**DUY TAN UNIVERSITY**

**INTERNATIONAL SCHOOL**

**NETWORK SECURITY**

**⅏🕮⅏**



***TOPIC:***

**IOT PENTEST**

**Teacher : Anand Nayyar**

**Class : CMU-CS 426 BIS**

Student 1 : Nguyễn Thanh Hiệp

IDStud 1 : 2321112713

Student 1 : Phạm Thanh Thúy

IDStud : 2320118028

**TABLE OF CONTENTS**

ACKNOWLEDGGEMENT……………………………………………………………………I

INTRODUCTION……………………………………………………………………………..II

DETAILS OF THE TOPIC ……………………………………………………………….….III

DIAGRAMS…………………………………………………………………………….……IV

STEPS IOT PENTEST …………………………………………………………………….…V

CONCEPTS………………………………………………………………………………...…VI

FRAMEWORK……………………………………………………………………………….VII

LASTEST INFORMATION COVERRAGE TO TOPIC……………………………………VIII

CONCLUSION…………………………………………………………………………..……IX

ANALYSIS…………………………………………………………………………………….X

REFERENCES……………………………………………………………………………..…XI

**INTRODUCTION**

It will not be long before all our household electronic devices can actually be linked with one another over the internet. Bluetooth, Wi-Fi and mobile communications are already available in nearly every household. The Internet of Things is becoming more and more important, and is gradually becoming part of our everyday life. Our household appliances will be able to both exchange and process information. Our lives will therefore become easier, faster, more comfortable, connectable and smarter.

Analysts assume that more than 30 billion devices will be connected to the internet by 2020. Developers of Internet of Things (IoT) devices are faced with a decision: Should they extend existing devices with older micro controllers and thus make an internet connection and Cloud-based applications possible, but accept possible security risks in doing so? Or should they use newly developed micro controllers that have been designed especially for IoT devices? This is because different requirements are placed on IoT devices. These requirements include low power consumption, high computing performance, different interfaces and also – with increasing importance – security. Many IoT devices are powered by batteries. Great emphasis is placed here on a long operating time between the charging processes although low power consumption is also a desirable goal for the devices running off the mains. This enables the development engineers to keep the size of the overall device and the utilized components – cooling elements for example – small. Numerous sensors and input units, e.g. touch recognition and graphic displays, require enormous processing power. In addition to USB and Ethernet, many of the new IoT devices use radio connections. They mainly include Bluetooth Low Energy (BLW) and Wi-Fi. And last but not least, IoT devices will also maintain the personal privacy of the user while also providing protection against digital attacks, unauthorized access and data changes.

Generally speaking, the security requirements relating to IoT devices are higher than those for devices not connected to the internet. Recently, it has been possible to read an increasing number of reports in which IoT devices have been misused for digital attacks. In 2016, for instance, hackers misused thousands of IoT devices for Distributed Denial-of-Service (DDoS) attacks and paralyzed large areas of the internet. Companies such as Amazon, Netflix, Spotify and Twitter, to name but a few of the large enterprises, were affected by these attacks. For manufacturers and service providers, these kinds of attacks and outages not only mean an image loss, but also have immense commercial losses. IT security researchers presume that similar Distributed Denial-of-Service (DDoS) attacks using IoT devices will increase in future.

**DETAILS OF THE TOPIC**

1. **WHAT IS THE INTERNET OF THINGS ( IOT)?**

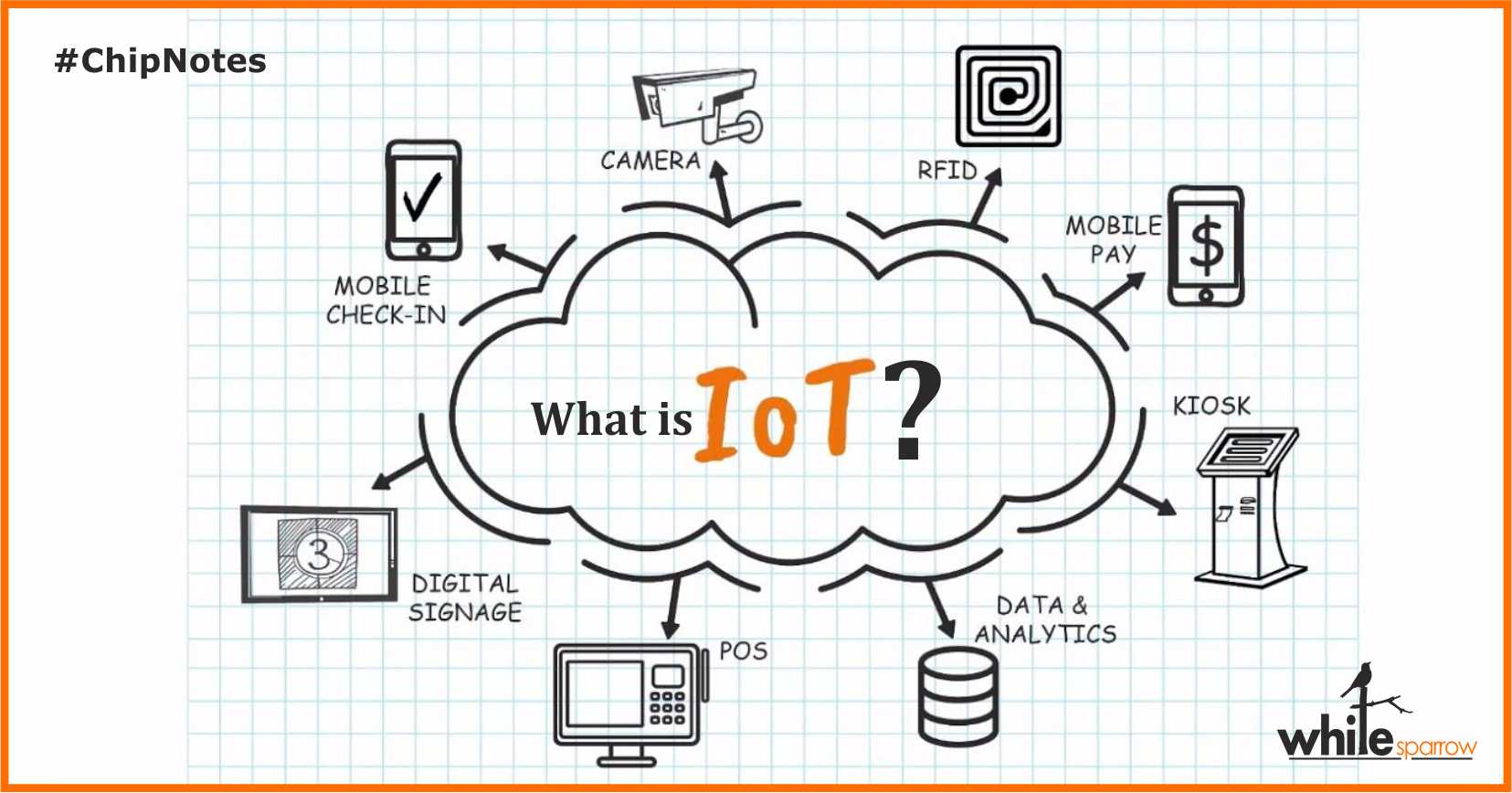


Figure 1: IoT

Communication between electronic devices is now well-established albeit often in a simple form. When we, for example, order something over the internet, we can track our parcel using our smartphone and know where the shipment is at any particular time. If our printing cartridge is running low, the printer can automatically trigger the order for a replacement.

Some human interaction is still required in the described cases. Thanks to the Internet of Things, this human effort will no longer be necessary. The IoT is not just being talked about, it has now become part of our everyday life. When looking specifically for a definition of the term, it can quickly be seen that there is more than one definition for the IoT [3]. Different institutions regard this term in different ways and one quickly stumbles from one keyword to the next. Terms such as IoT, Industry 4.0, M2M – the interlinking of machines and systems is referred to in many ways and in reality all mean the same:

IoT means communication between smart devices without any human intervention. It must be assumed here that IoT devices often have an IP address and can be reached as a separate unit in the local network and over the internet as well. The Internet of Things is not just a technology, it also combines a whole range of different technologies. These devices are therefore able to acquire, evaluate and exchange data, and also use the data for further activities. This abundance of possibilities will accompany us in daily life and reduce our workload. However, this also means that there is a whole range of new digital attack scenarios associated with such refined technological developments.

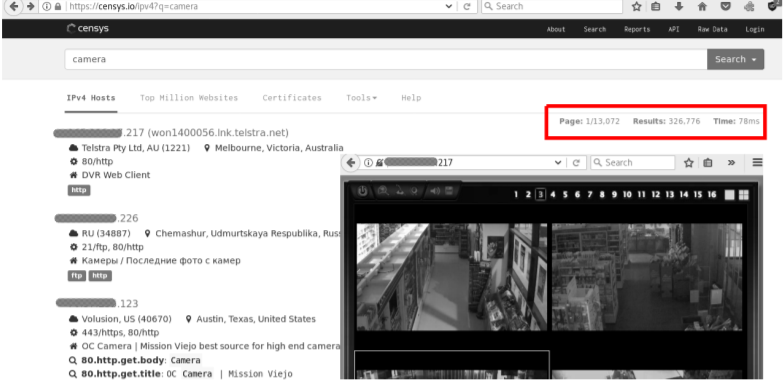


Figure 1: Censys: IoT search engine with potential hazards.

## WHAT IS PENTESTRATION TESTING ( PENTEST) ?

A penetration test, also known as a pen test, is a simulated cyber attack against your computer system to check for exploitable vulnerabilities. In the context of web application security, penetration testing is commonly used to augment a [web application firewall (WAF)](https://www.imperva.com/products/web-application-firewall-waf/).

Pen testing can involve the attempted breaching of any number of application systems, (e.g., application protocol interfaces (APIs), frontend/backend servers) to uncover vulnerabilities, such as unsanitized inputs that are susceptible to code injection attacks.

Insights provided by the penetration test can be used to fine-tune your WAF security policies and patch detected vulnerabilities.

1. **WHAT IS THE IOT PENTEST?**

A connected device is a complex solution, with various potential entry doors for an attacker. A connected device security audit (or pentest IoT) includes tests on the entire object ecosystem, i.e. electronic layer, embedded software, communication protocols, server, web and mobile interfaces. Server-side, web interfaces and mobile applications tests are not specific to IoT, however they are important tests as they are particularly high-risk areas. The tests on the electronic side, embedded software and communication protocols concern vulnerabilities more specifically the IoT.

There are three specific types of attacks on connected objects and embedded systems. Software attacks, non-invasive hardware attacks and invasive hardware attacks. The first take advantage of software vulnerabilities, the second recover information from the hardware without damaging it while the third involve opening the components and therefore destroying them in order to be able to extract secrets. While the first two types of attacks do not require many resources, this is not the case for invasive attacks, for which very expensive equipment is required.

Here are ten concrete tests conducted during the security audit of a connected device, illustrated by some mediatized and emblematic examples. For each of the points discussed below, there are many tools and methods that take advantage of very different vulnerabilities. This is therefore a non-exhaustive list.

## PENTEST METHODS:

## What is Penetration Testing? | Pen Testing Methodologies and Tools ...

Figure 2: pentest methods.

### External testing:

External penetration tests target the assets of a company that are visible on the internet, e.g., the web application itself, the company website, and email and domain name servers (DNS). The goal is to gain access and extract valuable data.

### Internal testing:

In an internal test, a tester with access to an application behind its firewall simulates an attack by a malicious insider. This isn’t necessarily simulating a rogue employee. A common starting scenario can be an employee whose credentials were stolen due to a [phishing attack](https://www.imperva.com/learn/application-security/phishing-attack-scam/).

### Blind testing:

In a blind test, a tester is only given the name of the enterprise that’s being targeted. This gives security personnel a real-time look into how an actual application assault would take place.

### Double-blind testing:

In a double blind test, security personnel have no prior knowledge of the simulated attack. As in the real world, they won’t have any time to shore up their defenses before an attempted breach.

### Targeted testing:

In this scenario, both the tester and security personnel work together and keep each other appraised of their movements. This is a valuable training exercise that provides a security team with real-time feedback from a hacker’s point of view.

## MISJUDGED RISK OF THE IOT:

## This article also deals with IoT devices in the literal sense, i.e. with “intelligent” devices that are connected to the internet and can also potentially be targeted “from the outside”, that is to say not just from within a closed company network.

## Although this produces new value-added networks, new digital risks are also created. Possible dangers arise at four different levels. The first level is the hardware level. This relates to the IoT device interacting with its environment. The interaction usually takes place through sensors. It is therefore possible to manipulate certain kinds of signals in such a way that sensors are disturbed and malfunctions result. It cannot be ruled out that attackers may be able to extract sensitive data from the device when accessing the hardware. Dangers at the second level, the network level, relate to the exchange of information between the components. The attackers may induce the utilized software to trigger undesirable activities. The dangers at the third level concern the back end or Cloud solution where the various services are to be found.

## In addition to the requirements, data can therefore also be stored which are needed for the IoT products to function. These products may contain vulnerabilities, e.g. on account of the use of outdated software through which the IoT device could be misused for malicious activities. The application level is the fourth level and relates to the vulnerabilities in the operation of a web-based application or a mobile device.

## HIDDEN AND APPARENT DANGERS:

## Owners of an IoT device can quickly become the victims of malware. This was recently the case when IP cameras were affected by a vulnerability in which the infection lasted less than two minutes . Since the majority of devices susceptible to the malware “Mirai” used in this case did not have a non-volatile memory, the devices could be freed from the malware after interrupting the power supply. This is not a permanent solution, however, since these devices will be infected again within a short space of time. In principle, it makes sense to generally change the preset passwords of the devices.

## In November 2016 customers of Deutsche Telekom were the victims of a botnet attack and in the course of which would have almost been part of an even larger botnet. An attempt was made to attack the router over port 7547 and to link this with the Mirai botnet. Maintenance servers were able to contact the router via this remote maintenance port and notify the server that a software update is available. An attempt was made in this way to bring additional malware into the device. Fortunately though, the attack did not function perfectly with the result that the around 900,000 attacked routers of Telekom could not be misused for an even larger botnet, but only that some routers simply failed. Nevertheless, Deutsche Telekom suffered enormous financial losses for which the responsible 29-year-old British hacker received a suspended prison sentence of one year and eight months from Cologne District Court at the end of July 2017. This attack highlights the fact that many attacks are not apparent to the individual user.

## Many IoT systems can be found using the search engines Shodan and Censys. Even private cameras, weather stations and installations of public utility companies can be controlled.

## Figure 1 shows the search engine Censys with a search request for IP cameras. Inadequately protected cameras can provide a detailed insight into the private sphere. However, not only can IP cameras be explored, but also critical equipment control systems. This represents an immense security risk. If these systems get into the wrong hands, the consequences can be serious. With regard to IT security in particular, operators of critical infrastructures must be aware of the risks associated with interlinking devices.

## The Wikileaks press release entitled “Vault 7: CIA Hacking Tools Revealed” describes the nightmare of security experts: In this way networked automobiles will be used as a potential murder weapon. The following sentence in particular stood out in the leaked documents of the CIA: “In October 2014 the CIA also wanted to infect a vehicle control system used in modern automobiles and trucks”. The purpose of this control was not specified, but it would allow the CIA to carry out assassinations that could almost not be perceived as such. In the document quoted by Wikileaks as the source, there are also considerations by the CIA regarding attack targets with IoT devices. However, the documents do not show whether the secret service has in fact already managed to manipulate a networked vehicle and infiltrate the control system. The fact that this is possible was demonstrated by the security experts Charlie Miller and Chris Valasek when they hacked the Cherokee Jeep from Chrysler in 2015. The attacks on the entertainment system in the vehicle are also interesting. It was possible, for example, to monitor the vehicle location without having to attach a separate transmitter for this purpose. Even the microphones in the vehicle could be tapped to follow the conversations between the passengers. At present, automobile owners can hardly protect themselves against attacks or improve the IT security of their automobiles. However, it is recommended that diagnostics systems running over ODBII be unplugged if possible because the connector plugs operated over ODBII constitute a gateway for attacks against the vehicle.

## 

Figure 3: IoT smarthome.

## However, the attackers do not always need highly complex knowledge of the possible attack vectors. For example, a Smart Lock user was quite astonished when a neighbor suddenly entered the apartment without needing a key. With Smart Lock the house door can be controlled, for example, using a smartphone app. If the smartphone (or tablet) is now in the house and a language assistant is also active, a tilted window is all that is needed to open the door. The neighbor was therefore able to unlock the front door by calling out “Hey Siri, unlock the front door”. Only the instruction is required here, the language assistant is not able to differentiate between persons on the basis of their voice.

## BASIC IOT ARCHITECTURE:

Tapping into IoT penetration testing, security engineers may wrongly consider this domain less challenging, as the IoT environment doesn’t have the most common vulnerability: human error (according to CompTIA, this is the major cause for [52% of security breaches](https://www.comptia.org/about-us/newsroom/press-releases/2015/03/31/organizations-changing-strategies-and-tactics-as-security-environment-gets-more-complex-new-comptia-study-finds)). Most Internet attacks involve a user clicking a malicious link or opening an infected email. With IoT environment, there is no one to lure, so it’s harder to break into. This supposition is deceptive. Here’s what CSO says about IoT breaches in 2017: “Aruba Networks, Hewlett Packard Enterprise wireless networking subsidiary, has revealed that 84 percent of companies have already experienced some sort of IoT breach in a new study involving over 3,000 companies across 20 countries”. Intruders have more opportunities to breach an IoT system, as its architecture comprises a number of elements that become potential hacker’s targets.

Typically, an IoT architecture consists of the following components:

* **Things:** Smart devices equipped with sensors and actuators.
* **IoT field Gateways**: Border elements that provide connectivity between things and the cloud part of an IoT solution.
* **Cloud gateways**: Components facilitating data compression and transmission between the gateways and cloud servers.
* **Streaming data processor**: An element ensuring a smooth transition of input data to a big data warehouse and control applications.
* **Data storage**: Consists of a data lake (stores unprocessed data in the form of “streams”) and a big data warehouse (stores filtered and structured data, as well as context information about smart devices, sensors, commands from control applications).
* **Data analytics**: A unit that uses information from the big data warehouse to establish data patterns and gain meaningful insights.
* **Machine learning**: Generates and regularly updates models based on the historical data accumulated in a big data warehouse which is used by control applications.
* **Control applications**: Components that send automatic commands and alerts to actuators.
* **Client-server system**: Consists of a user business logic component (the server side), a mobile application and a web application (the client side).

Full-scale IoT penetration testing goes beyond smart devices and should cover all IoT system elements.

## 

## WHY AND HOW SHOULD YOU ADOPT IOT?

## First and foremost, don’t adopt IoT for IoT’s sake. Instead, focus on specific business use cases and outcomes, using IoT as the basis for an overall strategy to achieve those goals. For instance, in industrial organizations, these may be to improve operational efficiencies and reduce costs through increased asset or service uptime; lower warranty costs; or device, device resource, and asset optimization.

## There are many roads you can take in pursuit of these goals, but to reap the greatest return it’s important to look at IoT not just as a journey but a maturity progression as your organization’s needs and adoption of IoT changes. Bsquare’s experience has shown that moving along the maturity path to full IoT can be broken down into five stages: device connectivity, real-time monitoring, data analytics, automation, and enhancing on-board intelligence. Each stage beyond the first provides increasing levels of benefits. Progression through all five stages culminates in a holistic, intelligent, automated IoT system that delivers the broadest range of positive outcomes for multiple business goals.

## Whether your organization needs to complete all five stages depends on the desired business outcomes. Having a clear strategy, plan for execution, and an understanding of what constitutes success will help determine how far your company needs to go.

## Many companies fail to realize that maximum ROI will only be achieved after completing most, if not all, of the stages, and are discouraged when they don’t reach the business goals they envisioned, Bsquare has found. They also have observed that many heavy industrial equipment organizations have the opportunity to extract the greatest value from their equipment after progressing through the entire IoT maturity model. The amount of time required to step through the maturity model and time spent at each stage will vary depending on many factors: the amount of equipment and devices already outfitted with sensors or other intelligence, communications and data storage infrastructure, the types of monitoring and analysis systems already in place, the ease of integration with existing enterprise systems, and organizational readiness, to name a few.

## So, what does each stage mean and how can you get started? Let’s examine each stage, how it builds on its predecessor, and the resulting benefits and potential pitfalls along the way.

## THE LAYERS OF THE IOT ARCHITECTURE:

Conventional IoT architecture is considered to have three layers which are the perception layer, the network layer and application layer. However, another layer was added to the list later and that is the support layer, which lies between the application layer and the network layer. There is another model for IoT layers, which is the model most people refer to when trying to understand the IoT architecture. This model includes seven IoT layers;

**Layer 1: The Things Layer**

This layer of IoT comprises of devices, sensors and controllers. Connected devices are what enable the IoT environment. These devices include mobile devices such as smart phones or tablets, micro controller units and single-board computers. The connected devices are the real endpoint for IoT.

**Layer 2: Connectivity/Edge Computing Layer**

Layer 2 is the connectivity/ edge computing layer, which defines the various communication protocols and networks used for connectivity and edge computing. It is a distributed architecture where IoT data is processed at the edge of the network.

**Layer 3: Global Infrastructure Layer**

Layer 3 is the global infrastructure layer, which is typically implemented in cloud infrastructure. Most of the IoT solutions integrate with cloud services. A comprehensive set of integrated services, IoT cloud can provide businesses with useful insights and perspective on customers.

**Layer 4: Data Ingestion Layer**

Layer 4 is the data ingestion layer, which includes bigdata, cleansing, streaming and storage of data.

**Layer 5: Data Analysis Layer**

Layer 5 is the data analysis layer and relates to data reporting, mining, machine learning etc.

**Layer 6: The Application Layer**

Layer 6 is the application layer, which comprises of the custom applications that is actually making use of the things data.

**Layer 7: People and Process Layer**

Layer 7 is the people and process layer. This includes people, businesses, collaboration and decision making based on the information derived from IoT computing.

## FIVE STAGES OF IOT:

## 

Figure 3: IoT Stages.

## Stage 1: Device Connectivity and Simple Data Forwarding

## Every IoT story starts with a smart, connected device. Often companies begin their IoT progression by outfitting their equipment and devices with sensors that collect all manner of information and data. Often referred to as telematics, machine-to-machine (M2M) or IoT products and solutions, a gateway or similar communications device transmits sensor data from the device to a location where the information can be stored for future use.

## Most of these systems operate with very limited communications bandwidth, particularly in industries with remote or highly mobile environments. This means only a small subset of the data produced is actually collected. Even in situations where bandwidth is not an issue, the cost of transmitting and storing such vast amounts of data, which may or may not be important, can be prohibitive.

## All of this is a crucial first step in enabling devices to form the foundation of an IoT solution. Without the device data, none of the subsequent stages are possible. However, simply adding sensors and intelligence to equipment and collecting data requires investment but doesn’t produce business benefits. A system that performs some type of monitoring or analysis is required to begin to extract value from the device data.

## Stage 2: Real-time Monitoring

## As data is collected, it must be monitored in real-time and visualized to begin enabling the use cases that lead to desired business outcomes. Common use cases for industrial equipment are: condition-based maintenance to improve operational efficiency and reduce service costs, device utilization information to help guide future product design and improve regulatory compliance, and IoT device management for enhanced device integrity and lower operating costs. Monitoring and alerting can help companies gain awareness of equipment status and start to adopt and refine business processes that improve outcomes.

## 

## In many cases, the monitoring system is a dashboard that provides basic information, data visualization, and simple alerting. For instance, the dashboard displays an alert when a temperature gauge exceeds a specified threshold so the person monitoring the dashboard can take steps to diagnose and service the issue.

## At this stage, data is often called “actionable data,” but the action must be taken by humans, who simply can’t stay on top of the vast amount of monitoring and alerts that take place on the flood of data generated as real-time monitoring is rolled out to more and more equipment. One of the biggest hurdles along the way is that the dashboard-human operators approach cannot scale effectively. Software needs to perform monitoring and analysis – not humans.

## With real-time monitoring, companies can see some level of condition-based maintenance; however, using this approach often results in an unacceptably high rate of false positives or false negatives. These are error codes or conditions that indicate a problem but when the equipment is serviced no problem is found (false positives), or error conditions that are not correctly flagged (false negatives). Basic dashboard solutions simply cannot detect complex conditions and events as they attempt to apply simple logic to complex equipment.

## To extract real value from machine data, complex event processing capabilities in the form of data analytics are required. These go beyond simple event processing (if a then b) to apply rules and analyze multiple data sources to gain valuable insight and truly actionable data.

## 

## Stage 3: Data Analytics

## Data analytics can deliver insight, predictions and optimization, and reduce unnecessary false positives by an average of 25%, but culling insights from rivers of data is an involved process. In order to support IoT, many different types and formats of data are likely to be needed. To effectively use all these different data sources, an extra ingest step is required to align the data; for instance, transforming all temperature data from Fahrenheit to Celsius or enriching data to give context. There are several elements required for a successful data analytics system:

## Data discovery – First and foremost, it’s important to have the right kinds of data to support the desired business use cases and outcomes. The appropriate types of data can be identified and collected by determining whether additional sensor data is necessary to provide essential device information, and what business systems may require integration. Only then should complex event processing capabilities for in-depth analysis and actionable insight be applied.

## Machine learning – Algorithms can then be applied to the large pool of data to accomplish much of the heavy lifting required to identify correlations and patterns that may have been done manually in the past.

## Cluster analysis – Once machine learning is applied to the data, groups of equipment that behave similarly can be identified to help understand how the environment is working.

## 

## Digital model – This is a representation of how the equipment behaves. Beyond the basic anatomy of a particular piece of equipment with a given number of sensors, the behavior of how those sensors interact with each other leads to insight.

## Taken together, these elements provide truly valuable insight that allows appropriate actions to be taken – in many cases automatically. In the example of condition-based maintenance, the ability of complex event processing logic to examine historical data and other contextual or system information (such as equipment specifications) in concert with machine sensor data provides much more accurate profiles and event prediction than human operators ever could.

## The insights gleaned at this stage further a company’s ability to make progress towards several use cases: predictive failure for increased asset uptime and elimination of false negative and positive reports, condition-based maintenance, device optimization for improved asset performance, and device utilization.

## While insights alone may provide more sophisticated dashboards and a better picture of what’s occurring in the organization’s environment, the system is still reactive – not proactive. The human scalability challenge outlined in Stage 2 steadfastly limits IoT success as companies often underestimate the fire hose effect of events, conditions, and data streams generated by a growing amount of connected complex equipment. Automation solves this problem.

## Stage 4: Automation

## Once rules and automation are applied, complex actions can be orchestrated across multiple areas such as integration into inventory, support, or service ticketing systems. Additionally, there may be data collection rules that can be changed. For instance, if the device is healthy, less data is collected and transmitted, but when the rules monitoring the system’s health start to determine that conditions are approaching a potential anomalous condition, data fidelity can be increased by collecting more data. Rules can be enabled for improved safety, as well. If the system recognizes a specific set of conditions, it can shut a machine down without waiting for a control room operator to respond.

## Rules must be dynamic in nature. It’s a widely held belief that analytics is a “once and done” or infrequent exercise. The reality, particularly in the world of complex heavy equipment, is that analysis is only as good as the moment it is put into place. At the outset, the system collects data, performs analysis, gains insight, and turns it into a set of rules that gets applied to the real-time environment. Everything goes along smoothly.

## But fast-forward to 30, 60, or 90 days later and rule effectiveness begins to drift. In many cases equipment may be operating in challenging environments and can be affected by external factors. There may be hardware or software revisions that change equipment behavior enough that the rules aren’t as applicable as they were during the original analysis. Automation can help manage rule drift.

## Organizations completing this stage can achieve the full benefit of several use cases, including condition-based maintenance and device utilization – and for some businesses, this may be sufficient. Significant benefit also can be realized for predictive failure, data-driven diagnostics, and device optimization. Yet companies can gain additional value across these use cases, as well as IoT device management, by making their equipment and devices even smarter.

## Stage 5: Enhancing On-Board Intelligence

## Distributed intelligence, edge computing, edge intelligence, edge analytics. These terms all describe the same fundamental concept – processing data on or very close to the connected equipment in addition to functions performed in gateways or the cloud. Rather than moving the data to the logic, on-board intelligence brings the logic to the data. In the case of complex industrial machinery, much of today’s connected equipment already has computing capabilities that can be tapped to perform data analytics and automation directly on the equipment, in real time.

## 

## By adding analytics and automation to equipment, all data can be analyzed for greater accuracy. In a typical scenario where limited data is transmitted for analysis, decisions are being made on a tiny percentage of the available data. By having 100% of the data available, results are faster, more accurate, and eliminate the need, and associated costs, to transmit and store unnecessary data.

## On-board intelligence brings the IoT maturity model full circle, allowing industrial organizations to gain maximum ROI and business benefit from predictive failure, datadriven diagnostics, and device optimization. Further, true IoT device management becomes a reality as on-board intelligence monitors for conditions in order to identify events and then automates actions directly on the equipment for better predictive accuracy and more rapid response time. It also enables valuable functionality when equipment loses connectivity. For example, machinery can be rapidly and automatically shut down when there is an unsafe condition. Other important benefits are the enablement of over-the-air (OTA) software updates for improved device integrity and OTA operational configuration to dynamically alter asset behavior as requirements and environments change.

## FULL IOT MATURITY AND ADOPTION – STILL EARLY:

## Where do most organizations stand on their way to IoT maturity? Bsquare has found that the majority of companies have implemented the first stage, with a slightly smaller majority moving on to the second stage and actually doing something with the data. Very few have progressed further than real-time monitoring of data and are missing out on the benefits of data analytics, automation, and ultimately on-board intelligence.

## It’s only when the insights gained from analytics are applied to the company’s business models and processes that the true value of IoT can be achieved. Analysis achieved with predictive failure and data-driven diagnostics can reduce unplanned downtime. And, when service is required, it can ensure the right technician with the right skills, tools, and information can solve the problem correctly the first time. Insight gained as the system gathers and processes more intelligence can be used to help improve business processes and product development.

**Steps, Concepts, Framework and Example**

**1. Typical IoT pentest**

A typical IoT penetration test (Attacker Simulated Exploitation) would involve the following components:

**Attack Surface Mapping**

Our entire team spends between 1-2 days to perform an in-depth Attack Surface Map of your solution.

In this stage, we prepare a highly detailed architecture diagram highlighting all the possible entry points for a malicious dedicated attacker.

**Firmware reverse engineering and binary exploitation**

It’s important to ascertain if hardware and chip makers have fully implemented security best practices within the firmware and operating system.

To do this, the team will test the built-in security of the device firmware and its update distribution process, such as cryptographically signing firmware updates and using authentication capabilities in hardware devices to verify signatures. At the OS level, the team should examine software boot sequences, code execution, application core dumps and data confidentiality protections. As part of this analysis, security engineers will also need to examine memory to ensure sensitive data is properly erased by the application.

Reverse engineering firmware binaries

Encryption analysis and Obfuscation techniques in use

3rd party libraries and SDKs

Binary reverse engineering and exploitation

Debugging binaries to gain sensitive info

Encrypted:

Possible XOR? Identify by Hexdump and look for repetitive strings

Use XOR decryption script to get the decrypted firmware

Non-XOR? Manual Analysis using a disassembler

Dump from the flash memory while the firmware is in decrypted state

Non-Encrypted:

Extract file system using Binwalk (binwalk -e firmware.bin)

Analyze configuration files and hardcoded sensitive values and certificates (manually + automated tool - Firmwalker)

Disassemble individual binaries using Hopper/Binary Ninja/IDA Pro

Identify vulnerabilities such as command injection and backdoors in the disassemblers

Analyze strings present in the binary

Analyze the function list

Look for Xrefs to system()

Emulate individual binaries using qemu (sudo chroot . ./qemu-arch -L optional-lib-path -g gdb-port binary-to-emulate)

Attach GDB-Multiarch (or IDA Pro) to the emulated binary

Set breakpoints at functions like strcmp and analyze the context (registers, stack and disassembly) at that point.

Identify overflow based vulnerabilities (pass a large string and see if the program crashes)

Exploit the overflow by forming a ROP chain (use ROPGadget to find useful gadgets)

Modify firmware image using FMK (Firmware-Mod-Kit):

./extract-firmware.sh to extract the firmware

Add backdoor or bindshell to the extracted firmware and ensure that it starts at bootup. Build the firmware using build-firmware.sh with the flags -nopad -min

Upload the modified version of the firmware to the target device

If the DUT accepts the new firmware, it is missing firmware integrity verification

If it does not accept, look for code sample in the firmware (or other components) where it's checking for signature verification.

Obtaining the firmware

Vendor website and support forums

Sniffing the package during OTA update

Reversing mobile application

Dumping from the device using H/W Exploitation tactics

**Hardware based exploitation**

The security team should begin its analysis by evaluating physical and hardware controls to see if these are sufficient to prevent an attacker from tampering with the platform’s components and their normal execution flow.

Each underlying component must be examined for reverse engineering and subversion capabilities. For instance, remnant JTAG, SWD and USB interfaces that provide a “way in” are often useful for interacting with the underlying hardware. Techniques to circumvent hardware modules that enforce trust and protect sensitive data are of particular interest.

Assessing hardware communication protocols such as UART, SPI, I2C etc.

JTAG debugging and exploitation

Logic sniffing and bus tampering

Dumping sensitive information and firmware

Proprietary communication protocol reversing

Tampering protection mechanisms

Glitching and Side-Channel attacks

Security features included in the hardware

Recon

FCC-ID database : Look up the FCC ID of the DUT on fccid.io

Identify the chipsets being used

What frequency does the device operates on

Internal/External pictures

Public searching of previous accessible resources of the device

Look for research done on previous versions of the device if that's available to get an idea of the process

Teardown

Identify the kind of enclosure holding the packaging together and use appropriate tools to open it

Look for screws beneath the rubber pads

Pry open the device

Apply heat (make sure to not damage the device)

Cut/Remove the enclosure (as the last option)

XRays/Focused Ion Beam Workstation

Exploitation

What does the PCB Reveal

Identify different chips and part numbers

Look for datasheet to figure out more information about a given chip

What protocol does it work on

Data storage size

Support for communication interfaces

Can you see any possible interfaces?

UART - usually in pair of 3 or 4 with one of the pins being GND

Identify GND using continuity test

Identify Tx and Rx with voltage fluctuations (Tx will have more fluctuations during bootup due to device bootup messages)

Connect to Attify Badge once pinouts are identified Run "sudo python baudrate.py" Hit Ctrl+C once you see readable characters Save conf Hit enter to get shell

SPI - presence of a flash chip

Once pinout is identified from the datasheet, connect the pins to Attify Badge either using Mini Test Probes or pulling out the chip and putting on an adapter

You can use tools like spiflash.py or flashrom to dump the entire flash image / firmware (sudo python spiflash.py -r firmware.bin -s size-to-dump)

JTAG - set of 6,12,13,20 pin headers (or could also be scattered across the board)

Use JTAGulator to identify the pinouts

Connect to all possible JTAG pins Run "sudo screen /dev/ttyUSBx 115200" Set target voltage (using "V") Hit "B" to start the scan

Use Attify Badge once pinouts are known

TDI => TDI TDO => TDO TCK => TCK TMS => TMS

Run openocd with the configuration file obtained from openocd github repo (or on your local system) sudo openocd -f badge.cfg -f target-device-chip.cfg

Run GDB and connect to :3333 to gain debugging access to the target device

hbreak strcmp or break interesting-function You can even use memory manipulation commands to change the contents of memory at any given address (this could be used to change boot args)

Unfamiliar protocol

Research the chip to identify the possible communication protocols

Use a logic sniffer to see what data is being communicated (can use different analyzers and see which one shows something meaningful)

External interfaces like USB (can plug in a keyboard and brute force with special key combinations)

Backdooring

Upload malicious firmware to the device using Flash re-write or using JTAG

Add your own new component for persistent hardware backdooring on the device

**Web**

Vulnerabilities in the web dashboard - XSS, Injection based attacks, IDOR, Authorization and Authentication bugs and more

IDOR (Insecure Direct Object Reference)

Check for permission level bugs (admin, user, superadmin)

Typical bugs - XSS, SQLi, XXE, XSRF etc.

**Mobile**

Mobile application security issues identification and exploitation for Android and iOS - Platform related security issues, App reversing, Binary instrumentation techniques to gain sensitive information etc.

Find out information such as how the firmware is downloaded from the remote endpoint and flashed to the device

Protocol and encryption (key could be found here or by dumping the flash contents in h/w) used for communication

Reverse the API communication

What kind of data is being stored on the mobile device

Use Frida to perform runtime manipulation and analysis

Automated mobile application security tools

Replay based attacks

Insufficient authentication and authorization checks when communication with the device

Here’s what the team will look for with each of these:

• Storage level data — Proper use of native APIs for features like key stores; avoiding insecure storage of dangerous client artifacts (ex: user credentials, personal information or other sensitive application data); and properly erasing sensitive data.

• Transport level data — Vulnerabilities related to information disclosure, tampering and spoofing in the traffic between the mobile app and any remote systems.

• Authentication/authorization — Implemented authentication protocols, certificate validation, password policy enforcement and account lockout mechanisms. It should also examine data access controls, segregation (and principle of least privilege), confused deputy attacks and the accessibility of hidden functionalities.

• Session management — Resiliency of persistent sockets when faced with a severed connection. The entropy, length, timeout and rotation of session identifiers to see if they are susceptible to preset identifiers, brute force, session fixation, etc.

• Data validation — Any open ports, interfaces, IPC channels or other input modes that can be leveraged by an attacker or malicious application. Exposed interfaces should be fuzz tested to see how they handle erroneous input via filtering, sanitation and validation. Key vulnerabilities in scope: XSS, SQLi, command injection, mishandled exceptions and memory corruption attacks (RCE or DoS).

**API**

Web application testing begins with the network and operating system to make sure the underlying platforms are securely configured.

Next, the team will move on to the web application layer — this requires significant attention and will comprise the majority of the engagement. For this part of the pen-test, it’s important to play multiple roles: first, as an attacker without valid credentials to the web application, and, secondly, as users who have valid credentials. In the latter instance, the testing should be conducted across all user roles in order to fully examine the app’s complicated authorization controls. This should test a user’s ability to access another user’s information within the same role, as well as a user’s ability to access another user’s information at a higher role (vertical privilege escalation).

API based security issues

RE the Thick Client to find vulnerabilities such as Command Injection and Overflows

Communication to the remote endpoint

Reversing the APIs

**Wireless**

A wireless configuration review should be conducted to validate the security and configuration of wireless communication protocols used for local device communication, such as ZigBee, 6LoWPAN and Bluetooth LE.

The security review begins by identifying device roles, cryptographic primitives, encryption keys, authentication and other algorithms related to security. After taking inventory of various security components, run an analysis of common attacks like man-in-the-middle, replay, unauthorized network commissioning and then (if applicable) fuzz test the protocol stack.

**Network, Cloud**

All back-end platforms used to exchange data with IoT networks, applications, devices and sensors should be tested to see if an attacker is able to gain unauthorized access or retrieve sensitive information. These include any external cloud services (Amazon EC2, Google CE, Azure VM) or APIs.

Use network diagrams, documentation and cloud management console access to evaluate the security of the platform’s cloud deployment. Assess the security architecture and deployment by examining the following major components: key security architecture design assumptions, current network topology, inventory of existing security technologies, security policies, guidelines, and procedures, instance group policies, network access controls, and network segmentation, remote access and virtual private networks, authentication controls including two-factor authentication and single sign-on, datastore encryption and key management, containerization technologies such as Docker and Rocket, and logging and monitoring deployments.

Cloud-based and vulnerabilities in the backend systems

Check for running services on the device

Outdated services

Password brute-forcing and cracking

Unfamiliar port open?

Sniff the network communication

**Radio security analysis**

Assessment of radio communication protocols

Sniffing the radio packets being transmitted and received

Modifying and replaying the packets for device takeover attacks

Jamming based attacks

Accessing the encryption key through various techniques

Radio communication reversing for proprietary protocols

Attacking protocol specific vulnerabilities

Exploiting communication protocols such as BLE, ZigBee, 6LoWPAN, zWave, LoRa etc. through insecurities and vulnerable implementations

Raw Radio Communication protocol

Identify the frequency using HackRF / RTL-SDR (Notice the spikes in GQRX when device sends bursts of data)

Capture the data being transmitted at that frequency and process it in GNURadio to obtain meaningful information

Use hackrf\_transfer to replay the captured data

BLE

Identify the BLE devices around you and their addresses (using a BLE dongle)

Identify characteristics and services of the target device

Capture the BLE traffic while interacting with the target device using Ubertooth One (Or with 2 BLE dongles with projects like BTLEJuice)

Clear-text traffic

Relay based attacks

Write data to the target devices's BLE Characteristics using Gatttool

Capture the initial pairing packets and use crackle to decrypt traffic (if encrypted)

ZigBee

Find the ZigBee channel on which the DUT is operating on

Capture communication using zb\_dump and analyze in wireshark

Perform replay based attacks using zb\_replay

Identify keys in the captured communication

**PII data security analysis (optional)**

Ensuring that customers data are kept with highest security standards

Ensuring that no PII information is being leaked through any channels - web, mobile, hardware or radio

Additional assessment of data-at-rest and data-at-transit

Providing you with a PII report

**Report preparation**

Preparing an in-depth report including both technical details, non-technical summary and an executive summary

Providing you with all the scripts, Proof of Concepts, exploitation techniques, demos or code snippets that were created during the engagement

Categorizing the vulnerabilities based on criticality for your given product and user use-case scenario

**Re-assessment**

Once the bugs have been patched, we perform an in-depth reassessment to ensure that the bugs have been fixed security

Also checking for the fact that the patches did not introduce any additional vulnerabilities

**IoT penetration testing tools**

The arsenal of IoT penetration testing tools comprises familiar names widely used in traditional security testing: the Metasploit framework for penetration testing, scanners (Nmap, Burp Suite, ZMap, Nessus), script languages (e.g., Python, Perl). In addition, it’s a good practice to create custom IoT penetration testing tools that will suit a particular environment (e.g., to support real-time operating systems).

**2. Open source for Iot Pentest**

**IoTGoat**

IoTGoat is a deliberately insecure firmware created to educate software developers and security professionals with testing commonly found vulnerabilities in IoT devices. <https://owasp.org/www-project-internet-of-things/>

The IoTGoat Project is a deliberately insecure firmware based on OpenWrt and maintained by OWASP as a platform to educate software developers and security professionals with testing commonly found vulnerabilities in IoT devices. The vulnerability challenges are based on the OWASP IoT Top 10 noted below, as well as "easter eggs" from project contributors. For a list of vulnerability challenges, see the [IoTGoat challenges](https://github.com/OWASP/IoTGoat/wiki/IoTGoat-challenges) wiki page.

Getting started

Several methods exist to get started with hacking IoTGoat.

For those looking to extract the filesystem, analyze configurations and binaries statically, download the latest precompiled firmware release from <https://github.com/OWASP/IoTGoat/releases>. Refer to [OWASP's Firmware Security Testing Methodology](https://scriptingxss.gitbook.io/firmware-security-testing-methodology/) to help with identifying vulnerabilities.

For dynamic web testing and binary runtime analysis, the quickest way to get started is downloading the latest "IoTGoat-x86.vmdk" (VMware) and create a custom virtual machine using the IoTGoat disk image. Select the following operating system details Type: Linux Version: Linux 2.6 / 3.x / 4.x (32-bit) and Enable PAE/NX in virtual machine settings. Both the .vmdk and .vdi have been tested in the latest VirtualBox release (April 2020) for Windows 10, Ubuntu 18.04 LTS, and MacOS Mojave. Refer to [OWASP's Web Security Testing Guide](https://github.com/OWASP/wstg/tree/master/document) and [ASVS](https://github.com/OWASP/ASVS) projects for additional guidance on identifying web application vulnerabilities

Emulate firmware with opensource tools (e.g. Firmadyne, ARM-X Framework, and FAT) that leverage QEMU to virtualize IoTGoat locally.

Use the IoTGoat-raspberry-pi2-sysupgrade.img firmware to flash on a Raspberry Pi 2 (BRCM2708 & BRCM2709).

Refer to the [Getting started](https://github.com/OWASP/IoTGoat/wiki/Getting-started) page for additional details and screencaptures.

Building from source

OpenWrt can build many different CPU platforms and boards. Building from source gives users the flexibility to flash IoTGoat on supported OpenWrt hardware. Ensure 10-15GB disk space is available with at least 4GB of RAM and a supported Linux distribution such as Ubuntu 18.04. Use the following steps to get started with building custom firmware.

Do everything as a normal user, don't use root user or sudo when building!

$ git clone https://github.com/OWASP/IoTGoat.git

$ cd IoTGoat/OpenWrt/openwrt-18.06.2/

$ ./scripts/feeds update -a

$ ./scripts/feeds install -a

$ make menuconfig # select your preferred configuration for the toolchain, target system & firmware packages.

$ make # Build your firmware with make. This will download all sources, build the cross-compile

The first build will take some time to complete and will vary based on the provided internet connection for downloading the toolchain. Once a successful build is complete, the compiled firmware will be placed in the following directory IoTGoat/OpenWrt/openwrt-18.06.2/bin/targets/ depending on the target selected in menuconfig. For example, IoTGoat Raspberry Pi 2 firmware will be located in the following directory IoTGoat/OpenWrt/openwrt-18.06.2/bin/targets/brcm2708/bcm2709. IoTGoat build configuration files are made availble for x86 (.config-x86) and Raspberry Pi 2 (.config-rpi) platforms.

**AttifyOS**

AttifyOS is a distro intended to help you perform security assessment and penetration testing of Internet of Things (IoT) devices. It saves you a lot of time by providing a pre-configured environment with all the necessary tools loaded. The new version is based on Ubuntu 18.04 64-Bit - that also means that you'll receive updates for this version till April 2023.

Disclaimer - Make sure you only tests the target for which you have proper authorization of. You hold all responsibilities of what you decide to do with it.

**PI-HOLE**

Pi-hole is a DNS sinkhole that protects your devices from unwanted content, without installing any client-side software.

From the Pi-hole overview:

Easy-to-install: versatile installer, takes less than ten minutes

Resolute: content is blocked in non-browser locations, such as ad-laden mobile apps and smart TVs

Responsive: speeds up browsing by caching DNS queries

Lightweight: runs smoothly with minimal hardware and software requirements

Robust: command line interface quality assured for interoperability

Insightful: responsive Web Interface dashboard to view and control Pi-hole

Versatile: optionally functions as DHCP server, ensuring all your devices are protected automatically

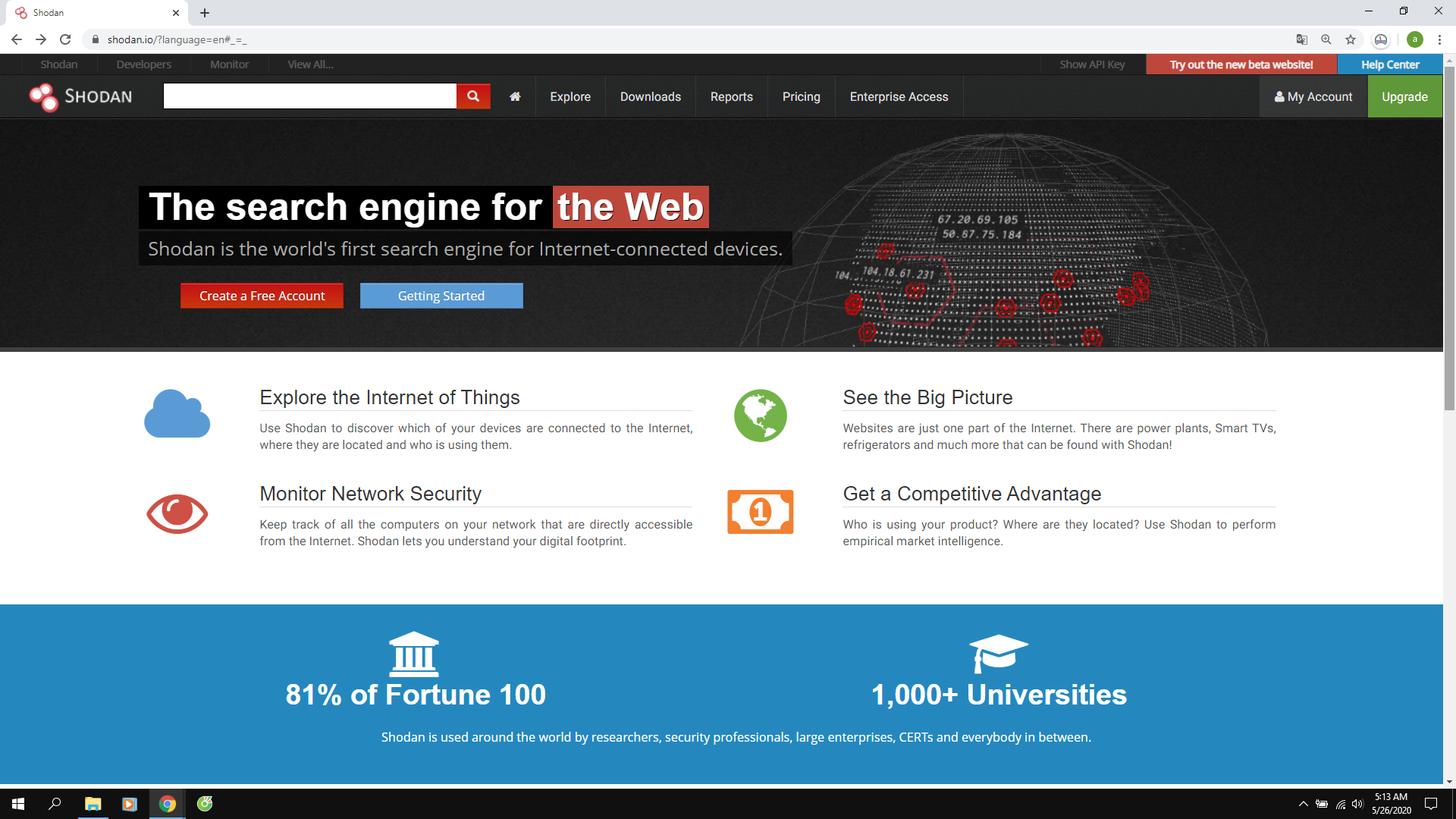
Scalable: capable of handling hundreds of millions of queries when it is installed on server-grade hardware

Modern: blocks ads over both IPv4 and IPv6

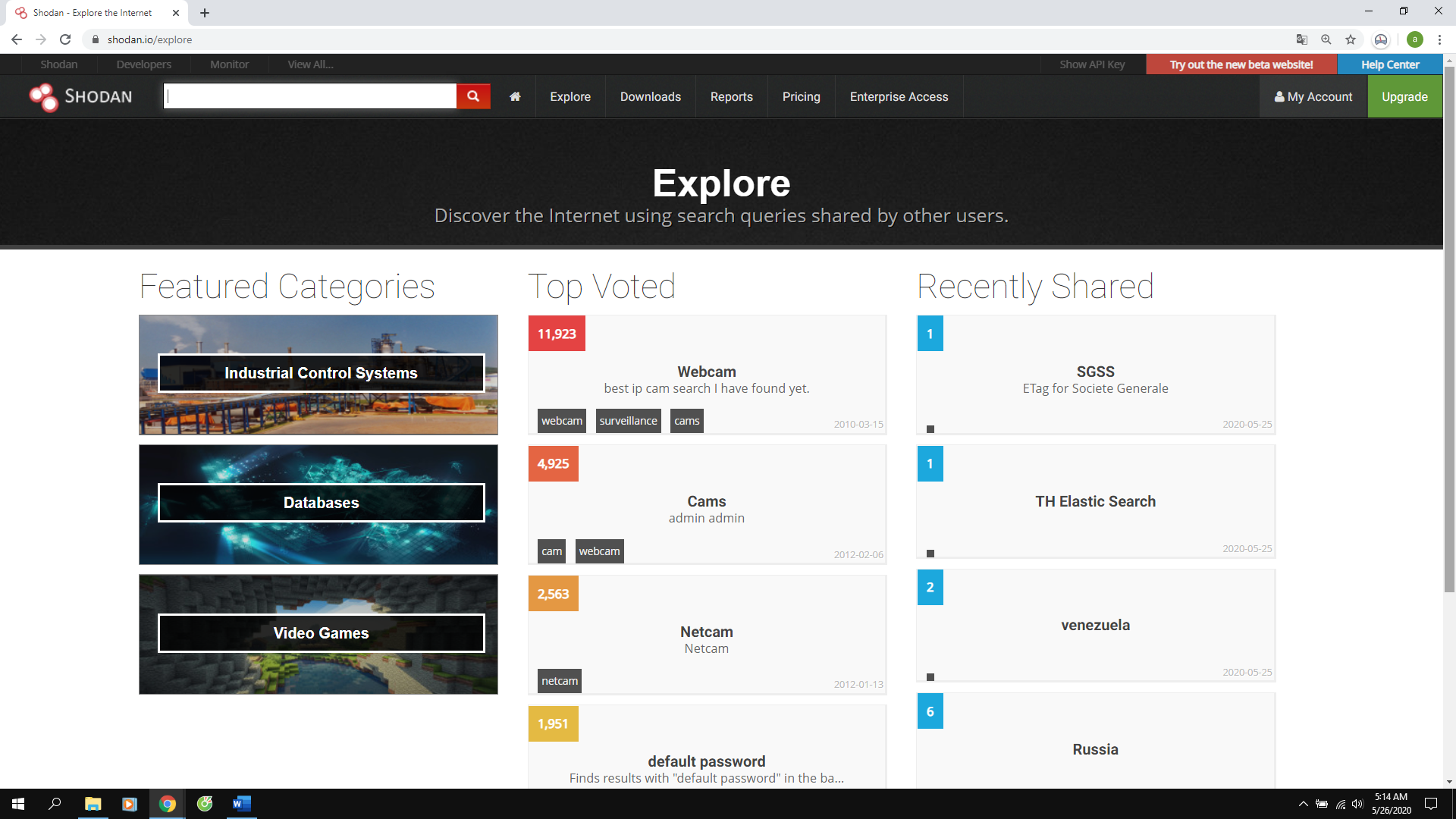
Free: open source software

**3. Practical IoT security**

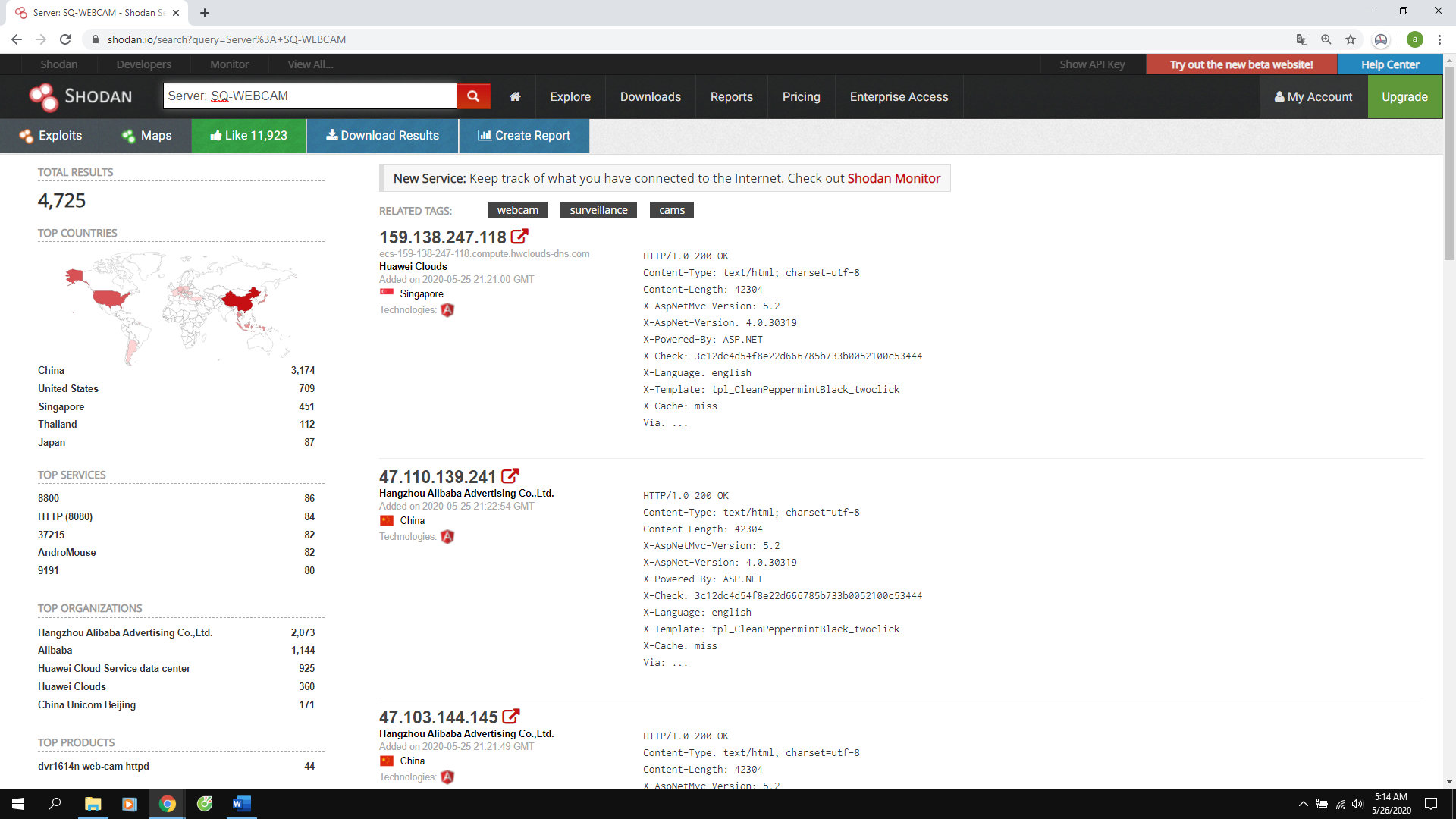
**INFORMATION GATHERING WITH SHODAN**



First Create a Free Account on shodan. After that, login to the newly created account After logging in to shodan, click on getting started.



shodan will then turn to the explore page and you can see many different type of database on shodan like webcam, default password, Elastic,etc.



Here, You can see a lot of cameras being public on shodan.



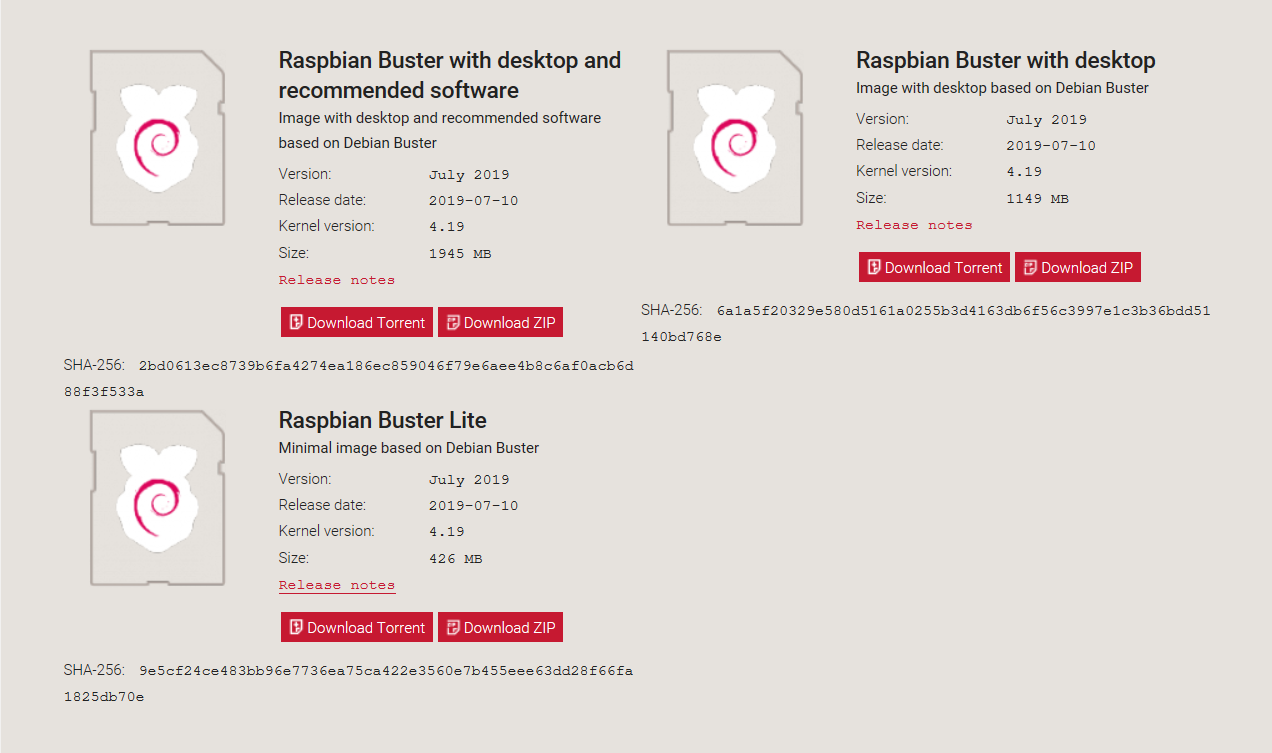
Click on any IP, you will see more details about the camera.

**Raspberry Pi Pi-Hole Installation**

**Downloading Raspbian Buster**

First of all, visit the [Raspbian download page](https://www.raspberrypi.org/downloads/raspbian) to download an image file (.img) of the latest version of [Raspbian Buster](https://maker.pro/raspberry-pi/tutorial/raspbian-buster-software-update-for-raspberry-pi).

The Desktop + Recommended Software version comes with everything you need to start working on most projects, so I recommend it for beginners. If you don’t need more than a simple desktop, the second option is best. If you plan on running your Pi in a headless setup, you may not even need a desktop and should download the third option.

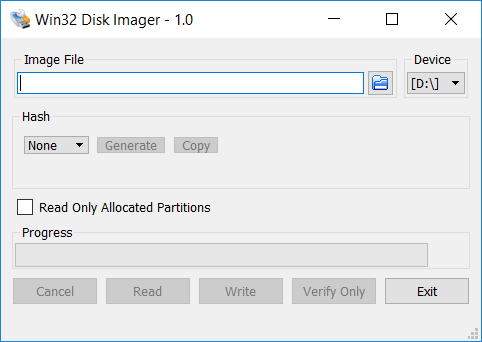


Once you have downloaded an OS, you can move on to the next step.

**Flashing the SD Card**

For this step, you will need to download an OS Etching software. Though the Raspberry Pi Foundation recommends Etcher, I recommend using Win32DiskImager for Windows users. The newest version of Etcher bricked a couple of my SD cards, and the support team wasn’t very helpful.

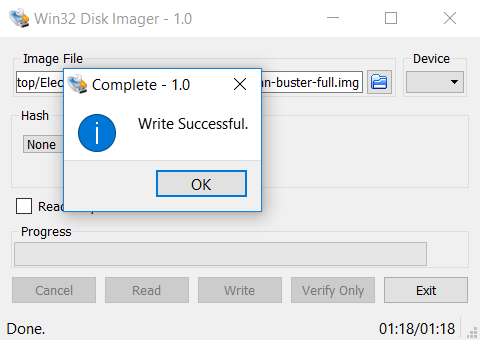
While the OS etching software downloads, go ahead and unzip the version of Raspbian you downloaded. You will find a .img file inside. After extracting that file, copy its location as you will need it for the next step.



*Win32DiskImager with SD card as the selected device*

Open up your preferred disk imager, then select your SD card. Verify that it is your microSD card instead of another drive by double checking the drive name in your File Explorer.

Once you are confident in your selection, paste the location of your .img file in the program. Win32DiskImager lets you browse for the file if you did not copy the address before. Click “Write”, then wait for the process to finish. It should take less than two minutes, so feel free to get up for a quick stretch in the meantime. After this is done, you should get a “Write Successful” message like the one below.



Now you can safely eject your microSD card and move on to the final step.

**Booting up the Pi 4**

Insert your flashed SD card into the slot in the bottom of your Pi 4. Plug in your Pi with an appropriate USB-C cable, and hook up your monitor and keyboard. You should see the colorful startup splash screen, then a couple of Raspberry Pi logos at the top right of your screen while your Pi boots up for the first time.

Raspbian will boot significantly faster than NOOBS because NOOBS is heftier and delays startup on purpose to give you a bigger window to select and install alternative operating systems. Once it boots up, you should see the beautiful default wallpaper and the Raspberry Pi setup wizard. Simply follow the on-screen instructions to set up your language, location, and network settings.

**Install Pi-hole ad blocking on Raspberry Pi**

Pi-Hole is a software running on Linux operating system that acts as a DNS server filtering ads out of websites when you visit that website. It is similar to the extensions you use on your browser to block ads like "ADblock".

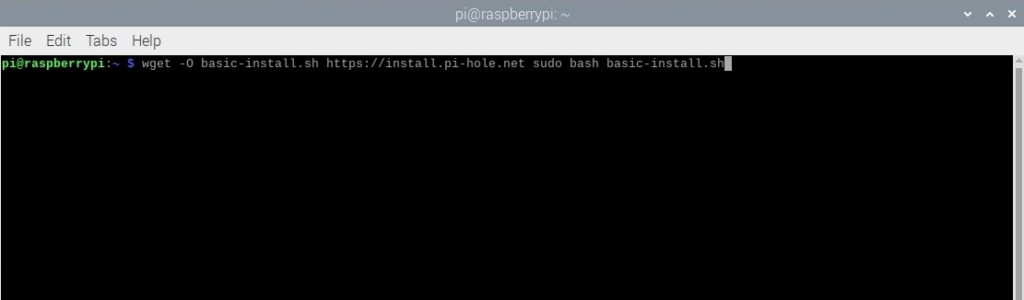
The difference is that if you install "adblock" on any device, only that machine will block ads. If you install a Pi-hole on a Raspberry Pi, your whole family will always block ads without worrying about which device.

**Step 1** : Open the terminal window of the operating system and type in the command line below, everyone

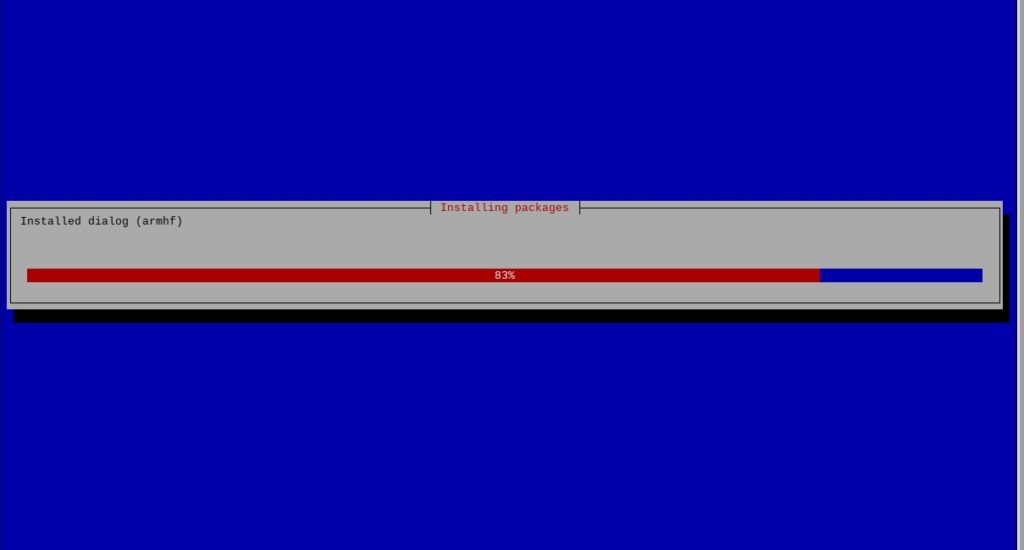
*wget -O basic-install.sh https://install.pi-hole.net*

*sudo bash basic-install.sh*

If you are lazy to type or are afraid to type wrong, just copy-paste into the terminal window. Then press Enter.



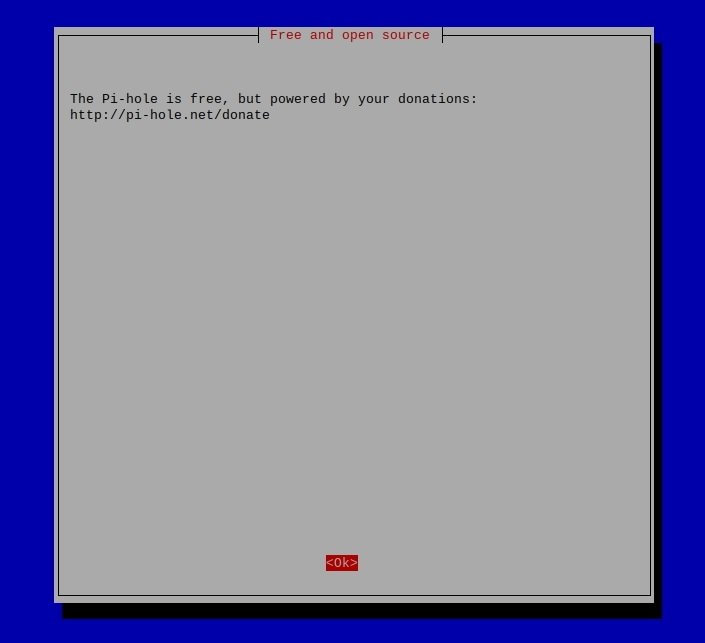
**Step 2** : After step 1, you just leave it, it will automatically install.



**Step 3** : Press enter.



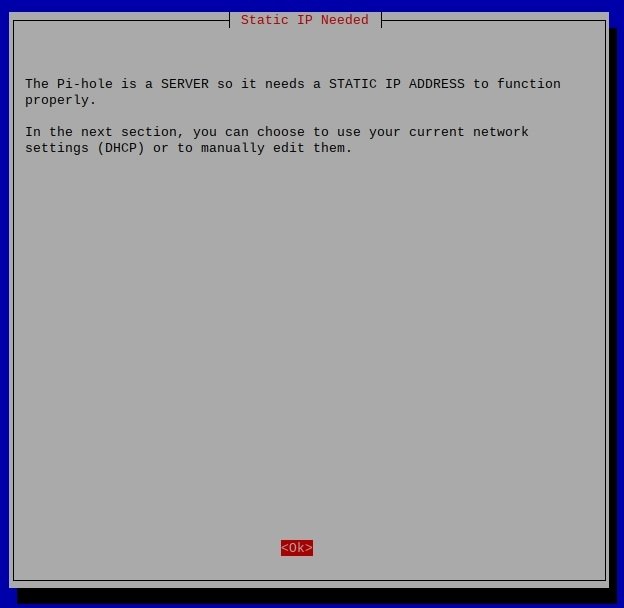
**Step 4** : so this step press Enter.



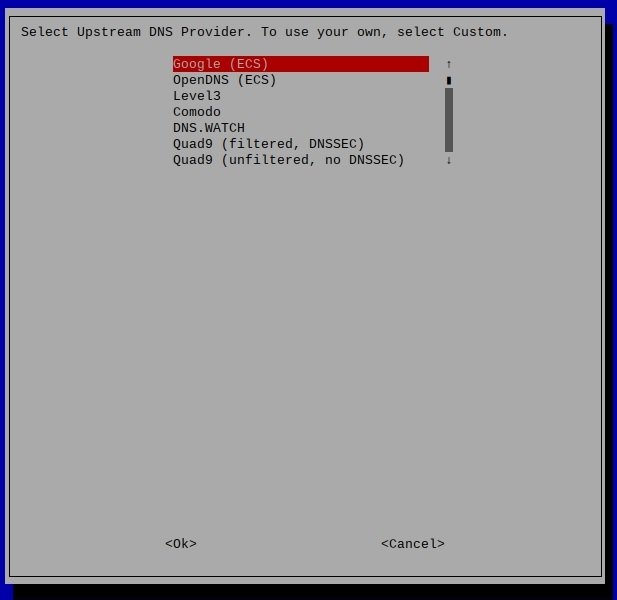
**Step 5** : This step Pi-Hole appears that the server blocking this ad needs a static IP to operate. Oke simple, like static IP has static IP always.

To set the static IP for this Raspberry Pi, go to the total router, then go to Finland => DHCP it will show the devices that are receiving ip from the total router. You just need to select "Make static ip" for the Raspberry Pi is ( depending on the type of modem / router that will have a static ip setting for different devices, you should refer to the attached document or google or Ask here I will help). If you do not set a static ip for the Raspberry Pi, then the latter will be a bit annoying.

Press Enter to continue through step 6.



**Step 6** : This step, you do not need to do anything, choose a great Google because later you like to change it. Enter through the next step.



**Step 7** : Pi-Hole uses the list of "block ads" from the 3rd party. That means you can feel free to add anyone's list to Pi-hole as long as you feel like it. This you just enter through the next step for it quickly, because later add something less in the admin interface is gone.



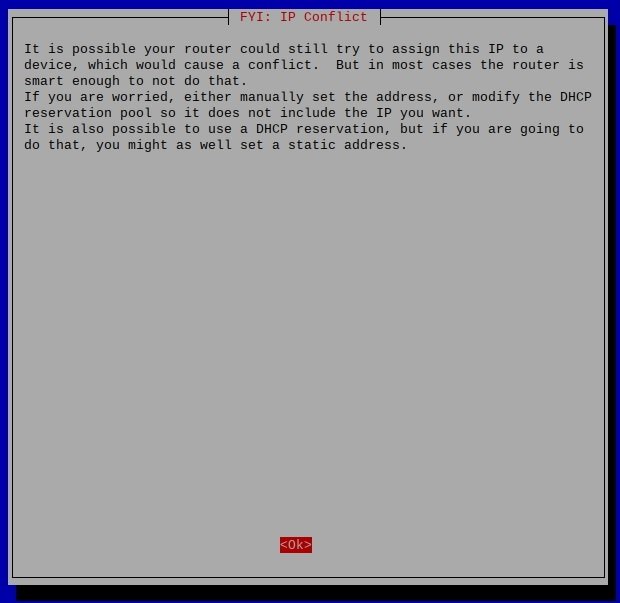
**Step 8** : Which IP address do you want Pi-hole to block ads from? Only this step you care about doing. Enter through the next step because the system has already selected both.



**Step 9** : The system displays Raspberry's current IP and default gateway.



**Step 10** : This step does not need to care, enter through the next step.



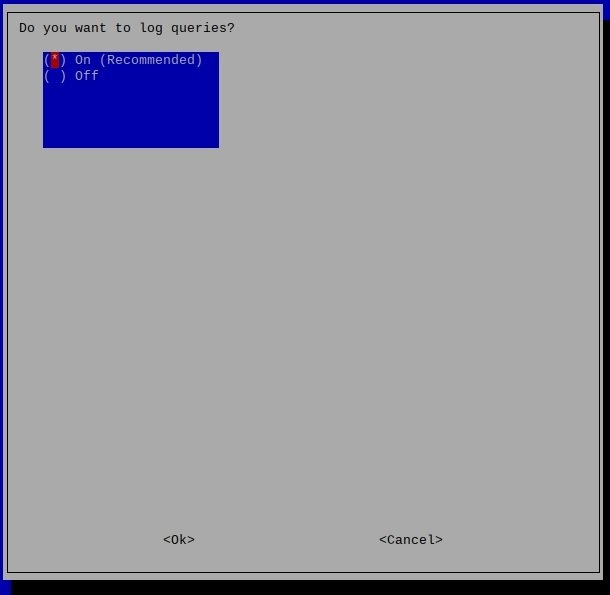
**Step 11** : Pihole asks if you want to install the admin interface to monitor all parameters and manage pihole through the website interface,  Press enter through the next step.



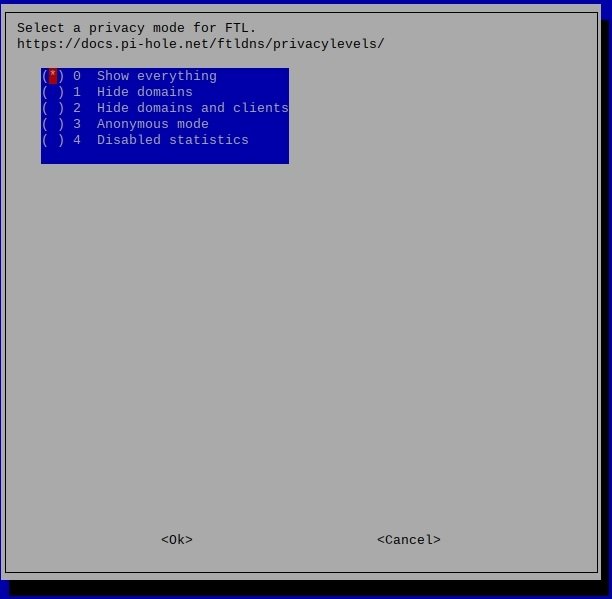
**Step 12** : The default is ON, enter through the next step for early market okay.



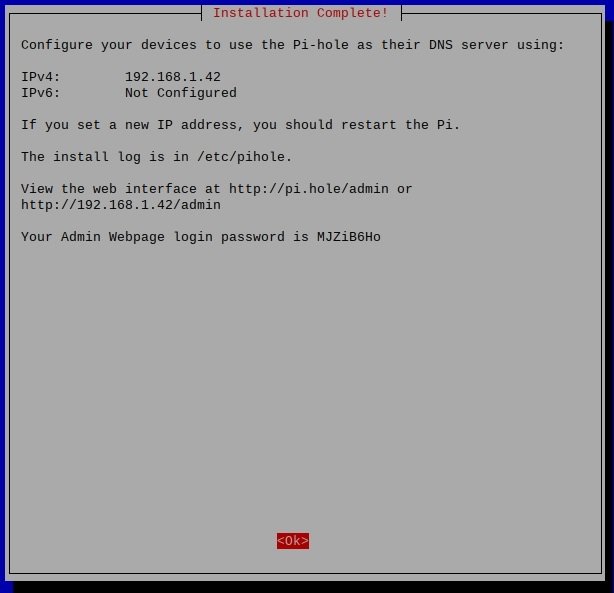
**Step 13** : You want to look through the admin interface, you have to turn ON to log then.



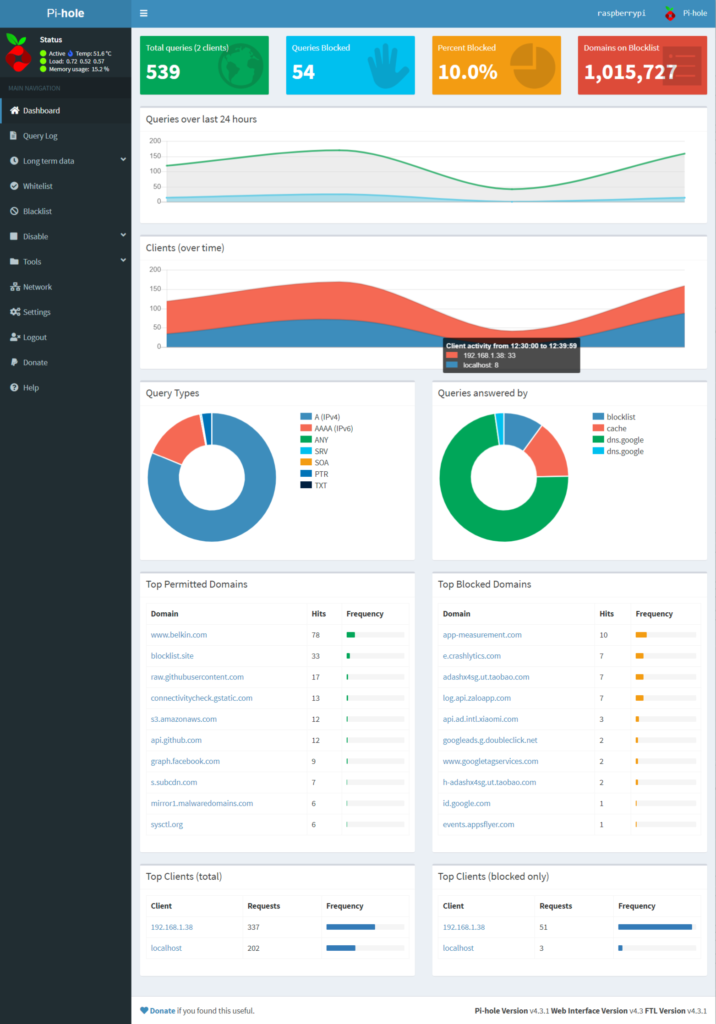
**Step 14** : Press Enter .



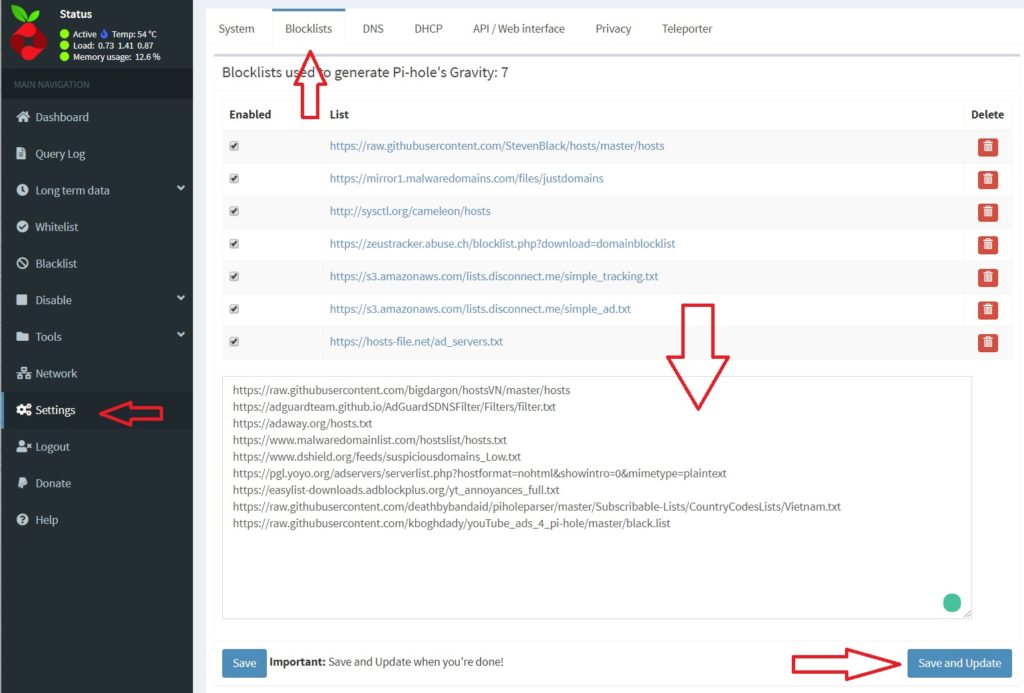
**Step 15** : To change this default password, open the terminal and type this line "sudo pihole -a -p" and then enter. Type the 1st pass, enter and retype 2nd time same as 1st and enter.



**Step 16** : The admin interface will look like this, the parameters will be updated in real time. Now go to settings to add or remove ad sources.



**Step 17** : To add or remove sources of "ads block" you choose Settings => Blocklists => as shown in the picture.



**Step 19** : For the system to work, you need to change the default DNS in the total router. You will see the DNS, you enter the static IP of the Raspberry. This static IP you have installed in step 5.

**Latest Information coverage to IoT Security**

**1. Vulnerability of IoT**

OWASP IoT Top 10 2018:

I1 Weak, Guessable, or Hardcoded Passwords:

Use of easily bruteforced, publicly available, or unchangeable credentials, including backdoors in firmware or client software that grants unauthorized access to deployed systems.

I2 Insecure Network Services:

Unneeded or insecure network services running on the device itself, especially those exposed to the internet, that compromise the confidentiality, integrity/authenticity, or availability of information or allow unauthorized remote control.

I3 Insecure Ecosystem Interfaces:

Insecure web, backend API, cloud, or mobile interfaces in the ecosystem outside of the device that allows compromise of the device or its related components. Common issues include a lack of authentication/authorization, lacking or weak encryption, and a lack of input and output filtering.

I4 Lack of Secure Update Mechanism:

Lack of ability to securely update the device. This includes lack of firmware validation on device, lack of secure delivery (un-encrypted in transit), lack of anti-rollback mechanisms, and lack of notifications of security changes due to updates.

I5 Use of Insecure or Outdated Components:

Use of deprecated or insecure software components/libraries that could allow the device to be compromised. This includes insecure customization of operating system platforms, and the use of third-party software or hardware components from a compromised supply chain

I6 Insufficient Privacy Protection:

User’s personal information stored on the device or in the ecosystem that is used insecurely, improperly, or without permission.

I7 Insecure Data Transfer and Storage:

Lack of encryption or access control of sensitive data anywhere within the ecosystem, including at rest, in transit, or during processing

I8 Lack of Device Management:

Lack of security support on devices deployed in production, including asset management, update management, secure decommissioning, systems monitoring, and response capabilities.

I9 Insecure Default Settings:

Devices or systems shipped with insecure default settings or lack the ability to make the system more secure by restricting operators from modifying configurations.

I10 Lack of Physical Hardening:

Lack of physical hardening measures, allowing potential attackers to gain sensitive information that can help in a future remote attack or take local control of the device.

Researchers offer many tools and ways to search for hacker-friendly IoT devices. The most effective methods have already been tested by botnet creators. In general, the use of certain vulnerabilities by botnets is the most reliable criterion for assessing the level of security of IoT devices and the possibilities of their mass exploitation.

Searching for vulnerabilities, some attackers rely on the firmware (in particular, those errors that were discovered during firmware analysis using reverse engineering methods). Other attackers start looking for vulnerabilities searching for the manufacturer’s name (it can be determined by the first three octets of the MAC address) or the OS version (most devices report it in a network response.)

In any case, for a successful search, some kind of a distinctive feature of a vulnerable device is needed, and it would be nice to find several such features.

Let’s take a look at the standard process of finding vulnerable devices:

First of all, you need a database of vulnerabilities. There are several good sites, for example, Rapid7 or MITRE. These sites help to find vulnerabilities related to certain types of IoT devices.

The most promising in terms of successful exploitation are the following types of vulnerabilities:

Vulnerabilities found after the manufacturer stopped supporting the device and releasing patches.

Recently discovered vulnerabilities that do not have patches yet, or if most users have not had time to install such patches.

Architectural bugs and hardware vulnerabilities that cannot be fully fixed by software patches like Spectre/Meltdown.

Vulnerabilities affecting several models or even types of devices at once, for example, those that have to do with an imperfect component of the web interface or with the communication protocol.

Next, it is necessary to study all the minor technical details associated with selected vulnerabilities as well as the devices affected by them. You need to read all the available documentation in search of unique markers or details of the mistakes made by the manufacturer. It is necessary to determine the features that distinguish the selected devices from the mass of other similar ones. For example, the response from the vulnerable device contains a line with the number of a specific OS version, or maybe a non-standard port is used.

The next step is to prepare advanced search queries for Google (Google Dorks) and specialized search engines for the Internet of Things:

Shodan

Censys

ZoomEye

To prevent script kiddies, we will not cite IPs of vulnerable systems, and detailed queries that make it possible to find low hanging fruits in one click. However, the treasure lies on the surface. It is enough to carefully read the description of the vulnerability and add one or two search filters.

The Shodan and Censys services perform additional screening of evil researchers. Without registration, they show only a short list of search results, limit the number of queries per day, and do not allow them to be refined effectively. All the fun usually begins after the first hundred results, or even further.

Many researchers use scripts to speed up their search for vulnerable IoT devices. To use them (as well as to use your own custom scripts), researchers need to register with Shodan and Censys.

Now, you have to check the search results and (if necessary) sift them using additional queries. Such a need arises almost always; therefore, scripts are also often used to parse results.

It is not a problem to prepare a set of tools for connecting to vulnerable IoT devices. In most cases, a browser will be enough. To control cameras and DVRs, it is sometimes needed to install an old version of Java Runtime Environment (JRE) and a specific video codec. You will also need Telnet and SSH clients. Sometimes you will need specific software like Cisco Smart Install Client.

So, finally, it is time to perform a test connection and try to change the settings. The latter is not recommended, because you can easily run into a trap - honeypot. Interpol also needs to improve its crime detection stats in the field of information security, and not too cautious security researchers may get caught.

**2. The Security Threats to IoT Devices**

1. Botnets

A botnet is a network that combines various systems together to remotely take control over a victim’s system and distribute malware. Cybercriminals control botnets using Command-and-Control-Servers to steal confidential data, acquire online-banking data, and execute cyber attacks like DDoS and phishing. Cybercriminals can utilize botnets to attack IoT devices that are connected to several other devices such as laptops, desktops, and smartphones.

Mirai botnet has displayed how dangerous IoT security threats can be. The Mirai botnet has infected an estimated 2.5 million devices, including routers, printers, and smart cameras. Attackers used the botnet to launch distributed denial of service attacks on several IoT devices. After witnessing the impact of Mirai, several cybercriminals have developed multiple advanced IoT botnets. These botnets can launch sophisticated cyber attacks against vulnerable IoT devices.

2. Denial of service

A denial-of-service (DoS) attack deliberately tries to cause a capacity overload in the target system by sending multiple requests. Unlike phishing and brute-force attacks, attackers who implement denial-of-service don’t aim to steal critical data. However, DoS can be used to slow down or disable a service to hurt the reputation of a business. For instance, an airline that is attacked using denial-of-service will be unable to process requests for booking a new ticket, checking flight status, and canceling a ticket. In such instances, customers may switch to other airlines for air travel. Similarly, IoT security threats such as denial-of-service attacks can ruin the reputation of businesses and affect their revenue.

3. Man-in-the-Middle

In a Man-in-the-Middle (MiTM) attack, a hacker breaches the communication channel between two individual systems in an attempt to intercept messages among them. Attackers gain control over their communication and send illegitimate messages to participating systems. Such attacks can be used to hack IoT devices such as smart refrigerators and autonomous vehicles.

Man-in-the-middle attacks can be used to attack several IoT devices as they share data in real-time. With MiTM, attackers can intercept communications between multiple IoT devices, leading to critical malfunction. For instance, smart home accessories such as bulbs can be controlled by an attacker using MiTM to change its color or turn it on and off. Such attacks can lead to disastrous consequences for IoT devices such as industrial equipment and medical devices.

4. Identity and data theft

Multiple data breaches made headlines in 2018 for compromising the data of millions of people. Confidential information such as personal details, credit and debit card credentials, and email addresses were stolen in these data breaches. Hackers can now attack IoT devices such as smart watches, smart meters, and smart home devices to gain additional data about several users and organizations. By collecting such data, attackers can execute more sophisticated and detailed identity theft.

Attackers can also exploit vulnerabilities in IoT devices that are connected to other devices and enterprise systems. For instance, hackers can attack a vulnerable IoT sensor in an organization and gain access to their business network. In this manner, attackers can infiltrate multiple enterprise systems and obtain sensitive business data. Hence, IoT security threats can give rise to data breaches in multiple businesses.

5. Social engineering

Hackers use social engineering to manipulate people into giving up their sensitive information such as passwords and bank details. Alternatively, cybercriminals may use social engineering to access a system for installing malicious software secretly. Usually, social engineering attacks are executed using phishing emails, where an attacker has to develop convincing emails to manipulate people. However, social engineering attacks can be simpler to execute in case of IoT devices.

IoT devices, especially wearables, collect large volumes of personally identifiable information (PII) to develop a personalized experience for their users. Such devices also utilize personal information of users to deliver user-friendly services, for example, ordering products online with voice control. However, PII can be accessed by attackers to gain confidential information such as bank details, purchase history, and home address. Such information can enable a cybercriminal to execute an advanced social engineering attack that targets a user and their family and friends using vulnerable IoT networks. In this manner, IoT security threats such as social engineering can be used to gain illegal access to user data.

6. Advanced persistent threats

Advanced persistent threats (APTs) are a major security concern for various organizations. An advanced persistent threat is a targeted cyber attack, where an intruder gains illegal access to a network and stays undetected for a prolonged period of time. Attackers aim to monitor network activity and steal crucial data using advanced persistent threats. Such cyber attacks are difficult to prevent, detect, or mitigate.

With the advent of IoT, large volumes of critical data are effortlessly transferred among several devices. A cybercriminal can target these IoT devices to gain access to personal or corporate networks. With this approach, cybercriminals can steal confidential information.

7. Ransomware

Ransomware attacks have become one of the most notorious cyber threats. In this attack, a hacker uses malware to encrypt data that may be required for business operations. An attacker will decrypt critical data only after receiving a ransom.

Ransomware can be one of the most sophisticated IoT security threats. Researchers have demonstrated the impact of ransomware using smart thermostats. With this approach, researchers have shown that hackers can turn up the temperature and refuse to go back to the normal temperature until they receive a ransom. Similarly, ransomware can also be used to attack IIoT devices and smart home. For instance, a hacker can attack a smart home and send a notification to the owner to pay a ransom.

8. Remote recording

Documents released by WikiLeaks have shown that intelligence agencies know about the existence of zero-day exploits in IoT devices, smartphones, and laptops. These documents imply that security agencies were planning to record public conversations secretly. These zero-day exploits can also be used by cybercriminals to record conversations of IoT users. For instance, a hacker can attack a smart camera in an organization and record video footage of everyday business activities. With this approach, cybercriminals can acquire confidential business information secretly. Such IoT security threats will also lead to major privacy violations.

To mitigate their effects, business leaders need to be updated about IoT security threats and create a holistic cybersecurity strategy before utilizing IoT infrastructure for their organization. For this purpose, they can hire a dedicated team of cybersecurity professionals who can take care of all security concerns. Alternatively, if business leaders wish to carry out cybersecurity techniques independently, they can start by ensuring that all their confidential data is encrypted and their systems are regularly audited for security purposes. Businesses can also deploy modern technologies such as big data, blockchain, and AI to enhance their cybersecurity efforts.

**3.** **Cloud IoT security controls**

Given the variety of cloud-based services that support IoT deployments, each cloud and stakeholder endpoint plays a vital role in securing the multitude of transactions. This section provides a brief listing of recommended IoT security controls and services that your organization should consider. Basic controls such as authentication and encryption to the cloud are supported by all of the CSPs, but you should carefully review and consider your CSP based on their offerings in other areas.

Most CSPs bundle the services in different ways. Your organization can either directly or indirectly obtain and benefit from these services based on unique package offerings. These services can be combined in different ways to build powerful, transitive trust relationships throughout your virtualized infrastructure.

**Authentication (and authorization)**

Considering authentication security controls, your organization will need to handle

most or all of the following:

1. Verify administrator authenticity for individuals accessing administrative functions and APIs (multi-factor authentication is preferred here, given the enormous sensitivity of administrative controls on your virtual infrastructure).

2. Authenticate end users to cloud applications.

3. Authenticate cloud applications (including IoT gateways and brokers) from one to the other.

4. Directly authenticate IoT devices (that have the requisite security and functional resources) to gateways and brokers.

5. Proxy-authenticate end users from application provider to another.

A variety of authentication mechanisms are supported by CSPs. Amazon AWS and Microsoft Azure are described in the following sections.

**End-to-end security recommendations**

Consider the following end-to-end security recommendations in your IoT cloud deployment:

• Ensure that security is not lost at the gateway. Ideally, end-to-end authentication and integrity protections should persist from the CSP to the IoT devices with the gateways simply acting as pass-throughs. Although this is not always possible, take alternate defensive actions when deployed sensor nodes rely upon the gateway to validate the authenticity and integrity of firmware updates and commands.

• Apply the rigor of secure software development practices to the web services and databases that serve the IoT devices.

• Sufficiently protect the cloud applications that support the analysis and reporting workflows.

• Apply secure configurations to the databases that feed the analysis and reporting applications.

• Apply integrity protections to the IoT device data. This requires the use of integrity protections on data transmitted from the IoT device to the gateway as well as the gateway to the cloud.

• Leased devices will operate within the customer environment and service providers will not want to inadvertently infect their customer networks with malware (and vice versa). Segregation of these devices on customer networks should be enforced when possible. This use case opens up potential for fraud and/or theft from stealing services, and as such it is important to design the devices in a manner that prevents tampering. This can be accomplished using tamper-evident or tamper-responsive protections that are described in resources such as NIST FIPS 140-2.

• Protect against denial of service attacks by using robust, properly configured load balancing application gateways (a number of superb industry solutions exist for this now).

• Provide assurances that the data being transmitted to the IoT devices (or gateways) is authenticated by the devices themselves.

• Encrypt the data when needed.

• Transactions and messaging between devices themselves (M2M) must be authenticated (and integrity protected)

• In all cases, service providers should be able to track the privacy controls

associated with information generated by a person or by a device that can be tied to a person. In the case of the medical device, has the patient been notified and authorized the use of not only the data generated while in medical offices, but also for any data that is uploaded to the cloud by connected devices? Notifications should also include any organization that the data may be shared with.

• Maintaining control of data through to destruction is not possible when the data may have been passed on to potentially many other organizations. However, service providers should make attempts to obtain privacy agreements with peer organizations. Additionally, assess the adequacy of the security controls implemented by those other organizations.

• Implement flexible access controls (use attribute-based access controls for higher resolution access decisions).

• Tag data for privacy protections.

• Provide notifications on data use.

**Maintain data integrity**

How can you assure the integrity of data that will be used for myriad purposes and by potentially many stakeholders? In the context of an enterprise IoT system, the ability to trust the data collected is critical. This drives a need for the following:

• Authentication and integrity controls applied to the IoT devices to ensure rogue devices cannot transmit data into the cloud.

• Secure configuration of gateway devices. Gateway devices may be installed on-site or operate in the cloud, but these gateways devices process large quantities of data and as such should be secured via:

° Security logging and analysis in a SIEM.

° Secure configurations (operating system, database, application).

° Firewall protection.

° Encrypted communications on each interface. This requires the use

of encrypted communication on the cloud-facing interface. This is typically accomplished with Transport Layer Security (TLS) and an

appropriate ciphersuite. On the sensor-facing interface, encrypted RF communications is strongly recommended.

° Strong authentication using PKI certificates if possible.

• Software security measures for the web service that interfaces with and collects data from the gateways or devices.Cloud Security for the IoT

• Secure infrastructure configurations (for example, web server) supporting the IoT web service.

**Secure bootstrap**

In order to have confidence in the credentials used by a particular device to authenticate to services and gateways, care must be taken during the initial provisioning of trust to devices. Depending on the criticality of a particular device, bootstrap can occur at the vendor, or in-person by a trusted agent. Completing bootstrap and enrollment results in the ability to provision operational certificates to devices in a secure manner (and over a network).

**Security monitoring**

IoT gateways/brokers should be configured to look for suspicious behavior of the endpoints. As an example, MQTT brokers should capture messages from publishers and subscribers that may signal malicious behavior. The MQTT Specification Version 3.1.1 provides examples of behaviors to report:

• Repeated connection attempts

• Repeated authentication attempts

• Abnormal termination of connections

• Topic scanning

• Sending undeliverable messages

• Clients that connect but do not send data.

**4. Blockchain for IoT security**

The application of blockchain in IoT security enables a direct information sharing between connected devices instead of communication via a centralized network, thus decreasing the susceptibility of IoT to cyber-threats. Currently, the highest rate of blockchain adoption among IoT-enabled businesses in the US is seen in pharmaceuticals, transportation and energy sectors, according to a Gartner survey . All these industries rely on transportation of physical goods, and the majority of companies that have successfully adopted blockchain are veterans of the IoT industry.

Perhaps the most promising way of successfully combining the two technologies is to install chips in every IoT device. For example, the alliance of Ubirch, a blockchain-anchoring security specialist, G+D Mobile Security, and IoT carrier 1NCE has developed an IoT security service that leverages the power of blockchain and sensor-embedded chips to significantly increase security. Data no longer travels from sensors to the cloud to be approved, which single-handedly eliminates the possibility of the ‘man in the middle’ type of hacking attacks. Now the information is sealed by a private key directly on the device and is anchored in a public blockchain, which means that data about every access to a particular sensor is forever recorded on a ledger. Adopting smart contracts for this purpose also opens up more opportunities for enhancing

Such applications of blockchain and enterprise IoT can also significantly increase standards in other industries such as pharmaceuticals and food. For example, many medical machines have to be kept under regulatory-approved temperatures. IBM’s Food Trust has been in the works for a few years already, but IoT sensors can bring even more exciting opportunities when combined with blockchain.

IBM and Samsung have developed a PoC for a blockchain‑enabled IoT system called ADEPT (Autonomous Decentralized Peer‑to‑Peer Telemetry). It uses smart contracts and peer-to-peer messaging to create a distributed IoT network. ADEPT may find its application in a smart home, where a Samsung washing machine, for example, can become a semi-autonomous device capable to perform self-service and maintenance. If the machine goes out of order, it will notify operator about the breakage and install software updates on its own. Using ADEPT, the washing machine can communicate with other smart devices in a network to optimize energy efficiency. For instance, it may postpone a cycle of washing for several hours if the TV is on. Moreover, it allows the machine to manage the supply of detergent it uses: pay for the order itself and receive a delivery confirmation from the retailer. The washing machine’s owner then will receive a notification about the purchase on the smartphone. This is less futuristic than it may sound, and the systems similar to ADEPT may gain the market soon.

Chronicled, a self-proclaimed IoT and blockchain laboratory, applied a combination of these two technologies to pharmaceutical supply chains. The developed solution allows drug makers, wholesalers, and hospitals to monitor each step of drug shipment and makes it difficult for criminals to unload stolen medicines.

It can be used to create secured mesh network that will allow IoT devices to connect securely and reliably avoiding the threats of device spoofing and impersonation.

Every IoT node can be registered in the blockchain and will have a blockchain id which will uniquely identify a device in the universal namespace. For a device to connect another device, one will use the blockchain id as URL and will use its local blockchain wallet to raise an identity request. The wallet will create a digitally signed request and send to the target device which will use blockchain services to validate the signature using the public key of the sender. In this way, M2M authentication can take place without the need of any centralised arbitrator or service.

For a device that is constrained by a resource can be connected to proxies where the wallet can be stored. This will introduce some form of aggregation but it will be fairly limited.

The above possible solution will be applicable to a wide range of IoT services. Some of the examples will be intelligent healthcare connected vehicles, logistics, transportation etc.

Although there are many advantages of adopting blockchain for IoT security, the technology is far from perfect. Being a technology enabler for Bitcoin, blockchain does its job well in the cryptocurrency realm: protecting sensitive financial data when moving currency from one person to another. However, IoT implies control over a network of devices, where multi-layered security has to be put in place.

One of the major roadblocks on the way to adoption is paradoxically linked to one of the proposed advantages of blockchain: every action on the network has to be approved by other network participants for it to go live. For example, in case of an obvious security breach through one of the connected devices, denying access to that device would significantly decrease the negative impact of spreading the malware. On a bigger scale, with thousands of ‘things’ connected to a large network, it can be difficult to receive consent from the majority of entities.

It could seem that this can at least be implemented for smaller systems like smart homes. Unfortunately, another challenge emerges: home devices don’t usually have enough computing power to maintain a blockchain.

There is no easy way to address this challenge, but a custom blockchain platform can be a solution. Blockchain developers have to ensure that corrupted devices could be instantly eliminated from the network without the need for a conventional blockchain consensus. Organizations should thoroughly investigate their privacy requirements and choose a proper blockchain type or request the development of a custom one until ready-made solutions appear on the mass market.

**5. Smarthome security**

1. Give your router a name.

Don’t stick with the name the manufacturer gave it — it might identify the make or model. Give it an unusual name not associated with you or your street address. You don’t want your router name to give away any personal identifiers.

2. Use a strong encryption method for Wi-Fi.

In your router settings, it’s a good idea to use a strong encryption method, like WPA2, when you set up Wi-Fi network access. This will help keep your network and communications secure.

3. Set up a guest network.

Keep your Wi-Fi account private. Visitors, friends and relatives can log into a separate network that doesn’t tie into your IoT devices.

4. Change default usernames and passwords.

Cybercriminals probably already know the default passwords that come with many IoT products. That makes it easy for them to access your IoT devices and, potentially, the information on them. Are you considering a device that doesn’t allow you to change the default password? Then consider a different one.

5. Use strong, unique passwords for Wi-Fi networks and device accounts.

Avoid common words or passwords that are easy to guess, such as “password” or “123456.” Instead, use unique, complex passwords made up of letters, numbers, and symbols. You might also consider a password manager to up your security game.

6. Check the setting for your devices.

Your IoT devices might come with default privacy and security settings. You might want to consider changing them, as some default settings could benefit the manufacturer more than they benefit you.

7. Disable features you may not need.

IoT devices come with a variety of services such as remote access, often enabled by default.

If you don’t need it, be sure to disable it.

8. Keep your software up to date.

When your smart phone manufacturer sends you a software update, don’t put off installing it. It might be a patch for a security flaw. Mobile security is important, since you may connect to your smart home through mobile devices. Your IoT device makers also may sent you updates — or you might have to visit their websites to check for them. Be sure to download updates and apply them to your device to help stay safe.

9. Audit the IoT devices already on your home network.

It could be time to upgrade that old security camera. Take time to check if newer models might offer stronger security.

10. Do the two-step.

We’re talking authentication. Two-factor authentication — such as a one-time code sent to your cellphone — can keep the bad guys out of your accounts. If your smart-device apps offer two-factor authentication, or 2FA, use it.

11. Avoid public Wi-Fi networks.

You might want to manage your IoT devices through your mobile device in a coffee shop across town. If you’re on public Wi-Fi — generally not a good idea — use a VPN. For instance, Norton Secure VPN offers a number of privacy and security features for both public and home Wi-Fi.

12. Watch out for outages.

Ensure that a hardware outage does not result in an unsecure state for the device.

No doubt more IoT devices are coming and will angle for a place in your home. If they make your life more convenient — even happier — great. But don’t forget to secure your increasingly smart home and your IoT devices.

**6. IoT Security for Smart Cities**

Smart cities promise greater operational efficiency, increased environmental and financial sustainability and a new level of responsiveness to the needs of residents and visitors. However, this new “intelligence” requires sending information from millions of internet-of-things sensors to the cloud, significantly increasing cybersecurity risks. Governments planning a smart city strategy must accept that they may become a favorite target of attackers.

While a SASE can establish basic cyber hygiene, smart cities must still practice “a healthy lifestyle.” This includes putting the right people and processes in place to minimize risk. For example, because human error is a common cause of breaches, agencies train everyone who handles data on their individual responsibilities for protecting that data, including securing social media and email accounts, backing up data where required, keeping all devices and applications not managed by the agency up to date and learning to avoid phishing and spear-phishing scams.

Many agencies have made great strides in promoting these healthy practices; however, most still ignore perhaps the most important cyber defense preparation: a robust incident response (IR) plan that can minimize data loss, the financial impact and the time to restore processes and services.

An effective IR plan requires:

Preparedness – Build a quality IR team. Security engineers should have the technical skills to uncover and defend against attacks, as well as the social skills to connect with colleagues when a breach occurs. Team members may also include compliance officers, human resources managers, attorneys and public relations specialists. The team should be involved in helping to develop cyber policies and an IR communications plan to ensure timely delivery of information to internal and external stakeholders.

Breach detection – Arm the IR team with 24/7 monitoring and detection tools that can help them understand the type of incident, whether it's malware, phishing or denial-of-service. Since many incidents are discovered by external parties, have a visible point of contact who can be easily reached.

Triage – Determine whether attackers are still in the network and what the current threat level is. Combine all available information from the network with the latest threat intelligence to formulate the most effective response.

Remediation – Find and eradicate the cause of the threat. Back up all affected systems to preserve their current state for later forensics and then proceed with any required service restoration. Communicate with stakeholders and demonstrate the agency's control of the situation.

Building an effective cyber defense is hard, but governments can ensure their smart city initiatives maximize benefits and minimize risks by starting with a SASE solution to create a solid foundation of good cyber hygiene and then establishing healthy cyber practices that include a robust incident response plan.

## References:

## https://medium.com/@dtembe/setting-up-an-open-source-siem-for-iot-standard-devices-applications-logs-pt-1-4bf1dcf28f8b

## https://www.vaadata.com/blog/pentest-iot-10-hardware-software-tests/

## https://blog.beyondsecurity.com/security-testing-the-internet-of-things-iot/

## https://github.com/Banjopkr/WQ7Tpanadapter

## https://learn.adafruit.com/freq-show-raspberry-pi-rtl-sdr-scanner/installation

## https://www.instructables.com/id/Install-and-Setup-Shinken-Network-Monitor-on-Raspb/

## https://hakin9.org/metasploit-on-raspberry-pi-tutorial-free-course-content/

## https://www.home-assistant.io/docs/installation/centos/

## https://www.peerlyst.com/posts/beginner-s-guide-to-pentesting-iot-architecture-network-and-setting-up-iot-pentesting-lab-part-1-nitesh-malviya?fbclid=IwAR2z3ioWfeTxOtGRi3WMrmjkJYdde2vmbnNI-M8XKuO-kfXJDtOIUK3d1Fg

## https://www.peerlyst.com/posts/zeek-ids-installation-on-raspberry-pi-part-1-david-lagace-cissp-cism-pmp?fbclid=IwAR33nJHZ6sJu60CLpjRP-eYp37oqPZqAiHdxbK2ZKxOTpeFVl5wb3Cirizw

## https://jwgoerlich.com/principles-for-designing-security-capabilities/

## https://shahmeeramir.com/conducting-an-iot-pentest-6fa573ac6668

## https://github.com/V33RU/IoTSecurity101

## https://www.iotpentestingguide.com/

## https://www.pentestpartners.com/security-blog/iot-security-testing-methodologies/

## https://www.networkworld.com/article/3321919/securing-the-iot-has-become-business-critical.html

## https://hakin9.org/script-kiddie-nightmare-iot-attack-code-embedded-with-backdoor/?fbclid=IwAR0-Q26Q9wvpBt1L\_N36Tu89Y5WG2ZxrodzqpZ1AnThEDHbjN-MymMM3hG8

## https://pentestmag.com/iot-security-how-to-search-for-vulnerable-connected-devices/?fbclid=IwAR03\_-NmbWc-Xz4krxmCe7lwQtszk0fswRdL9IkN54uSY7K5CKuTz2VhULs

## https://www.kitploit.com/2019/10/iot-implant-toolkit-toolkit-for-implant.html?fbclid=IwAR1trwAa5Jui1AQvgWBRcLwuvf5T6\_kth63X8GFYfww\_G0DoSv5MwmxIfOc

## https://www.kitploit.com/2020/01/iotshark-monitoring-and-analyzing-iot.html?fbclid=IwAR2i66Cx6zlmuVulGbWjwCw\_-B5JVtnZQNUq0feWMKJMf6jaE33aVSIiias

## https://autome.me/what-is-iot-security-and-how-to-achieve-it/?fbclid=IwAR0qi-ijzFYTwmqy5J1GTA5WcG9LxsVCIoILqjrPvkqUyrmlE4UHhosl8JU

## https://www.hitachi-systems-security.com/blog/privacy-data-protection-challenges-in-the-iot/

## https://www.mindedsecurity.com/index.php/services/testing/iot-security

## https://www.praetorian.com/expertise/internet-of-things

## https://www.scnsoft.com/blog/iot-penetration-testing

## https://www.wizlynxgroup.com/hk/en/cyber-security-hong-kong/iot-security-testing-services

## https://www.xmind.net/m/UZ2e/

## https://pi-hole.net/2020/05/10/pi-hole-v5-0-is-here/?fbclid=IwAR0cMlWEsaWnqMpMsO8RVm424Yh\_fpouLhox1qBcdkP67ep8PCxDRtWJMRE#page-content

## https://www.smarthomebeginner.com/pi-hole-setup-guide/

## https://github.com/V33RU/IoTSecurity101/blob/master/README.md

## https://github.com/nebgnahz/awesome-iot-hacks/blob/master/README.md

## https://github.com/fkie-cad/awesome-embedded-and-iot-security/blob/master/readme.md

## https://syhack.wordpress.com/2019/09/23/iot-pentesting-approach-methods/

## https://www.slideshare.net/YogeshOjha4/kazhackstan-doing-the-iot-penetration-testing-yogesh-ojha

## https://resources.infosecinstitute.com/pentesters-guide-to-iot-penetration-testing/#gref

## https://blog.attify.com/how-to-iot-pentesting/

## https://www.i-scoop.eu/internet-of-things-guide/iot-security-forecasts/

## https://www.mitre.org/publications/project-stories/homing-in-on-the-challenge-of-securing-the-internet-of-things

## https://awakesecurity.com/datasheets/awake-covers-the-mitre-attack-framework/

## https://blog.checkpoint.com/2020/01/20/preventing-zero-day-attacks-using-mitre-attck-framework/

## https://www.armis.com/resources/iot-security-blog/mitre-attack-techniques-for-ics-practical-applications-part-2/

## https://cyberx-labs.com/blog/telling-the-full-story-with-the-mitre-attck-for-ics-framework/

## https://www.trendmicro.com/vinfo/us/security/news/internet-of-things/the-iot-attack-surface-threats-and-security-solutions

## https://www.fortinet.com/blog/threat-research/iot-is-the-weakest-link-for-attacking-the-cloud.html

## https://www.securitymagazine.com/articles/91750-more-than-84-million-ddos-attacks-targeted-it-infrastructures-cloud-mobile-networks-iot-devices-in-2019

## https://dl.acm.org/doi/fullHtml/10.1145/3376123

## https://iotify.io/top-5-challenges-in-iot-platform-testing/

## https://www.infoworld.com/article/3448256/thinking-about-the-security-of-iot-and-the-cloud.html

## https://www.avsystem.com/blog/iot-cloud-platform/

## https://www.marketsandmarkets.com/Market-Reports/iot-security-market-67064836.html

## https://fr.barracuda.com/resources/Barracuda\_Cloud\_Generation\_Firewall\_F\_DS\_IoT\_US

## https://wso2.com/iot/

## https://www.endian.com/de/community/

## https://payatu.com/blog/munawwar/trendNet-wireless-camera-buffer-overflow-vulnerability

## https://www.novelbits.io/bluetooth-version-5-2-le-audio/

## https://notknowiot.com/devices/esp32/esp32-the-secure-boot.html?fbclid=IwAR09IMXfHdZCIIc3UiphHSGwUMsl5fa0T5r5PeVr7\_9RgtMuypOvjWEaVZA

## https://pentestmag.com/iot-security-how-to-search-for-vulnerable-connected-devices/?fbclid=IwAR1npi\_93-1DH94r5b0j4c0PkV3OGxX3cW-LHfWyg70-UstBIbI5JsNnrCI

## https://github.com/IoTSecurity101/BLE-UAE

## https://www.allerin.com/blog/8-types-of-security-threats-to-iot

## https://www.itransition.com/blog/blockchain-iot-security

## https://us.norton.com/internetsecurity-iot-smart-home-security-core.html

## https://gcn.com/articles/2019/12/17/sase-smart-city-security.aspx