2A.
- **SD Card (Secure Digital Card)**: The Raspberry Pi does not have any locally available storage accessible.

• The working framework is stacked on a SD card which is embedded on the SD card space on the Raspberry Pi.

- GPIO (General Purpose Input Output): General –purpose input/output (GPIO) is a non-specific pin on a coordinated circuit to know is an input or output pin which can be controlled by the client at run time.

  • GPIO capabilities may include
  1. GPIO pins can be designed to be input or output.
  2. Input values are meaning (normally high=1, low=0).
  3. Yield values are writable/meaningful.
  4. Input values can frequently be utilized as IRQs (interrupt request).

- **DSI Display x:** The Raspberry pi Connector S2 is a display serial interface (DSI) for connecting a liquid crystal display (LCD) panel using a 15-pin ribbon cable.
- **Audio Jack:** A standard 3.5 mm TRS connector is accessible on the Rpi for stereo sound yield.
  - Any earphone or 3.5mm sound link can be associated straightforwardly.
- **Status LEDS:** There are 5 status LEDs on the Rpi that demonstrate the status of different exercise

---

8. Explain the Smart City IOT architecture.

Ans:

The answer to this question is as big as Bengaluru.

Just Give Up Da

---

9. Explain the different Pins/Parts of Arduino Uno Board.

Ans:

Digital Pins

In addition to the specific functions listed below, the digital pins on an Arduino board can be used for general purpose input and output via the pinMode(), digitalRead(), and digitalWrite() commands. Each pin has an internal pull-up resistor which can be turned on and off using digitalWrite() (w/ a value of HIGH or LOW, respectively) when the pin is configured as an input. The maximum current per pin is 40 mA.

- Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. On the Arduino Diecimila, these pins are connected to the corresponding pins of the FTDI USB-to-TTL Serial chip. On the Arduino BT, they are connected to the corresponding pins of the WT11 Bluetooth® module. On the Arduino Mini and LilyPad Arduino, they are intended for use with an external TTL serial module (e.g., the Mini-USB Adapter).
- External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
- PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function. On boards with an ATmega8, PWM output is available only on pins 9, 10, and 11.
- BT Reset: 7. (Arduino BT-only) Connected to the reset line of the Bluetooth® module.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.
- LED: 13. On the Diecimila and LilyPad, there is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

Analog Pins

- In addition to the specific functions listed below, the analog input pins support 10-bit analog-to-digital conversion (ADC) using the analogRead() function. Most of the analog inputs can also be used as digital pins: analog input 0 as digital pin 14 through analog input 5 as digital pin 19. Analog inputs 6 and 7 (present on the Mini and BT) cannot be used as digital pins.

Power Pins

- VIN (sometimes labelled "9V"). The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin. Note that different boards accept different input voltages ranges, please see the documentation for your board. Also note that the LilyPad has no VIN pin and accepts only a regulated input.

- 5V. The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- 3V3. (Diecimila-only) A 3.3 volt supply generated by the on-board FTDI chip.
- GND. Ground pins.

Other Pins

- AREF. Reference voltage for the analog inputs. Used with analogReference().
- Reset. (Diecimila-only) Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.