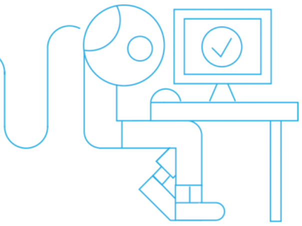
# OWASP Top 10 for IoT - Explained

## Table of Contents

## Introduction

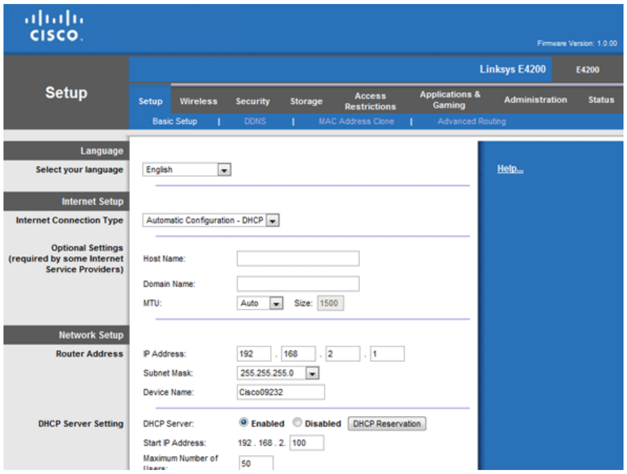
Even though the software industry has been dealing with security issues since the web introduced itself almost 30 years ago, IoT manufacturers who have not had this struggle in the past are now stepping in to a world of pain which they can probably avoid if they use the lessons learned in the past.  Internet of Things is no longer a thing of the past. OWASP have raised the flag to encourage and assist manufacturers to build their devices with security in mind and avoid repeating the same mistakes the IT industry has been dealing with for a few decades.

This document re-caps the recommendations available at OWASP and tries to give it more context and clarification. Each item is followed by a description and the recommended actions.



## Insecure Web Interface

A web interface is defined as a control panel to interact between a user and software running on a web server. Web interfaces are popular as they are easily accessible from any computer running any operating system and are simple to build/modify. Most house hold devices we use today to communicate with the internet have some kind of web interface. For example your home internet router web interface is accessible using a default IP defined by the vendor. You will need a username and password to access and control the settings.



Language
Select your language: English

Internet Setup
Internet Connection Type: Automatic Configuration - DHCP
Optional Settings (required by some Internet Service Providers)
Host Name:
Domain Name:
MTU: Auto Size: 1500

Network Setup
Router Address
IP Address: 192 . 168 . 2 . 1
Subnet Mask: 255.255.255.0
Device Name: Cisco09232

DHCP Server Setting
DHCP Server: ● Enabled ○ Disabled DHCP Reservation
Start IP Address: 192 . 168 . 2. 100
Maximum Number of Users: 50

When talking about 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 will be and already are using an interface of the sort.

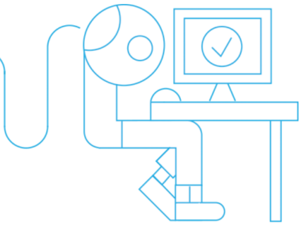
Considering Web interfaces have been in use for a long time, you would expect these to be built with security in mind. Vendors in the software industry have been struggling to keep access to such controls secure and along the years many solutions (some better than others) have been introduced to protect access to sensitive controls.

Let's take home security systems as an example.

You leave home in the morning. You are already at your office when you remember that you didn’t turn the alarm system on. Quickly you log into the web interface remotely and turn it on. Phew, thank god for the internet. But if you can turn it on remotely, who's to say no one else can turn it back off and maybe even unlock the door remotely (if your alarm system is extra smart )?

### What to look out for

1. Never allow use of default passwords – Enforce a password change upon setup of the "Thing"
2. Prevent Brute Forcing – Enforce an attempt limit and lock account down after x number of failed access attempts. Make sure to introduce a reset procedure that requires direct (non web) interaction with the "Thing"
3. Make sure the web application code is not susceptible to vulnerabilities such as XSS, CSRF, SQLi and others
4. Store credentials securely and do 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

Heard of the latest Starbucks breach? It was quite simple. User credentials were stolen (phished) and in other cases hackers were able to guess re-used passwords or basic passwords like the famous "password" or the ultimate numeric "123456" or the ever so complicated "qawsedrf". Once the hacker has their hands on the password the user was exposed and there was not much he could do to prevent funds being stolen from the account.

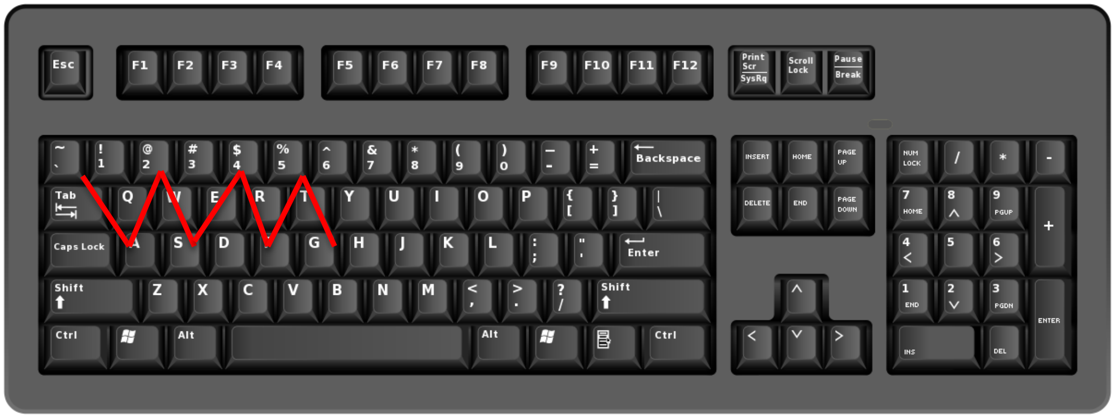

Esc F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12
Print Scr SysRq
Scroll Lock
Pause Break
~ ` ! 1 @ 2 # 3 $ 4 % 5 ^ 6 & 7 * 8 ( 9 ) 0 _ - + = Backspace
INSERT HOME PAGE UP
NUM LOCK / * -
Tab Q W E R T Y U I O P { [ } ] | \
DELETE END PAGE DOWN
7 HOME 8 9 PGUP +
Caps Lock A S D F G H J K L : ; " ' Enter
4 5 6
Shift Z X C V B N M < , > . ? / Shift
1 END 2 3 PGDN ENTER
Ctrl Alt Alt Ctrl
INS DEL

Figure 1 You are not special or original.....

There are ways to deal with phishing and credential theft and they are quite straight forward.

### What to look out for

1. Enforce strong passwords. They should include upper and lower case characters + numeric values and maybe even a symbol. They should definitely not be shorter than six characters.
2. Create user profiles and limit their permissions based on their access credentials
3. Implement two factor Authentication.
4. Write secure password recovery functions and processes
5. Force password expiration dates.
6. Do not allow use of default passwords.

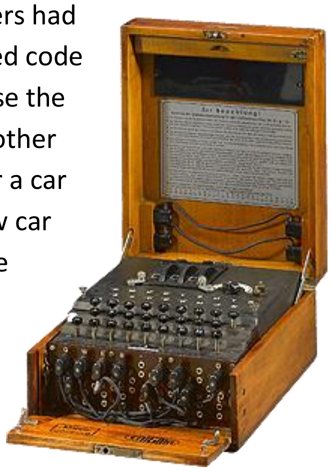
## Insecure Network Services

Network security has been around for a while now. Firewalls, Intrusion Detection Systems and Web Application Firewalls are used in most enterprise network security portfolios. They are the gate that protects outsiders from coming in and snooping around. What about IoT? How do they adapt to the landscape? Well, the fact that your fridge is now using the network does not mean that the network itself has to adapt. The network stays the same and the attack vectors stay the same, therefore its important to implement the same network security measures and solutions. You want to make sure that no one can actually make your "Things" un-responsive using Denial of Service attacks or attacks such as Buffer overflow and fuzzing.

### What to look out for

1. Make sure your application code is not exposed to such attacks. Good static analysis security testing solutions should be able to detect such attack potentials
2. Make sure that ports not in use are not open or accessible.

## Lack of Transport Encryption



What is transport encryption? Remember when we were kids and we wanted to pass secrets without anyone else understanding them? One method was whispering while others had their own code language. Encryption is exactly that, it’s a very sophisticated code language. There are different levels of complexity and its always best to use the most complex one available.  Your devices might be communicating with other devices and passing information you don’t necessarily want to expose. For a car manufacturer might be using a 3D printer to print initial prototypes of new car designs. The printer communicates with  the designers PC and receives the information over the network. If this information is passed in clear text, anyone who was able to access the network can grab the data and immediately understand or even re-use it for their own purpose.

### What to look out for

1. Use the latest and greatest encryption techniques for communication between "Things" and the web.

## 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 has 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 maybe data was unnecessarily kept in storage. Do you 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 the more devices are exposed to the net the more dangerous it becomes to store data.

Does a smart fridge need to know users' social security ID for it to be able to order milk**?**

### What to look out for

1. Don’t store data that you don’t need
2. Encrypt all stored data at rest and transport
3. Anonymize data where possible

## Insecure Cloud Interface

Ah, the "Cloud", how wonderful life is with all your data kept available wherever you go. No need to backup or 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. New interface means new passwords that need to be validates and enforced. 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 that all services you use on top of your own software.

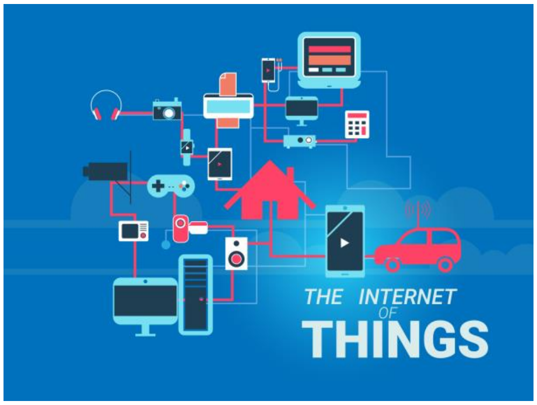
### What to look out for

1. Validate code vulnerabilities are addressed (XSS, SQLi, CSRF and others)
2. Enforce strong passwords. They should include upper and lower case characters + numeric values and maybe even a symbol. They should definitely not be shorter than six characters.
3. Force password expiration dates.
4. As for two factor Authentication.
5. Ensure cloud systems use transport encryption.

## Insecure Mobile Interface

Mobile devices sales have outnumbered computer sales during the past 3 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 holding probably most of your sensitive data, these devices are considered as 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 to provide important tools like web access, automatic crash notifications, remote system updates and other services, they also pose quite a significant risk in case the wrong user has taken control of the remote device. Imagine someone taking control of the car ignition or the car gas pedal. Not very pleasant but very possible: https://www.youtube.com/watch?v=eN7j90HtRPA



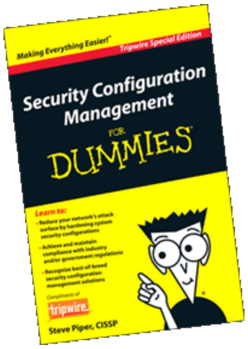
Mobile devices are everywhere. Medical applications used by doctors to communicate with medical equipment and register patients, home appliances, watering systems, air travel media centers and practically any modern industry.

### What to look out for

1. Apps should enforce high level of password security including 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 eaves dropping and data theft.
3. Do not collect any unnecessarry data and store required data encrypted and in a secure manner.

## 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 to 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

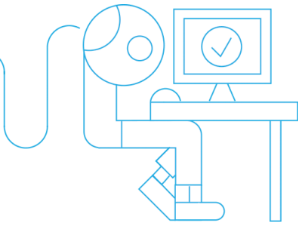
1. Enforce Application code allows password security options (two factor authentication, password expiration, no use of default passwords, high password complexity and account lockout mechanisms)
2. Validate applications are written with data encryption options (Enabling AES-256 where AES-128 is the default setting)
3. Audit logs and usage logs should be mandated as part of the application functionality
4. Security event notifications should be available to trigger and alert end users on operations which might introduce risks.

## 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 make sure new threats are protected against.

### What to look out for

1. Application/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 implemented using signed files.



## Poor Physical Security

The great thing about "things" is that we use them on a daily basis. That means that many of these 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). I wouldn't even call it hacking but rather basic theft. For the same reason most private homes have one or maybe two entrances at max, 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 application to allow local access only (no web access)