Redback Operations

Cyber Security Team

Group 3 - IoT Security

Completed by:

Hana Grace Coney

Stephanie Yanez

Thomas Robert O’Connor

Jacob Lummus

Tom Mirarchi

Summary

In many circumstances, devices that are linked to the internet can become an issue without the correct knowledge, tools, and people to prevent it. The Internet-of-Things (IoT) is continuously evolving throughout the world and will play a key role in the technology developed by Redback Operations. However, for the company to begin developing these technologies, they need to be familiar with what IoT is, as well as the components associated with it.

The following report focuses on three different components of IoT technology and security. These include:

- IoT Security
- Message Queuing Telemetry Protocol (MQTT)
- Sensors

The sole aim of this research document is to provide future guidance and knowledge to the Redback IoT Developers and the Cyber security team about essential IoT topics.

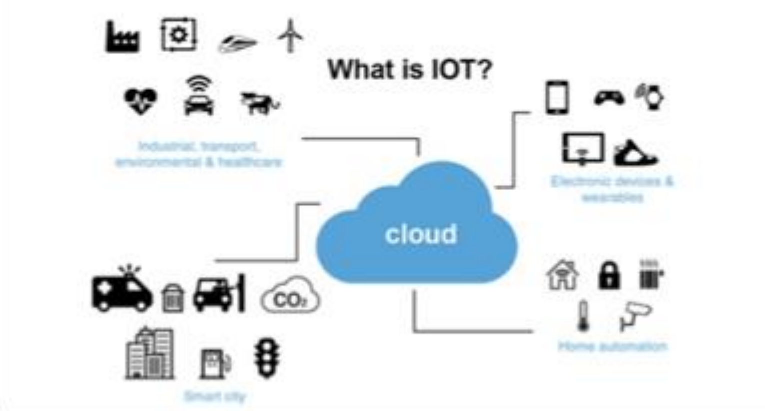
Table of contents

IoT Security and IoT Network Security

What is IoT?

IoT also known as the Internet of Things are connected devices embedded with different software and other technologies that transfer data and communicate over a network [1] [5]. Some examples include:

- Mobile phones
- Health monitors
  - Pacemakers
  - Watches


What is IOT?
Industrial, transport, environmental & healthcare
Electronic devices & wearables
cloud
CO2
Smart city
Home automation

IoT works through built in sensors in the device where it gathers specific data related to the specific needs and wants of the company. The data shared can then be analysed to observe what is useful and what is irrelevant to then develop the device further. IoT does not just include devices, it includes the software's and servers associated with its creation [10].

A quick and easy video that explains IoT can be found using this link:

https://youtu.be/6YaXKxXSli0

What is IoT Security? [2]

IoT Security are measures put in place to ensure that the devices, systems, and servers are not prone to exploitation when a vulnerability arises. The goal of IoT security is to:

- Protect data
- Prevent vulnerabilities and detect them when they arise.

In maintaining the goals, IoT security seeks to:

i. Secure smart devices

- Securing the hardware
- Updates
- Penetration testing

ii. Secure networks

- Encryption
- Strong authentication
- Firewalls
- VPN’s

iii. Secure data

- Encrypting data
- Relevant VS irrelevant data→ decide which data to collect
- Secure network communication

For IoT security to be set in motion, it is important for the cyber team to explore common cyber threats to begin with as a building block to start prevention methods to avoid those threats from becoming an issue, in addition to preventing other threats from occurring in the future. Common IoT security challenges include:

- Software and firmware vulnerabilities
- Insecure connections
- Data leaks
- Malware risks
- Cyberattacks

IoT security is essential because of the wide application of IoT systems. Any vulnerability found can lead to data breaches and system failures which can affect thousands of people as well as the companies involved.

Relations to Redback Operations

Redback Operations is a company looking to enhance the lives of people through the use of smart exercising. By using IoT, Redback Operations are provided with various options to develop technology devices to provide community driven exercises.

i. **Smart watches**

Smart watches are IoT devices, which Redback Operations can use as a fitness wearable. Fitness wearables can monitor the users heart rate, movement, calories, and many more biometric measurements. Also offering the option to connect the device to the user's mobile phone [11]. The company can use the data collected to assist in the development of a 3D outdoor simulator, providing a specific exercise plan tailored to the individual.

ii. **Future developments**

Redback Operations can use IoT with future company developments. Virtual Reality (VR) is a possible direction that will take place in which headsets can be created to work with 3D worlds for virtual exercise.

IoT Vulnerabilities

Vulnerabilities should be looked at within IoT to assess and secure potential risks associated with redbacks devices. With vulnerabilities arising, companies who do not address and provide solutions to them can possibly face legal actions [20].

CIA Triad (i.) and the relation to IoT vulnerabilities (a.) [3] [4] [6]

The CIA Triad are a model used for developing security systems. It stands for

I. Confidentiality

Security measures put in place to ensure that the data of users remain private.

a. If an IoT device/system has been compromised, the data embedded within could be accessible to unauthorised personnel's, breaching the confidentiality of the users/clients.

II. Integrity

Ensures that the data from users in the system is not modified and/or deleted by any unauthorised parties.

b. Exploits of devices can result in user data being accessed, which can then result in their data being tampered with.

III. Availability

Refers to how accessible the data of users are to authorised parties. If a party is authorised to access certain data and areas of the system, then it must be available to them when needed.

c. If breached in a system, authorised individuals may not be able to access certain areas that they once were available to. Hence, the attacker may use it to their advantage to access other user data as well as hack into other devices.

Weak Passwords

i. **Default passwords [7]**

Weak passwords are associated with IoT devices through the default password given to the client. Without the client changing that password, the device can be susceptible to exploitation.

ii. **User passwords**

Without creating strong passwords and using:

- Common passwords
- Short in length
- No special characters

for their device, hackers can use brute force and other passwords attack mechanisms to have access to a device system.

Software updates [4] [7]

There are diverse ways in which software can become an issue for IoT devices.

I. Without regular updates of the IoT software/firmware, it can become exposed overtime in which vulnerabilities can be increased.

II. Without running background checks for vulnerabilities on updates (i.e., penetration testing), it can potentially lead to risks of installing malicious codes into the device and compromise it.

III. Spam attacks [8]

- Hackers use spam attacks by sending mass emails containing malware to users through the form of software updates or packages. Users that are unaware of these dangers, pose the risk of installing them which can result in their device being hacked.

Access Control

Without limiting the Read, Write and Execute (RWE) permissions of the IoT system, anyone can have access to sensitive information which can have negative results. [9]

i. **Read**

   Individual's ability to read what is in the file.

ii. **Write**

   Individual's ability to modify the file contents.

iii. **Execute**

   Individual's ability to run the file contents. If it is a code program, then the individual will have the ability to run the program.

The challenges with IoT and access controls stem from employee to users, to the public. Without stable restrictions in certain areas, it will enable hackers to gain access to these permissions if the device is compromised [14].

Distributed Denial-of-Service (DDoS) Attacks [16] [17]

Because of the limitations with IoT resources i.e., storage and network capacity, the risk of DDoS attacks increases. DDoS attacks occurs when many devices are attacking a single server, crashing the device, its systems, and servers in which data can no longer be received and distributed [8]. IoT devices are common as IoT devices primarily use the internet to establish communications. If a DDoS attack has been successful, the device can be used as a botnet to infect other devices without the individual being aware. This can result in identity theft and other consequences if the attacker is successful.

Recommendations

Encryption [15]

If communication were being transferred in plain text, it would be easy for hackers to obtain and read sensitive data if they gain access to a network (most using a man-in-the-middle attack). Thus, encryption is an important prevention method in avoiding unauthorised individuals to compromise sensitive information.

End-to-end encryption ensures that only the sender and the receiver can access the data, using a special key in which the specific user can decode that data.

Passwords

Encryption should be used with user and employee passwords/credentials, data, networks and many more [15].

Users must also use strong passwords [18] for their device and account systems, and companies should implement strong password policies for users to be able to create their account and set up their device. Some guidelines can include:

⇒ Special characters

⇒ Certain amount of letters

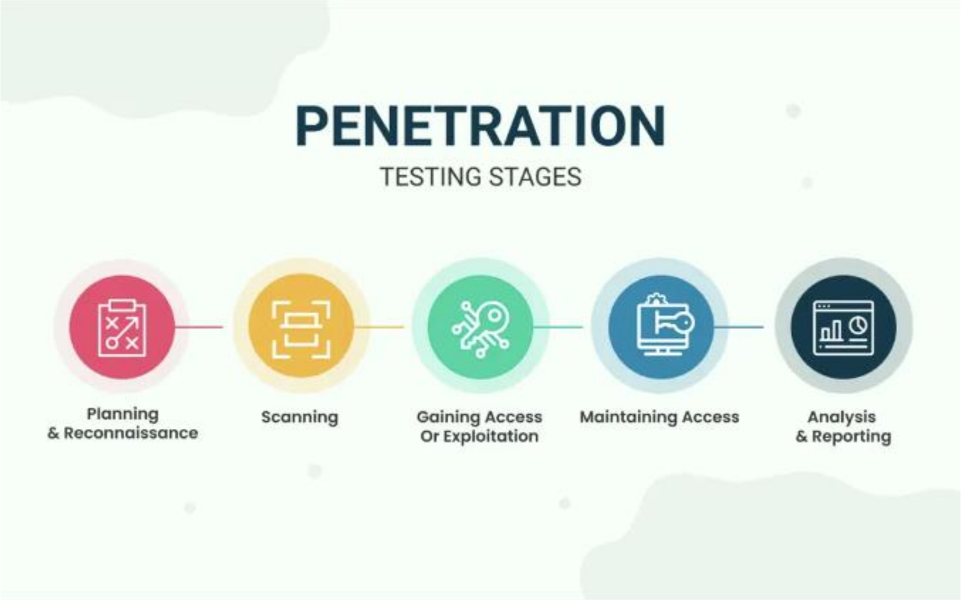
⇒ No common passwords

Regular software updates [1] [14]

Users should be provided with regular updates to patch existing bugs and vulnerabilities found in the current software. Through maintaining these updates, it will decrease the chances of virus’ infecting the device [8]. This updated software should also be pen tested to ensure no mistakes have been made in the process of creating it, as well as any vulnerabilities that may appear.

Penetration testing [12]

Penetration testing is also known as ethical hacking in which hackers (ethical) attack the system of a company. Through attacking the system, they are able to identify vulnerabilities and exploit them to then find solutions for those vulnerabilities, often following the penetration testing stages shown below:



Types of penetration testing include:

- Social engineering
- Network
- Web application
- Mobile application
- Software

Testing tools:

- Kali Linux
- Metasploit
- Wireshark
- John the Ripper

Access controls [14]

Access controls should be administered in all areas of the IoT system. Those in charge should ensure that only those permitted are able to have access to certain areas. Access controls involve the Read, Write and Execute permissions of the system in which the individual needs to address all parties involved with the IoT device/system/server including:

- Users
- Employees
- Public

Hardware security [7] [8]

Although hardware security does not seem like necessary aspect of IoT security, it essentially does have a key role. By securing the physical device itself by making in tamper proof, it can avoid hackers from stealing the device and tinker with it. Hardware security is found to be more effective [13] due to the difficulty of accessibility, making it more challenging for attackers to be successful. Hence, keeping up with the security of the device's hardware (electronic components) [19] will continue to ensure the security of the IoT device.

User awareness [14] [18]

Users should be aware of the dangers as well as the essential information concerning IoT and the device they are using. By informing consumers about regular updates, password changes and relevant news, it will prevent vulnerabilities from arising. Users should also be aware of any recalls surfacing if there is a fault in the device in which can be informed through any contact details provided by the user, usually in the form of an email or text message.

Training [1]

With the internet continuing to evolve, new cyber threats continue to increase. Hence, it is important for the security teams to be informed of any threats, keep up to date with systems (new or old), and continue to train and be provided with training to maintain their knowledge as well as to prevent any threats to the company’s system.

Cisco Router and Switch security practices

Many commands are universal across switches and routers, but some are unique to each other

Command	Explanation
Switch(config)#banner motd# https://study-ccna.com/configuring-cisco-banner-motd-login-exec/	This allows the user to create a banner message like "No unauthorized access allowed. Violators will be prosecuted to the full extent of the law#" This message will appear each time someone attempts to connect to the router or switch. The # symbols represent the start and the end of the message. You are able to use other special characters in place of the # if you would like to.
Router(config)#service password-encryption https://www.oreilly.com/library/view/hardening-cisco-routers/0596001665/ch04.html	This command encrypts all cleartext passwords by utilizing the Vigenère cipher
Router(config)#security passwords min-length {integer} https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/security/s1/sec-s1-cr-book/sec-cr-s1.html#wp4245155510	This requires all passwords to be of a minimum length as set by the {integer} parameter
Router(config)#login block-for {seconds} attempts {attempts} within {seconds} https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_usr_cfg/configuration/xe-16/sec-usr-cfg-xe-16-book/sec-login-enhance.html	This command will enter the router into a "quiet mode" where it will not accept any incoming connection requests for the amount of time specified in the first {seconds} parameter if the amount of failed login attempts within the timeframe set in the second {seconds} parameter exceeds the integer set in the {attempts} parameter.
Router(config)#login quiet-mode access-class {ACL} https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_usr_cfg/configuration/xe-16/sec-usr-cfg-xe-16-book/sec-login-enhance.html	This command can make computers or networks that are a part of the ACL specified in {ACL} exempt from the quiet mode
Switch(config)#enable secret {password} https://www.ccexpert.us/cisco-secure/enable-secret.html	This will require the user to input the password specified by the {password} parameter each time the "enable" command is used
Switch(config)#line con 0 Switch(config-line)#password {password} Switch(config-line)#login https://www.cisco.com/c/en/us/support/docs/ios-nx-os-software/ios-software-releases-110/45843-configpasswords.html	These commands will set a password on the console port, so each time someone connects to the console port on the switch or router after a logout or timeout, they will be required to enter the password as denoted in the {password} parameter

Switch(config)#line vty 0 4 Switch(config-line)#exec-timeout {minutes} {optional seconds} https://study-ccnp.com/cisco-exec-timeout-absolute-timeout-commands/	These commands will set a timeout of {minutes} minutes and {optional seconds} seconds if no activity is detected on the virtual terminal lines. Once a timeout has occurred, the user will be logged out.

Switch(config)#line vty 0 4
Switch(config-line)#absolute-timeout {minutes}
Switch(config-line)#logout-warning {seconds}
https://study-ccnp.com/cisco-exec-timeout-absolute-timeout-commands/
	The absolute-timeout command will timeout a user in a session as regardless of activity. They may be in the middle of inputting commands and the session will be cut after {minutes} minutes. As a courtesy, you can provide a warning to the user with the logout-warning command. This will provide a warning to the user of the impending timeout of the session {seconds} seconds before the session timeouts.

Switch(config)#hostname {name} Switch(config)#ip domain-name {domain} Switch(config)#crypto key generate rsa modulus {bit length} Switch(config)#username {username} privilege {level} secret {password} Switch(config)#line vty 0 4 Switch(config-line)#transport input ssh Switch(config-line)#login local https://www.cisco.com/c/en/us/support/docs/security-vpn/secure-shell-ssh/4145-ssh.html	These set of commands will enable SSH on the switch or router to be able to remotely configure the switch or router. Changing the name from the default name is required to enable SSH which can be completed with the first command. A domain also needs to be set with the second command. A random domain can be used such as “akjsdbkjasd.com” but it is recommended to use something relevant to what you are doing. The keys used are generated using the third command and the length of the keys are denoted with the {bit length} parameter. This can range from 360 to 2048 bits in length. The forth command is used to create an account with username {username} and password {password}. The “privilege {level}” portion is optional, but allows you to set privileges on the account being created. The privilege level can range from 0 to 15 with 0 being read only and 15 being full access. Different commands have different privilege level requirements by default and can be changed at a later time. The sixth command requires that those wanting to use the virtual terminal lines are required to use ssh. The seventh line requires the user to provide a username and password that is stored in the

	local database. Command four is how to create a new user for the local database.
Switch(config)#ip ssh timeout {seconds} https://docs.commscope.com/bundle/fastiron-08090-commandref/page/GUID-D09BE5D5-9FC0-4211-9DA9-51593803DD6B.html\	This command will timeout a ssh session if no input is received with {seconds} seconds
Switch(config)#ip ssh authentication-retries {retries} http://docs.ruckuswireless.com/fastiron/08.0.50/fastiron-08050-commandref/GUID-F4E1BF35-10CA-48AE-8B20-EC9F05B259A0.html	This command will only accept up to {retries} attempts to enter the password before the connection is terminated.
Switch(config)#int {interface name} Switch(config-if)#switchport port-security	These commands enable the port security functionality of the selected ports. This can prevent unauthorized access to a network.

Switch(config)#int {interface name}
Switch(config-if)#switchport port-security mac-address sticky
OR
Switch(config)#int {int name}
Switch(config-if)#switchport port-security mac-address {MAC address}
https://www.computernetworkingnotes.com/ccna-study-guide/switchport-port-security-explained-with-examples.html
	These commands will restrict the amount of hosts that are allowed to communicate on the specific port. The “sticky” keyword specifies that the switch will dynamically learn the devices connected to the switch on that specific port and only allow those devices to connect to the network using that port. The default maximum number of devices that the switch can learn is 1. Explicitly specifying the MAC addresses (in the form xxxx.xxxx.xxxx) will statically set the devices that are allowed to connect to the switch using that specific port

Switch(config)#int {interface name}
Switch(config-if)#switchport port-security violation {option}
https://www.computernetworkingnotes.com/ccna-study-guide/switchport-port-security-explained-with-examples.html
	If a device that is not allowed to connect to the network using the port attempts to, then the default for the port is to shutdown, preventing all access through the port. This can be changed by specifying either “protect” or “restrict”. Both options will prevent the port from shutting down and will drop any packets from unauthorized devices. “protect” will not log any dropped packets and only works if the “switchport port-security mac-address sticky” command is issued, and “restrict” will log the unauthorized packets. You can also revert back to “shutdown” if required.

Switch(config)#int {interface name}
Switch(config-if)#switchport port-security maximum {integer}
	This command will allow more devices to connect to a singular port, up to 132.

https://www.computernetworkingnotes.com/ccna-study-guide/switchport-port-security-explained-with-examples.html	
Switch(config)#no ip http server https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/https/command/nm-https-cr-book/nm-https-cr-cl-sh.html#wp3775424912	This command will disable the http server if it is running. You will want to disable any services that are not required as this can provide an attacker less potential gateways into a device
Switch(config)#int range {range of ports} Switch(config-if-range)#shutdown https://www.cisco.com/c/en/us/td/docs/ios/interface/configuration/guide/ir_ifrange.html	You will want to disable all ports that are not required to be enabled for normal operation. This will prevent an attacker from connecting a device to the network by simply plugging a device into a free ethernet port. If you would like to turn off all ports on FastEthernet 0/1 through to FastEthernet 0/24 and GigabitEthernet 0/1 through to GigabitEthernet 0/2, this can be denoted using “f0/1-24,g0/1-2”.
Switch(config)#vlan {vlan number} Switch(config-vlan)#name {vlan name} Switch(config-vlan)#exit Switch(config)#int {interface name} Switch(config-if)#switchport mode access Switch(config-if)#switchport access vlan {vlan number} Switch(config-if)#int {interface name connected to router} Switch(config-if)#switchport trunk encapsulation dot1q Switch(config-if)#switchport mode trunk Switch(config-if)#int {interface name connected to other switches} Switch(config-if)#switchport mode dynamic desirable https://www.networkstraining.com/how-to-configure-vlans-on-a-cisco-switch/ https://www.ciscopress.com/articles/article.asp?p=2181837&seqNum=8 Router(config)#interface {interface name}.{vlan number} Router(config-subif)#encapsulation dot1Q Router(config-subif)#ip address {ip address} {subnet mask}	Creating VLANs and assigning ports to those VLANs is a great way to implement privileges for users on those ports. From here, you can restrict users on different VLANs from accessing specific networks or IP addresses. Once VLANs are created and DHCP pools are created for those specific VLANs, you can then configure Access Control Lists (ACLs) on the router for these VLANs. When the VLANs are created, they will stay local to the switch, to allow the same VLANs that are on other switches and routers to communicate, the interfaces connecting them need to enter into trunking mode. Sub interfaces need to be created for each VLAN as well, this is to allow VLAN associated with the specific sub interface to communicate with other networks.

Router(config-subif)#int {interface name} Router(config-if)#no shutdown https://www.comparitech.com/net-admin/inter-vlan-routing-configuration/	

Router(config)#ip access-list standard {name/number}
Router(config-std-nacl)#{rule number} {ACL action} {"host"/wildcard} {IP of host/network}
Router(config-std-nacl)#exit
https://www.cbtnuggets.com/blog/certifications/cisco/networking-basics-how-to-configure-standard-acls-on-cisco-routers
 	
Router(config)#ip access-list extended {name/number}
Router(config-ext-nacl)#{rule number} {ACL action} {protocol} {source details} {source operator} {source port} {destination details} {destination operator} {destination port}
https://www.cisco.com/c/en/us/td/docs/app_ntwk_services/waas/waas/v401_v403/command/reference/cmdref/ext_acl.pdf
 	
Router(config)#interface {interface name}
Router(config-if)#ip access-group {ACL number/name} {direction}
	ACLs can be used in either extended or standard form. Standard only takes IP addresses of the source devices, being a host or network, and a wildcard address. An extended ACL on the other hand can accept a wide range of parameters including but not limited to the destination hosts(s), protocol, port number, and operator.
	The rule number is optional and specifies where exactly inside of the ACL that this particular rule will sit.
	{ACL action} specifies whether to permit or deny the traffic if it matches the rule by using the keywords "permit" and "deny".
	{"host"/wildcard} defines whether the rule will apply to a singular host with the "host" keyword or a group of hosts on a particular network by passing the wildcard address of a group of hosts on a network.
	{IP of host/network} is the specific IP of the host or network that you want the ACL to apply to
	
	Extended ACLs are implemented slightly differently which allow for finer control over what traffic is being passed through the network. The first 2 parameters are the same, but that is about all. {protocol} specifies whether the traffic is using "tcp" or "udp" but this is optional. {source details} matches the source of the packet and can be implemented in multiple ways, this can be specifying "any" to match all traffic, specifying a network such as "192.168.0.0 0.0.0.255", or by specifying a specific host by utilising the host keyword similar to "host 192.168.0.2". {destination details} is implemented in the same way as {source details}, but is instead to match the destination of the packet. Source and destination {operator} and {port} are both

<table>
<tr>
<td></td>
<td>optional but allow the ACL to match the port numbers of the packet. The operators that can be used are as follows:
<table>
<tr><td>eq</td><td>Equals</td></tr>
<tr><td>neq</td><td>Not equals</td></tr>
<tr><td>lt</td><td>Less than</td></tr>
<tr><td>gt</td><td>Greater than</td></tr>
<tr><td>range</td><td>Range</td></tr>
</table>
Port can specify the specific source or destination ports of the packets, with web traffic for example, you can use the specific port “80” or the keyword “www”. Some common ports do have keywords associated with them but many do not and will require the specific port number.

Once an ACL has been created, it will need to be applied to an interface on the router and will require a direction. One thing to keep in mind is that the direction is from the perspective of the router, not the network itself which is a common misconception. {direction} can simply be labeled as either “in” or “out”. “in” specifies traffic entering into the router through the interface that the ACL has been configured on, whereas “out” specifies the traffic leaving the router through the specific interface.

The router will check the traffic against the ACL. The first rule that applies will have the associated action take place. If none of the rules apply, than an implicit deny all will take place.</td>
</tr>
<tr>
<td>Switch(config)#int {interfaces that don’t require trunking}
Switch(config-if-range)#switchport mode access
Switch(config-if-range)#int range {all interfaces}
Switch(config-if-range)#switchport nonegotiate</td>
<td>To prevent attacks on VLANs, it is recommended to disable any form of automatic negotiation between switches that can share information about VLANs to an unauthorised user. You can do this by disabling auto-trunking negotiations with the “switchport nonegotiate” command and manually enabling trunking on</td>
</tr>
</table>

Switch(config-if-range)#int range {interfaces that require trunking} Switch(config-if-range)#switchport mode trunk	ports that require it using the "switchport mode trunk" command.
Switch(config)#vlan {random number} Switch(config-vlan)#name {random VLAN name} Switch(config-vlan)#exit Switch(config)#int {interfaces that require trunking} Switch(config-if-range)#switchport trunk native vlan {same random VLAN number} https://www.networkworld.com/article/2234512/cisco-subnet-tagging-the-native-vlan.html	You will want to change the native VLAN on the switches as well. The default is VLAN 1 and has been documented extensively as to how exploits can take place on this default setting. The native VLAN is the only VLAN traffic sent across the trunking interfaces on switches without a tag which is used to identify all traffic being sent between switches and what VLANs that they belong to. Untagged traffic creates security vulnerabilities, so you will want everything to be tagged by creating a vlan with no ports associated to it, and assigning that as the native VLAN. This ensures that all traffic is tagged and cannot be taken advantage of by known vulnerabilities.

Wildcard addresses/masks

A wildcard address is calculated by the inverse of the subnet mask. This means that 255.255.255.255 – subnet mask = wildcard mask. For example, if the subnet mask is 255.255.128.0, then the wildcard mask will be 0.0.127.255

If you would like to reverse any command that you have implemented, put the keyword "no" before the command.

Messaging Queuing Telemetry Protocol (MQTT)

The MQTT Transport Protocol

MQTT Fundamentals

MQTT is an open OASIS and ISO standard (ISO/IEC PRF 20922) for client-server, publish/subscribe type messaging transport protocol. The protocol runs over TCP/IP, or other over other network protocols that provide ordered, lossless, bi-directional connections [1].

The principle of this protocol focuses on minimizing network bandwidth and device resource requirements ensuring a reliable packet delivery even over low-bandwidth or unreliable networks [2].

MQTT offers three qualities of service for message delivery [3]:

1. **QoS o**
    a. Messages are delivered at most once according to the operating environment where the chance message loss remains intact and where it does not matter if the connection to the application reading sensor data is temporarily lost.
2. **QoS 1**
    a. It is otherwise known as “at least once” message delivery service where messages are assured to be delivered at least once and where duplicity may occur.
3. **QoS 2**
    a. It is otherwise referred to as “exactly once” message delivery service where messages are ensured to be delivered exactly once.

MQTT has three components [3]:

1. **A Publisher or Producer (An MQTT Client)**
    a. An MQTT client is responsible for opening network connections to the server, creating messages to be published, publishing application messages to the server, subscribing to request application messages that it is interested in receiving, unsubscribing to remove a request for application messages and closing network connection to the serve.
2. **A broker (An MQTT Server)**

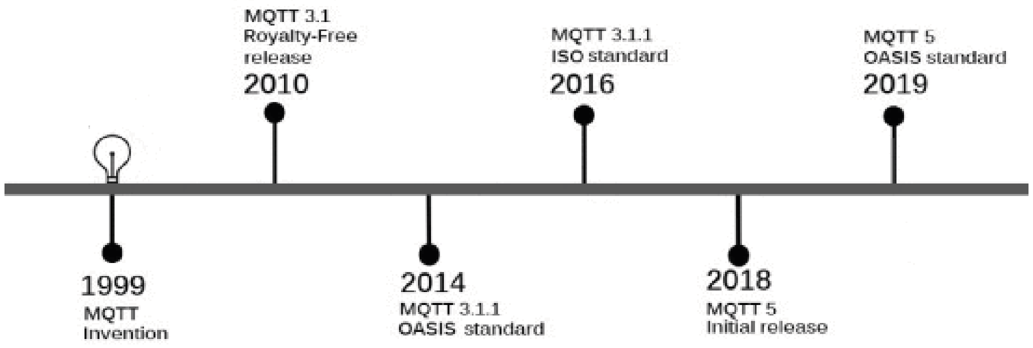
   a. An MQTT server is a program or device based on MQTT that acts as a post office between publishers and subscribers. An MQTT broker is responsible for accepting network connections from clients, accepting application messages published by clients, processing subscribing and unsubscribing requests from clients, sending application messages to clients as per its subscriptions, and closing the network connection from the client. Examples of such brokers include Mosquitto, HiveMQ and EMQ X.

3. **Consumer / Subscriber (Another MQTT Client)**
   a. **See 1a.**

Version History

First released in 1999, the MQTT protocol has experienced many developments. Its most current version is MQTT 5 which improved on previous versions. New features include [9]:

- A new expiration date has been added to the session, making it easier to use
- The extended authentication mechanism has reduced the need for TLS
- To bring greater efficiency a property called ‘Topic Alias’ was introduced. Though, this only has a limited effect as it does not work beyond network connections. However, MQTT was originally good at sending multiple messages while maintaining the same network connection.
- Power consumption can be reduced without assuming that the network connection is always connected
- By using the request/response sequence, it can notice a pattern in which one or more pieces of information possessed by the responder are searched and acquired by a query
- By using the requestion response sequence, end-to-end response confirmation has become possible
- End-to-end response confirmation makes it easier to detect lost messages and resend them
- Flow control makes it harder to lose messages
- The extended authentication mechanism has made it possible to support challenged-response sequences related to passwords and tokens.



Mosquitto Broker Fundamentals

As we now know, a component of the MQTT protocol is the requirement for a broker. A broker is that intermediate entity that enables MQTT clients to communicate and there are many providers or brokers offering this service with subtle differences.

One such broker is Mosquitto, it is an open-source MQTT broker written in C that supports versions 5.0, 3.1.1 and 3.1. Compared to other brokers, Mosquitto's largest advantage is that it is lightweight, scalable and is suitable for use on all devices from low power single board computers to full servers [4].

For your own reference:

- Mosquitto Documentation

Comparison to other Transport Protocol

MQTT may be the most ideal transport protocol option for IoT technologies. However, there are many other protocols that can be implemented. Other common transport protocols particularly known are:

1. **CoAP (Constrained Application Protocol)**
2. **XMPP (Extensible Message and Presence Protocol)**
3. **AMQP (Advanced Messaging Queuing Protocol)**
4. **DDS (Data Distribution Services)**

There are many other options in transport protocols. However, in this report it will cover the following advantages and disadvantages of the protocol above. [7][8][12][13]

1. **CoAP – Constrained Application Protocol**

The Constrained Application Protocol is a transport protocol that constrained, low-powered sensors and connected devices (IoT) via networks with low bandwidth and availability. CoAP can be considered as one of the most popular Machine-to-Machine (M2M) protocols as it extends to reach HTTP constrained devices.

CoAP is a **one-to-one protocol** based on the client/server model whereas MQTT provides supports for **many-to-many communication**.

Advantages	Disadvantages
▫ Lightweight ▫ Reduces power requirements ▫ Smaller packet size ▫ Multi-cast support ▫ Easily integrated with the Web ▫ Lower overhead	▫ Message unreliability (UDP does not guarantee ▫ Security – unencrypted default ▫ Lack of topic publication/subscription approach ▫ **Network Address Translation (NAT)** issues ▫ Complexity for mapping protocols

2. **XMPP – Extensible Message and Presence Protocol**

The Extensible Message and Presence Protocol is an open communication protocol designed to provide instant real-time communication. These could include instant messaging, voice and video calls.

The protocol supports **multiple communication patterns**.

These include **Asynchronous Messaging, Publish/Subscribe (same as MQTT) and Request/Response** sequence. XMPP is more notice as a **streaming protocol** to make it possible to exchange fragments of XML between two network endpoints.

Advantages	Disadvantages
▫ Persistent connection ▫ Decentralization – no central XXMP servers are needed ▫ Allows servers with different architectures to communicate	▫ No QoS mechanism ▫ Streaming XML has overhead ▫ XML content transport asynchronously ▫ Servers may overload with presence and instant messaging

3. **AMQP – Advanced Messaging Queuing Protocol**

   The Advance Messaging Queuing Protocol is an open standard application layer to pass messages between organizations and applications. The main use of AMQP is to develop unmatched communication between clients and broker parties.

   AMQP also uses the Publish/Subscribe sequences just like MQTT. Similarly, the publisher will hold the responsibility of generating the message whilst the client collects and administers it.

Advantages	Disadvantages
▫ Store-and-forward capabilities ▫ Uses QoS mechanism ▫ Space to evolve to work with different standards ▫ Offers secured connections with SSL protocols MQTT, CoA, HTTP and XMPP	▫ Low success rate with low bandwidth ▫ Not compatible with old versions ▫ Requires higher bandwidth than MQTT/CoAP/XMPP ▫ Resource discovery is not supported unlike CoAP/HTTP/XMPP

4. **DDS – Data Distribution Services**

   The Data Distribution Services is an API standard and middleware protocol that provides data connectivity between nodes. The protocol utilizes publish/subscribe based patterns like MQTT.

   DDS is optimized for the distribution processing. This means that it will directly connect to sensors, applications, and devices without the dependence of centralized IT (Information Technology) infrastructure.

Advantages	Disadvantages
▫ Suitable for real-time IoT ▫ Has powerful QoS ▫ Scalable extensible and efficient standard	▫ Support IP multicast ▫ QoS policies are only applied in strict DDS environment ▫ Events are originated per source in a real-time not multiple source

MQTT Advantages and Disadvantages

MQTT is significantly known for serving many IoT devices. The design of the protocol focuses on the support features, these may include:

i. **remote monitoring whilst providing low latency**
ii. **communication/messaging over fragile network**
iii. **efficient distribution of data to one or many distributors**

The popularity of MQTT had increased due to its compatibility amongst industries creating Machine-to-Machine (M2M) and Internet of Things (IoT). Yet, limitations do exist within this protocol and can be perceived from various aspects.

***Note:** Advantage and disadvantage can range depending on the versions of MQTT. The latest version, **MQTT 5**, became OASIS standard in 2019, however few users have yet transitioned from the previous version (**MQTT 3.1 and 3.1.1**). [8][11][13][14]

Redback Operations will need to inform the Cyber Security team of the version (**MQTT**) being utilised.

Pros:

Advantages	Description
Efficient Communication	Due to its **lightweight design**, the protocol makes it possible to transmit data with low energy usage. This makes it capable of providing a unified communication connection for a specific topic. The majority of IoT devices are well-known to not have powerful memory and processing power. It is designed to be flexible to exchange with message exchange and avoid confusion in the data infrastructure. Which makes MQTT a suitable option for its responsiveness is near real-time, text-based messaging application.
Message Delivery (details)	Supports an event-oriented paradigm with **asynchronous bidirectional** low latency push delivery of messages. To communicate, the publisher will need to know the broker's IP address, however the subscriber does not

	require to know anything. The subscriber does not need to know the details of the publisher’s network connection. This is to ensure the communication overhead is kept low on the device side. Both ends will then be capable of operating independently. Subscribers can be changed. There is no need for alteration of the publishing devices end.
Reliability	The options from **Quality of Service (QoS)** ensure to deliver flows in network traffics to specific capacity allocation and differentiated handling.
Low battery consumption	Works very well with **battery-powered devices** or **require low power consumption**. MQTT does not require access to an electricity grid. The protocol was designed to be capable in harsher environments, particularly areas that are deserted. The feature of this protocols use in low power is reduced as the amount data being transferred over the wireless link (**e.g., Bluetooth, Wi-Fi etc...**). **MQTT is known as a binary protocol**. This will mean there is less overhead. If the protocol maintains its connection with **TCP** (T**ransmission Control Protocol**), it will establish a connection for each published item of the data while avoiding the overheads. **If HTTP was applied in instead of MQTT**. HTTP would require polling. This means it would consume a lot of power, even in areas with no messages.
Security	Security for MQTT can be considered to have both **advantages** and **disadvantages**. MQTT is fortunately possible to implement security to its system. **Transmission Control Protocol (TCP)** is a common transport protocol that operates with MQTT, and TLS/SSL encryption would be the most suitable method to secure communication.

	However, MQTT is natively unencrypted. The protocol was originally not equipped with built-in security. Hence, adjustment will be needed to further strengthen the security.
Bidirectional	The **devices** can be used as both **publisher** and **subscriber**. Therefore, it will become possible to receive commands/orders just by subscribing to a specific topic on the broker. Data can also be received from other devices which could be an input into the data it publishes.

Many of the **publicized advantages** above rely on MQTT being connection oriented. However, since these actual terminals are under wireless environments such as cellular and Wi-Fi, or under firewalls, the connections are always connected.

Cons:

Disadvantage	Description
Operating on TCP (Transmission Control Protocol)	TCP was particularly designed for devices equipped with more memory and processing power. The protocol will **require more handshaking** to initiate communication links **before any kind of message exchange**. Additionally, this will also affect communication times, wake-up and the long-term battery consumption. Devices connected with TCP often keep sockets open for each other with a persistent session. **This will also require additional power and memory requirements.**
Failure of single point with broker	**Brokers can also lead to single point failure in the network.** In most common scenarios that are likely to occur. A broker device is connected/plugged into a wall socket with other devices that are battery powered. In an event of power loss/failure, the publishing devices would continue to

	operate but the broker would be offline. The network would be useless until the power recovers.
Lack of security	As referred to above in the advantages. **MQTT is not equipped with built-in security**. In other words, MQTT is primarily unencrypted by default. This means that the protocol is unsecure and needs to take additional action to ensure **TLS/SSL (Transport Layer Security/Secure Socket Layer)** is implemented. Otherwise, any communications over MQTT will easily exploited and hacked.
Scalability (Impacts on centralized broker)	The **broker can impact on the scalability** as there is an additional overhead for each device that is connected. The only method to expand network connections is to equip a local broker hub that can support it. By doing this, it will limit the expansion for each hub and spoke group.

MQTT Security

MQTT Security Mechanism Best Practice

The MQTT protocol itself only specifies a few security mechanisms because it is commonly built upon SSL/TLS security standards. According to IBM, three concepts are fundamental to MQTT security [IBM]:

1. **Identity** – Naming the client that is being authorized and given authority. A client identifier, user ID and/or public digital certificate can be used to prove identity.
2. **Authentication** – About proving the identity of the client. A client authenticates a server with the SSL protocol while an MQTT server authenticates a client with the SSL protocol, password, or both.
3. **Authorization** – Managing the rights that are given to the client. This is not part of the MQTT protocol rather, it is provided by the MQTT servers (brokers/clients) – what it authorizes depends on what they dictate.

From a layer perspective, the security mechanisms can be divided into 3 layers – each layer preventing various kinds of attacks. These layers are:

Network - You should create a secure and trustworthy connection by using a physically secure network or VPN for all communication between clients and brokers. This solution is suitable for gateway applications where the gateway is connected to devices on the one hand and with the broker over VPN on the other side.

Transport – By default, MQTT relies on TCP transport protocol which does not use encryption. If confidentiality is the primary goal and you can afford the additional bandwidth, TLS should be used for transport encryption. This method is a secure and proven way to ensure that data cannot be read during transmission and provides client-certificate authentication to verify the identity of both sides.

Application - On the transport level, communication should be encrypted, and identities authenticated. The MQTT protocol provides a client identifier and username/password credentials to authenticate devices on the application level. Authorization or control of what each device is allowed to do is defined by the specific broker implementation. Therefore, it is critical that an appropriate and effective authorization policy is configured at the server level.

Known MQTT vulnerabilities

Mosquitto’s website makes clear a list of past vulnerabilities and when they were patched. Their website also makes it easy to report a security vulnerability if one is discovered. This open line of communication between users and developers helps keep Mosquitto secure.

However, in February of 2022, Kaspersky discovered 33 vulnerabilities in the MQTT protocol with 18 of them critical. They highlight that when using MQTT, authentication is completely optional and rarely includes encryption. This makes MQTT highly susceptible to man in the middle attacks (when attackers can place themselves between two parties while they communicate), meaning any data transferred over the internet could potentially be stolen.

Now since 2014, a total of 90 vulnerabilities in MQTT have now been discovered, many of which remain unpatched to this day. [Kapersky]

Recommendations

MQTT is a great protocol for reducing energy requirements, simplifying connectivity, and providing servers with the ability to message connected clients directly. However, it is important to understand the fundamentals of the protocol and research to ensure a safe connection.

For example, MQTT libraries that do not use encryption by default should not be used, and the use of strong passwords and API keys is always critical. Since MQTT is nothing more than a protocol, it can be, and should be used on top of encrypted messaging systems such as TLS, rather than the default TCP. This would force devices to store server certificates while allowing individual clients to access certificates. Thus, only connections where both parties (server and client) have authentic certificates can be processed, and man-in-the-middle attacks are impossible.

Sensors

Raspberry Pi

Application used for Raspberry Pi identification

- Nmap
  - https://nmap.org/

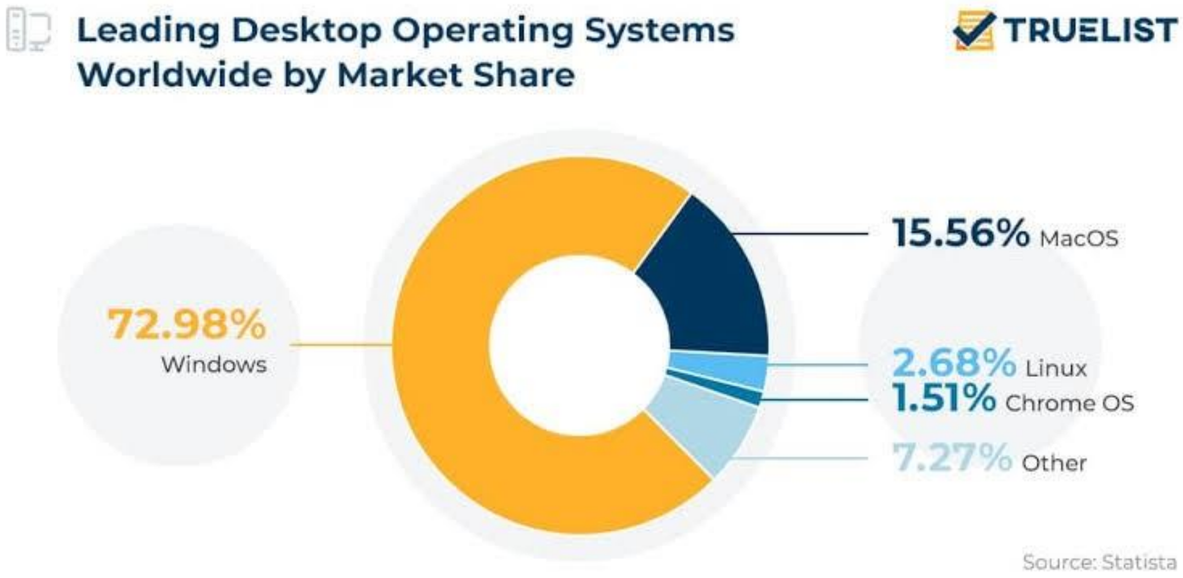
Application used for protecting against brute force attacks

- Fail2ban
  - https://www.Fail2ban.org/wiki/index.php/Main_Page

Nmap is an immensely powerful tool that can be used to enumerate devices connected to a network. It can also display information such as running services and the potential operating system running on the devices. Because of this and the vulnerability listed in CVE-2021-38759, it is possible to execute the command "sudo nmap -O -sT {network address in slash notation}" on a separate computer and this will list all the devices that are active, alongside what nmap believes to be the operating system on the devices and the services that are running and have ports open for network features. If a Raspberry Pi is found, it will appear as "Raspberry Pi Trading" next to the MAC address unless the MAC address itself has been changed. This is because the first 24 bits of the MAC address specify the vendor of the network interface. [5]

Once a Raspberry Pi and its services have been identified, it is possible with little knowledge to attempt to connect to the Raspberry Pi through these services and use the default username of "pi" and default password "raspberry." If these credentials are successful, then you will be able to perform functions on the Raspberry Pi that can access superuser commands.

As a Raspberry Pi is running its own distribution of Linux called Raspbian, which is a derivative of Debian [6], it is susceptible to malware attacks designed to infect Debian. However, in saying this, Linux based computers make up only a fraction of computers in comparison to other operating systems. Because of this, malware is more often designed to target operating systems that are more common such as Windows and MacOS. Its lack of popularity stems from it being a hard to use operating system. Besides this reputation, it is significantly more modular and malleable in a sense where it can be used in specific circumstances when a regular operating system such as MacOS or Windows cannot be used. Because of this, Linux is extremely common in enterprise settings such as servers, but uncommon in consumer devices including desktops and laptops.



After fixing CVE-2021-38759 by changing the default password, it is still possible to perform a brute force attack against the Raspberry Pi. As it is likely that you are still using the default pi account, the brute forcing process becomes significantly shorter as the username does not also need to be guessed. This is essentially the same as square rooting the number of guesses possible if you are using the same dictionary file for both username and password.

Using an application such as Fail2ban will help significantly limit the chance of brute forcing the correct password as it will ban a particular device from attempting to connect for 10 minutes if 5 failed authentication attempts are made within 10 minutes by default. This time can then be multiplied by 2^(n-1) where n is the number of times that a particular IP address has been banned, exponentially increasing the amount of time that a particular IP address will spend banned. This is not by default but can be easily configured by typing the following 2 commands into the /etc/Fail2ban/customisation.local file:

- bantime.increment = true
- bantime.factor = 1

These are just 2 basic commands that can be used to help prevent the password from being brute forced. It is also possible to implement formulas or randomness into the ban time to prevent a learning attacker from guessing the ban time which can help an attacker to minimise downtime and continue to brute force as soon as it knows that the ban time is over.

Fail2ban also works across many services and applications by default with including but not limited to sshd, apache-auth, php-url-fopen, openwebmail, proftpd, and courier-smtp. You are also able to add your own with a bit more understanding on how the program works. You can add your own service to be monitored by specifying "[{name of service}]" followed by the minimum of "logpath = {path to logfile}" in the /etc/Fail2ban/jail.d/{name of service}.conf file as this will allow Fail2ban to monitor that logfile for failed login attempts. This however is not enough as Fail2ban does not know how a failed login will look like. This is where you need to create a definition file called {name of service}.conf in the /etc/Fail2ban/filter.d/ directory. This file is where you create your definitions of what Fail2ban will look for to identify if a failed login attempt has been made. Upon making any changes, you need to make sure to restart Fail2ban for the changes to take effect. This can be done through the command "sudo service Fail2ban restart". [7]

An easy-to-understand definition file for WordPress where Fail2ban will understand what a failed login attempt will look like can be found here.

Wi-Fi

Applications used for Wi-Fi password cracking

- Wifite (Can capture WPA handshakes and crack passwords using the CPU)
  - https://github.com/kimocoder/wifite2
- Aireplay-ng
  - https://www.aircrack-ng.org/doku.php?id=aireplay-ng
- Aircrack-ng
  - https://www.aircrack-ng.org/
- Hashcat (Can crack passwords for a WPA handshakes file using a GPU which is significantly faster)
  - https://hashcat.net/hashcat/

An attacker can utilise these 4 applications to attack a Wi-Fi access point and its clients\network. Wifite is a remarkably simple to use application that can automatically obtain the password of an access point with only the use of a wireless network card. It can attack wireless networks using WPA or WEP encryption, it can attack networks using WPS, and it can also attack networks not broadcasting their SSID. By default, if an access point is utilising the WPA2 encryption standard, wifite would de-authenticate the users meaning that they would be temporarily disconnected from the network. This is in hopes that they would reconnect, perform the handshake with the access point and allow wifite to capture the handshake. This will allow wifite to begin performing an offline crack on the password with a dictionary attack. As wifite saved the captured handshake as a .cap file and defaults to using aircrack-ng, which utilises the CPU, wifite can be closed as to save computer resources and we can instead use a program like hashcat to crack the password as it utilises the GPU. This is significant as a GPU is much better at running tasks in parallel and can hence potentially attempt millions of passwords each second compared to thousands with a CPU. Aircrack-ng can be used to convert the .cap file to a .hccapx file which hashcat can then use with the syntax:

- aircrack-ng –j {.hccapx output file} {.cap file}

Hashcat can begin cracking the file using the syntax

- hashcat -m 2500 {.hccapx file} {dictionary file}

Pairing a program like hashcat and a dictionary file like rockyou.txt or rockyou2021.txt can prove to be a strong combination as passwords in both of those files have been used by real people before with 32 million and 82 billion passwords, respectively. Each of these files can be obtained very easily. rockyou.txt comes with a standard installation of Kali Linux, and rockyou2021.txt can be torrented after getting the link from 1 Google search.

It is possible to permanently de-authenticate a specific client connected to a Wi-Fi access point with wifite and aireplay-ng. You can perform reconnaissance of the devices connected to an access point with:

- wifite –nodeauth

This will still attempt to capture a handshake but will do so without de-authenticating the already connected clients. This will also show the MAC addresses of connected clients and the access point itself. With the specific client that we would like to target, we can use aireplay-ng with the command:

- aireplay-ng –deauth 0 –a {access point MAC address} -c {client MAC address} {wireless interface name}

Just by filling in the fields, we can permanently disconnect the client from the access point given that the attacker stays within Wi-Fi range.

By using the first method, we can also implement a proxy to spy on any information being sent across the network, which can include information generated by the sensors and health data which is sensitive data to the user. This is assuming that there is no encryption taking place on the data as it is being sent across the network. This breach in privacy can cause an attacker to steal and sell this information while causing Redback Operations to face potential fines, loss of trust from consumers, and can lead Redback Operations to bankruptcy.

By using the second method, the client will not be able to properly send data to the servers or the Raspberry Pi which can result in client downtime. This will annoy a customer and will tempt them to instead purchase from a competitor such as Zwift.

Application to change MAC address

- Macchanger (Can change the MAC address of the attacker's network interface to disguise the attacker)
    - https://www.kali.org/tools/macchanger/

Some access points have systems in place to prevent unauthorised devices from connecting to the network. This can be done throughout the network from a centralised device like a server to an end point like an access point. The way that these systems work is to check the MAC address of the connected device and see whether the device should be allowed on the network. This can simply be circumvented by seeing the MAC addresses of all Wi-Fi connected devices by using the "wifite --nodeauth" command and selecting the network that you would like to view. The MAC addresses of all connected devices will be enumerated in the terminal. You can then use one of these MAC addresses to bypass this authentication system of MAC address checking. This can be accomplished by executing these 3 commands:

- ifconfig {interface name} down
- macchanger –m {MAC address} {interface name}
- ifconfig {interface name} up

The access point that you are connecting to will now only see the new MAC address that you have input instead of the hardware encoded MAC address of your wireless networking interface.

To help negate the issues that plague WPA2, WPA3 does exist. This protocol is the latest offering from Wi-Fi Alliance and offers a handful of security improvements over WPA2. The prevention of offline dictionary attacks is enforced with a technology called Simultaneous Authentication of Equals (SAE) [8]. This requires an attacker to stay in range of the access point to crack the password which even then has become significantly more time consuming which makes it infeasible to crack. This, however, does not mean that it is impervious to attacks. Through the use of timing attacks, an attacker can take educated guesses as to what the password may be [9]. With this being said, there are only minimal tools available for cracking the WPA3 security standard at the time of writing which will prevent many script kiddies from attempting to attack a network, significantly reducing the potential number of attackers to those that know how to exploit WPA3. As WPA3 can be implemented with a firmware update to devices, I would recommend implementing this security protocol where possible.

Another device that can help with protecting the security of a Wi-Fi network is utilising the Wi-Fi 6E wireless standard. Wi-Fi 6E utilises the 6GHz frequency band which means that more than a firmware update is required, new devices that are capable of connecting to Wi-Fi 6E access points will be required [10]. As this technology is only relatively new and has only been cleared for use in Australia on the 3rd of March 2022 by the ACMA [11], there are few devices that do have support meaning that attackers physically cannot attack these networks without first upgrading their devices. Not to mention as frequencies increase, their power for both penetrating objects and travelling further decreases. Although it has not been thoroughly tested in Australia as the ACMA has only approved the lower end of the spectrum for use, it is estimated that Wi-Fi 6E will only travel approximately 5-8m [12]. This lack of range and different frequency used can prove vital to, at least of the time being, the reliability of the products to work without interference from adversaries trying to cause disruptions. 1 other benefit that Wi-Fi 6E can provide is that because it is using the 6GHz frequency, and as there are very minimal devices in a home that use this frequency, it will face less interference from other devices. This can prove useful in areas such as apartment complexes where there are many wireless signals in proximity at any given time.

Bluetooth

Applications used for Bluetooth interference

- Bettercap (predecessor is Ettercap)
  - https://github.com/bettercap/bettercap

Potential hackers would be able to breach any devices connected via Bluetooth by using a tool called Bettercap which is a successor to the tool Ettercap. Bettercap comes with a Bluetooth Low Energy suite which allows the hacker to look at nearby Bluetooth devices. The hacker can scan for mac addresses in range then use that to connect to the device and intercept any information it receives or sends. The hacker would also be able to write data to the device to exploit it or tag it to track the device over time. Once the hacker has started Bettercap using the "sudo bettercap." The network module will start which begins to search for devices on the same network. Using the command "net.show" the program will list all devices on the network as well as their IP and MAC address.

In order to start the Bluetooth sniffing module, the hacker would need to type "ble.recon on" which starts to discover devices. Using the command "ble.show" the program will show all devices it was able to discover with their MAC addresses exposed. The hacker will then be able to initiate a scan with any of the devices using the command "ble.enum [MAC address]" which while then provide the hacker with services that can allow the hacker to write data to the device provided those services on the device are exploitable. To write a value to the device, the user would have to type the command:

"ble.write [MAC address] [Field] [Value]"

Doing so can allow the hacker to exploit the device. The program can also allow the user to assign a value to a device to identify it. For example, if a device changes it is MAC address. The program would still be able to find the device through that unique value [14]

- Kali – BT Recon

Kali Linux could potentially be used as another tool to infiltrate Bluetooth devices. It has multiple tools built into its suite which can allow the hacker to interact with devices on the network. There is tool like "ifconfig" called "hciconfig" which instead of searching for wifi cards on your device it searches for Bluetooth cards and enables it to search for devices nearby. Using the command "hciconfig [name of device] up" will enable the Bluetooth device and allow it to begin searching. Running the command "hciconfig" again checks to see if the device is running [16]

Now to scan for devices using the tool “hcitool.” This tool looks for devices that are sending out discovery beacons. It can be used to do various tasks such as scans and data inquiries, however some commands require knowing the MAC address of the said devices to use them. Performing a scan using the command “hcitool scan” will use the Bluetooth technology to scan for nearby devices and display their MAC addresses in order to do more scans, perform inquiries or attempt to get the name of said device. Once the MAC address has been found, using the command “hcitool name [MAC address]” will display the name of the device. To learn more about the device, you can use the “inq” command: “hcitool inq [MAC address].” This will display clock offset and the class of the device. The class shows what kind of Bluetooth device it is which will then allow you to view the code for that class on the Bluetooth website [16]

Another tool that Kali Linux contains is called “sdptool.” This tool allows you to learn more about what is on the device and learn more about the properties. To start off type “sdptool browse” followed by the MAC address of the device found earlier when using “hcitool.” This will show more information about the protocol which could potentially lead to finding any vulnerabilities within the device. You can also find out if the device can be communicated with directly or if it is using MAC randomisation. [16]

Using that information, the next tool to use is “l2ping” which will allow you to ping the device using the MAC address regardless of if they are in discovery mode or not. Using the command “l2ping [MAC address]” the program will attempt to ping the device and if it comes back successful that would mean that the device is within range and can be reached. [16]

- Kali – BTScanner

Another tool which Kali Linux has which is more user friendly is called “btscanner.” It is more a GUI which may make it easier to discover devices. Once started up, typing in “I” will start an inquiry scan to find nearby devices and allows you connect to them or set a command. Once the scan is completed, pressing “enter” will display more details about the device. The name of the device, the owner and various other features are shown. This would allow the hacker to gain more information about the device be able to figure out if there could be any potential vulnerabilities with any of the protocols or features used within the device. [15]

Vulnerabilities	Description
Eavesdropping	○ Potential IoT sensors used could be at risk of hackers eavesdropping on data due to using outdated Bluetooth protocols [13]

	○ To mitigate this, make sure to use IoT sensors that use the latest Bluetooth version and protocols and to ban potential devices that using outdated protocols
Denial of service	○ These kinds of attacks can crash the device as well as the software and the servers that hold data [13] ○ Initiate code which turns off the sensors when they are not receiving data or have had no activity for a certain amount of time

References

IoT Security and IoT Network Security

[1] S. Shea and I. Wigmore, "What is IoT Security? - Definition from TechTarget.com", *IoT Agenda*, 2022. [Online]. Available: https://www.techtarget.com/iotagenda/definition/IoT-security-Internet-of-Things-security.

[2] A. Katrenko and E. Semeniak, "Internet of Things (IoT) Security: Challenges and Best Practices", *Apriorit*. [Online]. Available: https://www.apriorit.com/dev-blog/513-iot-security.

[3]"What is the CIA Triad?", *Forcepoint*, 2022. [Online]. Available: https://www.forcepoint.com/cyber-edu/cia-triad.

[4] K. Kandasamy, S. Srinivas, K. Achuthan and V. Rangan, "IoT cyber risk: a holistic analysis of cyber risk assessment frameworks, risk vectors, and risk ranking process", *EURASIP Journal on Information Security*, vol. 2020, no. 1, 2020. Available: https://jis-eurasipjournals.springeropen.com/articles/10.1186/s13635-020-00111-0.

[5]"What is the Internet of Things (IoT)?", *Oracle.com*, 2022. [Online]. Available: https://www.oracle.com/au/internet-of-things/what-is-iot/.

[6]"What is the CIA Triad and Why is it important? | Fortinet", *Fortinet*, 2022. [Online]. Available: https://www.fortinet.com/resources/cyberglossary/cia-triad#:~:text=The%20three%20letters%20in%20%22CIA,and%20methods%20for%20creating%20solutions.

[7] E. Boehm, "Top 10 IoT Vulnerabilities in Your Devices", *Keyfactor*, 2022. [Online]. Available: https://www.keyfactor.com/blog/top-10-iot-vulnerabilities-in-your-devices/#weak-guessable-passwords-4.

[8]"Are IoT Systems Too Easily Hacked, And What Would Be The Fix?", *Techvice*, 2022. [Online]. Available: https://techvice.org/blog/popular/iot-systems-hacked/.

[9] J. Wallen, "Classic SysAdmin: Understanding Linux File Permissions - Linux Foundation", *Linux Foundation*, 2022. [Online]. Available: https://linuxfoundation.org/blog/classic-sysadmin-understanding-linux-file-permissions/#:~:text=read%20%E2%80%93%20The%20Read%20permission%20refers,the%20contents%20of%20a%20directory

[10] J. Clark, "What is the Internet of Things, and how does it work?", *IBM Business Operations Blog*, 2016. [Online]. Available: https://www.ibm.com/blogs/internet-of-things/what-is-the-iot/.

[11]"Wearable Devices and the Internet of Things", *Au.mouser.com*, 2022. [Online]. Available: https://au.mouser.com/applications/article-iot-wearable-devices/.

[12] Cyril, "COMPREHENSIVE GUIDE TO PENETRATION TESTING (SECURITY TESTING)", *Secure Triad*, 2022. . Available: https://securetriad.io/penetration-testing/.

[13] F. Knott, "IoT Security: Hardware or Software?", *Archonsecure.com*, 2022. [Online]. Available: https://www.archonsecure.com/blog/iot-security-hardware-software.

[14] K. Sadique, R. Rahmani and P. Johannesson, "Towards Security on Internet of Things: Applications and Challenges in Technology", *Procedia Computer Science*, vol. 141, pp. 199-206, 2018. Available: https://www.sciencedirect.com/science/article/pii/S1877050918318180.

[15] M. Nichols, "Cryptography is an essential part of IoT design", *i-SCOOP*, 2022. [Online]. Available: https://www.i-scoop.eu/internet-of-things-iot/iot-cryptography/.

[16] Y. Al-Hadhrami and F. Hussain, "DDoS attacks in IoT networks: a comprehensive systematic literature review", *World Wide Web*, vol. 24, no. 3, pp. 971-1001, 2021. Available: https://link-springer-com.ezproxy-b.deakin.edu.au/content/pdf/10.1007/s11280-020-00855-2.pdf.

[17] A. Dahiya and B. Gupta, "How IoT is Making DDoS Attacks More Dangerous?", *Insights2Techinfo*, 2021. [Online]. Available: https://insights2techinfo.com/how-iot-is-making-ddos-attacks-more-dangerous/#:~:text=In%20summary%2C%20it's%20been%20shown,of%20IoT%20devices%20increases%20exponentially.

[18]"IoT Security Issues, Threats, and Defenses", *Trend Micro*, 2021. [Online]. Available: https://www.trendmicro.com/vinfo/us/security/news/internet-of-things/iot-security-101-threats-issues-and-defenses.

[19] I. Tudosa, F. Picariello, E. Balestrieri, L. De Vito and F. Lamonaca, "Hardware Security in IoT era: the Role of Measurements and Instrumentation", *2019 II Workshop on Metrology for Industry 4.0 and IoT (MetroInd4.0&amp;IoT)*, 2019. Available: https://www.researchgate.net/publication/334519662_Hardware_Security_in_IoT_era_the_Role_of_Measurements_and_Instrumentation.

[20]"The Security, Privacy and Legal Implications of the Internet of Things ("IoT") Part one – The Context and Use of IoT", *Data Protection Report*, 2015. [Online]. Available: https://www.dataprotectionreport.com/2015/05/the-security-privacy-legal-implications-of-the-internet-of-things-iot-part-one-the-context-and-use-of-iot/.

Message Queuing Telemetry Protocol

[1] "MQTT Version 3.1.1. Edited by Andrew Banks and Rahul Gupta. OASIS Standard", Oct. 2014, [online] Available: http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html

[2] S. Bandyopadhyay and A. Bhattacharyya, "Lightweight Internet protocols for Web enablement of sensors using constrained gateway devices", Proc. Int. Conf. Comput. Netw. Commun. (ICNC), pp. 334-340, Jan. 2013.

[3] Mishra, B., 2020. The Use of MQTT in M2M and IoT Systems: A Survey. *IEEE Access*, [online] 8. Available at: https://ieeexplore.ieee.org/document/9247996.

[4] Eclipse Mosquitto. 2022. *Eclipse Mosquitto*. [online] Available at: https://mosquitto.org/.

[5] Ibm.com. 2022. *IBM Docs*. [online] Available at: https://www.ibm.com/docs/en/ibm-mq/7.5?topic=m2m-mqtt-security.

[6] usa.kaspersky.com. 2022. *33 vulnerabilities found in the data transfer protocol for wearable medical devices*. [online] Available at: https://usa.kaspersky.com/about/press-releases/2022_33-vulnerabilities-found-in-the-data-transfer-protocol-for-wearable-medical-devices.

[7] O'Reilly Online Learning. 2022. *Analytics for the Internet of Things (IoT)*. [online] Available at: https://www.oreilly.com/library/view/analytics-for-the/9781787120730/db65d957-cf17-459c-a203-4b8234a14261.xhtml.

[8] D. Silva, L. Carvalho, J. Soares and R. Sofia. 2022. *A Performance Analysis of Internet of Things Networking Protocols: Evaluating MQTT, CoAP, OPC UA*. Available at:<https://eds.s.ebscohost.com/eds/ebookviewer/ebook/bmxlYmtfXzE1MDU5MTVfX0FO0?sid=60a5769d-4773-4c8a-8dca-b21098236746@redis&vid=1&format=EB&ppid=pp_17> [Accessed 16 May 2022].

[9] I. Hubschmann, "The Pros and Cons of Using MQTT Protocol in IoT", *Nabto*, 2022. [Online]. Available: https://www.nabto.com/mqtt-protocol-iot/. [Accessed: 16- May- 2022].

[10] D. Liadov, "MQTT v5 - New Features, Pros & Cons, Challenges", *MobiDev*, 2022. [Online]. Available: https://mobidev.biz/blog/mqtt-5-protocol-features-iot-development. [Accessed: 16- May- 2022].

[11] G. Hillar, *MQTT Essentials - A Lightweight IoT Protocol*. United Kingdom: Packt Publishing, 2017, pp. 9 - 58. Available:<https://eds.p.ebscohost.com/eds/ebookviewer/ebook/bmxlYmtfXzE1MDU5MTVfX0FO0?sid=0e11e9be-7f87-46d1-8320-6c307bba68f8@redis&vid=9&format=EB&rid=8> [Accessed 16 May 2022]

[12] Inc., "What is AMQP Protocol", *Wallarm.com*, 2022. [Online]. Available: https://www.wallarm.com/what/what-is-amqp. [Accessed: 16- May- 2022].

[13] G. Perrone, M. Vecchio, R. Pecori and R. Giaffreda, "The Day After Mirai: A Survey on MQTT Security Solutions After the Largest Cyber-attack Carried Out through an Army of IoT Devices", Scitepress, Italy, 2022.

[14] I. Skerrett and F. Raschbichler, "MQTT 5 Essentials", *Hivemq.com*, 2019. [Online]. Available: https://www.hivemq.com/blog/mqtt5-essentials-part3-upgrade-to-mqtt5-now/. [Accessed: 16- May- 2022].

Sensors

[5] hobbitakhilesh, "Introduction of MAC Address in Computer Network", *GeeksforGeeks*, 2021. [Online]. Available: https://www.geeksforgeeks.org/introduction-of-mac-address-in-computer-network/. [Accessed: 11- May- 2022].

[6] P. Wise, "RaspberryPi", *Debian Wiki*, 2013. [Online]. Available: https://wiki.debian.org/RaspberryPi#:~:text=Raspberry%20Pi%20OS%20(formerly%20Raspbian)%20and%20Debian,-The%20most%20often&text=Raspberry%20Pi%20OS%20is%20not,Debian%20on%20your%20Raspberry%20Pi's. [Accessed: 11- May- 2022].

[7] Webdock, "How to configure Fail2Ban for common services", *Webdock*, 2021. [Online]. Available: https://webdock.io/en/docs/how-guides/security-guides/how-configure-fail2ban-common-services. [Accessed: 18- May- 2022].

[8] Wi-Fi Alliance, "Security", *Wi-Fi Alliance*, 2022. [Online]. Available: https://www.wi-fi.org/discover-wi-fi/security. [Accessed: 24- May- 2022].

[9] Hak5, "Hacking WPA3 with Mathy Vanhoef & Retia", *YouTube*, 2021. [Online]. Available: https://www.youtube.com/watch?v=44I1wfgGT80. [Accessed: 24- May- 2022].

[10] Wi-Fi Alliance, "Wi-Fi Alliance® brings Wi-Fi 6 into 6 GHz", *Wi-Fi Alliance*. [Online]. Available: https://www.wi-fi.org/news-events/newsroom/wi-fi-alliance-brings-wi-fi-6-into-6-ghz. [Accessed: 24- May- 2022].

[11] Wi-Fi Coops, "It's Time for Wi-Fi 6E Down Under!", *Wi-Fi Coops*, 2022. [Online]. Available: https://wificoops.com/2022/03/09/its-time-for-wi-fi-6e-down-under/. [Accessed: 24- May- 2022].

[12] R. Shaw, "Wi-Fi 6E AX 6Ghz now approved in Australia. What does that mean for you?", *Cybershack*, 2022. [Online]. Available: https://cybershack.com.au/consumer-advice/wi-fi-6e-ax-6ghz-now-approved-in-australia-what-does-that-mean-to-you/. [Accessed: 24- May- 2022].

[13]"Bluetooth Attacks and How to Secure Your", Webroot.com. [Online]. Available: https://www.webroot.com/au/en/resources/tips-articles/a-review-of-bluetooth-attacks-and-how-to-secure-mobile-workforce-devices.

[14]"How to Target Bluetooth Devices with Bettercap", WonderHowTo, 2019. [Online]. Available: https://null-byte.wonderhowto.com/how-to/target-bluetooth-devices-with-bettercap-0194421/.

[15]"How to scan Bluetooth Devices in Kali Linux using Btscanner", Hack Today. [Online]. Available: https://www.hacktoday.com/how-to-scan-bluetooth-devices-in-kali-linux-using-btscanner/.

[16] J. Meyers, "BT Recon: How to Snoop on Bluetooth Devices Using Kali Linux", WonderHowTo, 2020. [Online]. Available: https://null-byte.wonderhowto.com/how-to/bt-recon-snoop-bluetooth-devices-using-kali-linux-0165049/.