INTRODUCTION

The coming year will bring large-scale IoT security breaches, with fleet management, retail, manufacturing, and government at the biggest risk, according to experts. The systems constructed and programmed as closed entities are now exposed to possible attacks. According to **Gartner**, more than half of major new business processes and systems will include an IoT component by 2020. 80% of **IoT apps** are not tested for vulnerabilities. **OWASP** encourages and assists manufacturers to build devices while following safe security practice in order to avoid facing the many threats the IT industry has faced in the past. This whitepaper recaps, simplifies, and explains the advice and tips from OWASP followed by descriptions and recommended actions.

INSECURE WEB INTERFACE

A web interface is defined as the control panel in between users and software running on a web server. Web interfaces are popular as they are easily accessible from any computer running on any operating system, adding to that web interfaces are simple to build and/or modify. Most household devices used today may communicate with the internet through some kind of web interface. For example, your home internet router web interface is accessible using a default IP defined by the vendor, requiring a username and password to enter and control your settings. While discussing IoT, in many cases, control and configuration may be required. Considering the ease of implementing a web interface for devices which are connected to the net, it's safe to assume that most IoT devices are already using an interface of sort. As web interfaces have been in use for a long time, one would expect them to have been built with security in mind. Vendors in the software industry have been struggling to keep access to web interfaces secure, even though many solutions have been introduced to protect access to such sensitive controls.

For example, let's use home security systems:
You may already be in the office and it suddenly occurs that you forgot to turn on your home alarm system. You then quickly log into your alarm's web interface and remotely turn your alarm on. But who's to say that if you can turn it on remotely, you can't turn it back off and perhaps even unlock doors in smart homes remotely?

Here are a few crucial things to look out for in order to prevent such scenarios

1. Never continue using the default passwords. You should perform a password change upon installation.
2. Prevent brute forcing. Enforce an attempt limit which will lock your account after X number of failed attempts. Be sure to have a reset procedure which requires direct (non-web) interaction with your device or application.
3. Verify that your web application code is protected from XSS, CSRF, SQL injections, and other known vulnerabilities.
4. Store credentials securely and make sure to not expose them over network traffic.
5. Use modern encryption techniques and don't settle for less than the latest encryption levels.

INSUFFICIENT AUTHENTICATION/AUTHORIZATION

The Starbucks data breach in May 2015 is an example of a breach where user credentials were stolen (phished). Hackers also were able to guess reused or basic passwords such as "password", the ultimate numerical combo "123456", or the ever so complicated "qawsedrf". Once the hackers had their hands on the passwords, the user was exposed and could do very little to prevent funds from being stolen from his/her account.

There are ways to deal with phishing and credential theft which are quite straightforward and simple:

- Enforce strong passwords. Passwords should include upper and lower case letters, numbers, and symbols. They should not be shorter than six characters.
- Create user profiles and limit user permissions based on their access credentials.
- Implement a two-factor authentication.
- Create a secure password recovery function and process.
- Force password expiration dates.
- Do not allow the use of default passwords.

INSECURE NETWORK SERVICES

Firewalls, intrusion detection systems, and web application firewalls are all used in most enterprise network security portfolios. They are all the fences which protect outsiders from entering and snooping around. But how does the IoT apply to this landscape? Well, the fact that your refrigerator is using the network does not mean that the network itself has to adapt. As the network and the attack vectors stay the same, it is important to implement the same network security measures and solutions. Make sure that no one can access your Things and may make them unresponsive using a Denial of Service (DoS) attack or may attack them with buffer overflow and fuzzing.

There are ways to deal with phishing and credential theft which are quite straightforward and simple:

1. Make sure your application code is not exposed to such attacks. Good static analysis security testing solutions should be able to detect such potential attacks
2. Ports which are not in use should not be open nor accessible

LACK OF TRANSPORT ENCRYPTION

Do you remember when you were a kid, and wanted to pass secrets to your friends without anyone else understanding them? One method used by kids is whispering. Another method is creating a unique code language - this is also known as encryption.
Encryption is a sophisticated coding language with different levels of complexity. It is recommended to use the most complex encryption level available. Your devices may be communicating with each other, and this you don't want to necessarily expose. For example, a car manufacturer may be using a 3D printer to print initial prototypes of a new car design, and by doing so, the printer communicates with the designer's PC by receiving information over a network. If this information is being passed from device to device in clear text, anyone who is able to access the network meaning that the communication can be grabbed, immediately understood and even reused for a different purpose.

There are ways to deal with phishing and credential theft which are quite straightforward and simple:

PRIVACY CONCERNS

Just a few weeks ago, a U.S. government office was attacked again and it is said that millions of past and present employees' private data had been stolen (allegedly by Chinese hackers). No one really knows how this happened, however, it's quite clear that data was not stored securely enough. And in some cases, the data may have been unnecessarily kept in storage. PII (Personal Identifiable Information) Data is probably one of the most valuable and sensitive assets and organization can store. Identity theft is on the rise and as more devices are exposed to the net, the more dangerous it becomes to store data. Does a smart fridge need to know the users' detailed personal information for it to order milk?

What to look out for:

1. Use the latest and greatest encryption techniques for communication between “things” and the web
2. Don’t store data that you don’t need
3. Encrypt all stored data at rest and transport
4. Anonymize data where possible

INSECURE CLOUD INTERFACE

The Cloud, making life easy by storing all of your data making it available wherever you are. No need to perform data backups or to remember anything. It’s all there, all the time and from anywhere. However the Cloud also introduces a whole new layer of risks. New code means new vulnerabilities which need to be validated and closed. And a new interface means new passwords which need to be validated and enforced. There is more communication and more encryption; in short, the Cloud is great but when you use it with your “thing”, as a manufacturer, you have a responsibility to make sure all used services are on top of your own software.

What to look out for:

1. Validate code vulnerabilities are addressed (XSS, SQLi, CSRF and others)
2. Enforce strong passwords. Passwords should include upper and lower case letters, numbers, and symbols. They should not be shorter than at least six characters
3. Force password expiration dates
4. Apply a two factor authentication
5. Ensure your Cloud systems use transport encryption

INSECURE MOBILE INTERFACE

Mobile device sales have outnumbered computer sales in the past years, and this is probably not going to change soon. Mobile might be one of the most popular web connected “things” out there today. Being both a “thing” and a handheld computer likely containing most of your sensitive data, these devices are considered the ‘crown jewel’ for hackers. Take a look at connected cars; these systems mostly run mobile operating systems and allow access to car controls. While these systems are very useful as they provide important tools such as web access, automatic crash notifications, remote system updates and other services. Additionally, they pose quite a significant risk in the case of a wrong user gaining control of the remote device.
Imagine someone taking control of the car ignition or gas pedal. Here is a video presenting a unpleasant possibility: https://www.youtube.com/watch?v=eN7j90HtRPA and https://www.youtube.com/watch?v=ARrlhlQiFzM
Mobile devices are everywhere. Medical applications used by doctors to communicate with medical equipment and register patients, home appliances, watering systems, air travel media centers and practically any modern industry.

What to look out for:

1. Apps should enforce a high level of password security including a two factor authentication, password expiration, no use of default passwords, high password complexity and account lockout mechanisms
2. Use transport encryption for any communication to avoid eavesdropping and data theft
3. Do not collect any unnecessary data and store required data encrypted and in a secure manner

INSECURE SOFTWARE/FIRMWARE

Once “things” are connected to the web, they will almost always have some kind of software running in the background. This software, like any other, might be exposed to zero day vulnerabilities, malware and other attack techniques. Therefore, you will want to make sure that the software is updated on a regular basis to ensure your device’s protection against new threats.

What to look out for:

1. Applications and software should be written to allow update capability
2. Update files should be processed in an encrypted manner
3. Updates need to be validated before implementation using signed files
4. Audit logs and usage logs should be mandated as part of the application functionality
5. Security event notifications should be available to trigger and alert end users on operations which might introduce

INSUFFICIENT SECURITY CONFIGURABILITY

When configuring a device, it is critical to allow the administrator to enforce strict security regulations. Imagine an industrial engineer setting up a turbine; the turbine has its own software which allows control of the turbine speed and scheduling. These settings can be controlled via a local interface to the turbine software. Would you want an engineer to be able to modify settings without consent of management or the relevant teams? The admin should be able to set and enforce specific security regulations which prevent modifications without proper approval.

What to look out for:

1. Enforce application code allows password security options (two factor authentication, password expiration, no use of default passwords, high password complexity and account lockout mechanisms)
2. Validate applications are written with data encryption options (Enabling AES-256 where AES-128 is the default setting)

POOR PHYSICAL SECURITY

The great thing about “things” is that we use them on a daily basis. That means that many of these connected devices can change physical ownership and can be used by multiple people over time. On top of device usage, there is also the aspect of how accessible a device is and what level of device access is really required. Do you need a USB port on your fridge at home? If so, do you need two USB ports?
Physical access to a device is probably the easiest way to infiltrate and create some kind of damage (depending on the device). This may be considered basic theft over hacking in many cases. For the same reason that most private homes have one or perhaps two entrances at most, you should not allow more than the required physical number of device access channels.

What to look out for:

1. Utilize a minimal number of device access ports (e.g. USB and network ports)
2. Sensitive application functions should not be accessible through USB
3. Consider writing applications to allow local access only (no web access)