



University of London

6CCS3PRJ Final Year

A Security Assessment of a Voice-based Virtual Assistant

Final Project Report

Author: Ronak Kapadia

Supervisor: Dr Jose Such

Student ID: 1722325

April 23, 2020

Abstract

Smart speakers with integrated voice assistants like Google Home or Amazon Alexa are becoming increasingly popular and its use widespread in the digital lives of many. Being often people's first standalone voice-controlled product, it offers a glimpse of the truly hands-free digital future. Praised for its potential to increase one's productivity to allow for, often automated, multitasking. Third-parties are finding ways to interconnect their own products to the smart home ecosystem, from a refrigerator to an entire home security system.

This project's outcome is to analyse and evaluate the privacy and security measures, or lack thereof, of a voice-based virtual assistant. This comprehensive report will outline and describe the findings, methodologies and the tools used to form this security assessment.

Originality Avowal

I verify that I am the sole author of this report, except where explicitly stated to the contrary. I grant the right to King's College London to make paper and electronic copies of the submitted work for purposes of marking, plagiarism detection and archival, and to upload a copy of the work to Turnitin or another trusted plagiarism detection service. I confirm this report does not exceed 25,000 words.

Ronak Kapadia

April 23, 2020

Acknowledgements

I'd like to take this opportunity to thank my supervisor, Dr Jose Such. His practical knowledge helped me tremendously in shaping this project in an effective and concise way. His guidance and feedback not only allowed me to continuously improve the project's deliverables, but also taught me crucial research and problem-solving skills, which I will use throughout my life.

Thank you, Dr Such.

I would also like to thank my family for their patience and, much-needed, moral support throughout my first major research project.

Thank you, Mum and Dad.

Contents

1	Introduction	2
1.1	Project Motivation	3
1.2	Project Aims/Objectives	3
1.3	Virtual Assistant	3
1.4	Project Scope	4
1.5	Report Structure	5
2	Background	6
2.1	Amazon Echo Dot Device	6
2.2	Architecture	7
2.3	Previous Vulnerabilities	9
2.4	Related Work	11
2.5	Penetration Testing	12
3	Design & Specification	14
3.1	Penetration Testing	14
3.2	Risk Assessment	15
4	Penetration Testing	17
4.1	Tools and Programs	17
4.2	Echo set-up	19
4.3	Echo to the cloud	25
5	Skill and Privacy	30
5.1	Third-party Skill	30
5.2	Privacy	30
6	Legal, Social, Ethical and Professional Issues	33
7	Conclusion and Future Work	35
	References	38

Chapter 1

Introduction

Throughout our existence, humans have searched for methods to increase efficiency and productivity. From prehistoric tools for construction to automobiles for transport. This search for efficiency is especially apparent in the modern iteration of the information age. From stationary PC desktops, to internet-connected watches.

Voice-based virtual assistants mark an important milestone in this thirst for technological innovation. For decades, we have been conditioned to use keyboards, mice, touchscreens and other physical interface mediums. Now, instead of advancing the portability or computational power, this relatively new technology seeks to change the way we, as consumers, interact with our digital devices.

These smart speakers with integrated voice assistants are devices that were designed to sit, inconspicuously, somewhere with sizeable traffic in the room/household. While idle, their only task is to wait and try to detect a user saying a 'wake word'. These wake words are specific words/phrases that activate the smart speaker's assistant and prepare it to record a voice command from the user. Once the command is issued, and understood by the assistant, it is analysed and a response returned (often verbally).

These types of devices have penetrated the consumer market quite rapidly and recently. Hence, research into the privacy and security implications of these 'always on' devices is, unfortunately, limited - and overlooked by the average consumer.

1.1 Project Motivation

Cyber-security has always remained an area in computing that I am fascinated with. From, how one small overlooked development decision could lead to an entire product being compromised and exploitable, to how encryption utilises the same drawbacks of modern computing to protect the data modern society depends on. This project topic brings two vital (and my favourite) research areas into one, security and emerging technology.

These virtual assistants being a relatively new technology means that my efforts and discoveries could lead to positive change, big or small, in this emerging market.

1.2 Project Aims/Objectives

The primary objective of this project is to deliver a security assessment of a voice-based virtual assistant. Essentially, the aim is to reach a well-informed conclusion on how a specific virtual assistant performs in various security tests, to determine how secure it is. The security assessment will involve; the analysis of existing work and using established penetration testing methodologies/techniques to perform specific tests.

Another vital aim is to assess the privacy protocols set in place to help protect users sensitive data. Data privacy is key in ensuring a secure and trustful partnership between end customers and manufacturers. Excellent security means very little when identifiable personal data is being misused.

1.3 Virtual Assistant

Cost and market share, were the main criteria when choosing a voice assistant. Most new smart speakers with integrated voice assistants start from a reasonable price, hence this was not a major concern. A voice assistant with established market dominance has two major advantages - first, there is likely more public resources available (e.g. scientific papers). Secondly, a large market share ensures that the findings from this project will have significant real-life implications for the users of such a device.

Amazon Echo Dot (3rd generation) with integrated Amazon Alexa voice assistant was the chosen product. A standalone device for £30 was within the project's budget. In 2018, Amazon Alexa powered devices controlled 68% [1] of the standalone voice assistant market within the UK. Although its growth and share is likely to slide in the coming years, it is still the dominant

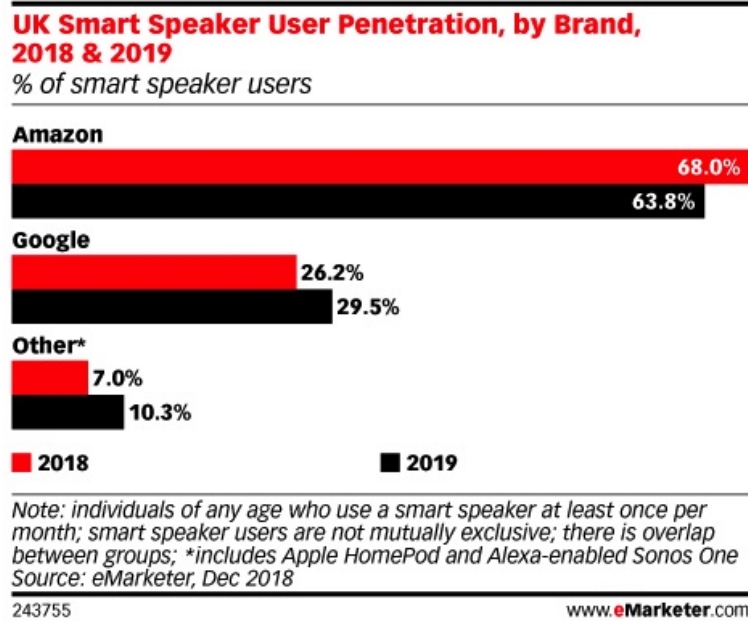


Figure 1.1: Originally from [1]

and most widely used assistant. Therefore, an appropriate subject for this project.

1.4 Project Scope

The scope of this project is to perform a security assessment of a commercial product. The area of cyber-security, in general, is rather large and complex - also often well guarded by manufactures/developers. Hence, why I will only be utilising publically available tools/information in the assessment. This being an undergraduate final project the main constraint will be time management with other academia commitments. Project funding is private, hence the need to limit expenditure where possible.

Two limitations have been set for this project:

1. No additional special hardware - e.g. Alexa remote, wireless antennas, third-party products etc.
2. No additional special software - e.g. pay to use tests

The Echo Dot and a smartphone (Alexa app), will be the only physical devices obtained for this project. The reason for no additional hardware (e.g. Alexa remote, wireless antennas,

third-party products etc.) is to try to keep testing within the scope of the project. Every extra device linked to the Echo Dot will add to the total amount of tests required, and since this is a resource and time-restricted project it will detract from the testing of Alexa itself. The same logic applies to special (pay to use) software. Each additional hardware/software opens up a 'Pandora's box', which will, unfortunately distract focus from the main objective. The lower need for additional resources also encourages and improves the replicability/reproducibility of ideas explored throughout the project.

1.5 Report Structure

To achieve the objectives set out earlier, the project/report has three main phases:

1. Assess the feasibility of known general attacks on Amazon Alexa, this will be primarily be accomplished through a comprehensive literature review.
2. Conduct a thorough penetration test on the Alexa assistant (Echo Dot).
3. Conduct a risk assessment based on data from 1. and 2. Risk assessments of attacks/vulnerabilities will be completed in their respective sections, the general risk assessment of the assistant will be completed in the Conclusion chapter.

The background review will follow this introductory chapter. The second chapter analyses various pieces of existing literature, ranging from scientific papers to articles. Aiming to provide readers with essential background knowledge regarding this topic.

The third chapter outlines what methodologies were used throughout the project.

Chapter four includes all the penetration testing completed, it is divided by the architecture of the Echo Dot. This is because tests were completed based on their positioning to the architecture. Chapter five includes all work completed related to the creation of an Alexa Skill and privacy aspects/concerns. The analysis and evaluation of testes and other explored concepts are completed within the respective chapter for the test.

The final chapter (six) includes the limitations and problems faced in the project, future work, and a conclusion.

Chapter 2

Background

This chapter aims to provide the reader with crucial background knowledge about the project topic. It is mainly comprised of a critical analysis of existing literature related to; the Echo architecture environment, previous Echo vulnerabilities and penetration testing research.

2.1 Amazon Echo Dot Device

2.1.1 Hardware

An Amazon Echo Dot itself is quite simple and includes some basic computing components. The device has four buttons on top, two for volume control (increase and decrease), disable microphone and an action button to active Alexa without saying the wake word. There is also a ring light on the top encompassing the Dot's circumference, it turns on then Alexa is activated - essentially a sign that Alexa is either speaking or listening. It's only input is a (USB??) charging port, and has a 3.5mm auxiliary port for output.

Within, the Dot houses; a total of four microphones, a 1.3GHZ Quad-Core 64-bit CPU by MediaTek (MT8516), a combination (depending on country) of 4BG RAM and 4GB eMMC flash memory and a MediaTek dual-band Wi-Fi and Bluetooth controller.

While most of the internals are quite basic and expected, it seems to have more memory than most would expect for a smart speaker. Up to 8GB of flash memory for a device that does very little computation locally, as well as a small and modest CPU - is quite odd.

2.1.2 Software

Details pertaining to the Dot’s firmware is limited, Amazon does not make this information widely accessible. Based on this post [2] the previous-generation Echo Dot runs on Amazon’s FireOS, in addition to many other Amazon devices, like FireTV and Kindle. Hence it is appropriate to assume that this device runs on FireOS as well. FireOS is a forked version of the Andriod operating system, created and maintained by Amazon.

2.2 Architecture

Understanding the smart speaker digital environment is vital before any security tests or assumptions are made. The environment is quite different from many other modern digital devices. For example with PC/laptops, depending on the task you are performing, part of it is being computed locally utilising your hardware; a separate part might be offloaded for calculation through the internet. Most laptops also offer a reasonably high degree of customizability, especially when it comes to aspects like OS configuration and program installation.

The primary way the Echo Dot¹ handles requests/commands is by sending and retrieving data to and from Amazon’s cloud servers. Very little is being computed on the device locally, services like weather and music use external services through the internet. When users speak to Alexa², the voice command analysis is being examined in the cloud. This intrinsic difference in computation provides a basic idea on how certain security loophole areas may be closed (e.g. downloading malicious programs locally) and some open (e.g. intercepting audio transmissions).

Smart speakers will often communicate the bulk of their request to the manufacture’s cloud service. They will then decode and fulfill the request either verbally through the speaker or another inter-connected device (e.g. smartphone app). The process is also similar to any third-party products or skills. Alexa Skills can be thought of as small extensions to the Alexa experience, they are similar to applications (apps) installed on a smartphone. Skills can be pre-installed, like weather and Amazon music; or they can be third-party, like Hive, Phillips Hue, etc. Instead of fulfilling the request through the manufacturer’s servers, it is sent to the appropriate skill/product’s servers to be processed. Then the request is either a verbal output (for skills) or via other hardware (third-party products).

¹‘Echo’, ‘Dot’ or ‘Echo Dot’ will be used to refer to the Amazon Echo Dot (3rd generation) smart speaker.

²‘Alexa’ refers to the Amazon Alexa virtual assistant.

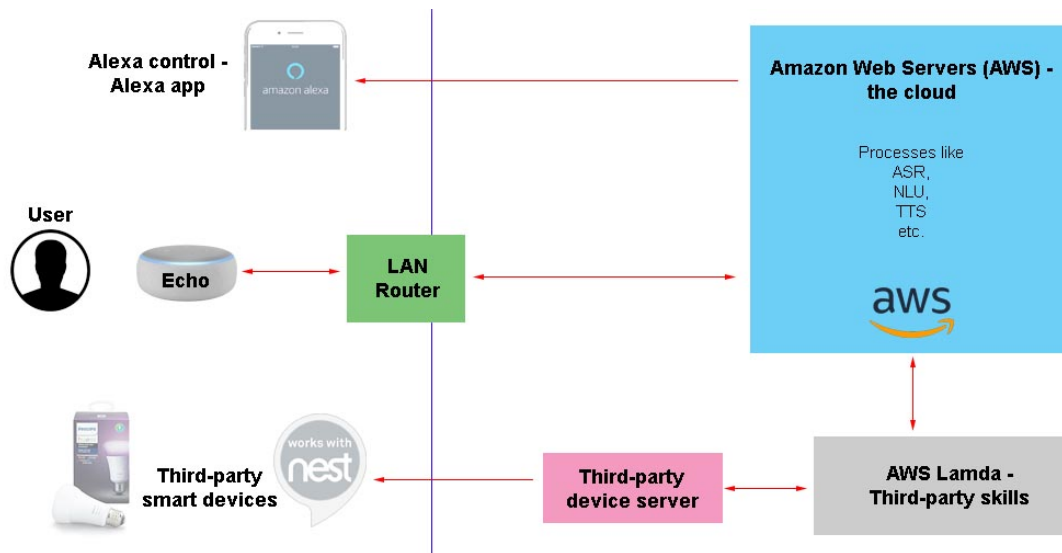


Figure 2.1: Overview of the Amazon Echo architecture

Figure 2.1 represents the Amazon Alexa architecture. When the user issues the wake word, the Echo activates, and once the speech that finished it sends the subsequent audio to Amazon’s cloud service³. The first system that encounters the transcript is Automatic Speech Recognition (ASR) this is where the audio is decoded formed into, multiple, likely strings of text. For example, the command ‘what is the weather like today’, can also be interpreted by ASR as ‘watt is the whether like two day’. Amazon then gives each possible string a score, this utilises machine learning algorithms to determine the most likely command issued at the given time. The string with the highest score is chosen.

The string of text then enters the Natural Language Understanding (NLU), this attempts to match intent to the string. The NLU’s main goal is to deconstruct the string into smaller data pieces (keywords) to help it decipher what the user actually wanted. With our previous example, the keyword ‘weather’ tells the NLU that this command is likely for the Amazon weather skill; ‘today’ then represents the time period.

Text-to-speech (TTS) is then used to formulate the verbal output for the user.

Since this type of environment is quite different from other modern digital devices, as most communication and processing is completed externally. This means some of the more ‘established’ and ‘conventional’ security attacks are infeasible - or at least severely reduced. Traditional

³<https://www.amazon.co.uk/gp/help/customer/display.html?nodeId=GA7E98TJFEJLYSFR>

devices, like most PC/laptops, are vulnerable to more diverse problems like [3];

- Operating system (OS) security vulnerabilities: As PC OS's have been in circulation longer, and more widespread, researchers have had more time and resources dedicated to its security testing. This penetration testing environment has led to many vulnerabilities being discovered and patched (fixed).
- Downloading malicious programs/applications: Users are only able to download and install approved Alexa skills through the skills store. Amazon performs security reviews⁴ on these skills before public release.
- Exploiting out-dated programs: Since over-the-air updating of Alexa skills and OS is used, updates are automatically pushed. Meaning the user rarely has to worry about using outdated firmware.

2.3 Previous Vulnerabilities

A large section of background research for this project was dedicated to previous Echo/Alexa vulnerabilities.

This being my first penetration testing experience, researching previously known bugs (vulnerabilities/exploits) would provide a realistic insight into Echo security testing. I can learn from how these vulnerabilities came to light and base my techniques on tests which were already successful in finding weaknesses. Essentially, having a foundation/plan rather than of going in blind.

Another reason for this research is to test some of these vulnerabilities, to test whether appropriate and effective fixes were developed.

The first vulnerability, discovered at DEF CON 26 by the Tencent Blade Team details a fascinating way to convert an Echo device into a listening tool on a local network [4]. This attack depended on a weakness in the in-built Echo program Whole Home Audio Daemon (WHAD). This program is used by Echo devices within the same network to communicate with each other. The attack requires two Echo devices; a target device and an attacker device. First, the team had to gain privileged access (root) to their attacker device by modifying the firmware which required removing its flash chip. They then needed to, already, have access to the target's Wi-Fi network. To ensure both Echos can recognise each other the attacker Echo must be linked

⁴<https://developer.amazon.com/en-US/docs/alexa/custom-skills/security-testing-for-an-alexa-skill.html>

to the target’s Amazon account, this was accomplished by using multiple web vulnerabilities with Alexa including HTTPS downgrade and URL redirection. Once connected the WHAD vulnerability allowed the group to hijack the target’s device and silently record conversations.

Team Fluoroacetate at Pwn2Own 2019 [5], showcased an exploit relating to the Echo Show 5⁵. They discovered that the device used an old, modified, version of the Chromium web browser. Once connected to a malicious hotspot, the team used an integer overflow bug in the browser to leak data from the speaker.

A security researcher discovered a way of utilising lasers to activate smart speakers [6]. The attack involves a high-powered industrial laser which is capable of altering intensity and frequency. The speaker’s microphones convert the light into an electrical signal, like sound. Varying laser intensity at specific frequencies allows for the attacker to give speakers certain commands, similar to normal voice commands, possibly from up to 70 meters away.

One method of exploitation is seeing more attention in recent times, this is through malicious third-party Alexa skills. Checkmarx’s researchers unearthed new ways to use Alexa skill development for eavesdropping, they exploited tools given to them by Amazon (skill squatting). Components like ‘reprompt’ and ‘outputSearch’ verbally inform the user before Alexa will close a skill, usually because the user fails to give the required verbal input. The parameters of these components could be set to empty output-speech (silence). This means that after most users would expect Alexa to stop listening, the Echo is still active and the skill developers can hear conversations (until the user stops Alexa).

A team from Security Research Labs (SRLabs) also found a new way to potentially phish users’ data [7]. They used a few quirks of the skill development process, two will be mentioned here. First, they took advantage of limitations on Alexa skill security reviews. During the skill certification process Amazon tests the security of skills to ensure they adhere to their policies, SRLabs discovered that this initial security review is the only test Amazon conducts - all subsequent alterations to skills go untested. They also managed to achieve a long-pause (verbally)



Figure 2.2: The *replacement character + dot + space*

⁵An Amazon Echo smart speaker with a physical screen, integrated Amazon Alexa

for their skill by making Alexa pronounce characters it was not designed to, specifically, they used Figure 2.2 (the *replacement character + dot + space*). To conclude, they successfully performed a skill squat attack by; creating a non-malicious skill to gain certification from Amazon, then later adding the malicious long-pause to avoid detection in the initial security test.

2.4 Related Work

This section includes an analysis of relevant work already published. It is primarily comprised of research papers relating to the security and privacy of voice assistants and other IoT devices.

Dr Such’s SPA security and privacy review paper [8] has contributed to various parts of this chapter and the project as a whole. Some of these areas include; third-party skill privacy, attacks based on architecture and smart speaker authentication.

Analysis, from researchers at the University of Campinas [9], outlines a major issue with online shopping through an Alexa integrated speaker. As many users leave the default wake word as ‘Alexa’, it is quite easy for one to accidentally either add items to their Amazon basket or to purchase them. Users are able to add a 4-digit PIN before checkout, but if they use up their maximum number of tries they can just simply restart the shopping skill to start again. Also, as the Alexa environment is primarily verbal communication, saying out-load any authentication details could jeopardise security if others (household members) are able to hear interactions with Alexa.

This paper [10] focuses on the authentication of users while communicating with smart speakers, and possible acoustic attacks. They deem the activation of such devices by a single wake word to be insufficient and weak, especially concerning when their tests reveal Alexa recognises and accepts verbal input via other machines (text-to-speech machines). This can introduce risks outside of the Echo environment as attackers may not need to target the Echo, but rather, another device nearby then issue verbal commands. Many third-party IoT device default names are not replaced by users, this could cause a security risk as an unauthorised agent may correctly assume the names of devices and issue commands like ‘Alexa, disable security camera one’ - similar risks to those associated with default passwords. The authors also propose certain remedies to the above problems, like adding real-time voice/speech authentication (biometrics) to ensure only authorised individuals can issue commands [11]. Motion-sensing could also be used to verify a human is in adequate proximity to the speaker in order to issue commands, although this may have other privacy implications.

Researchers from this paper [12, pg.5-7] realise that most traffic to and from the Echo is encrypted, but there are a few exceptions. Checking current network connectivity is unprotected, meaning network snoopers are able to detect smart speakers in a network. Firmware update pushes are also not secured, introducing the possibility of man-in-the-middle attacks. Dr Chung also outlines privacy concerns over encrypted traffic, again network-level attackers are able to analyse this traffic and discover certain patterns, like when the device is active and in use. Another paper [13, pg.4-5], from Princeton University, also evaluates the privacy implications of constant encrypted traffic from IoT devices.

Various papers [14] [15] [16] describe the privacy risks regarding misuse of third-party skills (skill squatting). In-communication skill switch is where a skill attempts to trick the user into thinking the skill session has ended and another started. For example, a skill may imply that it will close, then attempt to masquerade as an urgent firmware update trying to extract the user's Amazon password (via social engineering). Another threat includes faking termination, this is when a skill pretends it has been deactivated (long pause), but developers can still listen in. Deepak Kumar's small study found that over a third of speech errors (negative speech-to-text) were due to homophonous words. He notes, malicious skill developers could take advantage of this confusion to mask their skills. For example, a malicious skill 'Captial Won' could attempt to impersonate 'Capital One'.

2.5 Penetration Testing

Being my first encounter with security penetration testing, the first task was to familiarise myself with this field in an effective and timely manner.

Kali Linux⁶ was the chosen operating system for this project. Maintained by Offensive Security, it is a Debian-derived Linux distribution specifically designed for security penetration testing. It is a consistently high-ranking security testing OS, trusted by professionals around the world. Pre-installed with hundreds of testing tools, making it approachable for beginners; hence the chosen operating system.

A minor reintroduction to Linux Bash was needed before performing any tests on Kali Linux. Three YouTube videos were the main bulk of my Kali Linux introduction and penetration testing practise; Linux for Ethical Hackers (Kali Linux) by freeCodeCamp.org⁷, Network

⁶<https://www.kali.org/about-us/>

⁷<https://www.youtube.com/watch?v=lZAoFs75cs>

Penetration Testing for Beginners by freeCodeCamp.org⁸ and Learn Ethical Hacking With Kali Linux by edureka⁹. The Metasploit Unleashed course by Offensive Security¹⁰ was also used as supplementary material; used for sections relevant to this project, like ‘Information Gathering’ and ‘Vulnerability Scanning’.

⁸<https://www.youtube.com/watch?v=3Kq1MIfTWCE>

⁹<https://www.youtube.com/watch?v=0uvWRwLs5Zo>

¹⁰<https://www.offensive-security.com/metasploit-unleashed/>

Chapter 3

Design & Specification

Planning and following an organised structure is vital for this project. It allows for a coherent and systematic approach to security testing. The choices of two fundamental features will be described in this chapter; penetration testing methodology and risk assessment methodology.

3.1 Penetration Testing

There are many penetration testing methodologies currently in use, they include; Open Source Security Testing Methodology Manual (OSSTMM), CREST's Penetration Testing Execution Standard (PTES), OWASP's framework and many more. These methodologies are quite extensive, MITRE ATT&CK outlines over 260 techniques, due to the fact they were designed to be used in industry and across a wide array of environments. Soley using one of these as my method for my first penetration testing project would not only be unnecessary but also damaging. A large proportion of techniques outlined in the above methodologies are of limited use in this environment and fall out of scope for this project. Process manipulation attack (on databases), advanced password cracking and digital forensics tracing are a few examples of techniques not needed for this task. Instead, this project will follow a general five-step method inspired by Pentest People Ltd¹ and the Information System Security Assessment Framework (ISSAF)²;

1. Planning and initial snooping - This ensures testing will follow a systematic path and cover all aspects of the target. This stage also includes passive information gathering (independent research without target).

¹<https://www.pentestpeople.com/penetration-testing-methodology/>

²<https://www.futurelearn.com/courses/ethical-hacking-an-introduction/0/steps/71521>

2. Reconnaissance - Active reconnaissance and enumeration, involving scanning of the target.
3. Assessment/Testing - Penetration testing.
4. Documentation - This report will act as the documenting resource.
5. Reporting/Remediation - Reporting any weaknesses found to the manufacturer.

3.2 Risk Assessment

Many risk assessment structures/methodologies are available to follow with respect to security testing, for example; Microsoft STRIDE and OWASP Risk Rating Methodology (RRM)³. This project will primarily use the DREAD methodology [17], it comprises of 5 categories;

Damage - level of damage caused by attack.

Reproducibility - chance of performing the attack again (reproduce).

Exploitability - effort/resources required to launch attack.

Affected users - number of impacted stakeholders.

Discoverability - difficulty in finding the threat.

Each of the five categories will be associated with a number rating from 1 to 10. Throughout this report/project ‘Discoverability’ will be omitted, this is to reduce ambiguity as it does not play a significant role in the testing environment. DREAD is a useful risk assessment model as it not only simple to follow but also takes into account most of the key aspects of the risks of security vulnerabilities.

Overall Risk Severity				
Impact	HIGH	Medium	High	Critical
	MEDIUM	Low	Medium	High
	LOW	Note	Low	Medium
		LOW	MEDIUM	HIGH
Likelihood				

Figure 3.1: OWASP RRM

³<https://owasp.org/www-community/OWASPRiskRatingMethodology>

OWASP RRM's overall risk severity chart will also be used in some situations as it can be leveraged as a quick diagnostic tool. For example, it can be used to decide whether to continue pursuing a certain potential weakness or dedicating resources elsewhere.

Chapter 4

Penetration Testing

This chapter will outline and describe all security tests that were conducted during the course of this project. An evaluation of each test/attack will also be provided.

4.1 Tools and Programs

4.1.1 Wireshark and Environment

The chosen operating system (Kali Linux) comes pre-installed with hundreds of security tools/programs. However, for this project, only a small handful of these programs were used. This section will briefly describe the two main tools; Wireshark and Nmap.

Wireshark was the chosen network packet sniffer and analysing tool, version 3.2.1 was used for the majority of the project. Similar to Kali Linux, Wireshark is a highly popular tool used by pen testers around the world. It's intuitive graphical user interface and clearly defined features make it suitable for beginners like myself.

Before using it for testing, during the tool's research/practice period, a problem relating to the local network/environment was uncovered. As no additional hardware/resources were sourced for the project, the initial testing environment was planned to be the household's wireless (Wi-Fi) network. The internet router acted much like a network switch, this meant that a wirelessly connected device (e.g. laptop) was unable to analyse/sniff throughout the network. A switch, unlike a hub, is able to distinguish between different devices on the network. It does this by storing a CAM table which maps MAC addresses to different ports, with this it can efficiently

distribute data to the accurate owner. Since the data is being delivered straight to the device, and not to all on the network, it made packet sniffing harder. One workaround was to switch the Wi-Fi adapter and Wireshark to monitor mode, effectively turning the adapter into a local (hardware) sniffing tool. This is because now instead of only capturing packets on a local network, is capturing all packets in a given physical area (regardless of the network). By its very nature, this meant that was tremendous external noise in the Wireshark analysis. This was both due to neighbouring network's traffic also being captured and channel overlap. The local network and Wireshark were tuned to channel 6, as it was the least crowded channel. This, unfortunately, did not help with reducing the noise to an acceptable level. Filtering within Wireshark was also useless as packets were captured before being decrypted by the network's router, meaning MAC address filtering was infeasible.

The solution was to set-up another device as a Wi-Fi access point, Kali laptop (main testing laptop used for this project) was configured to act as the access point. Devices then had to connect through the laptop's signal to connect to the internet, thus Kali laptop being in a desirable position as a network sniffer/analyser.

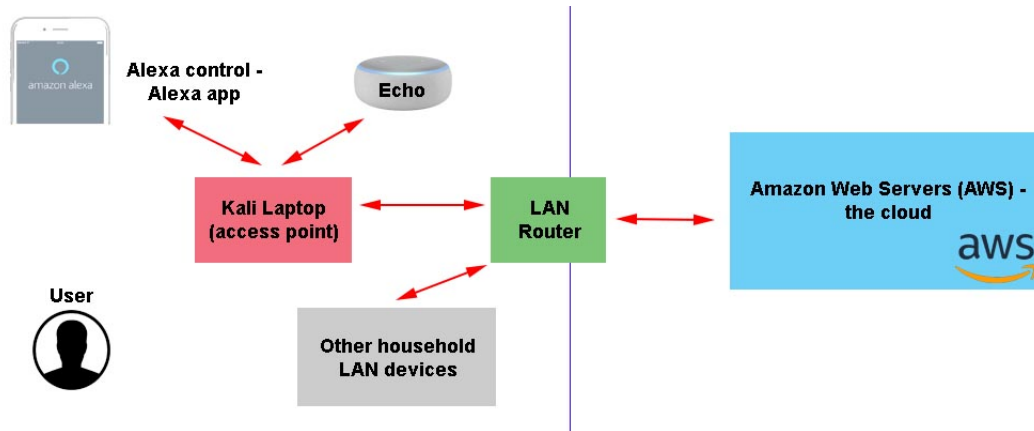


Figure 4.1: Home testing environment

4.1.2 Nmap and others

Nmap is a popular network scanning tool. It is primarily used to discover hosts and services on networks, it does this by sending packets out and analysing responses. For this project, Nmap was used as a port scanner. Speed -T4 was used for most applications as it was scanned deep enough relatively quickly.

Other tools like aprspoofer were also used, but their applications were more specific. Hence,

they will be mentioned during their respective test in this chapter.

4.2 Echo set-up

Amazon has made the Echo set-up process quite quick and hassle-free. One of the first sentences that Alexa says after powering on is to use the Amazon Alexa app to set-up the Echo. It is quite apparent that Amazon prefers user use the App and so they nudge people to do so. However, there are actually two methods to complete the set-up of an Echo device; a web browser (website) or smartphone app.

4.2.1 App

The smartphone app set-up process is the easiest. Plugin the Echo and launch the app, wait until the Echo is configured. Select the specific type of device to set-up.

The device will then activate a Bluetooth signal, through their phone the user must connect to the Echo's Bluetooth. Only one set-up process of a device can be open at any given time, hence external hijacking of the device is not possible. After this the user will need to connect the Echo Dot to the internet, they must select a Wi-Fi network. After some initialisation time, the device is connected to the internet and your app/account and is ready to use.

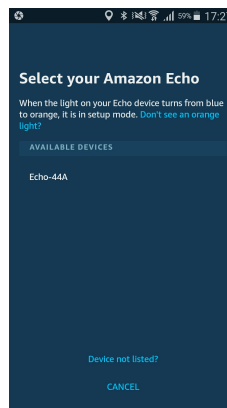
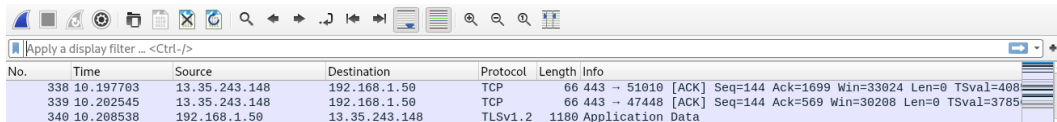


Figure 4.2: Amazon Echo selection via Bluetooth

Using Wireshark to monitor the session's traffic (Figure 4.3), it came to light that Amazon securely protects their traffic from the cloud to the Alexa app with TLSv1.2. Since I do not possess the private key I cannot decrypt this traffic. There were no gaps or potentially vulnerable points found during this set-up process.



No.	Time	Source	Destination	Protocol	Length	Info
338	10.197703	13.35.243.148	192.168.1.50	TCP	66	443 → 51010 [ACK] Seq=144 Ack=1699 Win=33024 Len=0 TSval=408
339	10.202545	13.35.243.148	192.168.1.50	TCP	66	443 → 47448 [ACK] Seq=144 Ack=569 Win=30298 Len=0 TSval=3785
340	10.208538	192.168.1.50	13.35.243.148	TLSv1.2	1180	Application Data

Figure 4.3: Alexa App traffic secured with TLSv1.2

4.2.2 Website

Amazon does not advertise, nor explicitly mention on their website, that users can register their device without a smartphone. Surfing old articles and archived Amazon support pages led to the discovery of alexa.amazon.com. This webpage is the online web browser (web app) equivalent to the Alexa app.

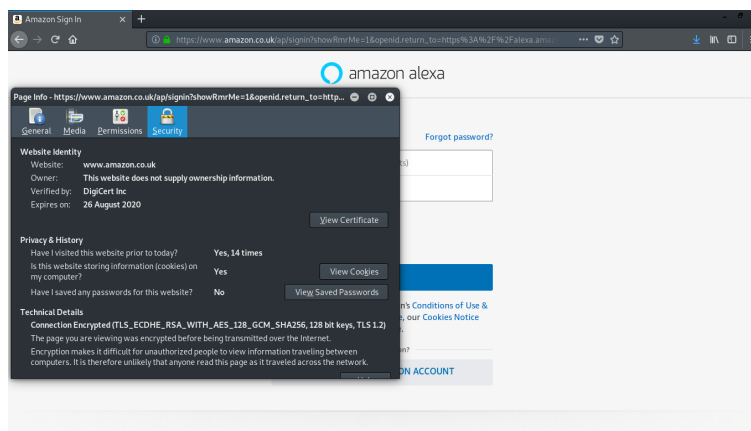


Figure 4.4: Log-in credentials secured via HTTPS

The initial alexa.amazon.com sign-in page is protected over HTTPS (Figure 4.4), this encrypts the credentials sent over the internet. Once logged in navigate to ‘settings’, and then to ‘Add new device’. This then provides the user with an options menu to select the specific Echo device they want to set up.

After this page is where the connection drops to a standard HTTP (Figure 4.5), meaning this session’s data is no longer secure as it isn’t being encrypted. This lack of protection also leaves customers vulnerable to Man-In-The-Middle (MITM) type of attacks. This is where a malicious agent who is between two parties, can snoop (see) or alter their communications. This can be accomplished by the attacker impersonating the server to the user, and impersonating the user to the server.

Unlike in the mobile app set-up, here the Echo device activates a wireless access point (not Bluetooth like with the app). This is a process adopted by many IoT devices (like Google Chromecast). The user must then join the new open and unprotected access point (Figure

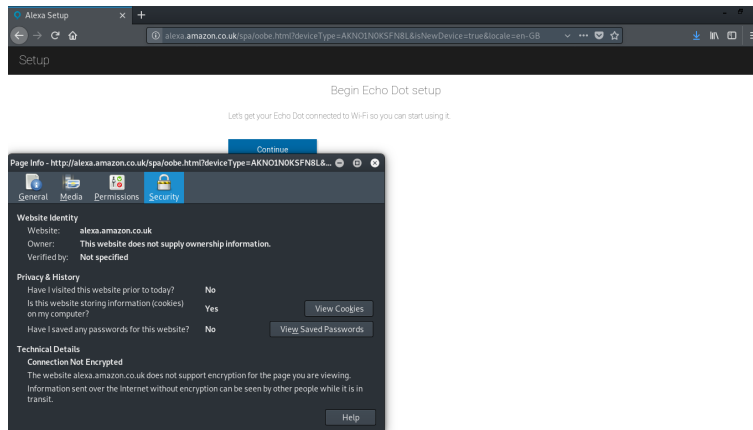


Figure 4.5: HTTP downgrade during set-up process

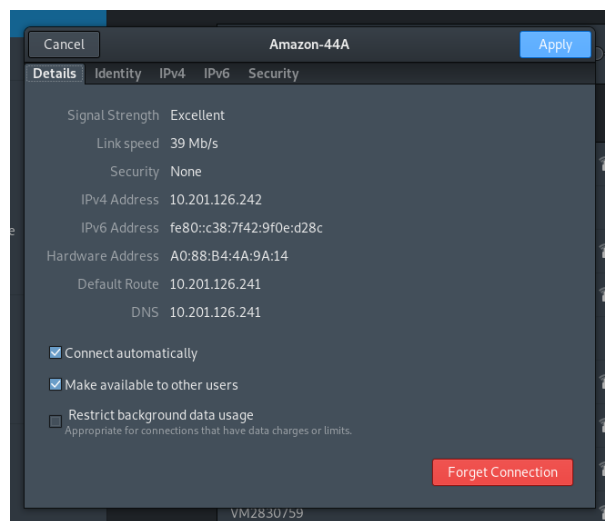


Figure 4.6: Amazon Echo Dot's access point

5.6). Once connected through the browser, you must select the Wi-Fi network the Echo device should connect to. Again, similar to the Bluetooth method, only one set-up session per device can be activated (at the same time) so hijacking the connection/device is not feasible. However, nearby malicious actors can sniff on the open network during the network connection phase. Once the network selection is completed, the user reconnects to the original network and waits for the device to initialise.

An additional aspect of the browser set-up process was the cookies Amazon had set. From the image we can see these last for a substantial amount of time (greater than 16 years), they are also not marked as http-only potential vulnerable to XSS cookie theft.

The following is a table that represents the DREAD risk assessment for the potential security risks of the three concerns found in the browser set-up; HTTP downgrade, Echo's open access point and cookie storage.

Name	Domain	Path	Expires on	Value	HttpOnly	sameSite
csrf	.amazon.co.uk	/	Mon, 22 Apr 2030 15:...	-316289120	false	Unset
DISPLAY_DE...	alexa.amazon...	/spa/	Fri, 31 Dec 9999 23:5...	false	false	Unset
sess-at-acbuk	.amazon.co.uk	/	Tue, 17 Apr 2040 15:3...	"Psif4HTi9R...	true	Unset
session-id	.amazon.co.uk	/	Tue, 01 Jan 2036 08:...	258-1595385...	false	Unset
session-id-ti...	.amazon.co.uk	/	Tue, 01 Jan 2036 08:...	2082787201l	false	Unset
session-token	.amazon.co.uk	/	Tue, 01 Jan 2036 08:...	"AdUFWqVB...	false	Unset
sst-acbuk	.amazon.co.uk	/	Tue, 17 Apr 2040 15:3...	Sst1 PQFeAA...	true	Unset

Figure 4.7: UPDATED SESSION cookies

Potential security risks			
-	HTTP downgrade for set-up	Echo using open a open access point	Session Cookies
Damage	A MITM attack would be dangerous and could cause damage. A passive/sniffing attack may not cause immediate risk, but could in the future or elsewhere relating to personal data misuse. A MITM attack that alters data to and from the device would be damaging. 8	Potential for sniffing is present. Misuse of personally identifiable data. No other concerns were noted. 2	Cross-site scripting (XSS) or other cookie hijacking. Damage moderate. 5
Reproducibility	A standard MITM attack is relatively simple to deploy, however, the main limiting factor is locality. 3	Simple to set up an appropriate environment. 3	Unable to decipher how useful unprotected cookies may be. 2
Exploitability	Standard, however having network access is the main drawback. 3	Sniffing attack requires very little expertise nor time. Local network is the main deterrent. 2	Minimal workload required to start attack. 5
Affected users	This type of attack occurs on an individual scale, hence less exploitable. 2	This type of attack occurs on an individual scale, hence less exploitable. 2	This type of attack occurs on an individual scale, hence less exploitable. 3

Nmap was used to perform a port scan during the set-up process, right after the Echo connected to the internet. The port scan found two open ports right after the Echo connected to the local network. These were 55442/tcp and 55443/tcp, [4] these attackers concluded that port 55442 is TCP HTTP server and port 55443 is TCP HTTPS server. The output (Listing 4.1) also seems to confirm that Linux/Android has a part in the device's OS (FireOS)

```
1  nmap -T4 -A -p- 192.168.1.55
2  Starting Nmap 7.80 ( https://nmap.org )
3  Stats: 0:01:20 elapsed; 0 hosts completed (1 up), 1 undergoing
   Service Scan
4  Service scan Timing: About 100.00% done; ETC: 18:22 (0:00:00
   remaining)
5  Stats: 0:01:21 elapsed; 0 hosts completed (1 up), 1 undergoing
   Script Scan
6  NSE Timing: About 0.00% done
7  Stats: 0:01:27 elapsed; 0 hosts completed (1 up), 1 undergoing
   Script Scan
8  NSE Timing: About 87.50% done; ETC: 18:22 (0:00:01 remaining)
9  Stats: 0:01:41 elapsed; 0 hosts completed (1 up), 1 undergoing
   Script Scan
10 NSE Timing: About 87.50% done; ETC: 18:23 (0:00:03 remaining)
11 Nmap scan report for 192.168.1.55
12 Host is up (0.0046s latency).
13 Not shown: 49145 filtered ports, 16388 closed ports
14 PORT      STATE SERVICE      VERSION
15 55442/tcp  open  nagios-nsc  Nagios NSCA
16 55443/tcp  open  ssl/unknown
17 MAC Address: 4C:17:44:71:54:14 (Amazon Technologies)
18 Device type: media device
19 Running: Google Android 5.X
20 OS CPE: cpe:/o:google:android:5.0
21 OS details: Sony Android TV (Android 5.0)
22 Network Distance: 1 hop
23
```

```

24  TRACEROUTE
25  HOP RTT      ADDRESS
26  1      4.58 ms 192.168.1.55
27
28  OS and Service detection performed. Please report any incorrect
    results at https://nmap.org/submit/ .
29  Nmap done: 1 IP address (1 host up) scanned in 141.68 seconds

```

Listing 4.1: Port scan during set-up

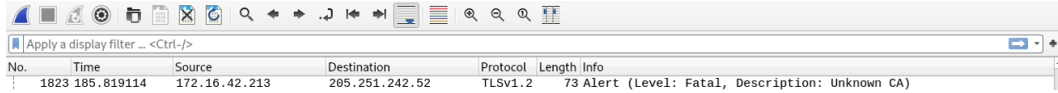
4.3 Echo to the cloud

This section of penetration testing primarily comprises of potential attacks on the device while communicating with Amazon’s backend server.

Before starting any digital tests, an experiment involving a low-grade laser was set-up. Mentioned in multiple articles [6] [18] and in the ‘Previous vulnerabilities’ section of this report, outlines an attack-type which uses lasers to activate Alexa without saying its wake word. No vulnerabilities were found during this experiment, Alexa did not activate while pointing a laser at its microphones. This was expected as in the above study, the researchers used industrial-strength lasers capable of altering intensity; the laser used of this project did not have these capabilities. Nonetheless, the damage from this type of attack could be considerable depending on the security/privacy setting of the victim/s Echo device. An attacker could activate the device, however, if the user has enabled secure settings like the 4-digit PIN before purchases, then the attack’s yield would decrease dramatically (damage = 5). This attack can be hard to reproduce as the attacker would both need to be relatively close to the device and have access to an industrial laser (reproducibility = 3). Apart from the laser and proximity requirements, a potential attacker would also have to have access to the Echo’s data stream, for example through packet sniffing, this traffic is protected with TLSv1.2 making the attempt infeasible (exploitability = 1). Affected users would be equal to 1 as this attack can only be completed on an individual and local basis. Overall, its DREAD score is 10 making it an inefficient attack.

As mentioned above Amazon protects the traffic between the Echo and its backend servers with TLSv1.2 making it difficult to exploit due to its encryption. Although a simple MITM

attack related to redirecting and altering packets would be infeasible due to TLSv1.2, it was tested. Following this procedure¹ and using the tool sslsplit, an attempt was made to insert myself between the server and Echo. The sslsplit tool helps as it automates the process of redirecting traffic to and from the endpoints. As suspected the Echo Dot successfully defended itself against the simple attack due to its use of TLSv1.2 protocol.



No.	Time	Source	Destination	Protocol	Length	Info
1823	185.819114	172.16.42.213	205.251.242.52	TLSv1.2	73	Alert (Level: Fatal, Description: Unknown CA)

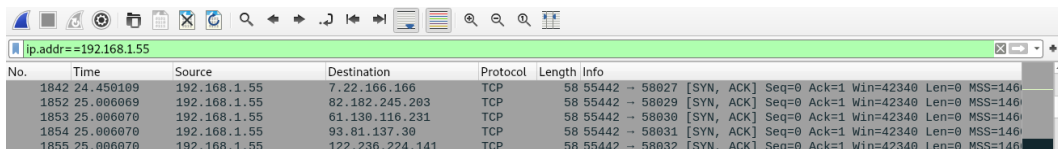
Figure 4.8: SSLsplit attack captured with Wireshark, certificate

Analysing the packet stream showed a ‘fatal certificate unknown’ error (Figure 4.8). Interesting the Alexa skill functioned as intended during this process, however, was slightly slower at responding and deactivating a conversion session. Also for approximately 2 seconds during the attack, the Echo’s ring light turned red, an indication of network problems. Also during the red ring light, Alexa could not be activated. Hence this test did show limited results as a Denial of Service (DoS) attack.

Another DoS attack was tested, this MITM attack was an SYN flood attack. This simple and well-documented attack involves an attacker, manipulating the 3-way handshake, by sending a large volume of SYN request packets to a target. The sheer volume overwhelms the memory capabilities of the device, resulting in the denial of any new (legitimate) requests.

```
1 hping3 -c 15000 -d 120 -S -w 64 -p 55442 --flood --rand-source
   192.168.1.55
2  HPING 192.168.1.55 (wlan0 192.168.1.55): S set, 40 headers + 120
   data bytes
3  hping in flood mode, no replies will be shown
```

Listing 4.2: hping3 bash output



No.	Time	Source	Destination	Protocol	Length	Info
1842	24.450109	192.168.1.55	7.22.166.166	TCP	58	55442 → 58027 [SYN, ACK] Seq=0 Ack=1 Win=42340 Len=0 MSS=1460
1852	25.006069	192.168.1.55	82.182.245.203	TCP	58	55442 → 58029 [SYN, ACK] Seq=0 Ack=1 Win=42340 Len=0 MSS=1460
1853	25.006070	192.168.1.55	61.130.116.231	TCP	58	55442 → 58030 [SYN, ACK] Seq=0 Ack=1 Win=42340 Len=0 MSS=1460
1854	25.006070	192.168.1.55	93.81.137.30	TCP	58	55442 → 58031 [SYN, ACK] Seq=0 Ack=1 Win=42340 Len=0 MSS=1460
1855	25.006070	192.168.1.55	122.236.224.141	TCP	58	55442 → 58032 [SYN, ACK] Seq=0 Ack=1 Win=42340 Len=0 MSS=1460

Figure 4.9: SYN flood attack, Wireshark

hping3 was the tool used to deploy this SYN flood attack. The command (Listing 4.2) specifies

¹<https://blog.heckel.io/2013/08/04/use-sslsplit-to-transparently-sniff-tls-ssl-connections/>

that 15000 packets (-c) each of size 120 bytes (-d), SYN (-S) will be enabled with TCP window 64 (-w). The target/Echo's HTTP port 55442 (-p) will be flooded (-flood), finally -rand-source lets hping3 generate a random IP address to spoof outgoing SYN requests and incoming SYN-ACK replies.

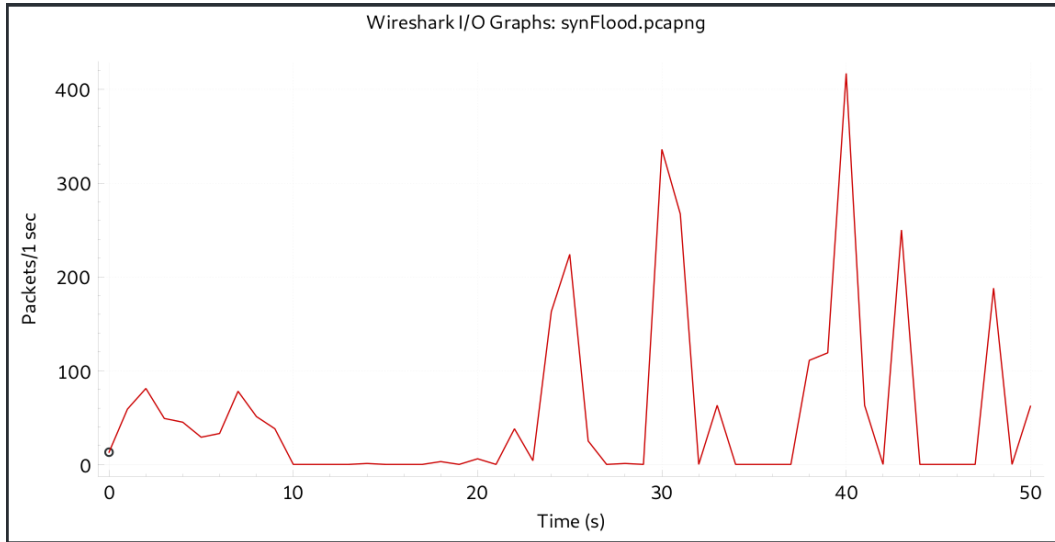


Figure 4.10: SYN flood attack graph, initial spike (before 10s) was a Alexa weather command

This attack was successful in freezing the Dot if Alexa is activated the query is frozen and Alexa does not respond. Trying to activate Alexa during or after the attack will prompt the response ‘Sorry I’m having trouble understanding right now, please try again later.’ this accompanied by the red ring light. The only way to remedy this is to restart the Echo’s power source and let it reconnect, this takes on average 5 minutes.

```
1 nmap -T4 -A -p- 192.168.1.55
2   Starting Nmap 7.80 ( https://nmap.org )
3   Note: Host seems down. If it is really up, but blocking our ping
   probes , try -Pn
4   Nmap done: 1 IP address (0 hosts up) scanned in 1.69 seconds
```

Listing 4.3: Port scan after SYN attack

Running a port scan during the attack, yielded in Nmap concluding the Echo was turned off.

The final attack tested was ARP spoofing. This attack involves sending fake ARP messages on a local network, to impersonate an IP address on the network (with attacker’s MAC address), which causes traffic meant for that IP to be diverted to the attacker. This specific test caused

another denial of service outcome.

[illegible]

Figure 4.11: arpspoof attack in process

arp spoof was that tool chosen to carry out the test. It requires a network interface (-i wlan0) and a target (-t) IP and host (-r) IP to initiate the attack. In this case, the network router will act as the target and the Echo as the host, the second command should swap the target and host IPs. The two commands were run in parallel to ensure a MITM position between the two points.

No.	Time	Source	Destination	Protocol	Length	Info
158	27.448396	192.168.1.51	62.252.60.161	TCP	66	[TCP Dup ACK 49#2] 39944 → 88 [ACK] Seq=1 Ack=1 Win=501 Len=0
159	28.085500	62.252.60.161	192.168.1.51	TCP	66	[TCP Dup ACK 50#2] [TCP ACKed unseen segment] 89 → 39944 [ACK] Seq=1 Ack=1 Win=501 Len=0
152	28.472336	192.168.1.51	72.21.194.180	TCP	54	[TCP Dup ACK 53#2] 34744 → 443 [ACK] Seq=1 Ack=1 Win=501 Len=0
153	28.566054	72.21.194.180	192.168.1.51	TCP	54	[TCP Dup ACK 54#2] [TCP ACKed unseen segment] 443 → 34744 [ACK] Seq=1 Ack=1 Win=501 Len=0

Figure 4.12: ARP spoof attack, Wireshark

This attack’s result was similar to the SYN flood attack. As it froze Alexa if she was activated during the attack, it also prompts the error message and red ring light. However, recovery after this attack much faster as arpspoof recovers and re-applies all ARPs after ending the attack.

All of these attacks are similar in outcome as they all result in denial of service. The damage caused by these attacks are moderate in nature, as DoS only prevent using the target, no data was altered nor acquired (damage = 5). These attacks rely on the attacker having some prior knowledge and already being connected to the local network, hence limiting its reproducibility to 4. Other than being on the existing network, these attacks are relatively simple to perform (exploitability = 6). Finally, the number of affected users could be exacerbated by variation in attack style, 4.

Running a port scan during normal operation yields the same open ports as during set-up,

55442/tcp and 55443/tcp.

Chapter 5

Skill and Privacy

5.1 Third-party Skill

During the beginning of this project, a small-scale third-party skill was developed. It was developed through Amazon’s online development environment using JavaScript (JSON) and AWS Lamda. The skill was called ‘Positive day’ and would read out randomise inspirational quotes daily. The aim was to embed malicious code/phrase into the skill that allowed the developer to listen to the user for longer than intended. Following [7], over 110 variations of special and unknown characters were attempted to uncover a bug with the skill framework, but yielded no results. The skill was based on the Alexa facts sample skill¹ with minor additions.

5.2 Privacy

Amazon recently filed for a new patent that would allow them to listen to users device just before activation [19]. It would allow for additional storage of verbal commands so Alexa can make better decisions. Some Echo Dot devices can have up to 8GB of flash memory. This is quite unusual for a device that otherwise relies on the internet for its computation.

The lack of voice recognition for Alexa commands, leaves a large gap in security in terms of access. Amazon may be hesitant to push any additional security feature/recommendation in fears that it could jeopardise the ‘care-free’ experience advertised to consumers.

Four types of tests were conducted to analyse traffic between the Echo (each ran for 30 mins), these are the average results. The first one was set up as a silent test (), minimal external (sound) noise and not Alexa activation. This test recorded that every 5 minutes for

¹<https://github.com/alexa/skill-sample-nodejs-fact>

so, the Echo would send (at least) 200bytes of data to Amazon. This is a substantial amount of data, for a period of time where Alexa is not in use. The second test () has the microphones muted, using the physical button on the Echo. The third test had the settings to send Amazon diagnostics data turned off, and the final test has muted mics and diagnostics off (). This data shows that despite the different environments the echo still sends data frequently to one another.

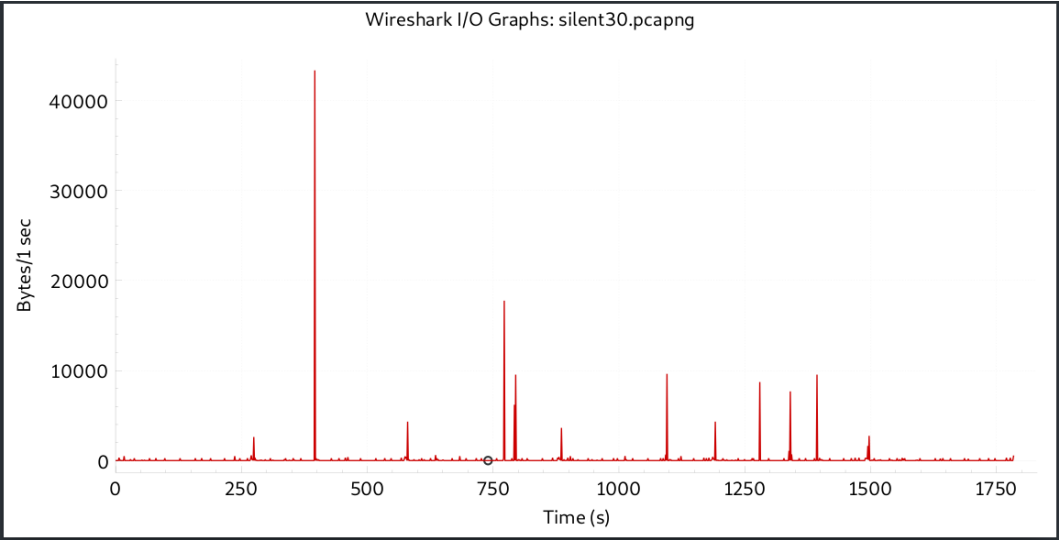


Figure 5.1: First slilent test

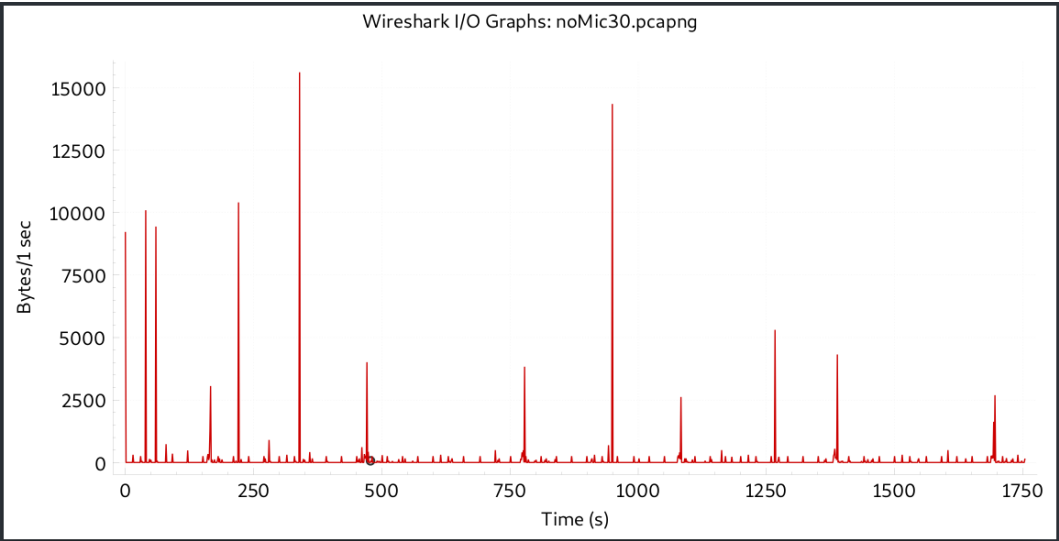


Figure 5.2: Second mics muted test

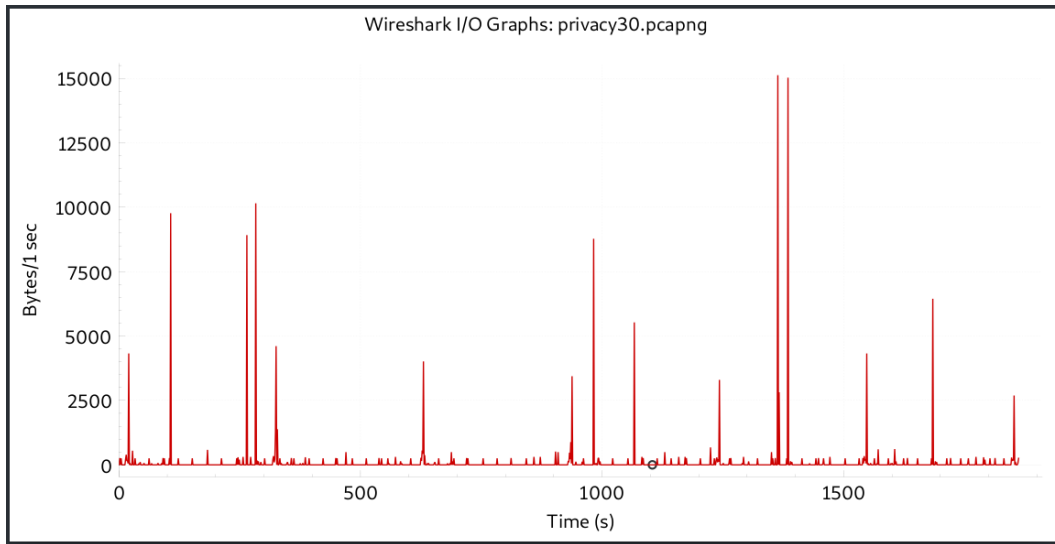


Figure 5.3: Third privacy settings changed test

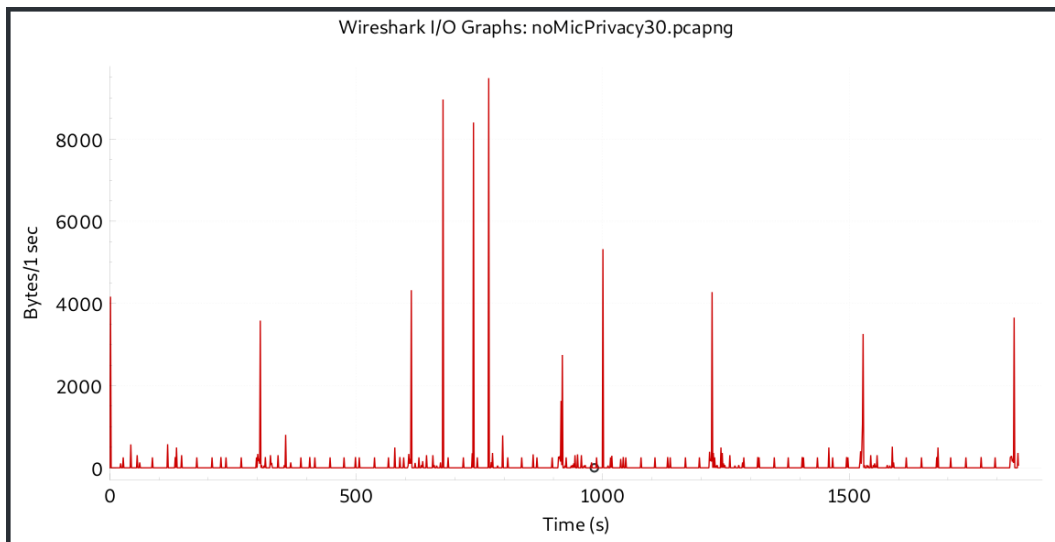


Figure 5.4: Fourth no mics and privacy settings changed test

Chapter 6

Legal, Social, Ethical and Professional Issues

Security and privacy being the two main parts of this project, it was highly important to plan for and mitigate any professional and ethical concerns that could have arisen. Because of this, the Code of Conduct and Code of Good Practice from the British Computer Society (BCS) was always followed can kept in mind.

“have due regard for public health, privacy, security and wellbeing of others and the environment”

“conduct your professional activities without discrimination on the grounds of sex, sexual orientation, marital status, nationality, colour, race, ethnic origin, religion, age or disability, or of any other condition or requirement”

To mitigate any form of discrimination or privacy issues (e.g. misuse of personal data) no external persons data was utilised during the course of this project. Much primary data (e.g. questionnaire for user story/case), for research, was not needed for this task; hence omitting personal data collection/misuse.

“carry out your professional responsibilities with due care and diligence in accordance with the relevant authority’s requirements while exercising your professional judgement at all times”

“have due regard for the legitimate rights of third parties”

A large area where privacy concerned could have arisen was during the creation and testing of the third-party skill. As mentioned several times prior, this skill was not deployed or published in a public setting - all activities were completed in local development mode. This meant no one was able to use the malicious skill, and no unethical data was collected. There were two reasons for this, first, to abide by the BCS's code to have due regard for the public's privacy. Secondly, ethical clearance was not pursued for this project. Hence the collection of, possibly, identifiable data would go against the collage's rules resulting in breaking the 'Duty to relevant authority' clause in the code. A local testing environment was used to also avoid any negative spillover effects on the public.

"Not claim any level of competence that you do not possess"

This project was my first penetration testing encounter, hence quite overwhelming but at the same time exciting. All tasks were completed based on my ability to complete them, those that were beyond my ability or incomplete have been omitted of this report. All credit/acknowledgement of ideas expressed that were not my own has been given (references).

Chapter 7

Conclusion and Future Work

7.0.1 Limitations

Many limitations arose during the course of this project. One is the testing environment, planning and setting up the alternative testing environment set the penetration testing start date a few days behind.

The exploits demonstrated in this paper are only feasible when the attacker is already within the local network of a target, e.g. MITM attacks and sniffing tools.

The testing of a third-party app consumed a considerable amount of time during the start of the project, despite very little results from it. It is important to note that all testing was completed in a local development setting, if the were to go live to the public with malicious code within there is a chance that it would have been noticed by Amazon's security tests. And this would have assumed customers would have actually used the skill.

7.0.2 Conclusion

This project was a fantastic learning opportunity, it is only strengthened my interest in cybersecurity.

From this project and my beginner security and penetration testing knowledge, the conclusion has been reached that the Amazon Echo Dot (v3) remains relatively a secure IoT smart speaker. Mainly due to Amazon protecting all traffic and keeping an up-to-date and secure skills framework[? , 3]

7.0.3 Future work

During this project, the third-party skill yielded no results and used up a large amount of limited time. Therefore, I would like to continue working on a third-part skill as I am no longer bound by the time-constraint of this project. Exploring all possible avenue of exploitability within the skill framework.

Additional testing of privacy settings and concerns will also be useful in the future.

The original ‘no additional hardware’ limitation, meant no other products could be tested. Testing the security of these devices could be interesting as many do not follow the strict guidelines and ethic that Amazon does. The Zigbee protocol and its potential exploits [20]*, is worth researching and testing.

Expanding my current penetration testing knowledge, I would like to revisit the MITM attacks and attempt to make them easier to conduct or more damaging, also attempt to find more.

References

- [1] B. Kinsella, “U.k. smart speaker usage doubles in 2018, growth expected to slow in 2019, amazon commands 68% market share,” 2019.
- [2] micaksica, “Exploring the amazon echo dot, part 1: Intercepting firmware updates,” 2017.
- [3] A. Montag, “Former nsa privacy expert: Here’s how likely it is that your amazon echo will be hacked,” 2018.
- [4] L. Yuxiang, Q. Wenxiang, and W. Huiyu, “Breaking smart speaker: We are listening to you,” tech. rep., Tencent Security Platform, 2018.
- [5] Z. Whittaker, “Two security researchers earned \$60,000 for hacking an amazon echo,” 2019.
- [6] A. Greenberg, “Hackers can use lasers to ‘speak’ to your amazon echo or google home,” 2019.
- [7] “Smart spies: Alexa and google home expose users to vishing and eavesdropping,” tech. rep., Security Research Labs, 2019.
- [8] J. S. Edu, J. M. Such, and G. Suarez-Tangil, “Smart home personal assistants: a security and privacy review,” tech. rep., King’s College London, 2019.
- [9] L. A. de Sousa Ferreira, L. Y. Schwarzstein, Y. Iano, and C. S. Domingues, “Alexa’s case: vulnerability issues in iot devices in residential automation,” tech. rep., University of Campinas, 2017.
- [10] X. Lei, G.-H. Tu, A. X. Liu, C.-Y. Li, and T. Xie, “The insecurity of home digital voice assistants – vulnerabilities, attacks and countermeasures,” tech. rep., 2018.
- [11] M. Kunz, H. R. Klaus Kasper, M. Möbius, and J. Ohms, “Continuous speaker verification in realtime,” tech. rep., 2012.

- [12] H. Chung, M. Iorga, J. Voas, and S. Lee, “Alexa, can i trust you?,” tech. rep., National Institute of Standards and Technology, 2018.
- [13] N. Aphorpe, D. Reisman, and N. Feamster, “A smart home is no castle: Privacy vulnerabilities of encrypted iot traffic,” tech. rep., Princeton University, 2017.
- [14] N. Zhang, X. Mi, X. Feng, X. Wang, Y. Tian, and F. Qian, “Understanding and mitigating the security risks of voice-controlled third-party skills on amazon alexa and google home,” tech. rep., 2017.
- [15] D. J. Major, D. Y. Huang, M. Chetty, and N. Feamster, “Alexa, who am i speaking to?,” tech. rep., 2019.
- [16] D. Kumar, R. Paccagnella, P. Murley, E. Hennenfent, J. Mason, A. Bates, and M. Bailey, “Skill squatting attacks on amazon alexa,” tech. rep., University of Illinois, 2018.
- [17]
- [18] J. Peters, “Your amazon echo or google home could be fooled by a laser ‘speaking’ words,” 2019.
- [19] T. Anderson, “Amazon alexa: ‘pre-wakeword’ patent application suggests plans to process more of your speech,” 2019.
- [20] M. W. Denko, “A privacy vulnerability insmart home iot devices,” tech. rep., University of Michigan-Dearborn, 2017.