Industrial Control Systems

Internet of Things/Industrial Internet of Things (IoT/IIoT)
Thing1 and Thing2

• No, literally. Today, most electronics or electro-mechanical devices can connect to the “world” via the internetwork. While this may seem “Cool” – think again.

• What is the difference between Electrical, Electronic, & Electro-Mechanical?

• And What are the risks of connecting these devices to the internet after all?
Electronics Vs. Electrical vs. Mechanical

- Made with Semiconductor materials like Silicon
- You would hear things like Diodes, transistors, resistors etc.,
- Think “Applied Electricals”, Integrated chips that are smarter with the “juice” ;)
- Pertains to Electrons

- Made with Conducting materials alike Alloys/metals etc., Copper
- Hear things like Wires, Fuse, Switch, Motor, transformers
- Think Lightning rods and other cool stuff!

- Any physical body/thing
- The “juice” here can be electrical, electronics or just mechanical – like the clock

---

How a Mechanical Watch Works

Mechanical watches have no battery, memory, or circuit.
Watch parts are machined to near-perfection at microscopic sizes and tolerances and can produce accuracy within 1-3 seconds per day.

---

Analog Circuits vs Digital Circuits

Example of Analog Signal

![Graph of Analog Signal](https://www.schoolelectronic.com/2012/01/difference-between-analog-and-digital.html)

Example of Digital Signal

![Graph of Digital Signal](https://www.schoolelectronic.com/2012/01/difference-between-analog-and-digital.html)

---


[https://animagraffs.com/mechanical-watch/](https://animagraffs.com/mechanical-watch/)
Internet of Things?

IoT and IIoT – Do we really need our personal coffee maker on the internet?
Industry 4.0

https://pattiengineering.com/blog/faqs-on-iiot/
What are Industrial Control Systems?

What do they control?
Cybersecurity permeates many aspects of our lives.
ICS Security, is it not the same as IT Security?

NIST Guide to Industrial Control Systems (ICS) Security
**FIGURE 3.** An example industrial IoT network.

https://www.researchgate.net/publication/326961617_A_Graph-Based_Security_Framework_for_Securing_Industrial_IoT_Networks_from_Vulnerability_Exploitations
Why A Smart City Framework?

Internet of Things blurs the line between Electrical and Mechanical Engineering, Operational, Architecture, and Design professionals can no more detach themselves from the matters of security

As they embark on designing infrastructure for cities and industrial systems, thinking about safety, security, and privacy becomes essential

What were secure through obscurity are now deemed unsecure for the very same reasons
United Nations Sustainable Development Goals

[Image: Logos for Sustainable Cities and Communities (11), Partnerships for the Goals (17), Responsible Consumption and Production (12), Industry, Innovation and Infrastructure (9)]

[Link: https://sustainabledevelopment.un.org/sdgs]
ICS close-up

- Process Control System (PCS)
- Distributed Control Systems (DCS)
- Programmable Logic Controllers (PLC)
- Supervisory Control and Data Acquisition (SCADA)
- Safety Instrumented Systems (SIS)
- Human Machine Interface (HMI)
- Remote Terminal Unit (RTU)

SCADA (Supervisory Control and Data Acquisition)

These systems are used in distribution systems such as water distribution and wastewater collection systems, oil and natural gas pipelines, electrical utility transmission and distribution systems, and rail and other public transportation systems.

SCADA systems integrate data acquisition systems with data transmission systems and HMI software to provide a centralized monitoring and control system for numerous process inputs and outputs.

SCADA systems are designed to collect field information, transfer it to a central computer facility, and display the information to the operator graphically or textually, thereby allowing the operator to monitor or control an entire system from a central location in near real time.

Based on the sophistication and setup of the individual system, control of any individual system, operation, or task can be automatic, or it can be performed by operator commands.
Distributed Control Systems (DCS)

- DCS are used to control production systems within the same geographic location for industries such as oil refineries, water and wastewater treatment, electric power generation plants, chemical manufacturing plants, automotive production, and pharmaceutical processing facilities.
- Typical control devices include Programmable Logic Controller, a Process Controller, a loop controller, a machine controller.
ICS Overview: Terms & Definitions

HMI: Human-Machine Interface

A human-machine interface is the user interface to the processes of an industrial control system. An HMI effectively translates the communication to and from PLCs, RTUs, and other industrial assets to a human-readable interface, which is used by control systems operators to manage and monitor processes. An HMI can range from a physical control panel with buttons to an industrial PC with a colour graphics display running dedicated HMI software.
Five terms to know well – Here’s a great article on Electrical Engineering Portal (EEP) website

Understanding Risks

Risk, Threat, Vulnerability, Impact, Likelihood
Look at the landscape

**Hazards**
- Fire
- Explosion
- Natural hazards
- Hazardous materials spill or release
- Terrorism
- Workplace violence
- Pandemic disease
- Utility outage
- Mechanical breakdown
- Supplier failure
- Cyber attack

**Assets at Risk**
- People
- Property including buildings, critical infrastructure
- Supply chain
- Systems/equipment
- Information Technology
- Business operations
- Reputation of or confidence in entity
- Regulatory and contractual obligations
- Environment

**Impacts**
- Casualties
- Property damage
- Business interruption
- Loss of customers
- Financial loss
- Environmental contamination
- Loss of confidence in the organization
- Fines and penalties
- Lawsuits
A bit more quantitative

Cyber Vulnerabilities

https://www.us-cert.gov/ics/content/overview-cyber-vulnerabilities

- Sending commands directly
- Changing the database
- Man-in-the-middle
- Exporting HMI Screen
# Smart Grid Threat Landscape

<table>
<thead>
<tr>
<th>Threat Type</th>
<th>Corporation</th>
<th>Cyber-criminals</th>
<th>Employees</th>
<th>Hacktivists</th>
<th>Nation States</th>
<th>Natural Disasters</th>
<th>Terrorists</th>
<th>Cyberterrorists</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical attacks</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unintentional damage</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Failures / Malfunction</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Eavesdropping / Interception / Hacking</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Legal</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nefarious activity / abuse</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Outages</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Damage / Loss (IT-Assets)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Disaster</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Involvement of Threat Agents in the threats

# Mobile Top 10 2016 - Top 10

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>M1 - Improper Platform Usage</strong></td>
<td>This category covers misuse of a platform feature or failure to use platform security controls. It might include Android intents, platform permissions, misuse of TouchID, the Keystore, or some other security control that is part of the mobile operating system. There are several ways that mobile apps can experience this risk.</td>
</tr>
<tr>
<td><strong>M2 - Insecure Data Storage</strong></td>
<td>This new category is a combination of M2 + M4 from Mobile Top Ten 2014. This covers insecure data storage and unintended data leakage.</td>
</tr>
<tr>
<td><strong>M3 - Insecure Communication</strong></td>
<td>This covers poor handshaking, incorrect SSL versions, weak negotiation, cleartext communication of sensitive assets, etc.</td>
</tr>
</tbody>
</table>
| **M4 - Insecure Authentication** | This category captures notions of authenticating the end user or bad session management. This can include:  
- Failing to identify the user at all when that should be required  
- Failure to maintain the user’s identity when it is required  
- Weaknesses in session management |
| **M5 - Insufficient Cryptography** | The code applies cryptography to a sensitive information asset. However, the cryptography is insufficient in some way. Note that anything and everything related to TLS or SSL goes in M3. Also, if the app fails to use cryptography at all when it should, that probably belongs in M2. This category is for issues where cryptography was attempted, but it wasn’t done correctly. |
| **M6 - Insecure Authorization** | This is a category to capture any failures in authorization (e.g., authorization decisions in the client, forced browsing, etc.). It is distinct from authentication issues (e.g., device enrollment, user identification, etc.). If the app doesn’t authenticate users at all in a situation where that should (e.g., granting anonymous access to some resource or service when authenticated and authorized access is required), then that is an authentication failure not an authorization failure. |
| **M7 - Client Code Quality** | This was the “Security Decisions Via Untrusted Inputs”, one of our lesser-used categories. This would be the catch-all for code-level implementation problems in the mobile client. That’s distinct from server-side coding mistakes. This would capture things like buffer overflows, format string vulnerabilities, and various other code-level mistakes where the solution is to rewrite some code that’s running on the mobile device. |
| **M8 - Code Tampering** | This category covers binary patching, local resource modification, method hooking, method swizzling, and dynamic memory modification. Once the application is delivered to the mobile device, the code and data resources are resident there. An attacker can either directly modify the code, change the contents of memory dynamically, change or replace the system APIs that the application uses, or modify the application’s data and resources. This can provide the attacker a direct method of subverting the intended use of the software for personal or monetary gain. |
| **M9 - Reverse Engineering** | This category includes analysis of the final core binary to determine its source code, libraries, algorithms, and other assets. Software such as IDA Pro, Hopper, etc., and other binary inspection tools give the attacker insight into the inner workings of the application. This may be used to exploit other nascent vulnerabilities in the application, as well as revealing information about back-end servers, cryptographic constants and ciphers, and intellectual property. |
| **M10 - Extraneous Functionality** | Often, developers include hidden backdoor functionality, or other internal development security controls that are not intended to be released into a production environment. For example, a developer may accidentally include a password as a comment in a utilities app. Another example includes disabling of 2-factor authentication during testing. |

https://www.owasp.org/index.php/Mobile_Top_10_2016-Top_10
Understanding the value of what we want to protect

Asset Characterization

• What asset (information) needs to be protected?
• Why does the asset need to be protected?
• Who has the responsibility for managing and protecting the asset (what are the roles, responsibilities, accountabilities and authorities)?
• If the threat actor compromised the asset, what realistic worst-case scenarios would result?
• What is the value of the asset?
• What is the criticality of the process or information to the business mission?
• What are the protection levels for confidentiality, integrity, and availability?
• What interconnections are required for the systems to perform?
• What methods are currently available for user access?
• What dependencies are present for system functionality?
• How does the information flow through the system, and through what mechanisms?

Recommended Practice

# MITRE Threat ATT&CK – Adversary tactics and techniques

ATT&CK Matrix for Enterprise

<table>
<thead>
<tr>
<th>Initial Access</th>
<th>Execution</th>
<th>Persistence</th>
<th>Privilege Escalation</th>
<th>Defense Evasion</th>
<th>Credential Access</th>
<th>Discovery</th>
<th>Lateral Movement</th>
<th>Collection</th>
<th>Command and Control</th>
<th>Elevation</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drive-by Compromise</td>
<td>AppleScript</td>
<td>bash_profile and bash;</td>
<td>Access Token Manipulation</td>
<td>Access Token Manipulation</td>
<td>Account Manipulation</td>
<td>Account Discovery</td>
<td>Audio Capture</td>
<td>Commonly Used Port</td>
<td>Automated Exfiltration</td>
<td>Account Access Removal</td>
<td></td>
</tr>
<tr>
<td>Exploit Public-Facing Application</td>
<td>CMSTP</td>
<td>Accessibility Features</td>
<td>Accessibility Features</td>
<td>Binary Padding</td>
<td>Batch History</td>
<td>Application Window Discovery</td>
<td>Application Deployment Software</td>
<td>Automated Collection</td>
<td>Communication Through Removable Media</td>
<td>Data Compressed</td>
<td></td>
</tr>
<tr>
<td>External Remote Services</td>
<td>Command Line Interface</td>
<td>Account Manipulation</td>
<td>AppCert Dlls</td>
<td>BITS Jobs</td>
<td>Brute Force</td>
<td>Browser Bookmark Discovery</td>
<td>Component Object Model and Distributed COM</td>
<td>Clipboard Data</td>
<td>Connection Proxy</td>
<td>Data Encrypted for Impact</td>
<td></td>
</tr>
<tr>
<td>Hardware Additions</td>
<td>Compiled HTML File</td>
<td>AppCert Dlls</td>
<td>AppCert Dlls</td>
<td>Bypass User Account Control</td>
<td>Credential Dumping</td>
<td>Domain Trust Discovery</td>
<td>Exploitation of Remote Services</td>
<td>Data from Information Repositories</td>
<td>Custom Command and Control Protocol</td>
<td>Data Transfer Size Limits</td>
<td>Data Encoded</td>
</tr>
<tr>
<td>Replication Through Removable Media</td>
<td>Component Object Model and Distributed COM</td>
<td>AppCert Dlls</td>
<td>Application Shimming</td>
<td>Clear Command History</td>
<td>Credentials from Web Browsers</td>
<td>File and Directory Discovery</td>
<td>Internal Spearphishing</td>
<td>Data from Local System</td>
<td>Custom Cryptographic Protocol</td>
<td>Exfiltration Over Alternative Protocol</td>
<td></td>
</tr>
<tr>
<td>Spearphishing Attachment</td>
<td>Control Panel Items</td>
<td>Application Shimming</td>
<td>Bypass User Account Control</td>
<td>CMSTP</td>
<td>Credentials in Files</td>
<td>Network Service Scanning</td>
<td>Logon Scripts</td>
<td>Data from Network Share Drive</td>
<td>Data Encoding</td>
<td>Exfiltration Over Command and Control Channel</td>
<td>Disk Structure Wipe</td>
</tr>
<tr>
<td>Spearphishing Link</td>
<td>Dynamic Data Exchange</td>
<td>Authentication Package</td>
<td>DLL Search Order Hijacking</td>
<td>Code Signing</td>
<td>Credentials In Registry</td>
<td>Network Share Discovery</td>
<td>Pass the Hash</td>
<td>Data from Removable Media</td>
<td>Data Collection</td>
<td>Exfiltration Over Other Network Medium</td>
<td>Endpoint Denial of Service</td>
</tr>
<tr>
<td>Spearphishing via Service</td>
<td>Execution through API</td>
<td>BITS Jobs</td>
<td>Dll Hijacking</td>
<td>Compile After Delivery</td>
<td>Exploitation of Credential Access</td>
<td>Network Sniffing</td>
<td>Pass the Ticket</td>
<td>Data Staged</td>
<td>Domain Fronting</td>
<td>Exfiltration Over Physical Medium</td>
<td>Firmware Corruption</td>
</tr>
<tr>
<td>Supply Chain Compromise</td>
<td>Execution through Module Load</td>
<td>BitKid</td>
<td>Elevated Execution with Prompt</td>
<td>Compiled HTML File</td>
<td>Forced Authentication</td>
<td>Password Policy Discovery</td>
<td>Remote Desktop Protocol</td>
<td>Email Collection</td>
<td>Domain Generation Algorithms</td>
<td>Scheduled Transfer</td>
<td>Inhibit System Recovery</td>
</tr>
<tr>
<td>Trusted Relationship</td>
<td>Exploitation for Client Execution</td>
<td>Browser Extensions</td>
<td>Essential Component Firmware</td>
<td>Hooking</td>
<td>Peripheral Device Discovery</td>
<td>Remote File Copy</td>
<td>Input Capture</td>
<td>Fallback Channels</td>
<td>Network Denial of Service</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

[https://attack.mitre.org/](https://attack.mitre.org/)
MITRE Threat Modeling

Figure 4. Key Constructs in Cyber Threat Modeling (Details for Adversarial Threats Not Shown)

System-of-Systems Threat
Some well-known cyber attacks on industrial systems

The IoT is poised to bring a new world of benefits to businesses operating industrial systems—optimized operations and supply chain management, improved business agility, new revenue streams and services among others. But the benefits also come with trade-offs, as some of the systems’ security and resilience is compromised. With these new threats, the benefits of the IoT are offset by the growing risk of cyber-attacks. The following are some of these attacks:

1. **Exploits**
   - **The Blubot Worm**
     - A worm created by German researchers to show that a botnet is possible.
     - The worm was able to infect industrial control systems connected to the Internet in a short period of time.

2. **The SCADA System**
   - A system used in critical infrastructure, such as power plants and water treatment facilities.
   - A vulnerability was discovered in the system that allowed an attacker to gain control.

3. **Smart Motors**
   - A motor that was used in a factory production line.
   - A vulnerability was found in the motor that allowed an attacker to gain control and disrupt production.

4. **German Steel Mill**
   - A steel mill in Germany that was targeted by a cyber-attack.
   - The attack caused significant damage to the mill's operations.

5. **Cyber-Attacks**
   - These attacks were actual attacks by hackers!

Implement Security into Your IIoT Ecosystems Now

According to the Industrial Internet Consortium (IIoT), only 25% of organizations have a clear IoT security strategy. Leaders are struggling most with data security (59%) and privacy (39%).

Overcoming these barriers is essential to the success of the IIoT. The following are tips for implementing security in your IIoT ecosystem:

1. **Security by Design**
   - Build security into IoT systems from the earliest possible stage.

2. **Information Security Priorities**
   - Leverage established standards and models, such as NIST, ISO/IEC 27001, and IEC 62443, to define your security framework.

3. **Use Proven Technologies and Standards**
   - Choose proven technologies and protocols that have been vetted and tested.

4. **Leverage the Cloud**
   - Use cloud-based services that have security as a core competency.

5. **Don't Give It Away**
   - Ensure that all components of your IoT ecosystem are secure and not vulnerable to attack.

With decades of experience in identity services provider and proving trusted, key infrastructure (PKI) and identity and Access Management (IAM) solutions, GlobalSign is uniquely positioned to help you build identity, management, and security into your IIoT (Internet of Things) ecosystem with minimal CARES and time to market. For more information, visit http://bit.ly/managed-lt
Humans are fallible!

The Four-Fold pattern of Preferences from Thinking Fast and Slow, Kahneman & Tversky, pg 317

<table>
<thead>
<tr>
<th>High Probability</th>
<th>Losses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fear of Disappointment</td>
<td>(e.g. 95% chance to lose $10000)</td>
</tr>
<tr>
<td>Risk Averse</td>
<td>Reject unfaourable settlement even with high chances of losing</td>
</tr>
<tr>
<td>Possibility Effect</td>
<td>Gains</td>
</tr>
<tr>
<td>Hopes of Large Gain</td>
<td>(e.g. 5% chance to win $10000)</td>
</tr>
<tr>
<td>Risk Seeking</td>
<td>Accept unfaourable settlement even with high chances of win</td>
</tr>
<tr>
<td>Low Probability</td>
<td>Pain</td>
</tr>
<tr>
<td>Buy Lottery</td>
<td>Accept unfaourable settlement even with high chances of win</td>
</tr>
<tr>
<td>Risk Averse</td>
<td>(e.g. 5% chance to lose $10000)</td>
</tr>
<tr>
<td>Fear of Large Loss</td>
<td>Reject unfaourable settlement even with high chances of losing</td>
</tr>
</tbody>
</table>
But we also work under many limitations

| Constraints (rings) within which businesses have to operate |
|-----------------|----------------|----------------|----------------|
| Law                          | Regulations    | Contractual Obligations | Geopolitics | Social-cultural |

Godha bapuji ©
Understanding security matters
Confidentiality, Integrity, Privacy, Availability, Authentication, Authorisation, Access Control
Cybersecurity

- Human Security
- Service Availability
- Integrity
- Economics, Finance, Banking
- Law & Order
- International Relations & Foreign Affairs
- Armed Forces
- Civil Life & Domestic Affairs

Provides optional Confidentiality

Strengthens Physical Safety through Law & Order, and Governance, Risk, Compliance, and Ethics
Understanding Cultures and Geopolitics of the world

Nations do not engage in conflict for fun – there are reasons – find those intentions, motives, instigations, interests, their strong and weak points to better secure critical infrastructure
How do we protect these systems then?

Understanding defenses, Business Continuity and Resilience Needs of each system
We must pick and choose

Prioritise what matters most; Think about what interests you most; Figure out what you can do the best in the available time
Can I help to protect our ICS?

Of Course! You must join in the workforce for the future! Skills required to defend our ICS

https://dragos.com/blog/industry-news/a-dragos-industrial-control-system-security-reading-list/

https://www.state.gov/students/

Understanding Network Protocols and how they might differ in Industrial systems

Understanding Policies and Safety Regulations in Industrial zones

Understanding how Electrical, Electronics and Mechanical devices work together in Industrial systems

Understanding Risks, Vulnerabilities, threats, and impact on communities due to an industrial system failure

https://ics-cert-training.inl.gov/learn
Learn Cloud
AWS IoT Cloud Services: https://aws.amazon.com/iot/solutions/industrial-iot/
Computer Sciences

Diagram:

- Computer Science (CS)
  - Computer Applications
  - Programming
  - Artificial Intelligence (AI) & Machine Learning (ML)
  - Hardware Engineering
  - Software Engineering (SE)
  - Data Sciences (DS)
  - Information and Communication Technologies (ICT)
  - Information Technology (IT)
  - Information Security (IS)
  - CS, CA, IT & ICT
An extremely simple and fun Case Study with AWS I played around with last year just for fun
AWS IoT with AWS Greengrass and Buttons

Raspberry pi 3 model B+
Through this PoC, I tried the following:

1. To use physical hardware to communicate and interact with AWS IoT
2. I used two physical hardware – AWS IoT button and a Raspberry Pi

This short framework consisted of AWS IoT and AWS Greengrass talking to each other securely via certificate-based authentication from Amazon’s trust center’s root certificate.

It also involves AWS Greengrass successfully communicating with AWS Lambda routines.

One of the interesting parts of this PoC was to use a physical hardware and in this case it was the IoT Button. We not only configured the IoT Button into the AWS IoT console but also onboarded the AWS IoT as part of the Greengrass Core devices through the device onboarding process for the groups.

The installation and setup of Raspberry Pi with AWS IoT and with Greengrass are very similar to EC2 instances with the only difference being the physical hardware setup involved in case of the Pi.

This PoC used the Pi as an example to see how easy or difficult it was to onboard a physical device into a Greengrass group and as a result of this PoC I conclude that except the hardware configuration part wherein you have to not only manage your hardware but also find the right API’s that can be used with AWS IOT, once this step is done – AWS Greengrass and Lambda make it easy straightforward to communicate with the devices. As per the documentation, new devices such as the such as sensors, motors can also be onboarded using the OPC-UA protocol mechanisms in Greengrass.

This project used MQTT and TLS for communication and security but Greengrass is capable of using other protocols such as MODBUS over TCP.
Workflow of my PoC

1. Press Amazon IoT Button
2. Greengrass receives this
3. Lambda is triggered
4. An SNS is executed to send SMS
5. MQTT sends SMS
6. Message is received on my phone
Useful information

- Great presentation by Boaz Ziniman @Goto Conference Amsterdam, 2018
  https://www.youtube.com/watch?v=FrH-EQfQkRU
- onboard your iot button using this link
  https://docs.aws.amazon.com/iot/latest/developerguide/iot-console-signin.html
- onboard your raspberry pi using this link
  https://docs.aws.amazon.com/greengrass/latest/developerguide/module1.html
- The instructions are for Greengrass from Module 2 onwards
- recordmydesktop
  sudo apt-get install recordmydesktop
  launch with recordmydesktop -no-sound on the commandine
  ctrl+c to quit
What's in my pi

- 1.2 GHz 64-bit quad core CPU, ARM v7 Debian Stretch
- 1 GB RAM
- 4 USB ports
- 1 micro-USB port
- 1 HDMI slot
- 3.5 mm mini jack for AV
- 1 ethernet port
- And all other usual stuffing that go into a raspberry pi 😊

- Bring your own wires, cables, standard keyboard, a mouse and a HDMI or DVI compatible monitor and adaptor

- I also have a NOOBS SD card with Raspbian setup: I used the instructions on Raspberry Pi website to set this up. It took a while a few hurdles but worked in the end.

- I did not find the pre-installed NOOBS card useful as the memory will need expansion later and the process is quirky.

- Take usual care as you would with any delicate electronic chip or board!
Getting your Hardware ready – a minimalist version!

- Use `sudo raspi-config` to enable SSH and VNC on your Raspberry so you can remote login.
- I have the RealVNC software on my Windows to connect as client into my pi.
- Always remember to safely shutdown or reboot your pi either from console or from the Raspberry menu.
Introduction and overview

- AWS SDKs, AWS IoT and AWS IoT SDKs
- AWS Greengrass core and SDK
- AWS IoT Button
- AWS Lambda
- AWS SNS with SMS
- AWS EC2 Linux instance
- Python and respective AWS Greengrass Core SDK software

My aim was to explore how Greengrass could be setup and how if at all make it work with AWS IoT and the buttons.

Greengrass is found under AWS Internet of Things but has an ecosystem of its own on the console settings.

Greengrass is a software that sits in between your IoT and the edge IoT devices brokering connections across different platforms and protocols, a typical case in an industrial infrastructure.

With greengrass your IoT device can now talk to IoT either using TCP or MQTT channels.

With Greengrass, your IoT devices need always not remain online and only connect to the cloud when they are due for an update.

- You will other AWS IoT SDKs as per your programming needs.
- I installed python and greengrass only supports 2.7 at present.
- `sudo pip install AWSIoTPythonSDK`
- I also installed the paho-mqtt libraries to able to use MQTT from the programs
- `sudo pip install paho-mqtt`
Details

- The greengrass root directory is typically /greengrass and the config file is config.json found in the /greengrass/config folder.

- Default MQTT/TLS port is 8883 – there are instructions to change this if needed

- Create a ggc_user and a ggc_group for Greengrass
  - `sudo adduser --system ggc_user`
  - `sudo groupadd --system ggc_group`

- Make the greengrass directory read-only for security

- Make the ggc_user the owner of certs and lambda functions

- All steps clearly given in the developer guide – I used it to setup my greengrass. DO NOT USE STEP 10 – it corrupted my kernel and I had to redo everything on my Pi, from SD card formatting!

- MANDATORY STEPS 11 and 12 : to create protection for hardlinks and symbolic links and to enable memory limits for Lambda function.

Before you run the daemon – it helps emotionally to check the dependencies

- cd /home/pi/Downloads
- wget https://github.com/aws-samples/aws-greengrass-samples/raw/master/greengrass-dependency-checker-GGCv1.7.0.zip
- unzip greengrass-dependency-checker-GGCv1.7.0.zip
- cd greengrass-dependency-checker-GGCv1.7.0
- sudo modprobe configs
- sudo ./check_ggc_dependencies | more
- cd /greengrass/ggc/core
- sudo ./greengrassd start
Greengrass setup complete on my EC2 core & the daemon is running !!
Now that Greengrass is up and running – I want to see if I can use Lambda to talk to my Greengrass. This is module 3 of the developer guide under the Getting started section.

Next, I have downloaded the required AWS IoT Greengrass Core SDK software in this case.
I zipped up the folder like the guide instructs.

```bash
sudo zip -r hello_world_python_lambda.zip greengrasssdk greengrassHelloWorld.py
```
Servicerole permissions error with Greengrass!

- This one took me a while to sort out, but it worked in the end!

- This role assignment is important for Greengrass and Lambda to work.
Next, I created the Lambda function and upload the deployment zip file.

We have successfully imported our greengrass helloworld zip file into our Lambda function code.
Here's one on my Raspy

- Choose the Greengrass group
- Choose Lambdas option on the left
- Choose the alias you need
- Create subscription

and Pi nodes. The next step is to create the topic subscription so that MQTT can handle.

On the Subscriptions page for Greengrass – choose Lambda Tab and it should show the Lambdas that you created in previous steps and for Services tab, choose IoT Cloud. See the following image.
Ah, the sweet sight of success, even with the hello/world!
I used the AWS IoT Button QuickStart guide to setup my new button with the AWS Console.

https://docs.aws.amazon.com/iot/latest/developerguide/iot-button-quickstart.html

Rampant technical glitches – I could see the IoT blueprints once, and not now – wonder what’s going on!

But used the GitHub code to try it out.

https://github.com/aws-samples/aws-lambda-iot-button
Configuring your Amazon IoT button is very simple!
Details on rules

- The lambda captures 3 events:
  - Single click
  - Double click
  - Long click

- It has three different messages for each clickType as you see below in the test on my phone.
This link provides a really simple json script to send a quick text message via Lambda through SNS→SMS options to my phone.

https://docs.aws.amazon.com/iot/latest/developerguide/iot-lambda-rule.html

I followed the steps using the blueprint as a start and onboarded my IoT Button into the Lambda as per the AWS IoT setup instructions within this page.

Using the following method

Of course, I had to have my SNS topic created and setup for SMS.
### GreengrassCoreSWOnEC2

<table>
<thead>
<tr>
<th>Deployed</th>
<th>Version</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dec 13, 2018 4:14:54 PM -0500</td>
<td>4a9d5f0a-8828-422e-90af-69f3f1e68315</td>
<td>Successfully completed</td>
</tr>
<tr>
<td>Dec 13, 2018 3:28:16 PM -0500</td>
<td>24795340-47eb-4ae0-a34e-e4f79c7e73bc</td>
<td>Successfully completed</td>
</tr>
<tr>
<td>Dec 13, 2018 3:20:54 PM -0500</td>
<td>b9f08b4e-b78a-47a1-92c4-501a4a424754</td>
<td>Successfully completed</td>
</tr>
<tr>
<td>Dec 13, 2018 3:18:43 PM -0500</td>
<td>5f0059e6-ba8e-421a-92c4-4c01a4a24754</td>
<td>Successfully completed</td>
</tr>
<tr>
<td>Dec 13, 2018 2:55:57 PM -0500</td>
<td>5a0619ce-1e92-450a-9b4b-b8573ae0585</td>
<td>Successfully completed</td>
</tr>
<tr>
<td>Dec 13, 2018 2:15:54 PM -0500</td>
<td>d6ec7f90-fb26-464b-8731-f165139f6e6</td>
<td>Successfully completed</td>
</tr>
<tr>
<td>Dec 13, 2018 2:18:18 PM -0500</td>
<td>20e080f4-4e1d-4a87-8023-42018ce4168</td>
<td>Failed</td>
</tr>
<tr>
<td>Dec 13, 2018 2:13:00 PM -0500</td>
<td>92990f05-1576-4bd0-8031-3779a559c56e</td>
<td>Failed</td>
</tr>
<tr>
<td>Dec 13, 2018 2:06:05 PM -0500</td>
<td>f8a5b97e-a58c-48be-a753-928fc0d6e77</td>
<td>Successfully completed</td>
</tr>
<tr>
<td>Dec 13, 2018 1:46:14 PM -0500</td>
<td>75786621-8161-477d-8e64-a558d99b0969</td>
<td>Successfully completed</td>
</tr>
</tbody>
</table>

### GreengrassGroupOnRaspy

<table>
<thead>
<tr>
<th>Deployed</th>
<th>Version</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dec 13, 2018 2:00:53 PM -0500</td>
<td>287caf15-42ef-4f53-a003-2b9169f66c6e</td>
<td>Successfully completed</td>
</tr>
<tr>
<td>Dec 13, 2018 1:45:50 PM -0500</td>
<td>5e09d5e6-c985-417e-b03c-47f3539a78d3</td>
<td>Successfully completed</td>
</tr>
<tr>
<td>Dec 13, 2018 12:16:19 PM -0500</td>
<td>5e09d5e6-c985-417e-b03c-47f3539a78d3</td>
<td>Failed</td>
</tr>
</tbody>
</table>
In my case this was "myIoButtonLambdaGG"

Once this is complete, Head over to AWS IoT → Test option and give your topic as "IoTbutton" = YOUR BUTTON's DSN found on the back of the hardware

Now lets test it:

Press the button once:

I see my message as per the IoT+Lambda script!

Press the button twice in quick succession to see the logic trigger for clickType=Double

Press the button long enough for a few seconds to trigger the third logic! Viola! We have successfully controlled our Lambda with our little IoT Button and both of these are being run by our Greengrass core. Here's the sample of the messages I got on my phone.

Lets now create a rule in the IoT console to trigger this lambda o the press of the Gripper friendly IoT Button!

Choose Act → Create

In the Create a Rule dialog box – setup the rule as you would for any MQTT rule. I specified a simple query to fetch everything!
What else should I learn?
• Learn MITRE’s threat modeling: a good intro is here:
  https://digitalguardian.com/blog/what-mitre-attck-framework
• Learn about Kill Chains – there are various – start here:
  https://www.varonis.com/blog/mitre-attck-framework-complete-guide/
  and here: https://medium.com/datadriveninvestor/att-ck-model-c40a113aab4
International Relations & International Security

- Take courses in International Relations
- Study how various nations perceive cybersecurity
- Take courses in International Security
- Study what international laws apply to the field of cyber security
- A good place to start would be the NATO website
- The Tallinn 2.0 manual is a great resource to understand cyber laws and other international laws that apply in a cyberspace conflict
Global Cybersecurity Index


Figure 4: Heat map showing geographical commitment around the world

Disclaimer: The designations employed and the presentation of this map do not imply the expression of any opinion whatsoever on the part of the ITU concerning the legal status of any country, state, territory or area and/or of its authorities, or concerning the delimitation of its boundaries or frontiers. Efforts were made to ensure this map is free of errors; however, there is no warranty the map or its features are either spatially or temporally accurate or fit for a particular use. This map is provided without any warranty of any kind whatsoever, either express or implied.

Source map: UN.org

The colours in the heat map above indicate differences in the level of commitment with high, medium, and low scores in a range of colours from light blue (peak commitment) to dark blue (low commitment). This is also reflected in the GCI groups in section 4.2.

4.2 GCI groups

Countries are classified according to their level of commitment: high, medium, and low.

1. Countries that demonstrate high commitment in all five pillars of the index.
2. Countries that have developed complex commitments and engage in cybersecurity programmes and initiatives.
3. Countries that have started to initiate commitments in cybersecurity.
US DHS VLP provides resources to those interested to make a career in the field
https://ics-cert-training.inl.gov/learn
The Learning Map

Understanding Key Terms  →  Establishing Priorities  →  Understanding Threat Landscape  →  Risk Preferences and Decision-making

Continuous Improvement  →  The People Factor  →  Communicating  →  The Technology Factor

Learning  →  Applying Theory to Practice  →  Recording Lessons from the field  →  Taking practice back into classroom