National College of Ireland

Pen Testing Framework for IoT Devices

MSc Research Project
Cyber Security

David Collins
Student ID: X09106081

School of Computing
National College of Ireland

Supervisor: Dr Paul Stynes, Dr Vanessa Ayala-Rivera

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	David Collins
Student ID:	X09106081
Programme:	Cyber Security
Year:	2021
Module:	MSc Research Project
Supervisor:	Dr Paul Stynes, Dr Vanessa Ayala-Rivera
Submission Due Date:	23/09/2021
Project Title:	Pen Testing Framework for IoT Devices
Word Count:	5360
Page Count:	19

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	*David Collins*
Date:	23rd September 2021

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☑
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☑
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☑

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Pen Testing Framework for IoT Devices

David Collins
X09106081

Abstract

Current Pen-Testing frameworks focus on locating defects in the design of Web Applications and Network servers, identifying the most likely attack vectors, and introduce mitigation strategies to minimise their damage potential. The challenge is to adapt a Pen-Testing framework that would be suitable for use with IoT devices. This research proposes to create a Pen-Testing Framework that would be aimed at IoT devices which due to memory constraints and smaller processing power would need to be handled differently in a penetration test. The Pen-Testing Framework was applied to a network of ESP32 Microcontrollers in an IoT project used to gathered temperature and humidity data from a sensor, check the state of a push-button switch and monitor the output from a Microcontroller. This is a typical IoT project with basic coding standards and containing minimal security considerations. The Pen-Testing framework will be used to evaluate the concerns outlined in the literature review paying particular attention to the evolving regulatory standards and guidelines.

1 Introduction

The IoT ecosystem is rapidly evolving and presenting security challenges that extend beyond traditional networking considerations. In this new reality, cheap components are being used in a cyberwar by criminals targeting a multi-billion-dollar industry. The danger posed by malware such as Mirai in bringing down major sites including CNN, Twitter, Netflix was a wake-up call. What was unusual about this attack was the ease with which the worm was capable of infecting such a large and diverse range of devices resulting in an astounding 1 terabit per second of traffic to bring down its targets (Gallopeni et al.; 2020). This research aims to investigate if the existing Pen-Testing framework is suitable for use with IoT devices. The exponential growth of IoT whereby in 2025 an estimated 75 billion of these ultra-low-cost devices will be connected to the internet coupled with the lack of regulations in this area has not encouraged the introduction of the much-needed secure software design principles. This may change with the introduction of The IoT Cybersecurity Improvement Act of 2020 (U.S. House. 116th Congress; 2020) in the US where the National Institute of Standards and Technology (NIST) has been mandated to identify and create best practice requirements that must be followed before federal U. S Government can purchase any IoT products in the future. One year after Mirai, The European Union Agency for Cybersecurity (enisa) produced a report Baseline Security Recommendations for IoT (ENISA; 2017) one of its recommendations was to raise awareness for the need for IoT security. The report highlights the fragmentation and slow adoption of regulation in Europe, due in part to the continuous emergence of new

technologies. The EU Cybersecurity Act 2019/881 (Council of European Union; 2019) has established a certification framework influencing the creation of standards through education and training, this recognizes the difficulties involved in the regulation of this industry. While Pen-Testing frameworks do exist today for Web and Networks Systems, are they suitable for use with constrained IoT devices? To address this research question the following specific sets of research objectives were derived

- Investigate the state of the art around current Pen-Testing Frameworks for Web and Networks.
- Design a suitable framework for testing IoT devices.
- Implement a Penetration Test around a Network of ESP 32 Microcontrollers using the framework.
- Evaluate the effectiveness of the framework to satisfy the evolving regulations.

This paper discusses related work in Section 2 with a literature review that has a focus on comparing the current Pen-Testing Framework suitability for use with IoT and highlighting the security concerns of experts in this domain. Section 3 outlines the research methodology that will be used in this research. Section 4 provides detail of the design of a Pen-Testing Framework that will cater for the proposed legal changes. Section 5 describes a practical implementation of the Pen-Testing framework. Section 6 provides an analysis of the Pen-Testing Framework and evaluates its effectiveness; Section 7 concludes the research and discusses future work.

2 Related Work

A state-of-the-art Pen-Testing Framework for IoT should address the security risks associated with these devices. The literature review examines the strengths and weaknesses of the top five Pen-Testing frameworks available today for Web and Networks and considers their suitability for use with IoT devices to ensure compliance with the latest regulations and guidelines. The Frameworks reviewed were The Penetration Testing Execution Standard (PTES) which focuses on the exploitation of potentially vulnerable areas. The Open Web Application Security Project (OWASP) identifies flaws and vulnerabilities due to unsafe practices. The NIST SP-800-115 framework offers more specific guidance on guaranteeing Information security. The Open-Source Security Testing Methodology Manual (OSSTMM) provides a framework created to support network teams in securing firewalls. The Information System Security Assessment Framework (ISSAF) placed greater emphasis on the planning and documentation of the penetration test. The literature review also explored Threat Modelling as it relates to the Pen-Testing framework and security concerns raised in academic papers related to IoT devices were explored.

2.1 Penetration Test Frameworks for Web and Networks

The IoT industry is rapidly evolving and converging with new technologies, fog and edge computing, big data, blockchain, embedded systems, Artificial Intelligence (AI) and Machine Learning (ML), According to a report published by (Boeckl et al.; 2019), there are several differences between traditional IT and IoT devices in terms of security

and privacy. IoT devices due to their constrained nature have less memory and smaller processing power and will interact differently with the physical world. These devices cannot be managed or monitored in the same way, additional controls are required to improve Cybersecurity. The report highlights the three high-level risk mitigation goals which also apply to the traditional IT systems: 1)Protect device security: 2)Protect data security and 3)Protect individuals privacy. Some of the challenges for IoT in meeting these goals include not having a unique identifier to access an asset management system, lacking the capability to report on their status or to detect intrusion into the system, update patches may not be readily available from the manufacturer or be easily uploaded, the lack of memory may prevent logging and monitoring, the concealment of credentials may not be possible, restricted access to some devices due to their physical location could pose a challenge. The goal to protect data may be hampered by a lack of encryption mechanisms on the device or an inadequate access control mechanism in place. The goal to protect individual privacy could be hindered without a proper interface to provide a privacy notice, register consent or provide redress to the user. The IoT device may gather and share PII information in an unsecured fashion that would be contrary to current standards and policies.

2.1.1 Penetration Testing Execution Standard (PTES) (Nickerson; 2021)

The PTES framework does not provide any detailed guidelines on how to perform a penetration test but it can act as a blueprint that can be adapted for use with IoT devices. The standard consists of seven sections as the basis for executing the test, Pre-engagement, Intelligence Gathering, Threat Modelling, Vulnerability Analysis, Exploitation, Post Exploitation, and Reporting. A technical guide containing a list of tools and software was produced to assist with carrying out this standard. The latest official version was produced in 2012 while version 2.0 is still under development according to the unofficial document Release 1.1 dated June 2021 (Nickerson; 2021). The Pre-engagement interactions of the standard define the scope, the metrics for the penetration test, and the rules of engagement unique to each test. This would be considered best practice to avoid scope creep. The standard does not specify a threat model to be used it just requires it to be consistent and reproducible so when applied to future tests the same results would be observed. Motivation Modelling is a recommendation in the consideration of threat analysis. Vulnerability Analysis, Exploitation and Post-exploitation sections of the standard provide a high-level consideration that would require specific adaptation for use with IoT.

2.1.2 Open Web Application Security Project (OWASP) (Wichers and Williams; 2017)

The OWASP Top Ten vulnerabilities framework explores the different paths that a hacker can take to compromise a Web Application or a Network, it was designed to protect a single application but can be adapted to provide the same level of protection for the IoT industry. The OWASP Internet of Things project published a top 10 list in 2018 intending to guide on security issues that should be avoided while building, deploying or managing an IoT system (Miessler; 2015), unlike the Web Application project this list does not provide any detail of the vulnerabilities which should form part of a penetration test but it can be used as a high-level guide to illustrate the type of attack scenarios which an attacker can use. The OWASP firmware security testing methodology (Arron; 2019) provides a methodology that would assist in the security assessment. It is composed of

nine stages. 1)Information gathering of all documentation of the device firmware. 2)Obtain the firmware. 3)Analyse the firmware. 4)Extract the firmware. 5)Analyse the file system contents for hardcoded credentials. 6)Emulate the firmware. 7)Dynamic Analysis. 8)Runtime Analysis. and 9)Binary Exploitation of any identified vulnerabilities.

2.1.3 NIST SP 800-115 Framework (Scarfone et al.; 2008)

The technical guide to information security testing and assessment was created by NIST to satisfy the requirements of the Federal Information Security Act (FISMA) of 2002 (U.S. House. 107th Congress; 2002). The guideline is used to help design and implement a penetration test to find vulnerabilities in a system and to verify compliance with policies and standards. According to (Scarfone et al.; 2008) the Informational security assessment methodology presented consists of four phases. The planning phase is where the rules of engagement are agreed and permission is given to protect the penetration tester from legal action. The discovery phase covers information gathering and vulnerability analysis by comparing discovered services against a database of known vulnerabilities. The execution phase checks the effectiveness of any control mitigations that may be in place and the final phase Post-Execution will report back on the findings and provides recommendations in improving the security of the system. This guide has a focus on Network Security however with the limited resources available to IoT devices and it is distributed nature this framework would require some modification to make it a suitable penetration testing framework for IoT.

2.1.4 Open Source Security Testing Methodology Manual(OSSTMM) (Barcel et al.; 2000)

The OSSTMM provides a methodology for a security audit that encompasses: Human, Wireless, Data networks, Cloud, Virtual machines, Mobile, Messaging protocols, It is a way that an organization can ensure compliance with regulations such as PCI-DSS, ISO/IEC 27001:2005, ISO/IEC 27002:2005, and ISO/IEC 27005:2008 it can serve as proof that the test has been carried out by providing comprehensive understandable verifiable metrics in its documentation and it would hold the certified analyst responsible for the test. The rules of engagement define the acceptable practices in the performance of the test and the handling of the results. Guidance is provided on five channels to determine how well the security processes work. The five channels are 1)Human security, 2)Physical security, 3)Wireless communications, 4)Telecommunications and 5)Data networks. The document provides us with the following axioms which hold with IoT (Barcel et al.; 2000).

- $There\ is\ no\ such\ thing\ as\ 100\%\ secure.$
- $Even\ if\ secure,\ if\ an\ attacker\ wants\ in\ badly\ enough,\ they\ will\ get\ in.$

The ability to ensure compliance with an agreed standard is a feature of this framework worth considering for IoT as the legal regulatory changes are evolving.

2.1.5 Information System Security Assessment Framework (ISSAF) (Rathore et al.; 2006)

The ISSAF framework follows a similar approach however it is not an active community the latest standard draft 0.2.1B was produced in 2006 and does not cater for the advances

that have occurred in network technology, it does however provide comprehensive testing guidelines which can still be used as reference material in carrying out a penetration test. The primary purpose of this framework was to evaluate the security of Networks systems and their application controls. It has three action areas Planning and preparation where the testing environment is set up, the Assessment stage to guide the tester into performing the test and the Reporting and clean-up stage where the results are communicated (Rathore et al.; 2006). This is the common thread in all the frameworks researched and would be considered best practice.

2.2 Threat Modelling

Application threat modelling is an important consideration for any Pen-Testing framework. It includes a description of potential threats to the system, a list of actions that would be required to balance these risks and ensure a secure design. NIST SP 800-30 provides a simple risk model example that best illustrates this approach. It follows five steps, 1) Decomposing the application to examine the code, identifying any recognized patterns of bad security practices, 2) Defining the assets to properly classify its data following its expected level of confidentiality. It is only by understanding the legal consequences of disclosure or understanding the technical implication for the loss of integrity or understanding how the loss of availability of the IoT sensor data can the risk be fully comprehended, 3) Exploring potential vulnerabilities that could allow a hacker access to the system or data, 4) Exploring potential threats affecting the Confidentiality, Integrity, or Availability of the resources or data, 5) Creating a mitigation strategy to minimize the impact of a data or system breach (Stoneburner et al.; 2002). OWASP poses the following questions to explain its threat modelling methodology. What are we building? This question would help to define the scope of the threat model. What can go wrong? This question identifies the main threats to the application. What are we going to do about it? Suggests that use case and abuse cases modelling has a role in the identification of the risks and the potential security controls that can be put in place to offset the business impact of the vulnerabilities (Kingthorn et al.; 2019). In June 2021 The Australian Government announced applying a risk-based approach to Cybersecurity its guidelines are described in the Australian Government Information Security Manual (ACSC; 2021) which is drawn from NIST SP 800-37 Rev 2 (NIST; 2018).

2.3 Security concerns identified for IoT devices

The UK Department for Culture Media and sports commissioned the PETRAS IoT research hub a group of 9 universities to provide recommendations on the development of IoT security (Tanczer et al.; 2018). (Blythe et al.; 2019) expanded on that research to explore the potential risks and security threats caused by IoT, both reports highlighted the following risks that need to be addressed. 1) Default passwords should not be used. 2) Software should be capable of being updated. 3) Software Integrity should be maintained. 4) Credentials should be securely stored. 5) All Communications should be encrypted and 6) All Input Data should be verified. They concluded that manufacturers do not provide adequate information on the security features of many IoT devices and this would be the major difficulty in standardizing a Pen-Testing Framework. A paper IoT-PEN: A penetration testing framework for IoT (Yadav et al.; 2020) documents some of the processes that could lead to code injection. It outlines some countermeasures

that could help, the use of Safe language to replace the unsafe function calls with more secure ones, using static source code analysers to find potential vulnerabilities. This framework uses directed graph techniques to explore only the paths that can be used to breach a targeted system. An examination of the National Vulnerability Database for the term IoT indicates the exponential growth of vulnerabilities targeting IoT devices over the last decade. The results of the search can be seen in Figure 1. The increase in IoT vulnerabilities has prompted bug bounty programs to pay out for the discovery of new vulnerabilities (NICULA and ZOTA; 2020). The IoT paradigm is broader than

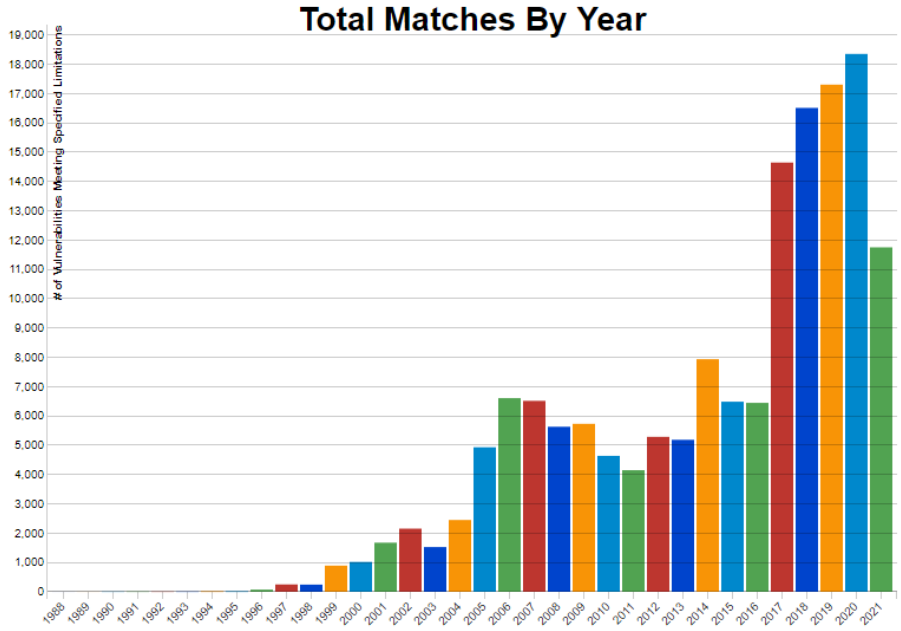


Figure 1: Growth of Reported IoT vulnerabilities

just IoT devices also include mobile devices, network connection fog and edge computing integration with the latest technologies data analytics, blockchain and AI, this will pose significant challenges in interrogating existing databases for the existence of a known vulnerability in an IoT system (Rytel et al.; 2020) highlights the need for publicly available structured information on IoT specific vulnerabilities in its conclusion it notes that the creation of an IoT-orientated database is in progress as part of the EU project VARIoT.

2.4 Evolving regulations for IoT Industry

In the baseline security report page 12 (ENISA; 2017), IoT has been defined as "a cyber-physical ecosystem of interconnected sensors and actuators, which enable intelligent decision making", at the centre of the European thinking on IoT is that information is critical for the decision making process and protection of personal data is high on its political agenda. The lack of skilled IoT cyber security personnel with the overriding time to market pressure exists for these IoT products. The challenge in defining horizontal baseline security measures to satisfy the emerging European regulations involves carrying out a risk assessment across critical information infrastructure with data classification critical to ensure GDPR compliance (Council of European Union; 2016) the good

practices that should be implemented are broken down by Security by design and Privacy by design. NISTIR 8259A (Fagan, Fagan, Megas, Scarfone and Smith; 2020) cross-reference the similar approaches taken by the two jurisdictions. In the United States, the NISTIR range of documents currently in draft guides manufacturers in complying with the new regulations. NISTIR 8259A outlines the 6 IoT Device security features that a customer should be aware exists before purchase by the federal government they are identified as 1) Device Identification: each IoT device should have a unique address. 2) Device Configuration: only authorized personnel can change software configuration. 3) Data Protection: encryption is present to protect data at rest or in transit. 4) Logical Access to Interfaces: access should be limited to authenticated users. 5) Software update mechanisms: is provided 6) Cybersecurity state awareness: logging and monitoring are in place to identify incident detection. NISTIR 8259B (Fagan, Marron, Brady, Cuthill and Herold; 2020) provides some additional guidance on the non-technical support required such as training and documentation. NISTIR 8259C (Fagan, Marron, Brady, Cuthill and Herold; 2020) provides transparency on how NIST will create the profiles for defining the IoT capabilities requirements which will be outlined in NISTIR 8259D (Fagan, Marron, Brady, Cuthill, Megas and Herold; 2020b). In Australia, the Australian Cyber Security Centre (ACSC) has produced an IoT Code of practice: Guidance for manufacturers The guidance provides useful examples of good and bad implementation of its thirteen principles which can be found here (ACSC; 2020). In addition to the previously mentioned requirements the following principles are recommended, Implement a vulnerability disclosure policy. Make systems resilient to outages, Monitor system telemetry data, Make it easier for consumers to delete personal data.

The European and U.S. requirements can be considered as a baseline a minimum standard to be implemented. A Pen-Testing Framework for IoT should incorporate the best practices from the other Frameworks and ensure that the baseline legal requirements are met.

3 Methodology

The network tested is outlined in Figure 2 It consisted of an IoT device the ESP-32 Wi-Fi module with a temperature/humidity sensor supplying the data. The choice of components was dictated by the popularity of this device and the quality of documentation provided by the manufacture The penetration test will be carried out using software that was installed on a Raspberry Pi 4 running Kali Linux. The use of the ESP32 as a software access point is intended to reduce noise in the network traffic, this traffic will be captured using WireShark and examined for data exposure. A command-line utility esptool will be installed on Kali Linux to allow querying of the chip's identity and firmware. The IoT project will gather temperature humidity and pressure data from a sensor, uses an existing push-button switch, and monitor outputs that can toggle Light-emitting diode LEDs. The data will transfer to a router using two protocols HTTP and MQTT both will be observed from suitable client's. The set-up resembles the project reported at the International Scientific-Practical Conference (Barybin et al.; 2019).

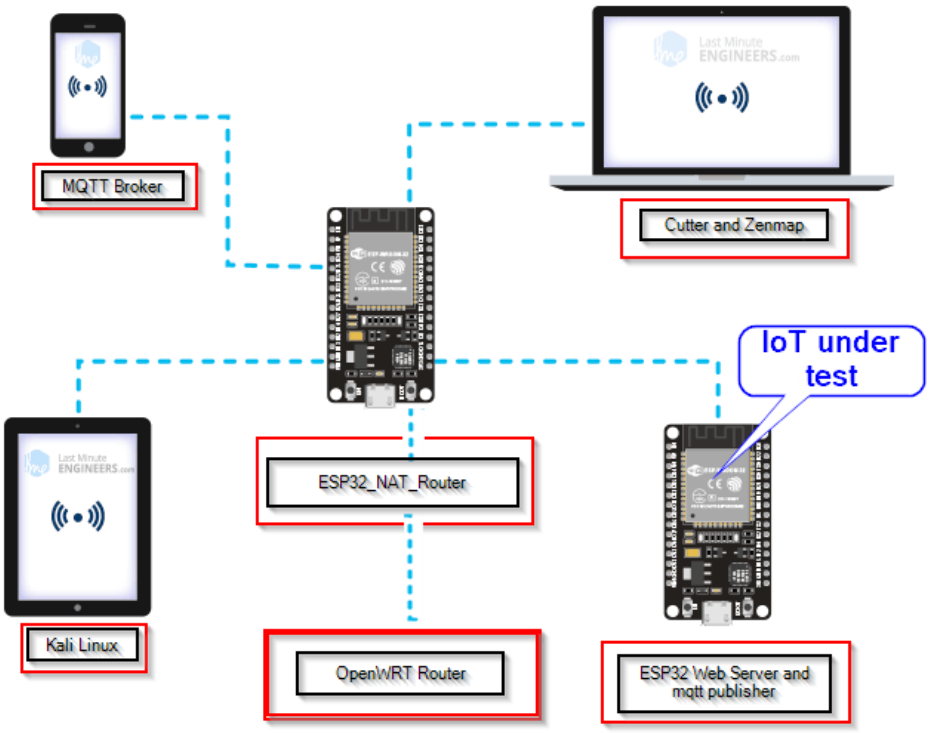


Figure 2: Lab Setup

4 Design Specification

In constructing a suitable Pen-Testing Framework for IoT the following concerns raised in the literature review were addressed. Both Black box and White Box approaches were used. For the Black box approach knowledge of the firmware is irrelevant and the testing is done on the system as a whole. For the White box approach, the firmware is examined for bad coding practice.

- 1 Device Identification should be unique. Test to reveal manufacturer and chip id.
- 2 Ensure firmware has not been tampered with by comparing hashes or comparing firmware against a local copy.
- 3 Software Analysis on firmware, perform static analysis to check for insecure coding practices.
- 4 Check Router Security for encryption protocols in use WEP, WPS, WPA. WPA2, WPA3. Check for use of default credentials.
- 5 Network scan for open/insecure ports.
- 6 Check network traffic for data exposure using WireShark.
- 7 Check if the facility is in place to update the firmware.
- 8 Has the manufacturer provided version control of documentation?
- 9 Has Cybersecurity state awareness been implemented?

5 Implementation

5.1 Available tools for carrying out Penetration Test IoT Devices

Tooling List		
Tool	ref	Description
esptool	[1]	Tool to interact with the ESP components
PlatformIO	[2]	Extension for Visual Studio Code
Cutter	[3]	Reverse Engineering Framework binary analysis
nmap	[4]	Network security analyser
RouterSploit	[5]	An open-source framework for embedded devices
ACRYLIC WiFi	[6]	WiFi Scanner
Wifite	[7]	Wireless Audit Tool
Wireshark	[8]	Network package capture tool

Table 1: Penetration Testing Tool list

Table 1 contains a list of software used in the implementation of the evaluation in section 6. Coding of the ESP32 Micro Controller was carried out using Visual Studio Code with the PlatformIO extension on Kali. [1] [2] [3] [4] [5] [6] [7] [8]

6 Evaluation

The Pen Test framework will be used to carry out the following tests on the laboratory set-up. The experiments carried out are listed in the design specification and reflect the evolving regulations and guidelines from Europe and the US.

6.1 Identification of IoT device

The information collected for Table 2 was obtained from using the following esptool commands.

python3 esptool.py chip_id : *python3 esptool.py flash_id.*

The features reported from the chip are A) WiFi, BT, Dual Core, 240MHz, VRef calibration in efuse, Coding Scheme None. B) WiFi C) WiFi, ADC and Temp sensor calibration in BLK2 of efuse. The Identity information retrieved reveals details of the Manufacturer and the chip id it also reports on the unique MAC address assigned by the manufacturer for this device.

[1] [1] https://github.com/espressif/esptool.
[2] [2] https://platformio.org/.
[3] [3] https://cutter.re/.
[4] [4] https://nmap.org/download.html.
[5] [5] https://github.com/threat9/routersploit.
[6] [6] https://www.acrylicwifi.com/en/wlan-wifi-wireless-network-software-tools/wlan-scanner-acrylic-wifi-free/.
[7] [7] https://tools.kali.org/wireless-attacks/wifite.
[8] [8] https://www.wireshark.org/.

Chip Type	MAC	MFR	Device	Features
ESP32-DOWD R1	30:ae:a4:9d:6f:5c	20	4016	A
ESP32-DOWDQ6 R1	84:44:a8:5f:87:64	20	4016	A
ESP32-DOWDQ6 R1 (CAM)	08:3a:f2:a9:15:bc	d8	4016	A
ESP32 DOWDQ6 V3	08:3a:f2:43:05:d4	20	4016	A
ESP32 DOWDQR V3	3e:61:05:0d:06:d4	68	4016	A
ESP32-C3-REV2	30:cd:a1:43:08:88	20	4016	B
ESP32-S2	7c:df:a1:08:d8:a2	20	4016	C
ESP8266EX	3c:61:05:d3:17:cb	d8	4016	B
ESP8266EX	d8:bf:co:ff:69:04	20	4016	B
ESP8266EX	e0:98:06:0f:16:c6	d8	4016	B

Table 2: IoT Device Identification

6.2 Ensure Firmware has not been Tampered with

The firmware for the ESP32-NAT-Router used in this experiment was installed using the ESP32 Download tool v 3.8.8. The application contains three binary files and all were flashed to an ESP32 Microcontroller. A penetration test should ensure that the firmware was not altered in any way one method to achieve this is with the esptool. The command can compare the firmware at a specific location in memory to a local file.

$$python3 - m \quad esptool.py \quad verify - flash$$

Figure 4 shows the result of the command verifying that the binary matches a file held locally another method would be to compare the hash of the firmware against a previously recorded value. The cutter tool would be able to retrieve the binary file hashes.

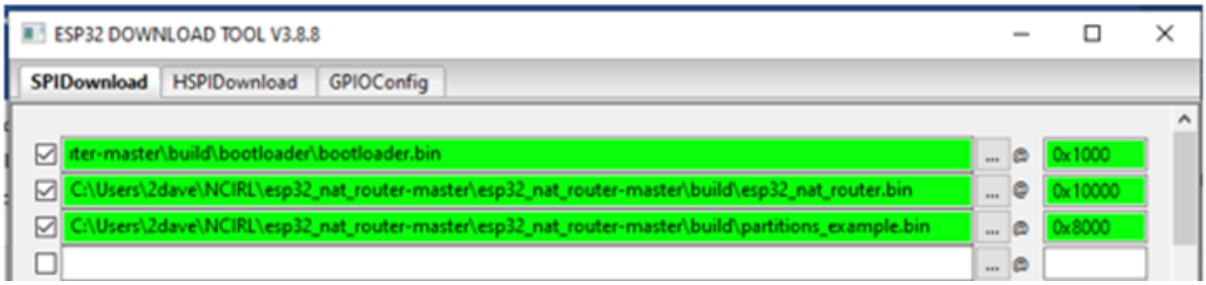


Figure 3: binary file locations

6.3 Software Analysis on firmware using Static Code Analysis tools

Visual Studio Code with the PlatformIO extension is a cross-platform solution for programming the next generation of IoT devices it provides C/C++ smart code linting and static code analysis with the capability to detect potential NULL pointer dereferences, out of bounds indexing of arrays, suspicious assignment among others. The choice of tool

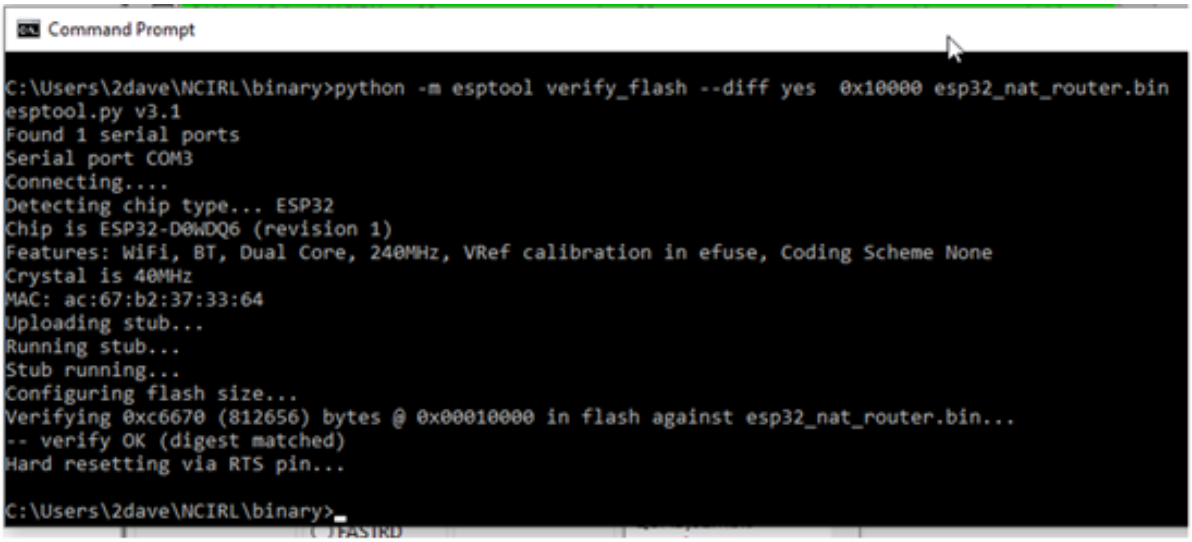


Figure 4: Verify binary file same as local

and the check options are selected in the PlatformIO configuration file in this test a linting tool clang-Tidy was selected Figure 5 describes the options available and how they can be configured. PlatformIO can inspect the memory of the firmware provide an analysis

Perform software analysis on the firmware	Platformio.ini configuration file
Options for the check tools are • Cppcheck • Clang-Tidy • PVS-Studio In this test Clang-Tidy was chosen this is a C++ Linter tool that can diagnose programming errors and style violations. https://clang.llvm.org/extra/clang-tidy/	[env:esp32dev] platform = espressif32 board = esp32dev framework = arduino lib_deps = ottowinter/ESPAsyncWebServer-esphome@^1.2.7 ottowinter/AsyncTCP-esphome@^1.2.1 adafruit/Adafruit Unified Sensor @ ^1.1.4 adafruit/DHT sensor library@^1.4.2 ayushsharma82/AsyncElegantOTA @ ^2.2.5 knolleary/PubSubClient@^2.8 monitor_speed = 115200 check_tool = clangtidy

Figure 5: Setup PlatformIO Code analysis

of the code display a report which can be used by developers in a secure software design methodology to improve the code quality. In this experiment, an ESP32 Microcontroller is programmed as a web server using the SPI Flash File System (SPIFFS) for generating the HTML CSS and JS. The binary file creates an application that gathers Temperature and Humidity from a DHT sensor and allows an asynchronous web server to display this data on a web page. It also publishes this data to an MQTT broker to demonstrate the security of multiple protocols. The results of the code analysis are shown in Figure 6.

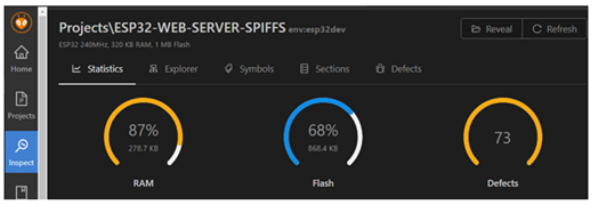
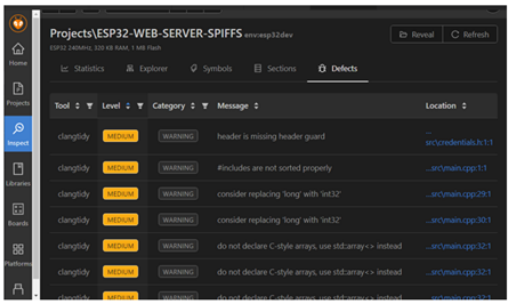
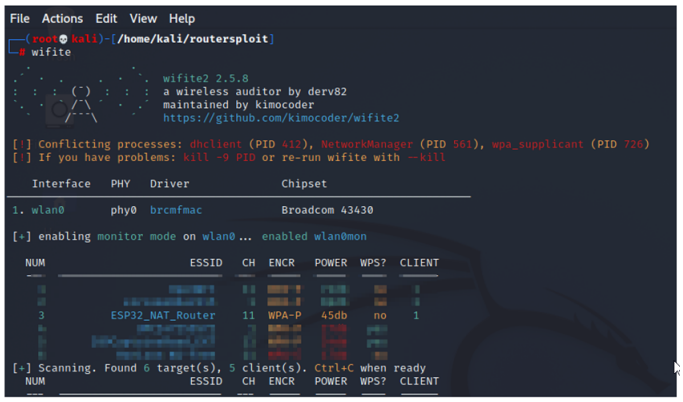
PlatformIO Inspect tool Results	Report using Clang-Tidy
	
Statistical result of Static code and memory analysis preformed on ESP32 firmware	**Report from Static Code Analysis 73 issues found in the firmware.**

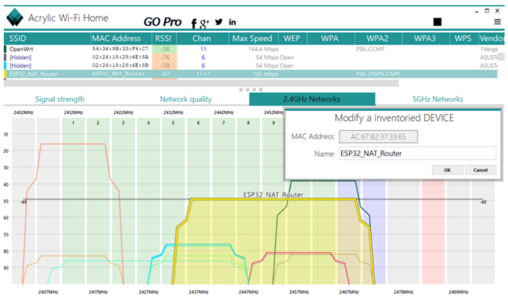
Figure 6: Results of Static Code Analysis

6.4 Check Router Security

For IoT security, it is necessary to check the security of the router. In this experiment, an ESP32 Microcontroller was used to set up a NAT network. The ESP32 supports the following protocols WEP, WPS, WPA. WPA2,WPA3. Both Acrylic and Wifite were able to detect the Security protocol Table 4 shows the results. Both programs reported similar results Encryption protocol used: WPA-P. Power output: approximately -46dbm. MAC Address: AC:67:82:37:33:65. SSID: ESP32 NAT Router Channels: 11. RouterSploit was

Table 3: Router check using Wifile, Acrylic





run on the ESP32 Router and a vulnerable Router. This program can check routers and other IoT devices for default credentials and other vulnerabilities. In Figure 7 RouterSploit could not verify the vulnerability and more tests would be required to check for issues. Figure 8 is an example of a confirmed vulnerable device with cmd access.

6.5 Network Scan

This experiment is using an ESP32 HTTP server on the right of Table 4 which is displaying data from a sensor and providing a mechanism to switch on or off a led. Node-Red is used on the left to display the same data that is transmitted using the MQTT protocol on port 1883. Data can be sent from injectable buttons to send messages back to the

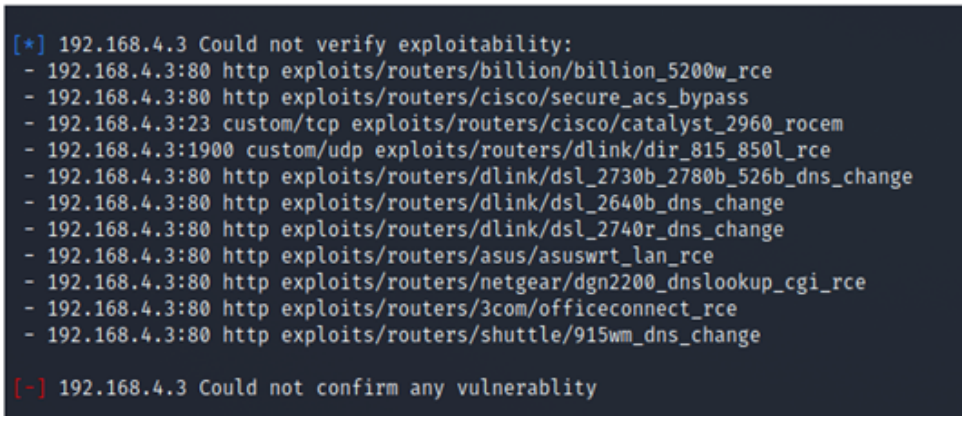


Figure 7: RouterSploit result for ESP32

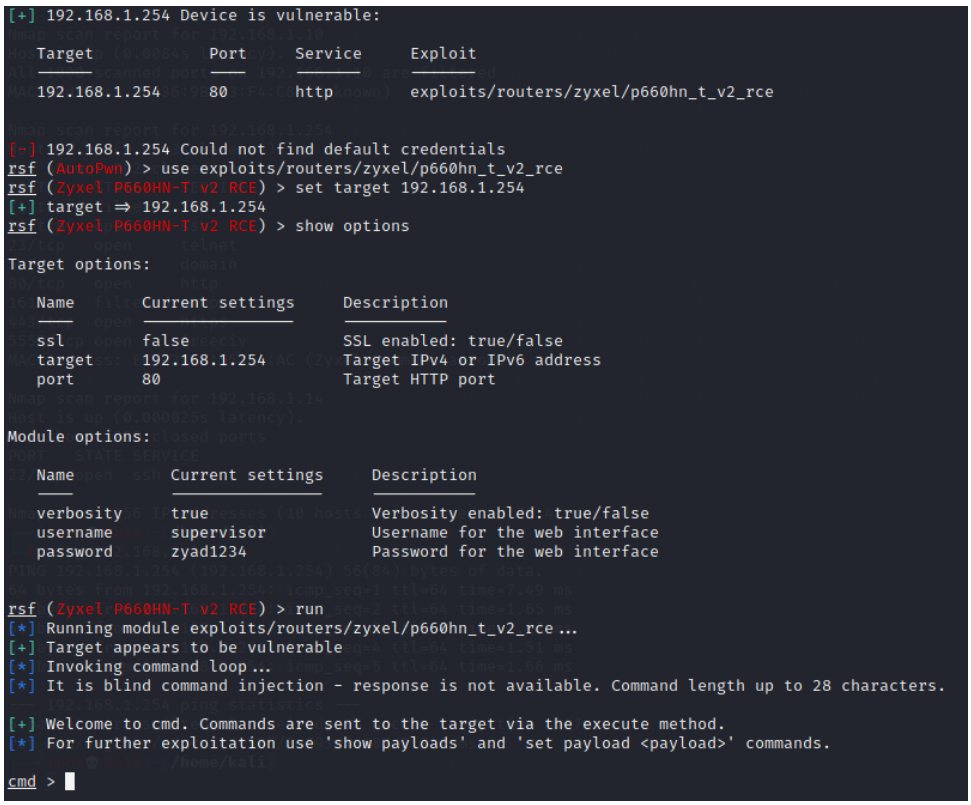
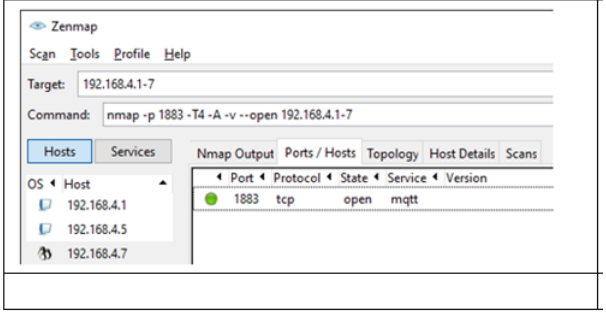
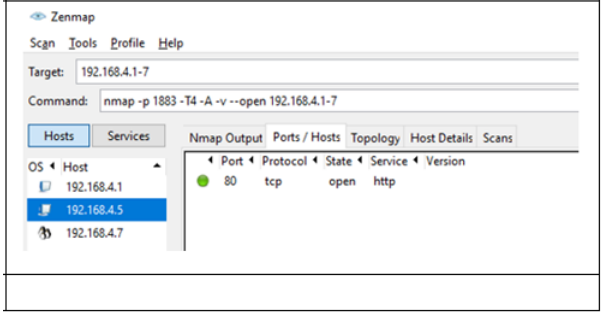


Figure 8: Vulnerable router RouterSploit

ESP32. Network scanning is performed on the network using Zenmap a Nmap security scanner with a GUI interface to provide a clear picture of the topology of the network and a clear indication of the open ports. The node containing the MQTT broker on port 1883 and the HTTP server on port 80 is illustrated. Both of these protocols are insecure and transmit data in cleartext. A penetration test would recommend that the secure versions of both protocols should be implemented HTTPS on port 443 and port 8883 for MQTT over TLS encrypting the data.

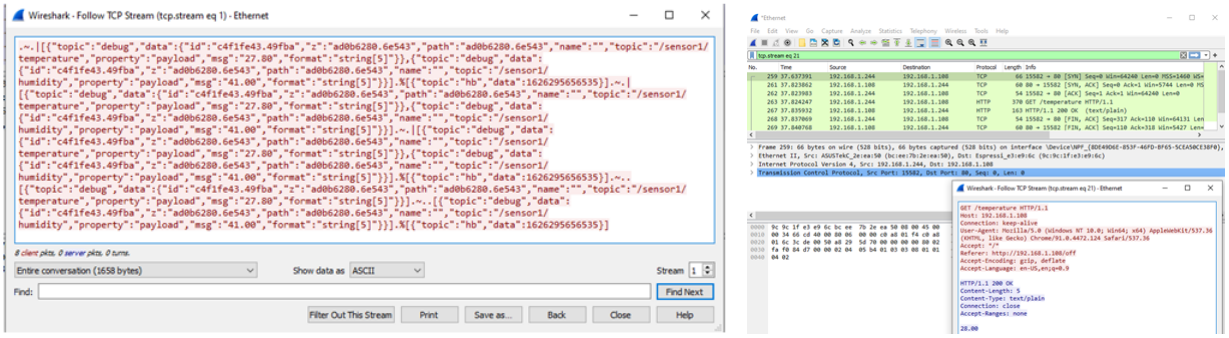
Table 4: Network Scan Results

MQTT broker PI on port 1883 insecure protocol	HTTP Server on port 80 insecure protocol
	

6.6 Check traffic for data exposure

The MQTT protocol on port 1883 is unencrypted as seen from the WireShark follow TCP stream on the left in Table 5 where the sensor temperature and humidity data can be seen. The unencrypted data from the HTTP server is seen on the right.

Table 5: WireShark Data Exposure



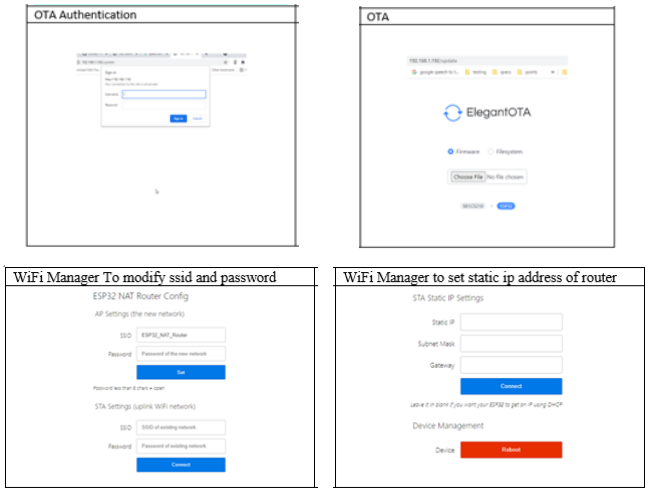
6.7 OTA update available

The latest guidelines for IoT development insists that there should be a mechanism in place to allow the firmware to either be updated or returned to a known state. Previously this would have involved the removal of the device to a lab where the update could take place. Libraries have been developed which would allow firmware updated to be carried out using wireless and an example of this is demonstrated below in Table 6. A WiFi-Manager can provide a mechanism to allow the update of new binaries and files with the option to return to a pre-update state should there be an issue with the upload. It can also allow the setting up of the local network and a facility to change networks or passwords as required.

6.8 Additional checks

In addition, the following check was carried out.

Table 6: OTA Capability, Wifi Manager



- Check for up to date documentation from IoT manufacturers. espressif documentation can be found online [9].
- Check Cybersecurity state awareness. Details of the ESP32 Application Level tracing Library logging can be found online in the API Guides [10].
- Check the end of life policy. Espressif has committed to approximately 12 years of support for its product [11]

6.9 Discussion

The experiments carried out in this paper were the minimum required to satisfy the evolving regulations in this area. esptool was started by Frank Ahlberg as an unofficial project and has been subsequently taken over by Espressif (Fredrik; 2021). Not all manufacturers have an equivalent tool with the same functionality that can be used to interrogate the chip and retrieve details for the identification or compare the firmware to ensure it has not been tampered with.

Some of the researched tools for use in this project showed great promise for performing penetration tests on IoT devices but were not used as they were no longer being actively developed. Tools that were originally written in python2.7 have not been upgraded for use with python3 this will become an issue as the sunset date for python2, Jan 1 2020 has passed, PENIOT is an example it targets IoT devices and supports all the protocols MQTT, CoAP, AMQP, and BLE it is however written in python2 and there appear to be no plans to port it to python3 (Berat; 2020).

Visual Studio Code with the PlatformIO extension is a cross-platform, multiple frameworks tool for embedded systems (Kravets; 2020) The flexibility of this tool allows white box testing of IoT projects to be performed simply and effectively. Since its introduction

[9]Espressif Documentation `https://www.espressif.com/en/support/documents/technical-documents`.

[10]ESP32 API Guide `https://docs.espressif.com/projects/esp-idf/en/latest/esp32s2/api-guides/app_trace.html?highlight=logging`.

[11]Espressif Longevity commitment `https://www.espressif.com/en/products/longevity-commitment`.

in 2015, it has constantly evolved and currently supports over 1000 boards from the leading manufacturers. The introduction of this tool will assist greatly with the development of a secure software development life cycle badly needed to improve Cybersecurity.

The check for up to date documentation from IoT manufacturers is to ensure compliance with the non-technical guidelines required by NISTIR 8259B (Fagan, Marron, Brady, Cuthill, Megas and Herold; 2020a). Espressif is proactive in developing a range of chips that are designed to meet the expectations of business leaders looking to develop into the latest technological areas of AI and ML. The documentation was easy to find and version control allowed access to earlier versions. The check for Cybersecurity state awareness can be hampered by the memory constraints of the IoT device. Espressif has published an API guide to assist with the logging and monitoring of its devices. The Check for end of life policy will become a consideration in the future It is an issue that can be easily tackled using policies and guidelines. It may not be possible to factory reset all these devices and wipe all personal data that used a write-once memory option. The question of how to keep track of these IoT devices over the long term or to deal with legacy projects to ensure proper disposal is not clear.

7 Conclusion and Future Work

In today world it is necessary to change the approach of cyber security from a prevent breach policy to an assume breach policy, the question can no longer be what to do to prevent a cyber-attack the focus has to shift to what to do when it does (Diogenes and Ozkaya; 2018). Pen Testing Frameworks for Web Applications and Networks are unsuitable for use with IoT due to the constrained nature of these devices which are designed for specific pre-defined long term tasks. The security implications highlighted by Mirai needs to be addressed to prevent similar types of attack. A suitable Pen-Testing Framework for IoT has to ensure that the following baseline security requirements are met.

- Disallow default credentials.
- Replace insecure protocols with secure ones.
- Ensure IoT devices can be uniquely identified on the network.
- Ensure that firmware has not been altered.
- Provide a mechanism to easily reset firmware or update the firmware on the device.
- Perform Static analysis or code review on the firmware before release.
- Protect data in transit and data at rest.

The penetration test carried out in this paper demonstrated how these requirements can be assessed on Espressif ESP32 devices. The solutions found may not work for other manufacturers. An equivalent ISSAF guideline may need to be produced to suggest how to test on other devices. The OWASP firmware security testing manual is a good starting point in this regard but it was found that over time other testing solutions became available while some were not updated, flexibility is required to allow the penetration tester the freedom to choose the most efficient test and tooling to carry out the task.

The effectiveness of The Pen Testing framework going forward can be enhanced if it was a requirement for manufactures to provide a tool similar to the esptool that would allow the interrogation of their chip to report on the unique identification and hash of the firmware ensuring that it has not been tampered with. The US, European and Australian Governments are taking the first steps into creating baseline security regulations and guidelines to regulate the industry. They are passing some of the responsibilities to the manufacturers to provide documentation that would allow developers to satisfy the new regulations and encourage the introduction of Secure SDLC principles.

This paper has just scratched the surface of what these powerful Microcontrollers can do, future work could involve testing the security of ESP-Mesh networks or Bluetooth Networks. It could also involve the end to end testing of the integration of IoT with Cloud and Fog Computing. The purpose of this paper was to illustrate how some of these evolving regulations could be implemented and tested using a suitable Pen-Testing Framework for IoT.

Acknowledgement

I would like to take this opportunity to thank my supervisors Dr Vanessa Ayala-Rivera and Dr Paul Stynes for their patience, assistance and guidance in producing this paper from them I have learned invaluable lessons concerning the technical structure of reports.

References

ACSC (2020). Iot code of practice: Guidance for manufacturers, [Online] Available : `https://www.cyber.gov.au/acsc/view-all-content/publications/iot-code-practice-guidance-manufacturers`. Accessed on: June, 11, 2021.

Arron, G. (2019). Firmware security testing methodology, [Online] Available : `https://github.com/scriptingxss/owasp-fstm/releases/tag/v1.0`. [Accessed on: Aug, 11, 2021].

Barcel, M., Klee, P., Ip, V., Chan, W., Spooner, R., Torres, M. A. D., Jankowski, R., Chuvakin, A., Torres, E., Hines, M. S. et al. (2000). Open-source security testing methodology manual, *Institute for Security and Open Methodologies, Tech. Rep* .

Barybin, O., Zaitseva, E. and Brazhnyi, V. (2019). Testing the security esp32 internet of things devices, *2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S T)*, pp. 143–146.

Berat, C. (2020). yakuza8/ peniot, [Online] Available : `https://github.com/yakuza8/peniot`. [Accessed on: Feb, 14, 2021].

Blythe, J. M., Sombatruang, N. and Johnson, S. D. (2019). What security features and crime prevention advice is communicated in consumer iot device manuals and support pages?, *Journal of Cybersecurity* **5**(1).

Boeckl, K., Boeckl, K., Fagan, M., Fisher, W., Lefkovitz, N., Megas, K. N., Nadeau, E., O'Rourke, D. G., Piccarreta, B. and Scarfone, K. (2019). *Considerations for managing*

Internet of Things (IoT) cybersecurity and privacy risks, US Department of Commerce, National Institute of Standards and Technology.

Council of European Union (2016). Regulation eu 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, [Online] Available : `https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32016R0679`. [Accessed on: Feb, 9, 2021].

Council of European Union (2019). Regulation eu 2019/881 of the european parliament and of the council of 17 april 2019 on enisa the european union agency for cybersecurity, [Online] Available : `http://data.europa.eu/eli/reg/2019/881/oj`. [Accessed on: Feb, 9, 2021].

Diogenes, Y. and Ozkaya, E. (2018). *Cybersecurity Attack and defense strategies*, Packt. ISBN: 9781838827793.

ENISA, E. (2017). *Baseline Security Recommendations for IoT in the context of Critical Information Infrastructures*, European Union Agency for Cybersecurity Heraklion, Greece.

Fagan, M., Fagan, M., Megas, K. N., Scarfone, K. and Smith, M. (2020). *IoT Device Cybersecurity Capability Core Baseline*, US Department of Commerce, National Institute of Standards and Technology. NISTIR 8259A.

Fagan, M., Marron, J., Brady, K., Cuthill, B. and Herold, R. (2020). Creating a profile using the iot core baseline and non technical baseline, *Technical report*, National Institute of Standards and Technology. NISTIR 8259C.

Fagan, M., Marron, J., Brady, K., Cuthill, B., Megas, K. and Herold, R. (2020a). Iot non-technical supporting capability core baseline, *Technical report*, National Institute of Standards and Technology. NISTIR 8259B.

Fagan, M., Marron, J., Brady, K., Cuthill, B., Megas, K. and Herold, R. (2020b). Profile using the iot core baseline and non-technical baseline for the federal government, *Technical report*, National Institute of Standards and Technology. NISTIR 8259D.

Fredrik, A. (2021). espressif esptool, [Online] Available : `https://github.com/espressif/esptool`. [Accessed on: Feb, 14, 2021].

Gallopeni, G., Rodrigues, B., Franco, M. and Stiller, B. (2020). A practical analysis on mirai botnet traffic, *2020 IFIP Networking Conference (Networking)*, pp. 667–668.

Kingthorn, R., Blankenship, H. and Turner, P. (2019). Application Threat Modeling | OWASP, [Online] Available : `https://owasp.org/www-community/Threat_Modeling`. Accessed on: June, 10, 2021.

Kravets, I. (2020). Professional collaborative platform for embedded development, [Online] Available : `https://docs.platformio.org/en/latest/`. [Accessed on: Feb, 10, 2021].

Miessler, D. (2015). Securing the internet of things: Mapping attack surface areas using the owasp iot top 10, *RSA Conference*.

Nickerson, C. (2021). The penetration testing execution standard. v1.1, [Online] Available : `https://buildmedia.readthedocs.org/media/pdf/pentest-standard/latest/pentest-standard.pdfl`. [Accessed on: August, 12, 2021].

NICULA, t. and ZOTA, R. D. (2020). Iot security, associated malware trends and vulnerabilities.

NIST, J. (2018). *Risk Management Framework for Information Systems and Organizations*, National Institute of Standards and Technology. NIST SP 800-37.

Rathore, B., Brunner, M., Dilaj, M., Herrera, O., Brunati, P., Subramaniam, R., Raman, S. and Chavan, U. (2006). Information systems security assessment framework (issaf) draft 0.2. 1b, *Open Information Systems Security Group* .

Rytel, M., Felkner, A. and Janiszewski, M. (2020). Towards a safer internet of things—a survey of iot vulnerability data sources, *Sensors* **20**(21): 5969.

Scarfone, K. A., Souppaya, M. P., Cody, A. and Orebaugh, A. D. (2008). Technical guide to information security testing and assessment., *Technical Report NIST SP 800-115*, National Institute of Standards and Technology, Gaithersburg, MD. [Accessed on: July, 12, 2021].
URL: *https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-115.pdf*

Stoneburner, G., Goguen, A. and Feringa, A. (2002). Risk management guide for information technology systems :: recommendations of the National Institute of Standards and Technology, *Technical Report NIST SP 800-30*, National Institute of Standards and Technology, Gaithersburg, MD. Accessed on: June, 11, 2021.
URL: *https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-30.pdf*

Tanczer, L., Blythe, J., Yahya, F., Brass, I., Elsden, M., Blackstock, J. and Carr, M. (2018). Summary literature review of industry recommendations and international developments on iot security, *PETRAS IoT Hub, Department for Digital, Culture, Media & Sport (DCMS)* .

U.S. House. 107th Congress (2002). H.r.3844 - federal information security management act of 2002, [Online] Available : `https://www.congress.gov/bill/107th-congress/house-bill/3844`. [Accessed on: Feb, 9, 2021].

U.S. House. 116th Congress (2020). H.r.1668 - internet of things cybersecurity improvement act of 2020, [Online] Available : `https://www.congress.gov/bill/116th-congress/house-bill/1668/all-info.` [Accessed on: Feb, 9, 2021].

Wichers, D. and Williams, J. (2017). Owasp top-10 2017, *OWASP Foundation* .

Yadav, G., Paul, K., Allakany, A. and Okamura, K. (2020). Iot-pen: A penetration testing framework for iot, *2020 International Conference on Information Networking (ICOIN)*, IEEE, pp. 196–201.