Issues in Information Systems
Volume 22, Issue 4, pp. 208-220, 2021

DOI: https://doi.org/10.48009/4_iis_2021_208-220

Security Analysis of the Amazon Echo Dot

Robert Brinson, *Georgia Southern University, rb07498@georgiasouthern.edu*
Hayden Wimmer, *Georgia Southern University, hwimmer@georgiasouthern.edu*
Queen Booker, *Metropolitan State University, queen.booker@metrostate.edu*

Abstract

Internet of Things (IoT) devices have the capacity to send data over Internet connections with small amounts of processing memory, limiting their ability to offer state-of-the art security functions. Because IoT devices can connect to so many different devices, the limited security makes them vulnerable to several types of attacks that could lead to access to sensitive personal data. Amazon introduced the Echo Dot in 2016 as a lower cost alternative to the popular Echo, making the Echo Dot available to many more households concomitantly increase the number of vulnerable households. Because of its potential reach, it is important to know if the Echo Dot is a secure device. In this study, we examine the vulnerability of the Echo Dot and compare our results to prior research on the Echo.

Keywords: IoT, Security, Design, Experimentation

Introduction

As Internet of Things (IoT) becomes more integrated into people's day to day lives, more personal data will be sent using the IoT technologies. This increase of sending personal data to and across technologies connected through the Internet provides more opportunity for attackers to gain sensitive data. These connected devices being brought into homes need to be tested to understand their true level of security. If these devices are easily compromised, then attackers may have access to new types of data they have not previously had access to.

With more than 200 million Amazon Echo Dots (Echo Dot) sold as of February 2020, the Echo Dot has become a popular IoT device used in households. (Sterling, 2020) The Echo Dot is a "voice-controlled smart speaker that uses an artificially intelligent (AI) personal assistant named Alexa to perform a broad range of activities. Alexa can play music, set alarms, control other "smart" devices in the home, play games, and answer questions by searching for information on the internet, just to name a few of its many skills." (Johnson, 2019) In the process of using these skills, the Alexa gathers or can be uploaded with a wide-ranging amount of personal data such as credit card information and daily schedules, which in the wrong hands, can be problematic for the owner.

With the newness of IoT devices, existing types of attacks are changing to target these devices and new types of attacks are being created to exploit the vulnerabilities inherent in devices with limited memory but sharing lots of information. The objective of this study is to present a methodology for testing the security of IoT devices and demonstrate the process with the Echo Dot. The rest of the paper is structured as follows. The next section will discuss research conducted in IoT security and specifically on the Echo. The third section discusses the methodology used to test personal IoT devices which is followed by the implementation using the Echo Dot. The fifth section will exhibit the results of the tests ran. The sixth

section will review some solutions to make IoT devices less vulnerable to certain attacks. The paper will close with a summary conclusion and next steps for this research.

Background and literature review

Internet of Things (IoT)

The internet of things (IoT) is a term used to describe the non-traditional Internet connected devices. These devices can send data, receive instructions or both. A wide range of electronics fit into the category of IoT devices such as wireless watches, wearable health devices, smart appliances such as TVs, refrigerators and lighting fixtures as well as specific devices such as Internet-enabled technologies like Alexa-style digital assistants; internet-enabled sensors that are transforming factories, healthcare, transportation, distribution centers and farms (Fruhlinger, 2020). We summarize some of the most recent articles on finding regarding IoT device vulnerabilities, noting particular vulnerabilities as related to the Echo.

IoT Security Attacks

Prior research has shown that IoT devices are particularly vulnerable due to their limited memory and their constant communication via the Internet to cloud servers. Deogirikar and Vidhate (2017) and Nawir, Amir, Yaakob, and Lynn (2016) present different attacks that IoT devices are susceptible to. They give details about several different attacks and rank them based on how dangerous they can be. They classified the various attacks based on their method for targeting networks. Once they were classified, they ranked them based on how damaging they can be to a network. Three of the attacks are discussed below: Vocal attacks, Distributed Denial of Service attacks and Domain Name System attacks. In addition, an examination of intelligence virtual assistants security and digital forensics are examined.

Vocal Attacks (VA)

Voice assistants is a key component for many IoT devices used in residential settings which leaves such devices open to vocal attacks. In their research, Lei, Tu, Liu, Li, and Xie (2018) tested if Home Digital Voice Assistants have proper security to protect against vocal attacks. They concluded such devices do not offer sufficient protection from voice commands given by someone other than the owner, and this can be exploited by attackers. They demonstrated adding a sensor to the Echo could prevent attackers from using acoustic attacks to target the Echo. They proposed creating a virtual security button in order to detect whether someone was present in the room with the Echo giving the command or if it was just a voice. The button works by detecting human motion based on Wi-Fi signals. This allows for minimal upgrading for Amazon since most houses are outfitted with Wi-Fi and the Echo uses Wi-Fi already. Their experiment with testing their virtual security button did show in lab and limited home environments that it is effective in properly activating the Echo with human motion. Their research was limited to smaller settings, and need to be tested on a larger scale, but it did respond to hand motions to alert the Echo (Lei et al., 2018). Mitev, Miettinen, and Sadeghi (2019) continued this research by testing if skills created with malicious intents could be downloaded accidentally due to Alexa misinterpreting the command. In their research,
they set two categories of skills: “benign skills” and “malicious skills.” Benign skills are normal skills found in the Amazon store. Malicious skills are skills designed to exploit the user or steal their data. Given the amount of effort Amazon goes to ensure security on the Alexa and the fact that the skills are produced by third-party companies, the question arises if attackers can compromise an Alexa device indirectly through these skills which may not have the same level of security as the Alexa itself. Their research also tests whether it is possible to hijack communications between the device and the user through their “malicious skills.” The authors of this experiment created a malicious skill and uploaded it on an Echo through the Amazon Apps store. This malicious skill operates by jamming the commands to the Alexa. In doing so, the user unknowingly is communicating with the skill and not the Alexa. This provides the attacker the ability to return fraudulent responses to the user. This is done by not directly targeting the Alexa

itself, but rather taking advantage of the third-party skills it supports. Their results showed that in order for this type of attack to work, it has to begin when the device is started up. Alexa devices operate on "wake words" which prompt them to begin actively listening for commands. If the attack does not begin jamming immediately after Alexa is woken up, Alexa will bypass the malicious skill and go to the benign skill it would normally run. Their research also found background noises can interfere with the jamming part of the attack. It is possible to jam the benign skills and return fraudulent responses to the user. This study also found it is possible to jam smart security systems through this attack as well. The user believes their security system is activated by the device when it is not (Mitev et al., 2019).

Distributed Denial of Service (DDoS) Attacks

A distributed denial of service (DDoS) attack is malicious attempt to disrupt the normal traffic of a targeted server, service or network by overwhelming the target or its surrounding infrastructure with a flood of Internet traffic. DDoS attacks achieve effectiveness by utilizing multiple compromised computer systems as sources of attack traffic. Exploited machines can include computers and other networked resources such as IoT devices. From a high level, a DDoS attack is like an unexpected traffic jam clogging up the highway, preventing regular traffic from arriving at its destination. ("What is a DDoS?" 2021into a reference).

Jones, Wimmer, and Haddad (2019) sought to show if it is possible to attack a someone using an IoT device on a virtual private network (VPN) and disable them using a DDoS attack. They performed their research on a PPTP VPN set on a Windows Sever 16. They used a Kali Linux VM to perform their attack. They first ran scans using nmap to find any vulnerabilities. They then used HPing3 to launch a DDoS attack against the device hooked into the VPN. They were able to show their attack was able to result in at least 80% packet loss over the network. They were able to show the ability to disconnect the user from the VPN using the DDoS over HPing3 (Jones et al., 2019). Overstreet, Wimmer, and Haddad (2019) sought to test whether the Echo is vulnerable to DDoS attacks. The authors used software through a Kali Linux VM for their experiment. In a local lab, they used nmap to scan and gather information about the device. They ran Wireshark to collect and analyze traffic on the network. They used Metasploit to launch a syn-flood DDoS attack against the Echo. Their scan using nmap allowed them to gain the MAC address and proper network information. They were able to take the Echo offline with the syn-flood attack (Overstreet et al., 2019).

Man in the Middle Attacks

As shown in Figure 1, a Man-In-The-Middle (MITM) is an attack where an attacker intercepts or peaks in on data transferring between two parties. When the victim's device sends the request to the destination, the request also goes to the attacker as well. The attacker can sniff packets of data being sent between the victim and the destination address, allowing them to get login credentials, as well as other sensitive information. Attackers can do this through a variety of methods, such as different software or ambiguous Wi-Fi hotspots. MITMs are used to gain information about a victim to access important data centers. Once attackers have necessary credentials or information, they can steal information or even alter and destroy valuable information.

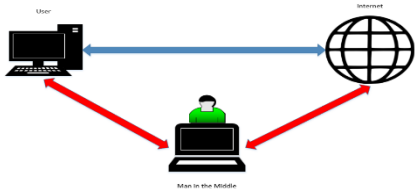

Figure 1. Man in the Middle Architecture

Launching a MITM requires the attacker to convince the two parties it is the proper destination. The attacker does this by spoofing IP addresses, causing the two parties to send their traffic to the attacker sitting between them. The attacker has to be careful pulling the traffic to itself. If the two parties realize their connection is interrupted, they will shut things down. In order to keep the traffic flowing, the attacker uses forms of packet forwarding to keep the two parties communicating. This allows for the traffic to flow to the attacker but continue moving to the final destination without alerting either party.

With the coming of HTTPS protocol, MITM attacks have become less effective. New security measures and encryption methods have made data harder to gain for attackers. However, MITM are still effective forms of attack. Areas, such as coffee shops that offer open WI-FI networks, give attackers the ability to sniff packets over an unsecure network or set up their own Wi-Fi hotspot for other patrons to connect to. MITMs are also effective to grab information sent over HTTP or other unsecure protocols.

C. Li, Qin, Novak, and Li (2017) demonstrated that MITM attacks remain an issue due to the use of software defined networks and their use in managing network flows. They assert that MITM attacks remain high as the use of SDNs "is a disaster for both the network service providers and their customers" because of the assumption that the SDN security either protect controllers themselves or make a strong assumption that the OpenFlow channel is already secured, making devices using them vulnerable to MITM attacks.

Dynamic Host Configuration Protocol (DHCP) Starvation

The Dynamic Host Configuration Protocol server is one of the most important features of a network. It is tasked with automating the process of issuing IP addresses, the local gateway information, and access to the local DNS address to any device seeking to connect to the network. Without this server, the only way a device could connect to the network is if this information is manually entered in. If the DHCP server goes down, it will make it difficult or impossible for devices to join a network. An attack to take a DHCP server offline is a DHCP starvation attack. This attack floods the server with requests to join by phony MAC addresses. The goal of this attack is to take all the available IP addresses through the volume of requests, crowding out any legitimate requests by devices to join the network. Once the available IP addresses are issued, no other device can connect to the network.

Intelligent Virtual Assistant (IVA) Security

Intelligent virtual assistants (IVAs), also known as virtual agents, allow users to engage with technology in a conversational manner. Conversations conducted with these technologies are human like. IVAs can understand answers for which they have not been explicitly programmed using Natural Language Processing (NLP). Through the use of machine learning and deep learning, they also build up a larger vocabulary, understand colloquial formulations, and give precise and correct answers to inquiries. (Nord, 2021) Because IVAs collect and send information to and through IoT devices, the use of IVAs represent the potential for exploitation.

Chung, Iorga, Voas, and Lee (2017) lays out instances where Intelligent Virtual Assistants have been exploited. They did not test anything directly in this paper but showed examples of where IVAs can be taken advantage of. They used an example of television sounds, such as adds to interact with the Amazon Alexa or the Google Home. They lay out how attackers can exploit these devices with Man-in-the-Middle and DDoS attacks. They also show how malicious and unintentional commands can cause harm to the owner of the device. This paper shows the need to improve the security of IVAs and their vulnerabilities. They used different cases to build their charge that these areas need to be further researched and improved (Chung, Iorga, et al., 2017).

Haack, Severance, Wallace, and Wohlwend (2017) ran tests against the security of the Echo. With Amazon outsourcing the development of skills to different companies, they highlight how little is stored on the Echo and display how the Echo performs commands given to it by the user. The different attacks yielded several

different results. The sound attacks showed Alexa can receive sounds from a wide variety of sources. It is difficult to create a distorted version of the wake word "Alexa" that is not noticeable to a normal person. It also found there are a limited number of wake words for Alexa, so it is not possible to attack using a made-up word. These attacks also showed how limited the Alexa device is in its capabilities, and how much it relies on third-party applications to perform tasks. Alexa can pick up on indirect commands and commands issued outside of its immediate vicinity. It will not give out personal information, but it can allow for tasks to be completed using personal information. An example would be someone yelling from outside of a window for Alexa to order something from Amazon. While the person cannot change the personal information to where the order will be sent to, they can still have the ability to order items without being in direct contact with the device. While Alexa's can be given pins to protect from ordering, given the weakness of these pins by the user and the seemingly unlimited attempts to guess the pins, attackers can still manipulate the Alexa through voice commands. Man-in-the-middle attacks are limited in what they can get from the Alexa, but the traffic can still be picked up by software such as Wireshark. Replay attacks against the Alexa were also unsuccessful (Haack et al., 2017).

Digital Forensics

Digital Forensics (DF) is the investigation of digital devices in order to collect, analyze, and preserve data from cybercrimes. DF is employed to counter data breaches, hacks, or other attacks on individuals or organizations. DF works to recover stolen information, as well as preserve evidence in such a way that is admissible in court proceedings. With cybercrime on the rise, DF is becoming used more, and is becoming relied on more by law enforcement.

S. Li, Choo, Sun, Buchanan, and Cao (2019) With the increased circulation of IoT devices, there has also been an increased focus on performing forensics. Since each device is different, finding a comprehensive model of forensics is difficult. In their research, the authors seek to create a comprehensive model to perform forensics on different IoT devices, and to demonstrate it using the Amazon Echo. They sought to create a forensics software that identifies the IoT device, acquires data from the device, analyze the data, and used to present the data. Their software connects to the Echo's SQLite files. Their software locates the data being stored in the companion app and allows for extraction. Their software was able to pull in the records from the Echo. They found success in getting records off the Echo. They were able to see certain metadata from the records and the records that were streamed across the Echo. Their software was limited due to its inability to extract private conversations or nonverbal tells of who is present near the Echo device (S. Li et al., 2019).

Chung, Park, and Lee (2017) also seek to display a viable forensics model, which centers around the combination of cloud technology with client-side forensics. Their model allows for the identification and procurement of data from the devices. They set up a testing environment with an Echo and uploaded data on different OS. They pulled the data from caches in the different OS into their software to get the data records. Their CIFT software relies on a custom user interface for the user to design their environment for their forensics process. The user then uploads the records into the software, and it is parsed for analysis. Their model did import data records from the Echo on different OS. It was successful for pulling in data from a local Echo from compatible software. They did not test outside of those parameters, and more work needs to be done. These data records were divided into smaller groups and the dashboard created by them did allow for custom interaction with the data (Chung, Park, et al., 2017).

While much research has been performed on IoT devices and different types of attacks, no prior research was found that examined the vulnerabilities of the Echo Dot despite its larger reach due to its smaller size and less expensive price than its predecessor Echo. This study expands the knowledge of IoT device security research by examining the security of the Echo Dot and examining the risks the wider population may face when using the lower cost product.

Research Study

The purpose of this research study is to investigate the security of the Echo Dot. The Echo Dot is a compact version of the original Amazon Echo. Its power supply comes from being plugged in to a fixed location. It does not have its own personal power supply built; it is typically kept in a fixed location. It has a built-in microphone and speakers to be able to interact with the user. The Echo Dot does not have the ability to process commands and give feedback on its own. Instead, the Echo Dot takes the command from the user and passes it along to the Amazon Cloud Services where it is processed and sent to Amazon's server. The response then passes back from the server to the cloud and then back to the Echo Dot to be broadcast back to the user. The Echo Dot is a Linux kernel without the ability to store much information on it. The Echo Dot architecture is shown in Figure 2.

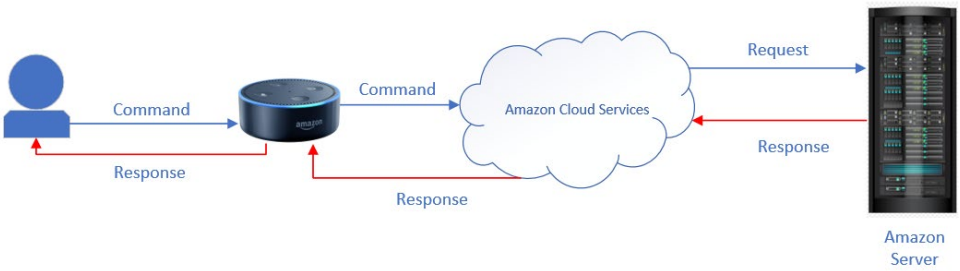

Figure 2. Echo Dot Architecture

Table 1 summarizes the literature review results for the Echo, which is the predecessor of the Echo Dot, and other IoT devices.

Table 1. Summary of Research Findings

Research	Type of attack	Finding
Lei et al (2018)	Vocal	Echo could be controlled by anyone other than the intended owner
Mitev et al (2019)	Vocal	Voice assistants can be hijacked by malicious skills apps
Jones et al (2019)	DDoS	Able to disconnect the user from the VPN using the DDoS over HPing3
Overstreet, et al (2019)	DDoS	They were able to take the Echo offline with the syn-flood attack
Haack et al (2017)	IVA	Alexa could pick up commands to the Echo from outside its immediate vicinity but could not change information stored such as delivery address
Li et al (2019)	Digital Forensics	was able to pull in the records from the Echo
Chung et al (2017)	Digital Forensics	was successful for pulling in data from a local Echo from compatible software
Li et al (2017)	MITM	Attacks are successful using vulnerabilities of middle man applications/networks

Based strictly on prior research on the Echo and its known vulnerabilities, we presume Amazon learned from its Echo implementations and other external testing and research and made improvements to the Echo Dot before introducing the product to the potentially larger market. Therefore, we presumed we would have difficulty hacking into or accessing the Echo Dot. Given the acknowledged lower quality of speakers, we focused our exploration on MITM, DDoS, DNS, and DHCP. Our research question was "Will the Echo Dot demonstrate less vulnerability to hacking attacks than previous IoT devices?"

The process used in this study begins with an initial scan of the Echo Dot and then proceeds to test for each of the vulnerabilities listed. We followed our attacks with a digital forensic analysis.

Methodology

For this study, an initial scan was performed on the Echo Dot. Then the device was tested for vulnerabilities against the threats found in the literature. For hacking purposes, two of the most important ports are port 80 and port 443 so we focused our efforts on those ports.

Scanning

The initial scan is a way to find out if a device on the network is live by pinging a device's IP address. Port scanning is used to find all live hosts on a network. When a user knows which hosts are live on a network, they can then find which ports are open on devices for communication. For hacking, finding live hosts on a network is how targets are found to attack. Port scanning is one of the main methods for hackers to perform reconnaissance. For our scanning, we used a tool through Kali Linux called Network Mapper (NMAP). NMAP is an open-source tool used for networking. We used NMAP to do reconnaissance on our Echo Dot to search for any vulnerabilities or access points we could exploit for other attacks. We issued commands using NMAP to scan our network, ping our Echo Dot, run a vulnerability scan on our Echo Dot, run a Protocol Scan on our Echo Dot, run a scan to acquire the OS, Scripts, Trace Routing, and Version Detection on the Echo device, and scanned to see if ports 80 and 443 were open.

Vulnerability testing

After verifying the device was active on the network, we started our analysis with a Man in the Middle (MITM) analysis and then proceeded with the vulnerability testing. Our testing took place in a local lab with a custom network set. We connected a 2nd generation Amazon Echo Dot into our network using the 2018 version of Kali Linux to carry out our pentesting. This version of Kali Linux was uploaded on bootable drives and connected to personal computers to create a VM interface to operate from. These computers were also hooked into our network.

MITM

For our MITM attack on the Echo Dot, we used two different methods. The first was a popular tool offered by Kali Linux called Ettercap. Ettercap allows the user to scan the network to find active IPs on the network. The user can set the gateway or other IP to spoof and target the victim's IP. We used Ettercap combined with Wireshark to monitor network traffic. We set the local gateway IP as our first target. We set the Echo IP address as our second target. We used the ARP Poisoning function of Ettercap and ran our attack. The second MITM attack launched was ARP Spoofing on the Echo Dot. ARP Spoofing is an attack used to forge ARP responses between two hosts to divert traffic to the attacker. It is a form of a MITM attack, so it follows the same structure.

We started by launching the command prompt and using the same "echo 1 > /proc/sys/ipv4/ip_forward" command to forward the packets to us. We then entered the command "arpspoof -i wlan0 -t 192.168.2.108 -r 192.168.2.1". For the command, we launched it across the wlan. It can be done over the ethernet if the network is wired. The "-t" followed by the IP address is our target IP address for the Echo Dot. The "-r" followed by the IP address is our Gateway address.

DHCP Starvation

For our attack on the Echo Dot, we used Yersinia. Yersinia is a pentesting tool in Kali Linux used to perform layer 2 attacks on networks. Once Yersinia was opened, the wlan option was selected as the global interface. DHCP mode was selected, and the "sending DISCOVER packet" option was selected. The "1" key was selected to launch the attack. We let the attack run for a few minutes before attempting to connect the Echo to the network. We tracked the traffic going across the network by running Wireshark in the background during the attack.

DDoS Network Flood

For our network flooding attack against the Echo Dot, we used HPing3. HPing3 is a penetration testing tool used to check security on a network. We coordinated several devices using HPing3 to go after the Echo Dot. We opened up the command prompt and entered the command "hping3 -c 99999 -d 999 --flood 192.168.2.108". The hping portion launches the HPing3 software. The -c represents the packet count, and in our attack, we were sending 99999 packets per second. The -d represents the data size, and in our attack each packet sent 999 bits. The - -flood is the speed at which the packets are being sent, and flood is the fastest option in HPing3. The IP address is the one of our Echo Dot. Each device entered this command simultaneously, flooding the Echo Dot with requests.

SYN Flood

Since servers and firewalls have gotten better at spotting traditional DDoS attacks by blocking abnormal amounts of uniform traffic from the same IP addresses, similar to the network flooding attack, we also launched a SYN Flood attack. In our vulnerabilities scan on the Echo Dot, we saw the ports 1080 and 8888 were open. Since these are higher numbered ports, these ports could have been left open by design to send traffic to and receive traffic from AWS. If one of these ports were left open for that purpose, we could cram malicious SYN requests into the open ports, blocking any connection to AWS. This would work if the ports were opened to communicate with AWS, allowing us to take advantage of open ports waiting for a response. We used a custom Python script to attack both ports individually with rogue packets.

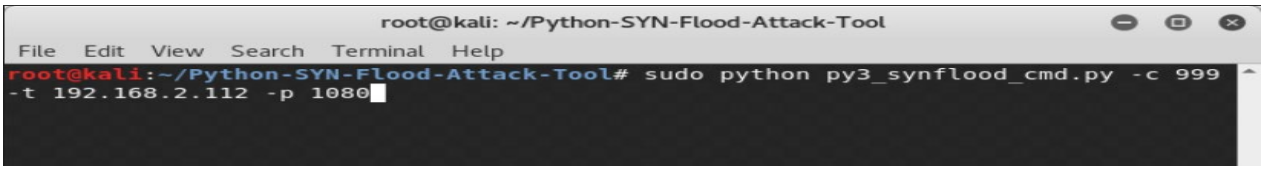

Figure 3 SYN Flood

Figure 3 shows the custom Python script. The first portion of the scripts call the attack. The -c represents the packet count. The -t is the target IP address, or in our case, the IP of the Echo Dot. The -p is the target port number. Once we launched the attacks from our devices, we issued commands to the Echo Dot in order to send traffic from the device.

DNS Hijack

When running our MITM attack, we noticed DNS Request by the Echo Dot to our Gateway. The 192.168.2.1 IP address is our gateway, and the 192.168.2.105 IP address is the Echo Dot. This prompted us to set up and carry out a DNS Hijacking attack. We knew that seeing DNS requests to our Gateway would allow us to reroute them to a local Windows server. We monitored the traffic through Wireshark to see where the Echo Dot was trying to reach. We configured our local Windows server to claim it was the destination the Echo Dot was searching for.

Digital Forensics

For our Digital Forensics investigation into the Amazon Echo Dot, we used a software package called Paraben's E3. This is a universal data processing suite that gives its users the ability to analyze data from electronic devices. Users can analyze devices, such as smartphones and laptops, or even extract data from the cloud. We could not directly connect the Echo into our computer to extract the data from it. Also, if this were possible, it is doubtful whether we would have been able to gain anything directly from the Echo due to its architecture.

We used the Cloud Forensics suite of E3 to gain access to the Echo Dot's data. Since we cannot directly hook the Echo into our computer and there is very little data stored on the Echo, we had to use a work

around. We set up a new case in the E3 software, and used the Cloud Import option to pull data in. We have an iPhone with the Amazon Alexa application installed with the credentials required on it. We backed the phone up into iTunes on the computer and encrypted it with a key known to us alone. We then selected the add an account option and then uploaded the backup files from iTunes into E3. We entered our encryption key and authenticated the credentials. Once this was added, all the information from the Echo Dot was imported into E3. We then had access and were able to examine the information that was passed over the Echo Dot.

Results/Findings

Scanning

We scanned our network and found the Echo Dot live on it. Once we found the Echo Dot was online, we pinged the device. The results showed there are two open ports on the device, port 1080 and port 8888. It also showed the number of closed ports on the device and showed how many ports are filtered. The filtered ports are masked during our scan, blocking us from seeing if they are open or not for us to access. In our command, the sudo nmap launches NMAP. The -sT portion is a switch that performs a full open scan used to find out which ports are open using TCP connection. We scanned for Common Vulnerabilities and Exposures (CVE) in the Echo Dot that we could exploit. None were found.

We then ran a Protocol scan on the Echo Device. This scan did not garner any different information that what was displayed in the pinging of the Echo Dot. We ran another scan to see the OS, Trace Routing, and Version Detection for the Echo device. The Echo Dot uses the Google Android OS. It did not display the version of protocols being used. It displayed the distance from the Echo Dot to the network. Since we could not tell if port 80 or 443 were open with our previous scans, we directly pinged the Echo Dot searching specifically for those ports. Both ports are filtered, so we were unable to tell if these ports were open or not.

MITM Analysis

We were able to see traffic over Wireshark, due to the encryption by Amazon, but we were unable to gain any useful information from the traffic between the two hosts in either our Ettercap attack or our ARP Spoofing attack. We were unable to pull any sensitive information from the Echo Dot.

While we were able to see traffic between the Echo Dot across the network, we were unable to gain anything of value from these attacks. Both forms of MITM we used showed traffic, but we could not see anything due to the encryption of Amazon.

DHCP Starvation

Once we let the attack run and attempted to connect the Echo Dot to the network, it was unable to connect. It began by spinning the light blue light around while attempting to connect. The device timed out and the light ring changed from blue to red. The Echo Dot gave the response: “I am sorry, I cannot connect to the Internet right now.” Once we ended the attack, the Echo Dot connected to the network.

DDoS Network Flood

Our devices flooded the Echo Dot with enough traffic to prevent it from being able to send requests to Amazon. The result was 100% packet loss by the Echo Dot, rendering it unable to process these requests. The Echo Dot was unable to connect to Amazon, successfully knocking the Echo Dot offline. When the device timed out while attempting to connect to Amazon, it stopped accepting vocal requests from users, rendering it useless.

SYN Flood

When running the attacks on the open ports, neither one was successful in blocking the Echo Dot. In both cases, the Echo Dot was able to receive our commands and issue responses from AWS. SYN requests were sent to the ports 1080 and 8888 during the attacks.

Since we were unable to jam the open ports with our SYN requests, we believe either these open ports are not used for communication between the Echo Dot and AWS, or if they are, the Echo Dot recognizes the IP addresses are not that of AWS, so the Echo Dot ignores them. Either way, our attack was unable to

block the connection of the Echo Dot to AWS, showing the Echo Dot is more than likely not susceptible to this type of attack.

DNS Hijack

The Echo Dot began sending its request to our local server, but it was unable to reach the Amazon servers. Once it realized that it was unable to reach the servers, the Echo Dot became unresponsive.

Digital Forensics

Once we obtained the data from the Amazon App on the iPhone, we ran it through the Paraben software. We preloaded commands and data on the Echo Dot through the app and voice commands. After running the data through Paraben, we were able to see the data we loaded. We then examined this data for a specific command we issued to the Echo Dot. This command was a malicious command given to the Echo Dot that brought back results. The record is highlighted in Figure 4.

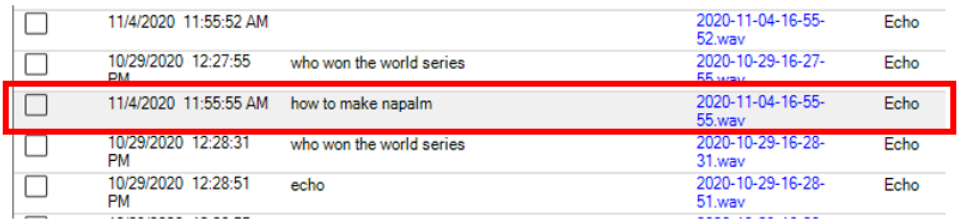

Figure 4: Malicious Command Extracted from Device

A summary of our findings shown in Table 2.

Table 2: Summary of finding

Type of attack	Previous Finding	Echo Dot
DDoS	Able to disconnect the user from the VPN using the DDoS over HPing3	Successfully attacked
DDoS	They were able to take the Echo offline with the syn-flood attack	Not successful in attack
Digital Forensics	was able to pull in the records from the Echo	Successful in pulling data from the Echo Dot
Digital Forensics	was successful for pulling in data from a local Echo from compatible software	Successful for pulling in data from a local Echo Dot from compatible software
MITM	Attacks are successful using vulnerabilities of middle man applications/networks	Not successful in launching MITM attacks

Based on the results of the study, the Echo Dot offers more protections than its predecessor Echo regarding DDoS and MITM but was as vulnerable with digital forensics. Based on the analyses we performed we reject our hypothesis that the Echo Dot would exhibit less vulnerability than the Echo and other previous IoT devices studied.

Conclusions and Implications for Practice

As stated earlier, for hacking purposes, two of the most important ports are port 80 and port 443. Not having the ability to see if they are open or having the ability to connect to them protects the Echo Dot owner's device. Hackers will have difficulty performing reconnaissance on the owner's Echo Dot. Ports are not needed in our MITM attacks, because the attacks sit between the device and the destination. In our case, the attacks sit between the Echo Dot and the Gateway. Due to the encryption of traffic between the Echo Dot and the AWS, our attacks are unable to pull the data. The architecture of the Echo Dot protects against MITM attacks. We conclude that owners of the Echo Dot do not need to be overly concerned with falling victim to this type of attack. The DHCP starvation attack is a layer two attack, so the Echo is limited with its abilities to directly defend against it. The Echo could be programmed to give more information as to why it cannot connect to the Internet. Instead of giving the feedback of the device cannot connect, add in the device cannot receive an IP address to connect to the Internet. This would allow the user to know it is not a flaw in the Echo Dot, but rather there is an issue with his or her DHCP server. The DDoS attacks performed on the Echo Dot were able to take it offline. As more IoT devices come online in the home and are connected to the Echo Dot, being able to disable the device would impact every IoT device connected it. In essence, an attacker could potentially take down a smart home if it is running by the Echo Dot. Amazon markets their device with the ability to control other IoT. Similar to the DDoS attack, we were able to take the Echo Dot offline using DNS Hijacking and a rogue server. We were able to show that by disconnecting the Echo Dot from Amazon's servers, the whole device is rendered useless. Again, for the user, if he or she uses his or her Echo Dot to control other IoT devices, taking it offline would disrupt those device's uses as well. To overcome this, we recommend Amazon hardcode their DNS server into the Echo Dots to protect against this attack. That way, rogue DNS servers cannot confuse the Echo while it is attempting to connect to AWS across local servers. While Paraben E3 is not widely available due to pricing, there are other digital forensics software available and that will be developed. Showing the ability to pull data from the Echo Dot using Paraben E3 demonstrates the Echo is able to be mined by this type of software. Amazon needs to find ways to increase their privacy settings in the Echo Dots to prevent data from being pulled by similar software.

While the Echo Dot proved slightly less vulnerable than the Echo, it was not secure. We recommend Amazon install either a feature to change the wake word to a custom one outside of the library of wake words provided or the ability of voice recognition to the owner of the device. Allowing for a custom wake word to be created for the Echo Dot would make it more difficult for nonauthorized users to directly interact with the Echo Dot. Creating voice recognition capability for the Echo Dot would also complicate the interaction of unauthorized users. As our digital forensics experiment showed, it is possible to upload malicious content on an Echo Dot and gain results. An unauthorized user could issue even more malicious commands, seeking results in areas such as child pornography or terrorism. If a forensics scan is done on the device, the user would be responsible for these commands, even if he or she is unaware of them. The ability to change the wake word to a custom one and the ability to set voice recognition would negate this from happening.

References

What is DDoS? (2021). Retrieved from https://www.cloudflare.com/learning/ddos/what-is-a-ddos-attack/.

Chung, H., Iorga, M., Voas, J., & Lee, S. (2017). Alexa, can I trust you? *Computer, 50*(9), 100-104.

Chung, H., Park, J., & Lee, S. (2017). Digital forensic approaches for Amazon Alexa ecosystem. *Digital Investigation, 22*, S15-S25.

Deogirikar, J., & Vidhate, A. (2017). *Security attacks in IoT: A survey.* Paper presented at the 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC).

Fruhlinger, J. (2020). What is IoT? The internet of things explained. *Networked World.* Retrieved from https://www.networkworld.com/article/3207535/what-is-iot-the-internet-of-things-explained.html

Haack, W., Severance, M., Wallace, M., & Wohlwend, J. (2017). Security analysis of the amazon echo. *Allen Institute for Artificial Intelligence*, 11.

Johnson, D. (2019) *What is the Amazon Echo Dot? Everything You Need to Know about Amazon's compact smart speaker.* Retrieved from https://www.businessinsider.com/what-is-amazon-echo-dot.

Jones, J., Wimmer, H., & Haddad, R. J. (2019). *PPTP VPN: An Analysis of the Effects of a DDoS Attack.* Paper presented at the 2019 SoutheastCon.

Lei, X., Tu, G.-H., Liu, A. X., Li, C.-Y., & Xie, T. (2018). *The insecurity of home digital voice assistants-vulnerabilities, attacks and countermeasures.* Paper presented at the 2018 IEEE Conference on Communications and Network Security (CNS).

Li, C., Qin, Z., Novak, E., & Li, Q. J. I. I. o. T. J. (2017). Securing SDN infrastructure of IoT–fog networks from MitM attacks. *4*(5), 1156-1164.

Li, S., Choo, K.-K. R., Sun, Q., Buchanan, W. J., & Cao, J. (2019). IoT forensics: Amazon echo as a use case. *IEEE Internet of Things Journal, 6*(4), 6487-6497.

Mitev, R., Miettinen, M., & Sadeghi, A.-R. (2019). *Alexa Lied to Me: Skill-based Man-in-the-Middle Attacks on Virtual Assistants.* Paper presented at the Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security.

Nawir, M., Amir, A., Yaakob, N., & Lynn, O. B. (2016). *Internet of Things (IoT): Taxonomy of security attacks.* Paper presented at the 2016 3rd International Conference on Electronic Design (ICED).

Nord, T. (2021) *What is an Intelligent Virtual Agent?* Retrieved from https://www.ultimate.ai/blog/ultimate-knowledge/what-is-an-intelligent-virtual-assistant-iva.

Overstreet, D., Wimmer, H., & Haddad, R. J. (2019). *Penetration Testing of the Amazon Echo Digital Voice Assistant Using a Denial-of-Service Attack.* Paper presented at the 2019 SoutheastCon.

Sterling, G. (2020) *More than 200 million smart speakers have been sold, why aren't they a marketing channel?* Retrieved from https://martech.org/more-than-200-million-smart-speakers-have-been-sold-why-arent-they-a-marketing-channel-2/.