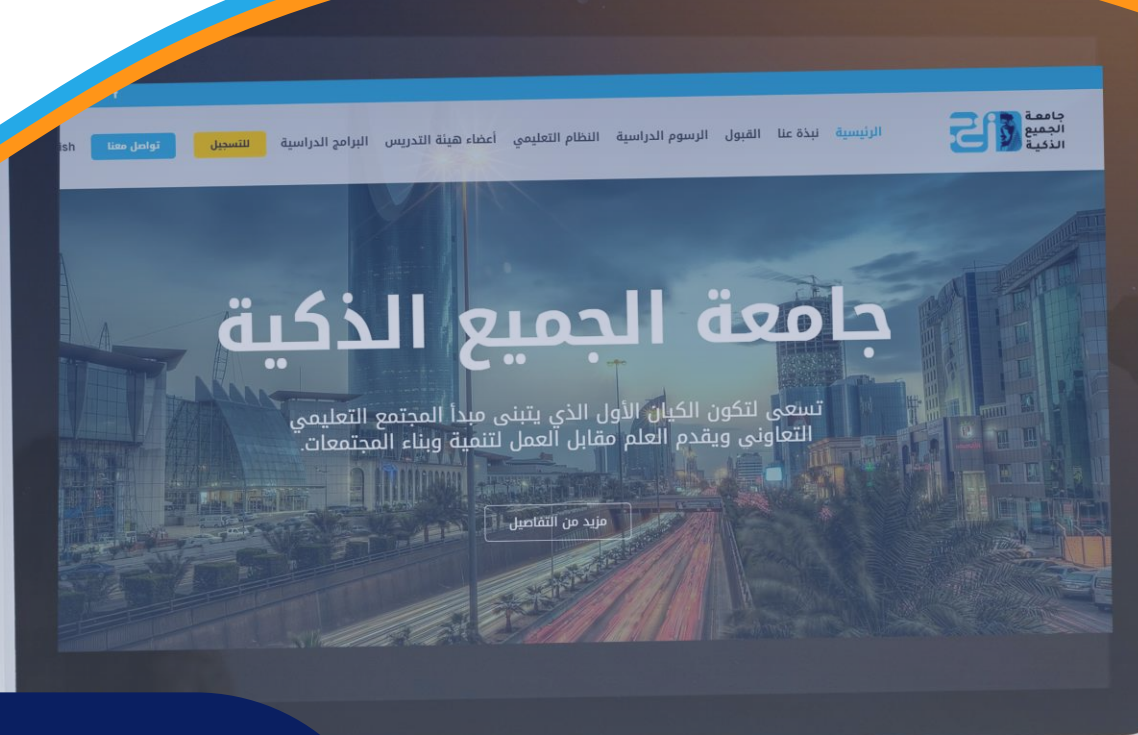




جامعة الجميع الذكية
EVERYONE'S SMART UNIVERSITY



IoT Security
Prepared by Dr. Abdelrhman Ahmed
Lecture 8

IoT Risk & Web/Cloud Interface
Security

As part of its Internet of Things Project, the Open Web Application Security Project (OWASP) has published a detailed draft list of IoT attack surface areas, or areas in IoT systems and applications where threats and vulnerabilities may exist. Below is a summarization of the IoT attack surface areas:

- **Devices:-** Devices can be the primary means by which attacks are initiated. Parts of a device where vulnerabilities can come from are its memory, firmware, physical interface, web interface, and network services. Attackers can also take advantage of unsecure default settings, outdated components, and unsecure update mechanisms, among others.



- **Communication channels:-** Attacks can originate from the channels that connect IoT components with one another. Protocols used in IoT systems can have security issues that can affect the entire systems. IoT systems are also susceptible to known network attacks such as denial of service (DoS) and spoofing.
- **Applications and software:-** Vulnerabilities in web applications and related software for IoT devices can lead to compromised systems. Web applications can, for example, be exploited to steal user credentials or push malicious firmware updates.



Having done numerous IoT pen testing engagements so far, It is confirmed that to conduct an IoT device pentest effectively; you will need to assess and map the entire attack surface for that given device. Assessing an IoT device attack surface does not vary a lot from web attack surface mapping, however here a lot many more components will be involved.

Below are the steps you would need to follow to map the attack surface of an IoT device:

- **Step 1:** The first step of Attack Surface mapping is to understand the entire IoT device architecture. This would require you to go through all the product manuals, documentation, information on the vendor's website and any additional detail you can find from any other source.

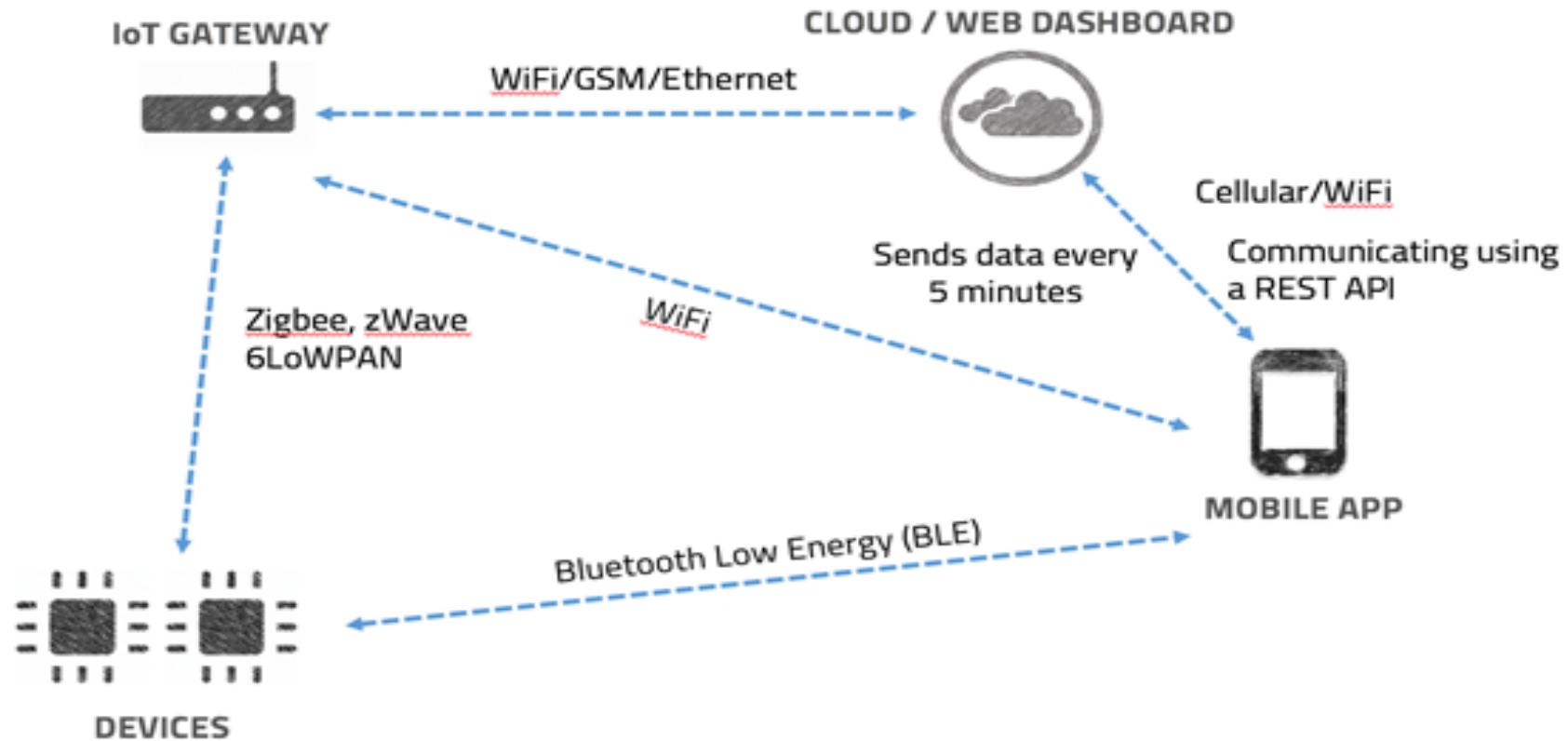
- **Step 2:** Build an architecture diagram for the given device specifying each of the components that is used in various categories we discussed above. If it is radio communication between two devices, show it with a dotted line and specify the communication protocol being used. If the application is communicating with the cloud to send and receive data using an API, mark it in the architecture diagram along with the kind of API which is used.

A sample architecture diagram is shown in the next slide.



Chapter 08

Map IoT Attack Surface



- **Step 3:** Once the complete architecture diagram has been prepared, think like an attacker. If you have to attack a particular component, what techniques would you use and what kind of information would you try to extract? List down all the specific test cases for each component in a spreadsheet.

The next Slide is a snippet of the attack surface list for the above architecture diagram

Once the above steps have been completed, we can now proceed to the actual execution of the attack vectors. Since we already have a clear idea of what technique we will use and what data we are after, our efforts will now be much more optimized for the time and would lead to better results.

Chapter 08

Map IoT Attack Surface

Component	Attack Surface	Cases	Expected Output/ Data Obtained
IoT Gateway	Hardware based attack vectors	Interacting using serial communication	Debug logs, shell access
		Firmware dumping techniques	Access to firmware, reversing and analyzing firmware for vulnerabilities
	Network communication to cloud	Sniffing Communication data	Understanding Communication messages, checking if data is being sent in clear text
		Modifying and replaying data	Identifying if the end web app is checking for authenticity and legitimacy of the message

“A snippet of the attack surface list for the above architecture diagram”

Many people have long associated IoT attacks with lower-level threats like distributed denial of service and crypto-mining botnets. But in reality, there are a growing number of ransomware, espionage, and data theft attacks that use IoT as the initial access point into the larger IT network, including the cloud. Advanced threat actors are also using IoT devices to achieve persistence inside these networks while evading detection, as was recently seen with the **QuietExit backdoor**.

An analysis of millions of IoT devices deployed in corporate environments, It was found that both high-risk and critical vulnerabilities (based on the Common Vulnerability Scoring System, or CVSS) are widespread. Half of all IoT devices have vulnerabilities with a CVSS score of at least 8, and 20% have critical vulnerabilities with a CVSS score of 9–10. At the same time, these devices also suffer from a number of basic security failures, in terms of password protection and firmware management.



While IoT risks can't be completely eliminated, they can be reduced. Here are several steps companies should take.

- Create a holistic and up-to-date asset inventory

In our research, we have found that 80% of corporate security teams can't even identify the majority of IoT devices on their network. That is an astounding number, and it shows how serious the problem is. If a company doesn't even know which devices are on its network, how can it possibly defend them from attack or protect its IT network from lateral movement after a successful IoT breach? IoT inventorying isn't easy, though. Traditional IT discovery tools were never designed for IoT. Network behavior anomaly detection systems listen for traffic on span ports, but most of the IoT traffic is encrypted, and even if it isn't, the information transmitted doesn't have enough identification details.

It's not enough to simply know something is an HP printer without any specifics, especially if it has vulnerabilities that need to be fixed. Legacy vulnerability scanners can help, but they operate by sending malformed packets, which aren't great for IoT identification and can even knock an IoT device offline.

A better approach is to discover IoT devices by interrogating the devices in their native language. This will allow an organization to create an inventory with exhaustive details about the IoT devices, such as device version, model number, firmware version, serial number, running services, certificates and credentials. This allows the organization to actually remediate these risks and not just discover them. It also enables them to remove any devices considered high-risk by the U.S. government, such as Huawei, ZTE, Hikvision, Dahua and Hytera.

- Password security is essential

Attacks on IoT devices are easy to carry out because many of these devices still have default passwords. We have found this to be the case in approximately 50% of IoT devices overall, and it is even higher in specific categories of devices.

For example, 95% of audio and video equipment IoT devices have default passwords. Even when devices don't use default passwords, we've found that most of them have only undergone one password change in as much as 10 years.

Ideally, IoT devices should have unique, complex passwords which are rotated every 30, 60 or 90 days.

However, not all devices support complex passwords. Some older IoT devices can only handle four-digit PINs, whereas others only allow 10 characters, and some don't accept special characters.

It's important to learn all of the details and capabilities of an IoT device, so effective passwords can be used and changes can be made safely. For legacy devices with weak password parameters, or no ability to provide any level of authentication, consider replacing those devices with more modern products that will allow better security practices.

- **Manage device firmware**

Most IoT devices run on outdated firmware, which poses significant security risks since vulnerabilities are so widespread. Firmware vulnerabilities leave devices exposed to attacks including commodity malware, sophisticated implants and backdoors, remote access attacks, data theft, ransomware, espionage, and even physical sabotage. Our research has determined that the average device firmware is six years old and roughly one-quarter of devices (25–30%) are end-of-life and no longer supported by the vendor.

IoT devices should be kept updated with the latest firmware version and security patches provided by the vendors. Admittedly, this can be a challenge, particularly in large organizations where there are literally hundreds of thousands to millions of these devices. But one way or another, it has to be done to keep the network secure. Enterprise IoT security platforms are available that can automate this and other security processes at scale.

However, sometimes device firmware should be downgraded, rather than updated. When a vulnerability is being widely exploited, and there is no available patch—since IoT vendors often take longer to issue patches than traditional IT device manufacturers—then it may be advisable to temporarily downgrade the device to an earlier firmware version that does not contain the vulnerability.

- Turn off extraneous connections, and limit network access

IoT devices are often easy to discover and have too many connectivity features enabled by default, such as wired and wireless connections, Bluetooth, other protocols, Secure Shell, and telnet. This promiscuous access makes them an easy target for an external attacker.

It's important for companies to do system hardening for IoT just as they have with their IT networks. IoT device hardening involves turning off these extraneous ports and unnecessary capabilities. Some examples are running SSH but not telnet, operating with wired ethernet, but not Wi-Fi, and turning off Bluetooth.

Companies should also limit their ability to communicate outside of the network. This can be done at Layer 2 and Layer 3 through network firewalls, unidirectional diodes, access control lists, and virtual local area networks. Limiting internet access for IoT devices will mitigate attacks that depend on the installation of command-and-control malware, such as ransomware and data theft.

- Ensure certificates are effective

In our research, we've found that IoT digital certificates, which ensure secure authorization, encryption and data integrity, are frequently out of date and poorly managed. This problem even occurs with critical network devices, like wireless access points, which means even the initial access point to the network isn't properly secured.

It's very important to validate the state of these certificates and integrate them with a certificate management solution in order to remediate any risks which might occur, such as TLS versions, expiration dates and self-signing.

- Watch out for environmental drift

Once IoT devices have been secured and hardened, it's important to make sure they stay that way.

Environmental drift is a common occurrence, as device settings and configurations can change over time due to firmware updates, errors and human interference. Key device changes to watch out for are passwords that are reset to default or other credential modifications that didn't come from the PAM, firmware downgrades, and insecure services which have suddenly been turned back on.

Security by design is a methodology that ensures security is a crucial objective at all stages of product creation and deployment. It addresses the challenge that, in many historic hardware deployments and instances of IoT design, security considerations were often included late in the design and prototyping phase. By prioritizing speed to market or other design considerations, security requirements can end up being added on.

The result?

This approach has led to serious security breaches in the past, as IoT security can not easily be retrofitted. Key examples are unsecured devices that enable attackers to breach an otherwise secure system or large numbers of connected devices that cannot be upgraded to deal with emerging threats.



Beyond knowing a few steps you can take with devices, it helps to choose a manufacturer that follows security by design. **Security by design is a set of principles within hardware and software development focused on securing the system and reducing the risk of a compromise.** Following these principles allows a manufacturer to know that they are protecting users and complying with the **European Union's General Data Protection Regulation (GDPR)**. Systems built using this method incorporate elements such as abiding by coding best practices, implementing authentication protections and deploying continuous testing.

The key reason that secure by design is so important is that software is typically considered first and foremost in terms of its function. Security becomes a secondary concern, and the developers must address security holes and vulnerabilities as an ongoing concern rather than building it with optimized security. With secure by design, you can be certain that the manufacturer is fixing security issues effectively and rapidly.

Security by design incorporates the following principles:

Secure defaults:- Create a secure experience standardly. Allow users to remove protections if desired.

Correctly repair security issues. Be careful about design patterns, which can introduce regressions when you attempt to fix your code. Test on all relevant applications.

Keep security simple. You want your code to be as simple as possible. It is easier to reduce your attack surface area in that context.

The principle of defense in depth:- While it may be reasonable to just have a single control, add more controls so that your defenses are deeper.

The principle of least privilege. Accounts should be given the minimum possible level of privilege in order to complete their business functions.



Do not trust services:- You may utilize outside providers for processing. Keep in mind, though, that services should not be trusted, by default.

Avoid security by obscurity:- You should not attempt to protect critical data simply by hiding key details. It is an insufficient security control.

Separation of duties:- Typically, administrators should not be users of an application. For instance, an administrator should not be able to buy from a storefront as a super-privileged user.

Secure failures:- Verify that your code never fails in a manner that makes the user an administrator by default.

Minimize attack surface area. The attack surface area should be restricted as much as possible. All features add risk. They should warrant it.

Privacy by design, a concept within the GDPR, is similar to security by design. The two core elements of privacy by design are:

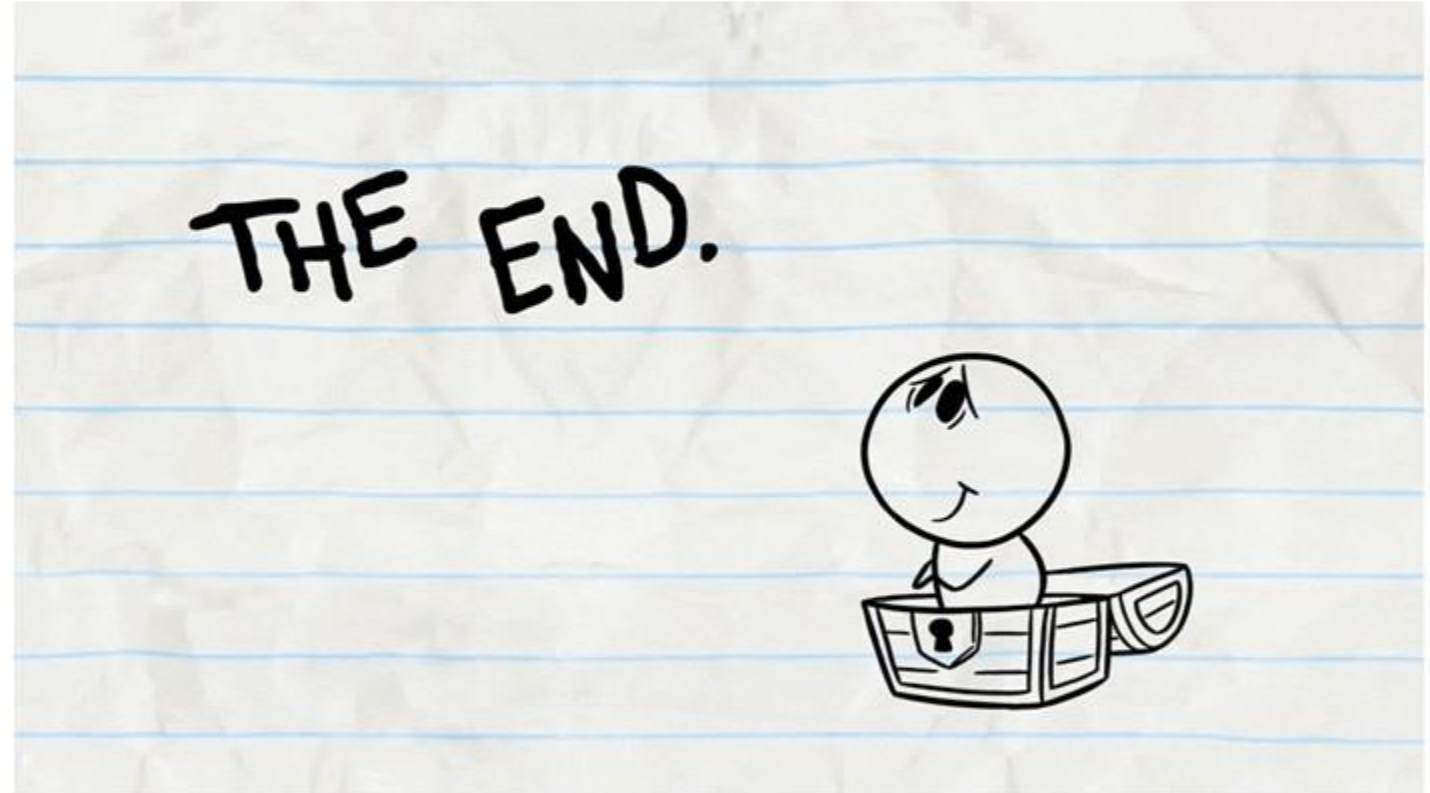
Data protection by default:- Data controllers should only store data as long as necessary, should only process data to the extent necessary, and should only process the specific data that is necessary and Data controllers should keep the processing of personal data as limited as possible by implementing pseudonymization and other safeguards.



1. The Thales Security by Design approach ensures that IoT security is considered at the beginning of project design. A successful security architecture ensures that devices are protected in the right location and at the right level to meet the needs of each implementation. In essence, you start with a security risk analysis. You list your threats, vulnerabilities, and the probability of an attack happening, and what the impact of that attack would be. Then decide on appropriate countermeasures.
2. Thales' security by design strategy continues with implementing a solid foundation of trusted digital device IDs and credentials, securely stored in the roots of devices, during manufacturing. Trusted credentials defend against device cloning, data tampering, theft, or misuse.

The End

Please feel free to contact me anytime if you have any questions



جامعة القادسية
EVERYONE'S SMART UNIVERSITY