School of Computer Science and Engineering (SCOPE)

B.Tech- Computer Science and Engineering

CSE3501 – Information Security Analysis and Audit

J Component

Final Report

Title: Security Analysis of Smart Home Devices and Mitigation With a Focus on ARP Poisoning

Submitted By

Jaivant Vassan – 19BCE2322

Divakar.A.S – 19BCE2337

Kabhilan S – 19BCE2339

B.S.Sandheep – 19BCE2341

Prithvi Saran S – 19BCE2344

Faculty: Dr. Vimala Devi K

Slot: L9 + L10

ABSTRACT:

The rise of smart technology in houses comes with many services and functions for everyday living. While a smart home is linked with significant promise in terms of comfort and risk treatment, it also adds new and adjusts current dangers. A smart home fundamentally is a communication network that links smart gadgets, sensors and actuators, enabling the owner to locally and remotely access, monitor and manage them. However, smart homes are presently encountering significant hurdles because to the underlying home automation systems, which are plagued by network security concerns. We have analysed the risks and vulnerabilities in these smart home devices and provided methods to mitigate such vulnerabilities with a focus on ARP poisoning. ARP poisoning (also known as ARP spoofing) is a form of LAN-based cyber-attack that includes delivering malicious ARP packets to a LAN's default gateway in order to alter the IP to mac address table's pairings.

PROBLEM STATEMENT

Smart homes are a typical instance of a dynamic ecosystem in which information and communication technology (ICT) penetration is high due to the use of ICT by various types of linked devices and locally or remotely distributed services.

Smart houses can be described from either a social or a technological standpoint. The former highlights the smart home's impact on human and societal requirements, whereas the latter defines the systems, processes, services, and smart gadgets that are linked to provide control over the home's environment.

A smart home can accommodate a wide range of components and entities, including utility providers, infrastructure providers, and third-party software or hardware manufacturers. Because of this diversity, the attack surface of the smart home is quickly expanding as additional security flaws are added, paving the path for an unstable and unsafe ecosystem.

ARP has a few fundamental security issues since it changes the host's ARP cache table in the absence of trustworthy mutual agreement processes while delivering request/reply messages. The attacker takes advantage of this lack of verification to manipulate the cache table by broadcasting fake ARP packets association his/her own MAC address to the IP address of the target or vice versa. The latter creates the perfect conditions for attacks like DOS and the former could lead to massive data leaks.

OBJECTIVE:

Our objective for this project is to perform a security analysis of Smart Home devices and assess their

1. Risks
2. Vulnerabilities
3. Possible Attacks

We also plan on discussing methods to prevent and mitigate these attacks with a focus on preventing ARP Spoofing attacks.

LITERATURE SURVEY:

Paper Details	Problem and Objectives	Proposed Methodology	Limitations
Vulnerabilities in IoT devices for smart home environment. Costa L, Barros JP, Tavares M. 5th International Conference on Information Systems Security e Privacy, ICISSP 2019	The researchers provide a technique designed to identify high risk vulnerabilities in smart home IoT devices in this article, with application examples of genuine flaws discovered in two commercially accessible products.	The method to identify vulnerabilities in smart home IoT devices proposed by the researchers is based on the PTES Standard.	The PTES standard required to be modified since it lacks several processes relevant to IoT smart home systems. Some of the information collection approaches proposed by the standard, for example, are aimed at acquiring information about individuals, such as a company's CIO.
A review of cyber security challenges, attacks and solutions for internet of	In this paper the researchers aim to review some recent articles regarding the most common issues of cybersecurity and	The paper proposes the use of methods like Virtual private Networks (VPNs), Encryption, Updated and regular protocol	Traditional security solutions for smart homes are inapplicable due to the heterogeneous components of smart

things based smart home Abdullah, Talal AA, Waleed Ali, Sharaf Malebary, and Adel Ali Ahmed Int. J. Comput. Sci. Netw. Secur. 19, no. 9 (2019): 139.	cyberattack that exploit the vulnerabilities of smart home environments and also provide suggestions and recommendations security methods to mitigate cyberattacks on smart homes.	and credential changes etc., to mitigate cyberattacks.	homes that have diverse types of smart home applications such as safeguarding houses, healthcare, energy, and convenience, as well as CPU and storage restrictions.
A New Detection and Prevention System for ARP Attacks Using Static Entry S. Hijazi and M. S. Obaidat IEEE Systems Journal, Sept. 2019	This paper proposes a new static IP-MAC approach for LAN specific client-server architecture, which does not need any extra constraint to fix ARP security problem. This approach was built for LANs under a static IP address settings. The experiments are conducted on different measurements of detection and prevention ARP attacks to find out the best results.	The idea used in this paper to prevent the ARP poisoning attacks is through a simple static entries solution where the dynamic IPMAC is mapped to the static state. There are many tools which are there to detect the arp attack for example ARPwner,wind ow ARP spoofed etc, But none of these software protect from ARP attack. In this particular paper they have adopted passive and active approach	1) The TCP is a stateful protocol. Thus, unlike ARP, replies will not be processed unless a request has been made. 2) The TCP uses sequence number and acknowledgement numbers for sequencing packets. These also provides authorization to packet. That is, a malicious node on the network cannot inject TCP packets in the communication.
Mitigation for Brute Force Attack against IP/CCTV Camera Login	The researchers aim to evaluate the different vulnerabilities of the IP camera which can be used by attackers to	First of all when user opens the portal of ip camera he/she will see the registration page where he/she has to	1. MD5 crypt's 128-bit output size would become the limiting factor in security. A brute force attacker could more easily find

Devang Thakar, Hepi Suthar IJRASET, Mar 2020	exploit and misuse the IP camera. Their finding shows the different vulnerabilities of the IP camera and not much strong mitigation system to prevent against those vulnerabilities. They aim to suggest a method to counter Brute Force attack and reiterate the need of good login system.	fill-up the details such as username, password, email id. Along with the basic details one more thing he/she has to input for registration which is an image of his/her choice. The only allowed file types for the image are PNG, JPG and JPEG. As soon as user enters the details and click on registration button. System will fetch the hash value of the image file using MD5 and SHA1 hashing algorithms. After fetching the hash value system will encrypt that hash value and finally store that value into the database. During login the user must upload the same image and only if the hashes match can the user login.	short strings hashing to the same value as a user's password than guess the actual password. 2. SHA-1 provides insufficient protection against collision attacks. An attacker could iterate over all possible combinations of secret key, creating a new hash until a matching hash was found.
Address Resolution Protocol Based Attacks: Prevention and Detection Schemes.	This paper discusses about the ARP poisoning attacks and focuses on reviewing various mechanisms developed for attack	Several solutions have been proposed for manipulate the ARP poisoning problem. The ARP watch, ARP Guard are manual solutions,	The Bombing Packets Attack, MAC, IP Cloning Attacks, the attacker can also pretend to be a receiver devise. This means impersonated

Francis Xavier Christopher D., Divya C. ICCBI 2018	detection and prevention with specified analysis to their advantages. Different attack detection and mitigation methods are evaluated in addition to comparison in terms of key parameters. This study helps in understanding the strategy employed for ARP attack detection and mitigation and developing a framework for improvement.	so these depend on administrator to process the ARP cache, which is achieved by specialized network tools. This solution involves assigning a static IP address to all hosts in the LAN, also setting VLAN (Virtual LAN) and so on. This technique laborious for administrators and there is no mechanism to distinguish between a malicious and genuine host, as well as this solution is unsuitable for DHCP environments.	an important entity like bank and obtain private information about user. In fact, ARP poisoning attacks violate all the security rules: confidentiality, integrity, and availability. Since the attacker
An analysis of security solutions for ARP poisoning attacks and its effects on medical computing. Prabadevi, B., Jeyanthi, N. & Abraham, A. Int J Syst Assur Eng Manag 11, 1–14 (2020).	This paper analyzes the existing defence systems against ARP attacks and proposes three different techniques for detecting and preventing the ARP attacks. The three techniques ensure security of traditional ARP and its impact in Medical computing where a single bit inversion could lead to wrong	Cross-layer Consistency Checking (CLCC), Timestamp and Counter based approach (TSCBA) and Extended TCBA in large data centre networks. (TCDCN)	TCDCN mitigation technique effectively performs the attack prevention by moving NULL MAC addresses, available MAC addresses, Multicast addresses detection before cross-layer inspection, thus reducing the computational time and cost involved in Data tables Scanning. However, it may still incur some considerable cost in

	diagnosis.		maintenance. TCDCN can detect the ARPbased DoS, MiTM, Cloning and host migration issues.
Towards Secure Smart Home IoT: Manufacturer and User Network Access Control Framework M. Al-Shaboti, I. Welch, A. Chen and M. A. Mahmood, IEEE AINA 2018	In this paper, the researchers propose an SDN-based framework for enforcing network static and dynamic access control, where manufacturers, security providers, and users can cooperate to enhance the smart home IoT security. They also proposed IPv4 ARP server as an NFV security service to mitigate ARP spoofing attack by replying to ARP requests in the network. They aim to implement a prototype to demonstrate the functionality of the framework against common attack scenarios (i.e. network scanning, ARP spoofing).	Proposed approach has three features: a) it allows the manufacturers to enforce the least privileged policy for IoT, and hence reduce the risk associated with exposing IoT to the Internet; b) it enables to enforce access policy as a feedback from security services; c) it enables users to customize IoT access based on social and contextual needs (e.g. only permits LAN access to the IoT through his/her mobile), which reduce the attack surface within the network.	The list of SDN challenges consists of: Controller placement, Scalability, Performance, Security, Interoperability and Reliability. SDN controllers must be wisely configured and the SDN's network topology authenticated to prevent manual errors and increase network availability. Applicable only for IPv4 ARP spoofing an nor IPv6. DPDK ARP server was able to handle only up to 50 parallel ARP requests

EXISTING METHODOLOGIES:

There are two techniques introduced to detect ARP spoofing: the passive approach and the active approach. The passive approach involves monitoring the ARP traffic and looking for inconsistencies in the IP-MAC mapping. The main drawback of this approach is the time lapse between learning and detecting spoofing. The detection that uses an active approach is injecting ARP packets into the network to probe for inconsistencies. The active technique is scalable, faster, intelligent, and more reliable in detecting attacks than the passive techniques.

A technique is in existence using ICMP requests by collecting and analysing the ARP packets, and then it uses ICMP request packets to probe for a malicious host, according to its response packets. The ARP spoofing detection architecture is divided into four modules.

1) The ARP packet sniffer module: This is meant to sniff all ARP packets from the Ethernet.

2) The IP-MAC mapping database: It compares two table entries in the database in order to check for a new MAC entry for the same IP entry. If found, it assumes it to be a spoof and then sends that IP to the ARP spoofing detection module.

3) The ARP spoofing detection module: It sends an ICM packet to the requesting IP address, and if a reply comes from that host, it decides if the host is legitimate or fake, with returns to the real MAC to update the database.

4) The Response module: It is used to alert the detecting ARP of a spoofing attack.

Detection using routing trace is used to find a change in the network movement path to protect the internal network. This technique detects ARP attacks through real-time monitoring (TTL) of the ARP cache table.

PROPOSED ALTERNATE METHODOLOGY:

A gateway-based approach to filter out the malicious ARP packets would be efficient. We plan to add an intelligent device capable of handling the packets about to be received by the devices in the network. Through IP - forwarding, the packets sent to the devices from other devices or vice versa could be made to pass through the aforementioned gateway, to undergo screening. While this gateway device would be capable of controlling the flow of the packets,(i.e.,block them from reaching the destination) it would need a reference table to check the authenticity of the the packets, we intend to create a script capable of identifying the fake packets by comparing the no. of request/response packets. The purpose of the script is to eliminate the need for manual updating of the reference table, as manual entries could prove inefficient when there is a large number of devices in the network.

ANALYSIS AND AUDIT:

SUMMARY:

For risk analysis section, we first figured out numerous developing and pre-existing hazards on diverse aspects. And we also studied impact and acceptance for various sorts of hazards. We also list out some risk vector categories that we have found. We also showed risk assessment categories of NIST. Then we performed IoT Risk Computation on several parameters. During study of vulnerabilities, we discovered out top 5 most exploited vulnerabilities on Iot Smart home devices. Then we collected all the vulnerabilities in present smart home gadgets. After we discussed about amazon echo and its vulnerabilities.

OBJECTIVES AND SCOPE:

The main objective of this report is to identify the vulnerabilities and Attacks such as Authentication bypass, WPA2 logic vulnerability etc on smart home devices and to prevent and mitigate it using appropriate methods and tools.

This audit report covers most of the threats, vulnerabilities, and Attacks on smart home devices. The result of this report provides a strong understanding on the security threats and the methods to prevent and mitigate it.

RISK ANALYSIS:

A risk, in general, is a deviation from a desired situation. With the wide range of technology available for home use, a wide range of targets and potential deviations emerge. In this section, we identify and describe the significant dangers connected with smart home devices, as well as their effect on impact and acceptance. In addition, we divide our analysis into developing and pre-existing threats. On the one hand, pre-existing risks are taken into account, such as those already addressed for households lacking smart home devices or services. Emerging risks, on the other hand, are risks that arise as a result of the integration of smart home applications into a household. They generally create or alter risks that are more difficult to quantify.

Emerging Risks:

Privacy: Emerging cyber threats relating to privacy and cyber security are among the most important dangers for smart homes, according to our research. The improper handling of personal user data obtained by smart homes is referred to as a privacy risk. As gadgets such as surveillance cameras and personal wearables become part of the smart home ecosystem, the most unwelcome outcome is privacy risks.

Cyber-Security: In contrast to the misuse of personal data associated with privacy concerns, cyber security risks refer to vulnerabilities and dangers in smart home device and service hardware, software, and data. Cyber-security risks can be classified into one of three categories: asset, vulnerability, or threat. The interaction of these three factors results in the determination of a specific cyber risk. A cyber risk is the potential damage to the smart home ecosystem caused by an attack that exploits certain vulnerabilities. Sensors, gateways, servers, application programming interfaces, mobile devices, and mobile device apps are examples of asset hazards. Certain categories of smart home architecture components, such as software, hardware, information, communication protocols, and human aspects, are prevalent. Overall, the dangerous assets are mostly those that are used and have their properties configured by the end user. Thus, cyber risks are mostly caused by software and mobile devices, as well as the apps and services that go with them.

Performance: A decline in the performance of a smart home product or service is associated with a growing performance risk. Typically, performance hazards arise from considerations about themes of greater technological interest and, as a result, have nearly universal applicability to all technologies. Technical reliability, warranties, and obsolescence are all risks that should be mentioned here.

Dependence: There is a concern that smart home technology will become a black box for normal homes, resulting in isolation, fraud vulnerability, or lock-in consequences. Initially, smart houses were intended to provide more control. However, use can lead to a lack of control, laziness, and other negative mental effects. Such dangers may have a negative impact on the consumers' peace of mind. Dependence risks are becoming more relevant and, for example, have a greater influence on overall risk perception than performance risks.

Access to Technology: On a societal level, additional concerns associated with smart home technology access emerge. This is a discrete yet cross-cutting issue from a risk standpoint. The exposure to today's pre-existing threats, such as fire and water, can be largely ascribed to socioeconomic conditions.

Social Isolation: Aside from the potential societal disparity in terms of technology access, smart home technologies and services may lead to an increase in technology-human interactions, displacing human-human connections. These considerations are strongly tied to worries about human detachment, which is a crucial factor in smart home acceptability. Users of smart homes may feel isolated from interpersonal contact, and this is especially true for older users or those with a distinct health focus.

Legal: Users perceive a risk connected with smart home vendors' lack of corporate accountability. These factors represent the user perspective and stem from unknown regulatory conditions or potentially short vendor lifetime, as the latter are frequently start-ups.
Time: The term perceived time risk refers to the amount of time that is wasted when using smart home technologies that may otherwise be spent on productive tasks.

Pre-Existing Risks:

Theft: On the one hand, eavesdropping on unprotected smart home devices provides criminals with a broader range of choices for committing crimes such as stalking or burglary. Insurance experts, on the other hand, see major benefits of smart home systems in terms of theft. One could conclude that criminal risk, mostly related with burglary and theft, is changing, but no consensus has been reached.

Waste of Resources: Smart houses are being pushed as a critical lever for meeting new climate targets. However, research suggests that increased data consumption as a result of smart home technology significantly raises global hazards, such as electricity usage or even everyday family work, and so reinforces unsustainable energy consumption.

Financial: Household damage frequently results in unexpected additional expenses or loss of income. The environment of smart homes broadens the potential sources of financial implications. According to research, the majority of cyberattack victims suffer financial losses. Similarly, as a result of potentially increasing dependence, there is a significant risk that smart home technology will lead to increased financial dependence. Thus, developing risks bring with them relevant new financial hazards, and many pre-existing risks eventually have a financial impact on the particular household.

Fire: With the increase of electronic devices due to the evolvement of smart homes, there is an increase chance of fire risk due to electronic or mechanical failure. Average fire-related insurance claims are the most expensive losses for households in the event of a mishap and it is further increase by the presence of expensive smart home devices.

Water: In contrast to fire losses, the likelihood of water damage is high and the severity is low. Flooding risk has its own important field of research that is heavily debating risk mitigation techniques.

Health: Many smart home use cases aim to improve people's health and well-being. In contrast to these advantages, it is uncertain whether smart home use introduces new health dangers. Health concerns about electromagnetic radiation are divisive. Such radiation becomes overly salient for high-risk perceivers, whereas radiation has a limited influence on moderate and low risk perceivers, negatively influencing total risk perception.

Overview of risks identified:

Impact – refers to the impact of smart homes on the risk

Acceptance – Describes the risks' influence on the acceptance of smart homes

Risk	**Impact**	**Acceptance**
Privacy	High	High
Cyber-Security	High	Low
Performance	High	High

S.no	Cloud-related	Real-time	Autonomous	Recovery
1	Cloud-computing platforms	Operational models in real time	Automated environments	Economic impact
2	Cloud technology skills	Customized products	Robotics and autonomous systems	Impact assessment
3	Cloud data centers	A platform for real time information	Robotics and artificial intelligence	SWOT (Strength, Weakness, Opportunities, Threat) analysis

Dependence	High	High
Access to Technology	Low	Low
Social Isolation	Low	High
Legal	Low	Low
Time	Low	Low
Theft	High	Low
Waste of Resources	Low	High
Financial	High	Low
Fire	High	High
Water	High	High
Health	Low	High

Risk Vectors:

There are four types of risk vector classes that have been identified: cloud-related, real time-oriented, autonomous, and recovery-related for Internet of Things (IOT) based smart home systems.

4	Cloud software	Digital real time and interoperable records	Robotics in IoT	Financial and fiscal state control
5	Cloud monitoring	Cyber-physical systems	Artificial intelligence and control systems	
6	Integration in cloud computing			
7	Cloud security networks			

Risk Assessment Categories of NIST:

NIST (National Institute of Standards and Technology) has come up with three main goals for IoT risk assessment:

(a) device protection,

(b) data protection, and

(c) user privacy

S.no	Device protection	Data protection	User privacy
1	Asset management	Strong encryption capability of IoT device	Disassociated data management
2	Vulnerability	Sanitation of sensitive	Informed decision

	management	data	making
3	Access management	Provide secure back-up	Processing permissions management
4	Incident detection	Verify the identification of other computing devices	Information flow management

Risk Rank for Smart Home Device Protection:

We prioritized the risks under consideration before determining the inherent risk and its impact. Risk impact is classified as high, medium, or low. A "high" impact rating, for example, indicates that the influence could be significant. The term "medium" denotes that the impact would be detrimental but reversible, as well as inconvenient. Low indicates that the influence would be minor or non-existent. The next stage is to assess the likelihood of the given exploit, taking into account the control environment in place.

IoT risk vector	Rank	Description/implication
IoT device does not have a unique built-in identifier	Medium	Remote access and vulnerability management are affected
IoT device's external dependencies are not revealed by the manufacturer	Medium	Managing the risk of external software and services are not possible
Patches or upgrades for the IoT device are not released by the manufacturer	Low	Known vulnerabilities cannot be removed
IoT device is not capable of having its software patched or upgraded	Medium	Known vulnerabilities cannot be removed

IoT risk vector	Rank	Description/implication
No vulnerability scanner that can run on or against the IoT device	Medium	Cannot automatically identify known vulnerabilities
The IoT device does not support the concealment of displayed password characters	Medium	Increases the likelihood of credential theft
The IoT device does not support strong credentials cryptographic tokens or multifactor authentication)	High	Tampering through credential misuse is possible
The IoT device does not support enterprise user authentication system	Medium	Each user needs more credentials
The IoT device is not able to log its operational and security events	Medium	Probability of detection of malicious activities are very less

IoT Risk Computation:

The purpose of this unique methodology is to calculate the cyber risk for IoT systems while taking IoT-specific elements into account, and then use this method to Smart Home gadgets to determine their risk level. The risk for any given device d is computed as follows:

$r(d)=w(d)\times s(d)$

where w represents the potential risk impact due to vulnerabilities/attacks and s represents the likelihood of the risk.

The following parameters are taken into account while calculating the risk impact.

a) Network structure: An unsecured network provides no protection and exposes all open traffic, resulting in the greatest risk impact. Insecure network services running on IoT systems that are also exposed to the Internet jeopardize information confidentiality, integrity, or availability, or allow unauthorized remote control.

b) Protocol type: The Internet of Things necessitates lightweight protocols such as 6LoWPAN and IEEE 802.15.4. Communication protocols such as MQTT, DSS, TCP, and UDP exist, as do connectivity protocols such as Wifi, Zigbee, Bluetooth, and RFID. Each protocol is vulnerable to attacks.

c) The number of heterogeneous systems engaged: If there are more intermediary systems involved, the risk's impact will be enormous. Critical IoT infrastructure systems with a greater number of heterogeneous devices are more vulnerable to cyber assaults, particularly network-related attacks.

d) Device security: An unsecured device is vulnerable to a wide range of threats. For example, in a malware attack, the amount of IoT devices that can be compromised is restricted to IP-based cameras for malware like Persirai and DVRs, routers, and CCTV cameras for malware like Mirai. The values are derived from the total number of IoT devices.

e) CIA type: If an attack compromises confidentiality, integrity, or availability, it will have a significant risk impact. If there is a replay attack (which affects confidentiality and integrity) and a DoS attack (which affects availability), the impact of the risk is substantial, and this might happen in the network layer of the implantable devices.

Risk Impact Parameters With Weights:

S. no	**Risk impact parameter (RIP)**	**RIP types**	**Weights (*W*)**
1	Type of network (nwt)	Unsecured network	10
		Network with minimum security	5
		Completely secured network	2
2	Protocol prone to attacks (prt)	Prone to more attacks	10

S. no	Risk impact parameter (RIP)	RIP types	Weights (W)
		Prone to fewer attacks	5
		Not prone to attacks	2
3	Count of heterogeneous systems involved (het)	More heterogeneous systems involved	10
		Few heterogeneous systems involved	5
		No heterogeneous systems involved	2
4	Device security (des)	Completely unsecured device	10
		Partially secured device	5
		Totally secured device	2
5	CIA type affected (cia)	CIA—all there are affected	10
		Only CI or IA or CA is affected	5
		Either C or I or A get affected	2

The risk impact *w* of device *d* can be derived as below.

w(d)=[nwt(d)+prt(d)+het(d)+des(d)+cia(d)]/5

To calculate the likelihood of the risk, the following parameters are considered

a. Count of previous attacks on the device (pat): If the device has a history of previous attacks, it is more likely that it will be attacked again.

b. An IoT layer that is subjected to several attacks (lyr): As previously stated, all layers of IoT are subjected to cyber threats, with the layer subjected to the most attacks receiving the most weight. A variety of attacks have been identified on the network layer of Smart Home devices.

b. IoT sector (scr): IoT is widely employed in industries, financial sectors, and healthcare sectors. It is critical to determine which industries are the most vulnerable to IoT threats.

d. Device risk factor (drf): There are a variety of smart home gadgets in use in households, each with a different risk factor. Devices that hold crucial personal and financial information, such as Amazon Echo, have a higher risk factor.

Risk Likelihood Parameters with Weights:

S. no	**Risk likelihood parameter (RLP)**	**RLP types**	**Weights (W)**
1	Past attacks on the device (pat)	Device underwent lots of past attacks	10
		Device underwent few past attacks	5
		Device underwent no attacks in the past	2
2	IoT layer with more attacks (lyr)	Network layer	10
		Application layer	5
		Physical layer	2
3	Sector (scr)	Healthcare	8
		Financial	7
		Others	5

S. no	Risk likelihood parameter (RLP)	RLP types	Weights (W)
4	Device risk factor (drf)	Amazon Echo, Google Home	9
		Smart Locks	8
		IP Cameras	6
		Automation System	4

The likelihood of risk can be derived as below:
S(d)=[pat(d)+lyr(d)+scr(d)+drf(d)]/4

VULNERABILITIES IN SMART HOME DEVICES:

Data breaches may cost businesses millions of dollars and put people in a vulnerable position. It's necessary to protect data and keep hackers at bay, but when your privacy at home is threatened by smartphone and Smart home gadgets, it's even more critical to prioritize safety.

The top five most exploited vulnerabilities in smart home devices are listed below.

1. Weak passwords:

Weak, readily guessed, or hardcoded and unencrypted passwords are the most often exploited vulnerability. It's remarkable to find hardcoded passwords in Smart home device source code after Mirai, the botnet that infected millions of Linux-based Smart home devices—but they still exist.
Any decent security expert will tell you that choosing a unique, multi-character password is essential for protecting your data from hackers.

Here are some suggestions for making a strong password:

- Make a password with at least 16 characters.
- Use two or more symbols (for example, @#$ percent).
- Include at least two numbers (e.g. 123456)
- Exclude characters with uncertain meanings ([] () /'" ', ;:. >)

- Make a long password that can be remembered by memory with the help of a mnemonic device.
- For each account or device, use a different password.

When two-factor authentication is feasible, you should use it. This adds an extra layer of protection to your devices and accounts.

Consider utilizing a password manager that can automatically create unique passwords for each account and reminds you to update your passwords on a regular basis. Weak and stale WiFi network passwords, for example, may jeopardize your whole home network, so changing them at least once every six months can help keep things safe.

2. Open or unsecured network services:

A cybercriminal's "in" might be via open or vulnerable network services like ports or guest networks. Guest networks let malicious actors traverse the network and check for additional weaknesses, thereby acting as a window into your network.

Smart home devices, like guest networks, are vulnerable to low-level hacking. When Smart home providers utilize open-source or reference-designed firmware without tweaking or updating fundamental templates, they often leave services like Telnet, which may be exploited to identify open ports, and others vulnerable to compromise. Checking for them or contacting a security expert, as well as sealing off anything that shouldn't be left open, will only help to protect your devices from these sorts of assaults.

Checking to discover whether you have any hacked Smart home devices might also help you uncover weaknesses in your network. It's extremely probable that your network already has malware, therefore employing a program like Minim to identify and neutralize issue devices with an appropriate reaction is a good idea.

3. Outdated devices:

Smart home device owners often disregard emails or notifications alerting them about security risks on their devices, resulting in hacked, out-of-date, or even legacy versions of device software. This is particularly important for gateway routers, which are often targeted by hackers.

Hackers were able to get usernames, passwords, credit card information, emails, and more by using the Krack (Key Reinstallation Attack) assaults against encrypted WiFi networks. Thousands of routers would still be vulnerable today if firmware upgrades were not available. And router manufacturers often provide monthly updates to secure home networks, so be sure to take advantage of these updates and keep current on the newest security measures.

4. Off-brand devices:

When it comes to smart home devices, the adage "you get what you pay for" is frequently accurate. Substandard "knock-off" or copycat smart home devices may infiltrate your network, gather data to transmit back to their maker, and get hacked far more readily than brand-name devices developed by much better InfoSec teams.

5. Poor physical security:

Physical security of your smart home devices is just as critical as keeping the software updated and locked down with a strong password, yet it's one of the most ignored parts of security. Because of their role, it might be difficult to safeguard all of the Smart home devices in a household (e.g. access points placed strategically for better signals, or the cable modem near the television). Many gadgets with WiFi or Ethernet capabilities, on the other hand, are left open and exposed to hackers.

To avoid leaving a door open, WiFi-enabled household equipment like as washers, refrigerators, and televisions should be programmed to be deactivated while not in use. Not only should Smart home devices be strategically located around the house, but your home should also be secured to prevent burglars from seeing inside to discover what gadgets you possess.

<u>VULNERABILTIES IN SMART HOME DEVICES:</u>

Devices	**Vulnerabilities**	**Attack difficulty**	**Impact**
Samsung, Windows, Google, Apple devices	Blueborne	Hard	Control devices
August lock	Plaintext BLE	Hard	Replay attack

Wemo switch	Port services available	Moderate	Root shell
Wemo devices	Authentication bypass	Moderate	Root shell
Ring’s smart doorbell	Plaintext credentials	Moderate	Leave Wi-Fi
Netvue HD camera	Stack buffer overflow	Hard	Remotely control
Lifx light, TP-Link camera, Nest thermostat, Linksys router, Sonos speaker, etc.	WPA2 logic vulnerability	Hard	Manipulate data
Sony android TV	Install unknown sources	Easy	Spy on users
Samsung SmartTV	Hacking tool (Weeping Angel)	Complex	Take over TV
Samsung SmartTV	Function vulnerabilities	Hard	Take over TV
LG SmartThinQ	Authentication logic vulnerability	Easy	Take over devices
Wink/Insteon hub	Plaintext credentials	Moderate	Root
Smart home App	Over-privilege	Moderate	Remotely control
BMW	Vehicle identification number and cross-site scripting vulnerability	Hard	Configure infotainment settings

Bmw, Mercedes-Benz, Chrysler	Internet-connected vulnerability	Hard	Full control
BMW	Bluetooth stack vulnerabilities	Easy	Unavailable resource

AMAZON ECHO:

Amazon Echo, frequently abbreviated to Echo, is an American brand of smart speakers manufactured by Amazon. Echo devices link to the voice-controlled intelligent personal assistant service Alexa, which will reply when a user speaks "Alexa". Users may alter this wake word to "Amazon", "Echo", or "Computer". The functionalities of the gadget include voice interaction, music playback, generating to-do lists, setting alarms, streaming podcasts, and playing audiobooks, in addition to delivering weather, traffic and other real-time information. It can also control many smart devices, operating as a home automation hub. The smart speaker has to utilize Wi-Fi to connect to the Internet since there is no Ethernet connector.

VULNERABILITIES OF ECHO DEVICE:

Various research has been undertaken to investigate Echo through physical access. For example, Ike Clinton et al. inverted the pins located at Echo’s bottom and debugged the device via the pins 1. Finally, they extracted the file system used by Echo and acquired the root power. Utilizing such root privilege on Echo’s file system, Mark Barnes installed a rogue program on Echo, then he created a root shell to access via the network, so that he sent microphone audio from the hacked Echo to his own server. Certainly, such effort may assist comprehend Echo’s internal myth or manage one’s own Echo in many ways, but cannot directly target others’ Echo remotely. Researchers from ISACA did a theoretical study on multiple attack surfaces of Echo, including network traffic encryption, firmware update, skill security, Alexa Voice API security, etc. No apparent vulnerabilities have been uncovered to date, which indirectly emphasizes the requirement of compromising voice control channel of Echo.

SQUATTERS AND MASQUERADERS:

Voice squatting is a tactic whereby a threat actor takes advantage or exploits the manner a skill or action is invoked. Let's take an example given from the researchers' white paper. If a user says, "Alexa, open Capital One" to run the Capital One skill, a threat actor might theoretically construct a malicious app with a similarly spoken name, such as Capital Won. The instruction intended for the Capital One skill is subsequently hijacked to execute the malicious Capital Won skill instead. Also, as Amazon is currently paying youngsters for saying "please" when directing Alexa, a similar takeover may occur if a threat actor uses a paraphrased name like Capital One please or Capital One Police.

"Please" and "police" may signify two very different things to humans, but for existing smart assistants, both phrases are the identical, since they cannot accurately identify one invocation name over another similar-sounding one.

Voice masquerading, on the other hand, is a strategy when a bad talent impersonates a genuine one to either fool users into handing up their personal information and account passwords or eavesdrop on talks without user knowledge.

Researchers found two methods this attack may be made: in-communication skill swap and fake termination. The former takes advantage of the mistaken idea that smart assistants quickly move from one ability to another whenever users activate a new one. Going back to our previous example, if Capital Won is already running and the user decides to ask "Alexa, what'll the weather be like today?", Capital Won then pretends to hand over control to the Weather skill in response to the invocation when, in fact, it is still Capital Won running but this time impersonating the Weather skill.

As for the latter, fake termination exploits volunteer skill termination, a feature whereby skills may self-terminate after giving a verbal response such as "Goodbye!" to users. A malevolent skill may be built to say "Goodbye!" yet stay operating and listening in the background for a certain duration of time.

ATTACKS AND MITIGATION:

1. ARP SPOOFING:

ARP spoofing is a type of attack in which a malicious actor sends falsified ARP (Address Resolution Protocol) messages over a local area network. This results in the linking of an attacker's MAC address with the IP address of a legitimate computer or server on the network. Once the attacker's MAC address is connected to an authentic IP address, the attacker will begin receiving any data that is intended for that IP address. ARP spoofing can enable malicious parties to intercept, modify or even stop data in-transit. ARP spoofing attacks can only occur on local area networks that utilize the Address Resolution Protocol.

ATTACK METHODOLOGY:

ARP spoofing attacks typically follow a similar progression. The steps to an ARP spoofing attack usually include:

1. The attacker uses ettercap to scan for the IP and MAC addresses of hosts in the target's subnet.
2. The attacker chooses its target and begins sending ARP packets across the LAN that contain the attacker's MAC address and the target's IP address.
3. As other hosts on the LAN cache the spoofed ARP packets, data that those hosts send to the victim will go to the attacker instead. From here, the attacker can steal data or launch a more sophisticated follow-up attack(MITM in this case).

CODE:

To enable IP forwarding on gateway device(terminal unix commands):

```
echo 1 > /proc/sys/net/ipv4/ip_forward
cat /proc/sys/net/ipv4/ip_forward
```

Gateway poisoned ARP packet detection program:

```
from scapy.all import Ether, ARP, srp, sniff, conf
def get_mac(ip):
    """
    Returns the MAC address of `ip`, if it is unable to find it
    for some reason, throws `IndexError`
    """
    p = Ether(dst='ff:ff:ff:ff:ff:ff')/ARP(pdst=ip)
    result = srp(p, timeout=3, verbose=False)[0]
    return result[0][1].hwsrc
def process(packet):
    # if the packet is an ARP packet
    if packet.haslayer(ARP):
```

```
        # if it is an ARP response (ARP reply)
        if packet[ARP].op == 2:
            try:
                # get the real MAC address of the sender
                real_mac = get_mac(packet[ARP].psrc)
                # get the MAC address from the packet sent to us
                response_mac = packet[ARP].hwsrc
                # if they're different, definetely there is an attack
                if real_mac != response_mac:
                    print(f"[!] You are under attack, REAL-MAC: {real_mac.upper()}, FAKE-MAC: {response_mac.upper()}")
            except IndexError:
                # unable to find the real mac
                # may be a fake IP or firewall is blocking packets
                pass


if __name__ == "__main__":
    import sys
    try:
        iface = sys.argv[1]
    except IndexError:
        iface = conf.iface
sniff(store=False, prn=process, iface=iface)
```

DEFENSE METHODOLOGY:

A gateway-based approach to filter out the malicious ARP packets would be efficient. We plan to add an intelligent device capable of handling the packets about to be received by the devices in the network. Through IP - forwarding, the packets sent to the devices from other devices or vice versa could be made to pass through the aforementioned gateway, to undergo screening.While this gateway device would be capable of controlling the flow of the packets,(i.e.,block them from reaching the destination) it would need a reference table to check the authenticity of the the packets, we intend to create a script capable of identifying the fake packets by comparing the no. of request/response packets. The purpose of the script is to eliminate the need for manual updating of the reference table, as manual entries could prove inefficient when there is a large number of devices in the network.

SCREENSHOTS:

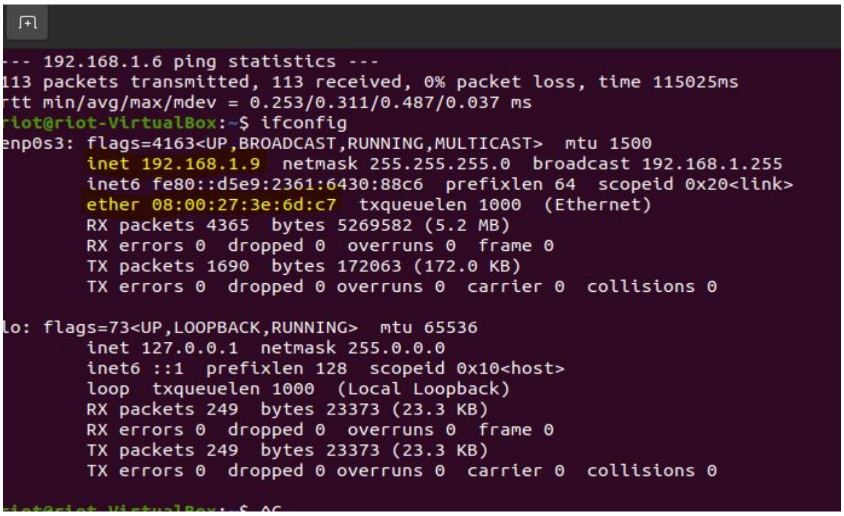
Server's ARP table(Before the attack):

```
Administrator: Command Prompt

C:\Users\RIOT>arp -a

Interface: 192.168.1.6 --- 0x7
  Internet Address      Physical Address      Type
  192.168.1.1           00-6d-61-ac-ea-03     dynamic
  192.168.1.4           08-00-27-f1-ba-21     dynamic
  192.168.1.9           08-00-27-3e-6d-c7     dynamic
  224.0.0.22            01-00-5e-00-00-16     static
  239.255.255.250       01-00-5e-7f-ff-fa     static

Interface: 192.168.56.1 --- 0xb
  Internet Address      Physical Address      Type
  224.0.0.22            01-00-5e-00-00-16     static
  239.255.255.250       01-00-5e-7f-ff-fa     static

Interface: 25.13.241.150 --- 0x17
  Internet Address      Physical Address      Type
  224.0.0.22            01-00-5e-00-00-16     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

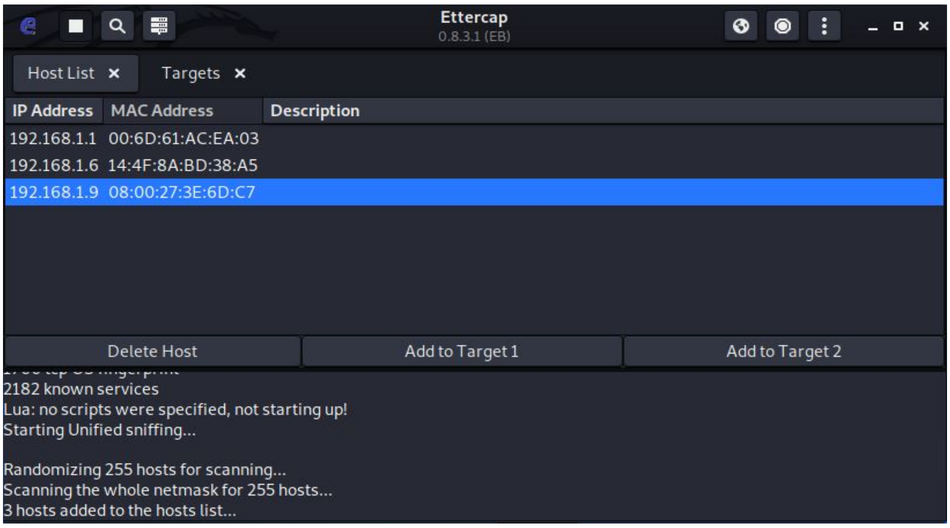
Server's MAC and IP address:

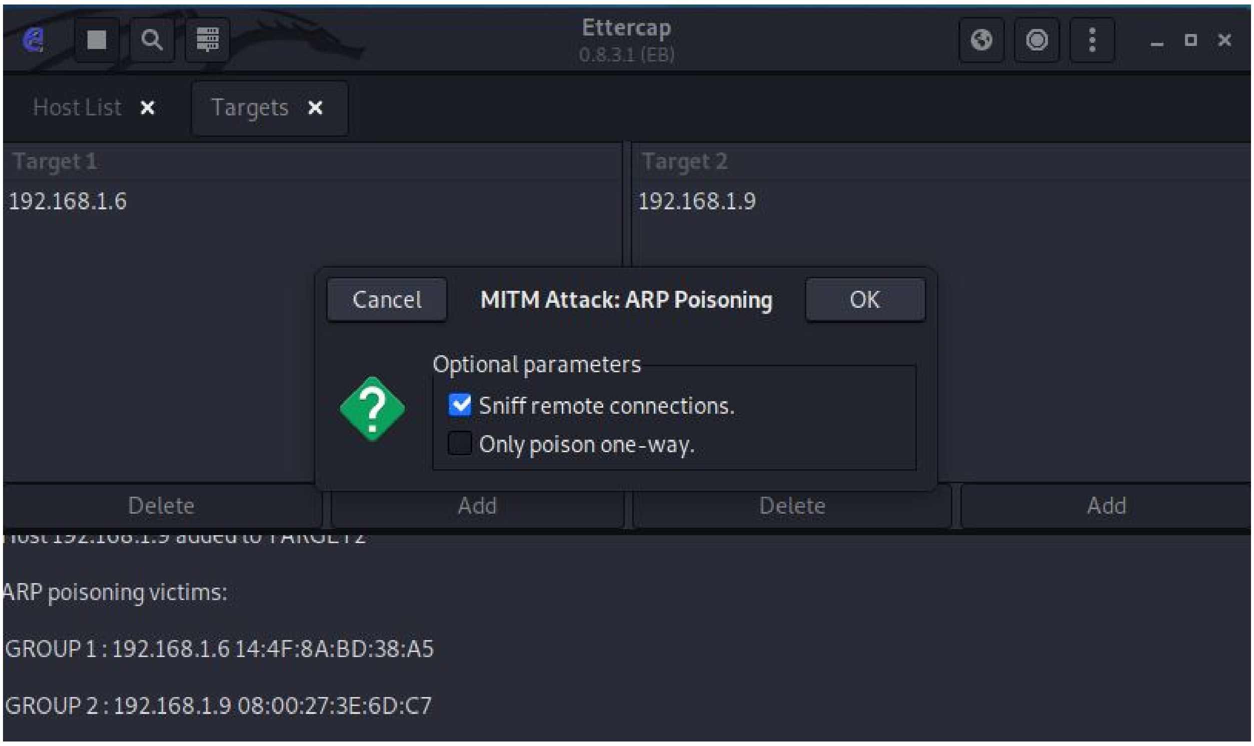
```
Administrator: Command Prompt

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Intel(R) Wireless-AC 9560 160MHz
   Physical Address. . . . . . . . . : 14-4F-8A-BD-38-A5
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::b414:8161:4b1c:3b21%7(Preferred)
   IPv4 Address. . . . . . . . . . . : 192.168.1.6(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : 15 November 2021 10:23:11
   Lease Expires . . . . . . . . . . : 17 November 2021 00:21:58
   Default Gateway . . . . . . . . . : 192.168.1.1
   DHCP Server . . . . . . . . . . . : 192.168.1.1
   DHCPv6 IAID . . . . . . . . . . . : 34885514
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-25-7F-BE-FD-C8-D9-D2-7F-89-A5
   DNS Servers . . . . . . . . . . . : 1.1.1.1
                                       1.0.0.1
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

Client's MAC and IP address:



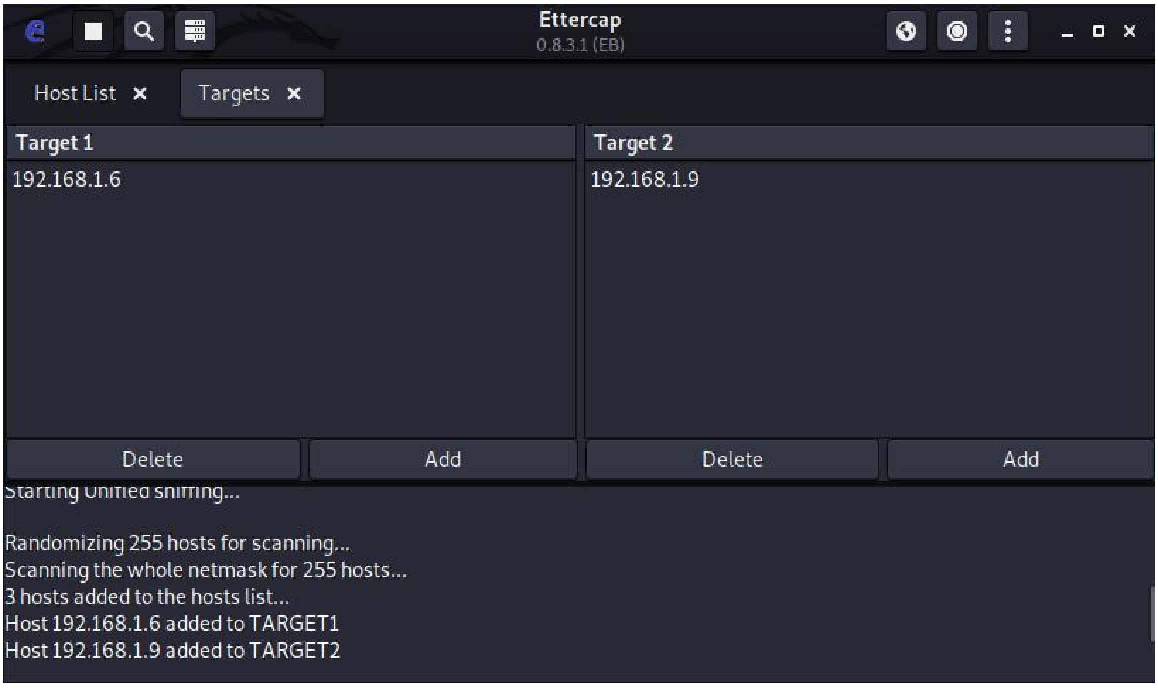
List of hosts on the LAN (gathered by Ettercap):





Client and Server added as targets in Ettercap:

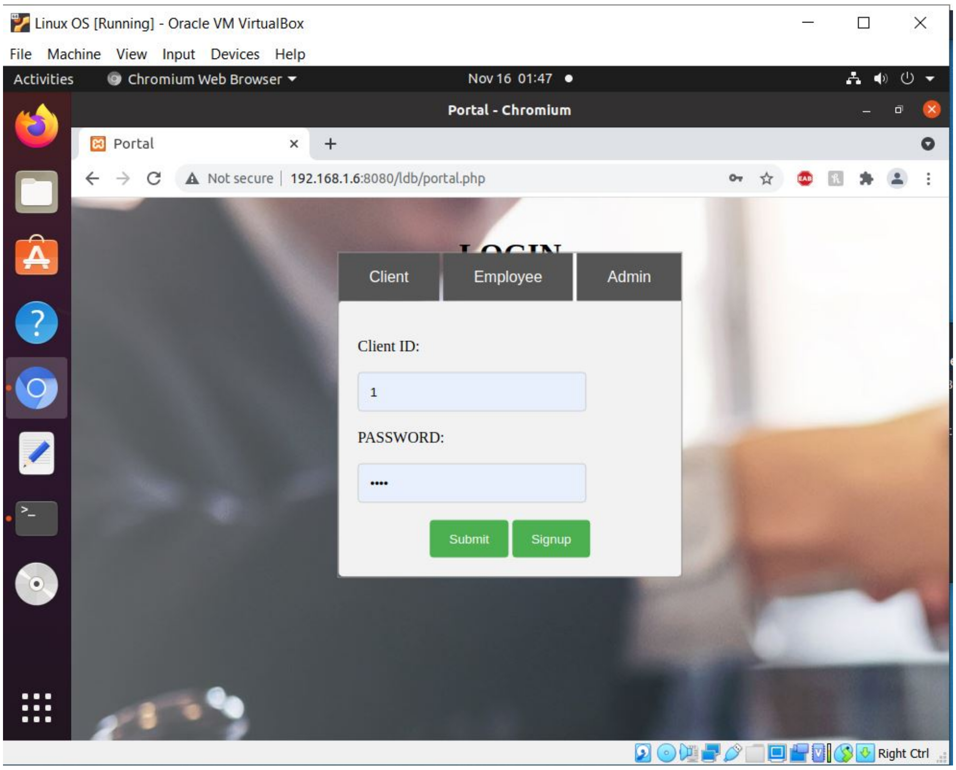
Initiating MITM attack by sending poisoned ARP packets to Server and Client:



Server and client’s ARP tables(After the attack):

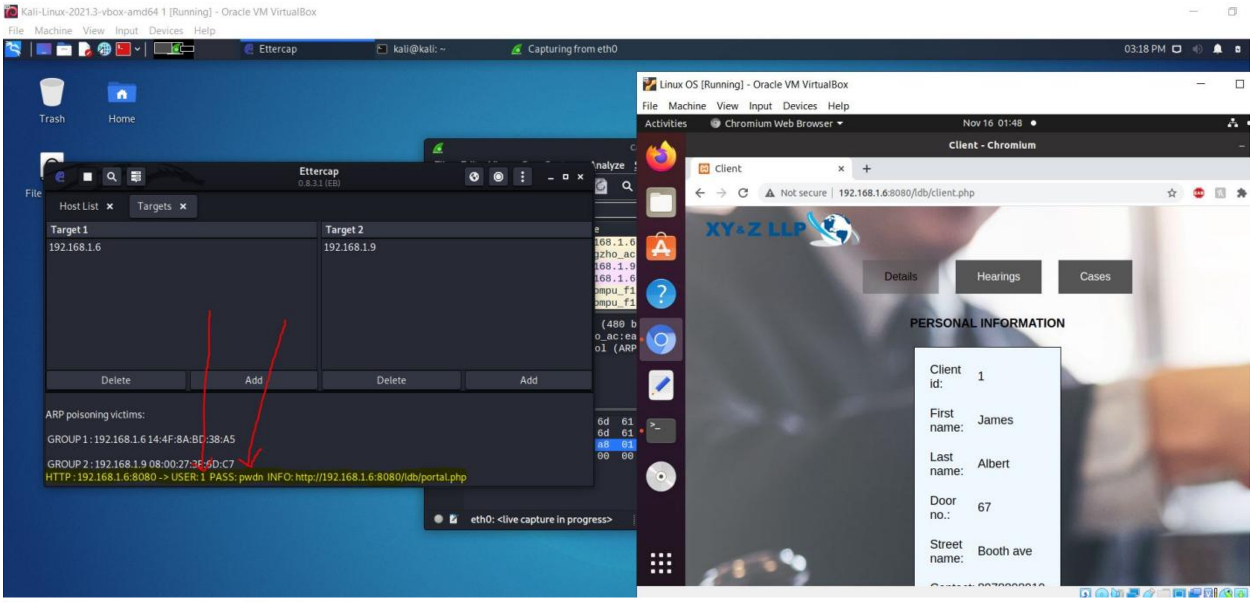
```
C:\Users\RIOT>arp -a

Interface: 192.168.1.6 --- 0x7
  Internet Address      Physical Address      Type
  192.168.1.1           00-6d-61-ac-ea-03     dynamic
  192.168.1.4           08-00-27-f1-ba-21     dynamic
  192.168.1.9           08-00-27-f1-ba-21     dynamic
  192.168.1.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  239.255.255.250       01-00-5e-7f-ff-fa     static

Interface: 192.168.56.1 --- 0xb
  Internet Address      Physical Address      Type
  192.168.56.255        ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  239.255.255.250       01-00-5e-7f-ff-fa     static

Interface: 25.13.241.150 --- 0x17
  Internet Address      Physical Address      Type
  25.0.0.1              7a-79-19-00-00-01     dynamic
  25.255.255.255        ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static

C:\Users\RIOT>
```

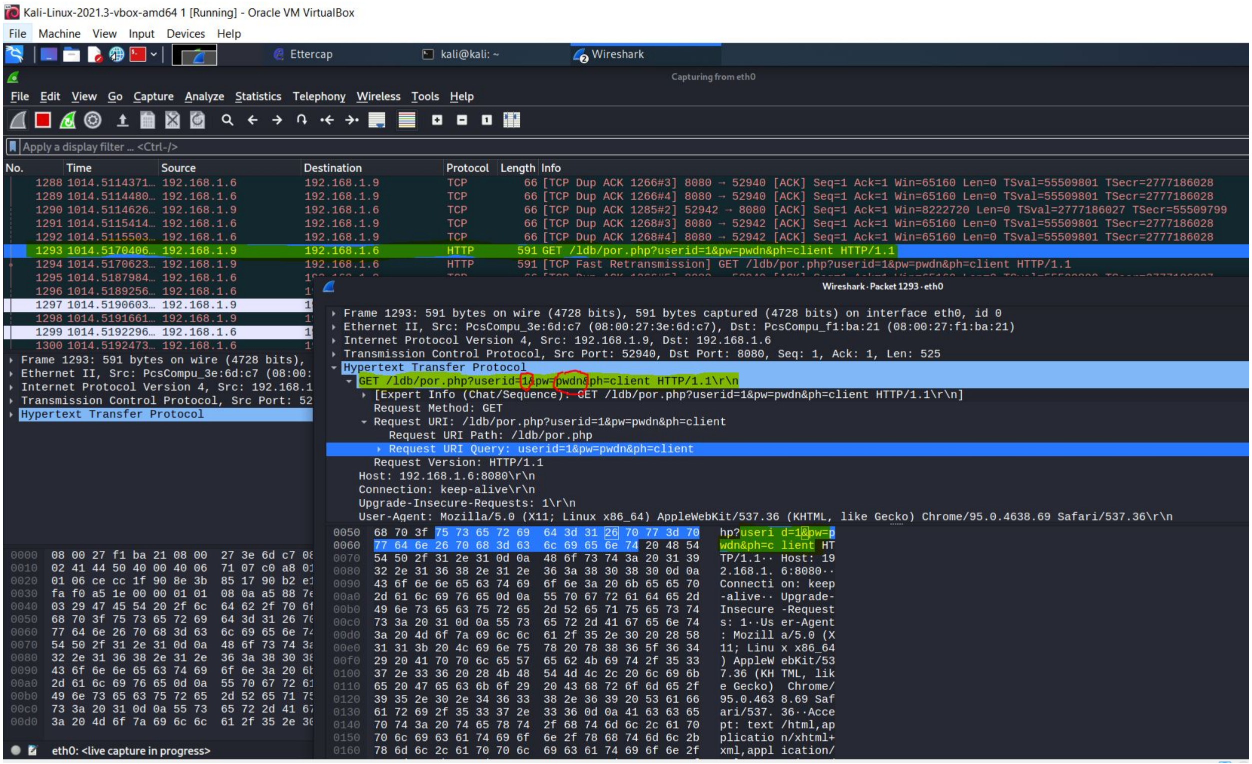
```
riot@riot-VirtualBox:~$ arp -a
? (192.168.1.1) at 00:6d:61:ac:ea:03 [ether] on enp0s3
? (192.168.1.6) at 08:00:27:f1:ba:21 [ether] on enp0s3
? (192.168.1.4) at 08:00:27:f1:ba:21 [ether] on enp0s3
riot@riot-VirtualBox:~$
```



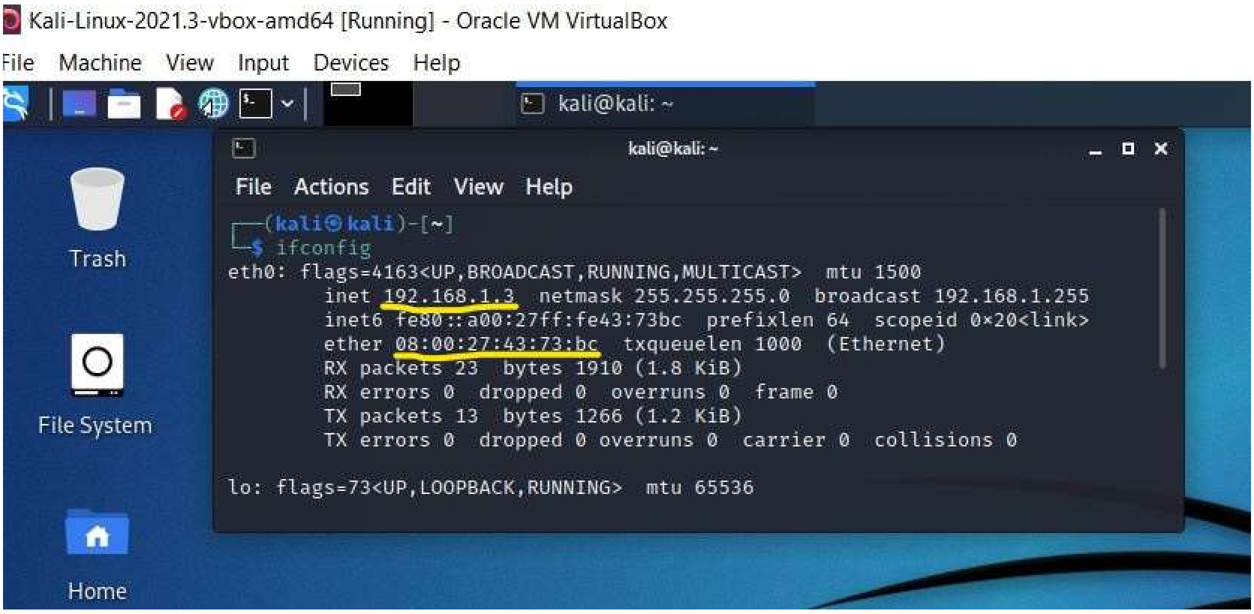
Accessing server from client device:

The attacker extracts sensitive information as the packets involved in the above communication go through the attacker's device:





A new device(gateway) is configured and added:

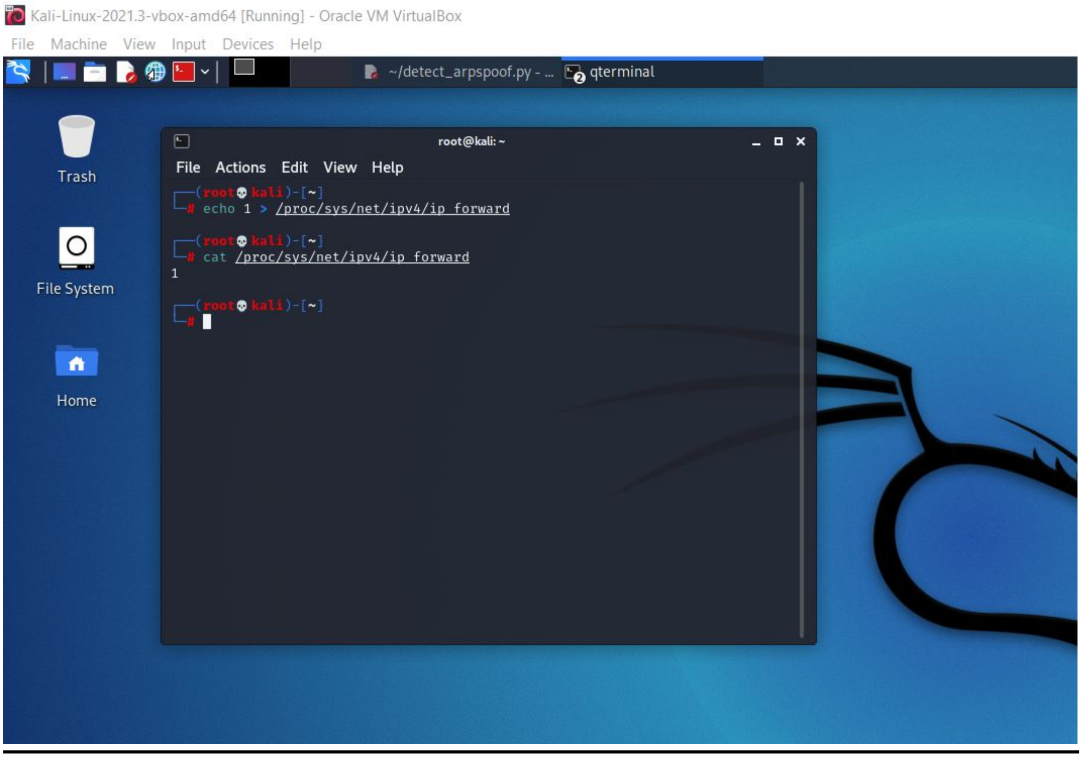


Client and server’s ARP tables(After adding gateway device to the network):

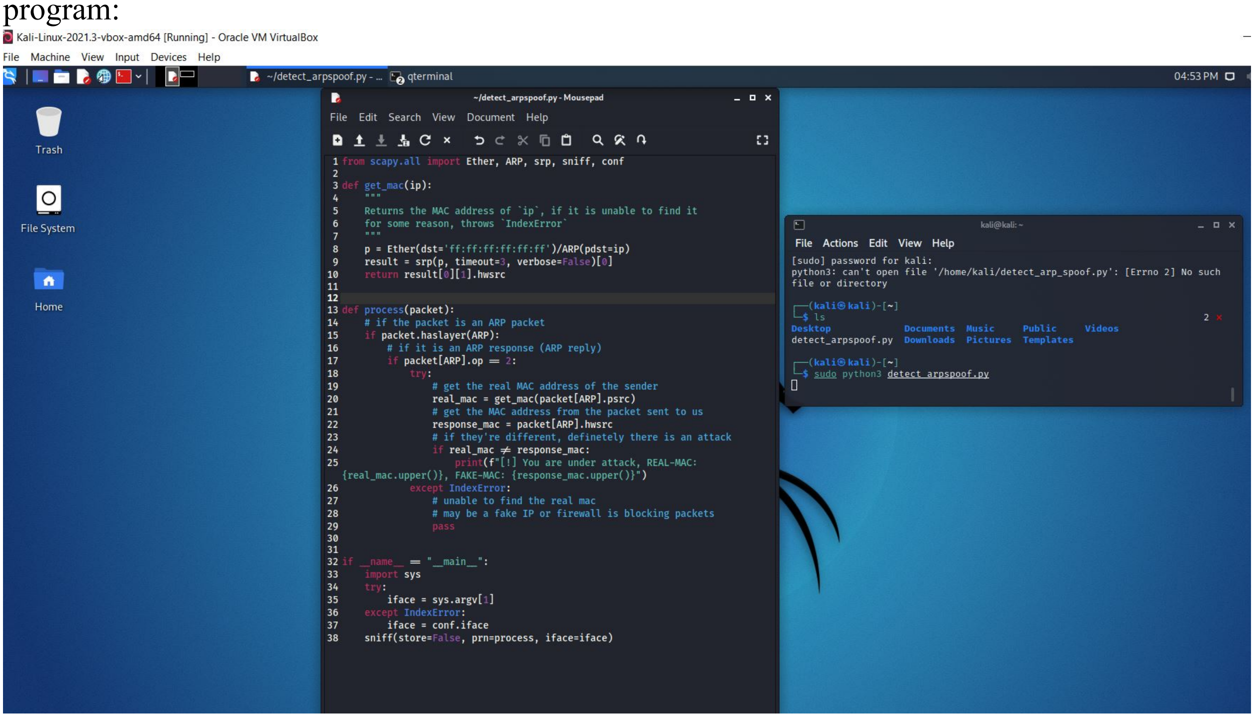
```
Administrator: Command Prompt

C:\Users\RIOT>arp -a

Interface: 192.168.1.6 --- 0x7
  Internet Address      Physical Address      Type
  192.168.1.1           00-6d-61-ac-ea-03     dynamic
  192.168.1.3           08-00-27-43-73-bc     dynamic
  192.168.1.4           08-00-27-f1-ba-21     dynamic
  192.168.1.9           08-00-27-f1-ba-21     dynamic
  192.168.1.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  239.255.255.250       01-00-5e-7f-ff-fa     static

Interface: 192.168.56.1 --- 0xb
  Internet Address      Physical Address      Type
  192.168.56.255        ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  239.255.255.250       01-00-5e-7f-ff-fa     static

Interface: 25.13.241.150 --- 0x17
  Internet Address      Physical Address      Type
  25.0.0.1              7a-79-19-00-00-01     dynamic
  25.255.255.255        ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

```
riot@riot-VirtualBox:~$ arp -a
? (192.168.1.3) at 08:00:27:43:73:bc [ether] on enp0s3
? (192.168.1.1) at 00:6d:61:ac:ea:03 [ether] on enp0s3
? (192.168.1.6) at 08:00:27:f1:ba:21 [ether] on enp0s3
? (192.168.1.4) at 08:00:27:f1:ba:21 [ether] on enp0s3
riot@riot-VirtualBox:~$
```

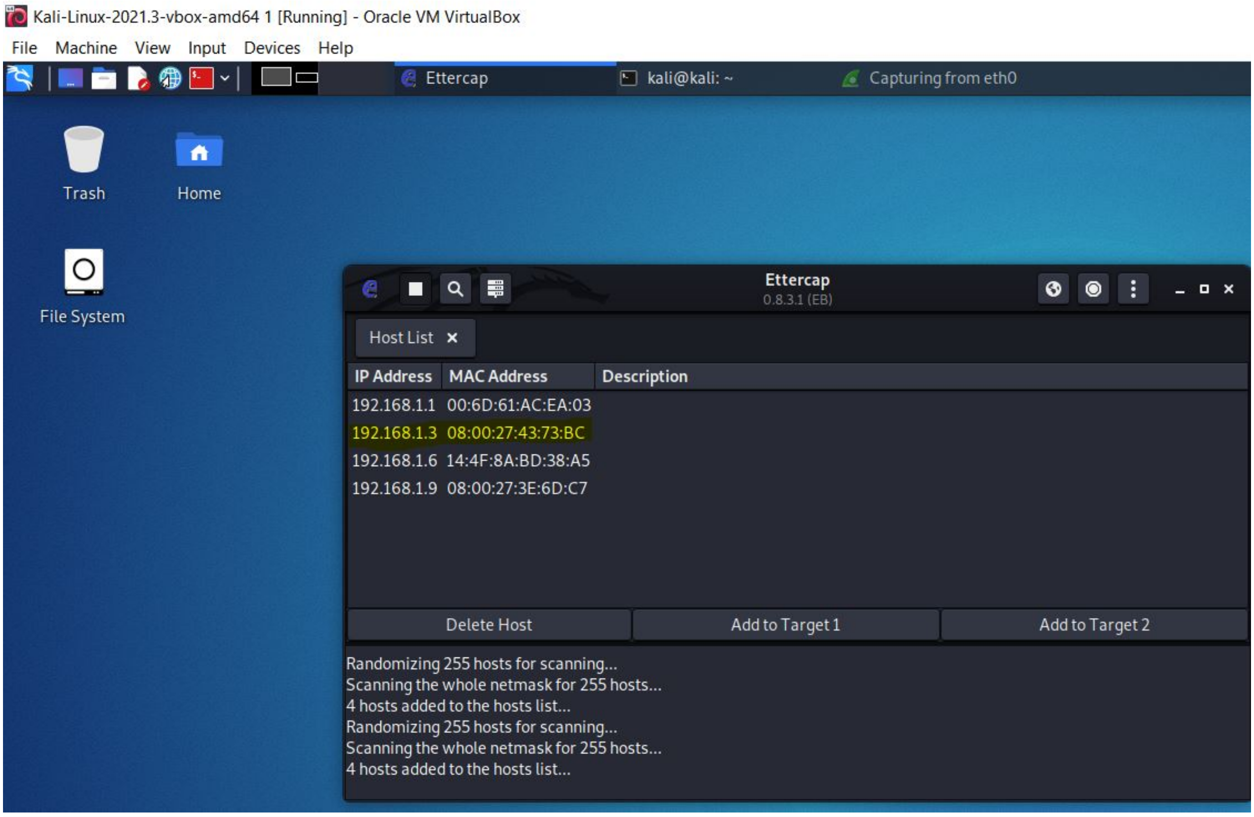
Enabling IP forwarding on the gateway device:

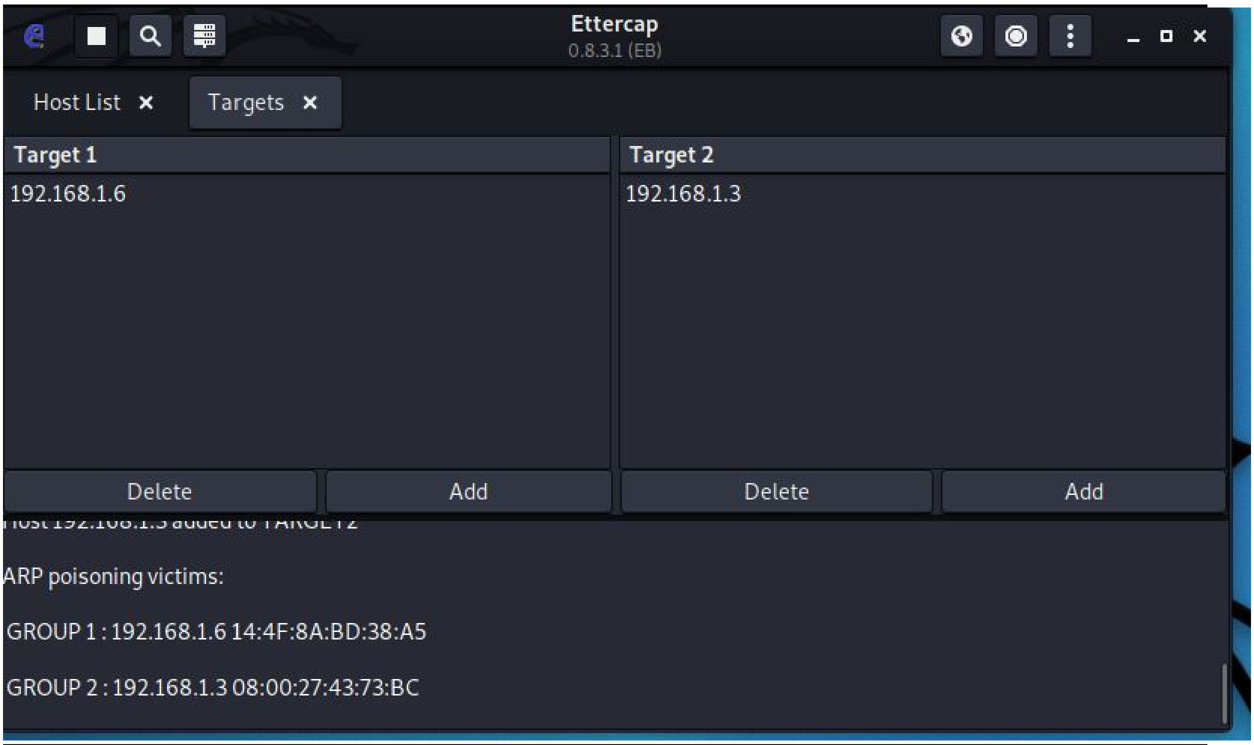


Gateway poisoned ARP packet detection program:

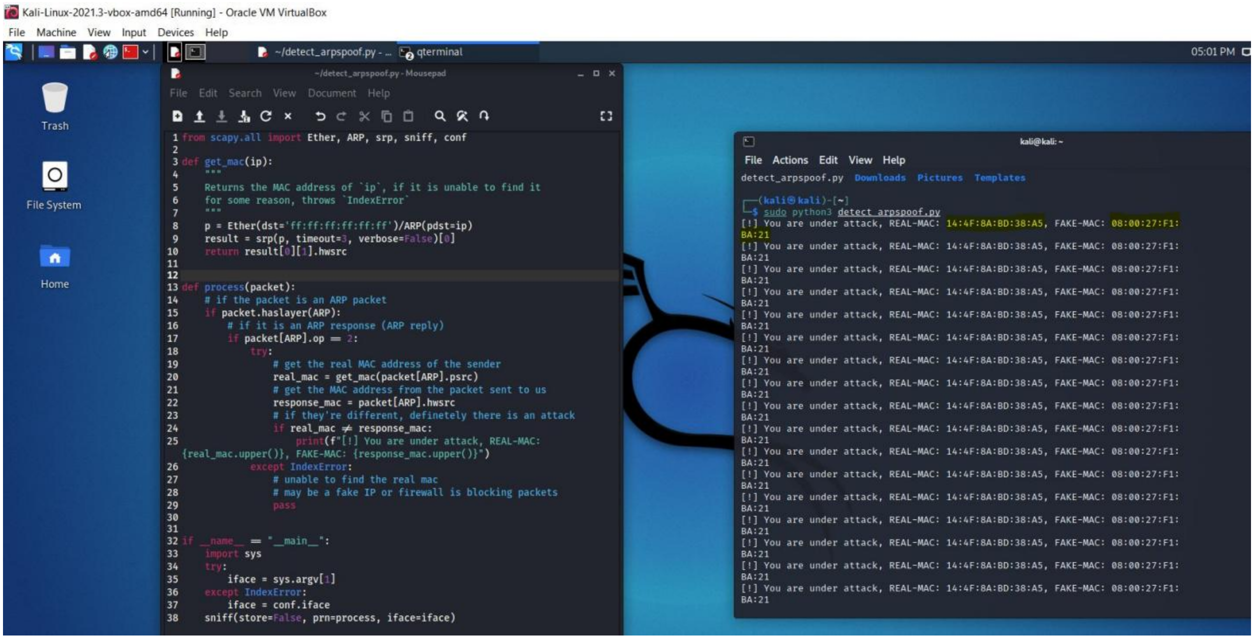


The gateway device is also added as a target and MITM is attempted:

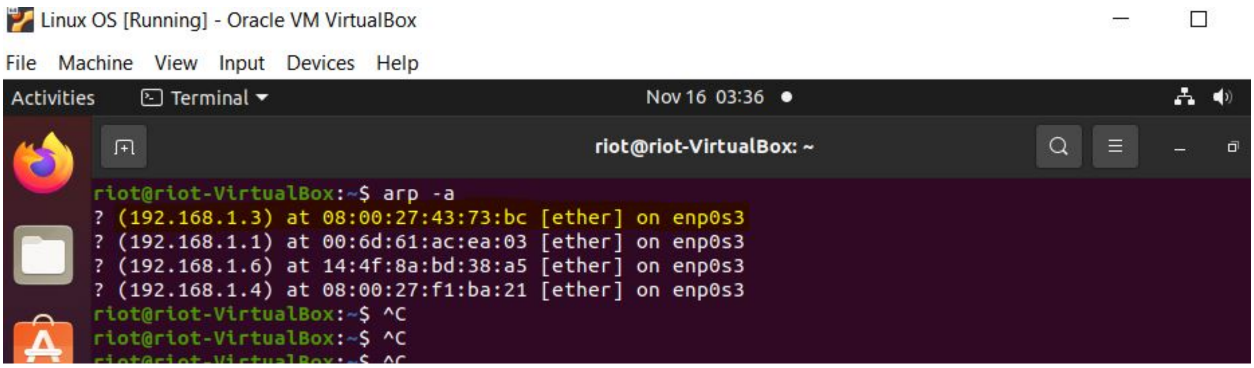




The poisoned ARP packets meant to modify the ARP tables such that the server's IP is mapped to the attacker's MAC address is detected by the python script thus we can take appropriate action:



Server and client’s ARP tables(unaffected even after the attack thanks to gateway and the python script):



```
Administrator: Command Prompt
  255.255.255.255       ff-ff-ff-ff-ff-ff     static

C:\Users\RIOT>arp -a

Interface: 192.168.1.6 --- 0x7
  Internet Address      Physical Address      Type
  192.168.1.1           00-6d-61-ac-ea-03     dynamic
  192.168.1.3           08-00-27-43-73-bc     dynamic
  192.168.1.4           08-00-27-f1-ba-21     dynamic
  192.168.1.9           08-00-27-3e-6d-c7     dynamic
  192.168.1.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  239.255.255.250       01-00-5e-7f-ff-fa     static

Interface: 192.168.56.1 --- 0xb
  Internet Address      Physical Address      Type
  192.168.56.255        ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  239.255.255.250       01-00-5e-7f-ff-fa     static

Interface: 25.13.241.150 --- 0x17
  Internet Address      Physical Address      Type
  25.0.0.1              7a-79-19-00-00-01     dynamic
  25.255.255.255        ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static

C:\Users\RIOT>
```

2. DENIAL OF SERVICE (DOS) ATTACK:

DoS is a difficult attack on computing machines that involves blasting them with requests for a set amount of time, causing them to crash, slow down, or shut down completely. DoS attacks may do more damage to IoT devices due to their restricted resources. The majority of IoT devices rely on low-cost hardware and IEEE 802.11-based networks. IoT devices are targeted by the most attackers due of their widespread use. Researchers are working hard to address these flaws in 802.11 networks by introducing new security standards such as WPA, EAP, 802.11i, and 802.1x to the protocol. When in infrastructure mode, an 802.11 network requires the wireless device to connect to an AP before data messaging can take place. Before talking with the AP, the device must first validate itself to the AP. If either the client device or the access point wants to disconnect from the other, they transmit a de-authentication frame. When client devices and APs communicate, these frames are unencrypted, and an attacker can easily spoof these frames, which contain the device and AP's unencrypted MAC addresses. Attackers can easily launch a DoS (de-authentication) attack using them to disconnect the client device from the AP.

MITIGATION:

DoS attacks in smart homes can be detected using a graph-based technique. Nodes represent connected devices, and edges represent communication between them in the graph approach. Even if a DoS attack disables one device, the system as a whole may appear to be operational. GODIT (Graph-Based Outliner Detection in the Internet of Things) claims to examine each entity (node) in the IoT network and study its performance in relation to the entire system. When compared to other DoS detection methods that require more parameters such as protocols and packet size, the GODIT approach uses only the source IP and destination IP to generate the graph of the network's data/traffic flow.

A Honeypot system operates as a decoy by imitating the behavior and features of the intended primary server. To work, the decoy needs three things: a computer, an application program, and some specified data. The DoS assault is forwarded to this fake server, which serves as a shield for the intended target server. The attackers are tracked and their activities are traced in order to research and evaluate them in order to prevent future attacks.

3. EAVESDROPPING ATTACK:

This type of attack is also known as sniffer. It's used to sniff network traffic in wireless networks that employ Bluetooth, IEEE 802.11x, or RFID to link IoT devices. It is carried out by imitating a legitimate IoT device and sniffing data from it. Before starting any form of assault on IoT devices, eavesdropping is a critical first step. An attacker can collect passwords, credit card numbers, emails, documents, browser history, login details, FTP login details, FTP documents, web addresses, and other personal information that users or devices would usually transfer over the network by launching this assault. This type of attack is used to acquire unauthorized access to information so that a De-authentication or man-in-the-middle attack can be launched. It collects all kinds of data, including encrypted data. To obtain information, a tool like Sniffer can be used to sniff packets. It is impossible to discover and exploit flaws in the wireless adapter of a system (i.e., a computer).

MITIGATION:

Unfortunately, because there are no disturbances or changes to the network, detecting and blocking passive network eavesdropping attacks is exceedingly difficult, if not impossible. Active attacks are easier to detect, but by the time network changes are recognized, data has typically already been intercepted.

Encryption: Encrypt email, networks, and conversations, as well as data at rest, in use, and in transit, first and foremost. Even if data is intercepted, without the encryption key, the hacker will be unable to decrypt it. Wi-Fi Protected Access 2 or WPA3 is recommended for wireless encryption. HTTPS should be used for any web-based communication. While most data can be encrypted, network traffic metadata like endpoints and IP addresses can still be collected using a sniffer.

Authentication: In order to prevent faked packets from being utilized in IP spoofing or MAC address spoofing attacks, it's critical to authenticate incoming packets. Authentication standards and procedures should be used. TLS, Secure/Multipurpose Internet Mail Extensions, OpenPGP, and IPsec are just a few of the cryptographic technologies that involve authentication.

Security Technologies: To prevent eavesdropping assaults, you'll need firewalls, VPNs, and antimalware software. Configure routers and firewalls to reject any packets with faked addresses using packet filtering.

4. VOICE CAPTURING ATTACKS:

The Alexa lacks any form of voice-based authentication, allowing any voice within a home environment to interact and command Alexa. As a result, Alexa can be triggered by any voice that contains wake words. This has resulted in the creation of remote assaults, which exploit the lack of authentication by broadcasting commands via devices such as televisions, radios, and speakers.

Voice instructions played by any speaker can erroneously prompt the Echo device to respond due to a lack of effective user voice verification. Injecting fake radio signals, substituting one TV channel with the given video stream, and fooling a wireless speaker into playing valid Echo instructions are all ways attackers can launch a remote voice assault.

Dolphin Attack: A dolphin assault is a type of remote attack in which Alexa is triggered by inaudible commands rather than audible ones. Voice commands are modulated on ultrasonic carriers to achieve this inaudibility. Although dolphin assaults require ultrasonic transducers to be within 2 meters of the Echo device, making them a less prevalent hazard than remote device attacks, there is still fear that these attacks will be able to extend their attack range in the future.

Man-in-the-Middle: A Man-in-the-Middle attack intercepts a user's conversation with their voice assistant without their knowledge. Command jamming is the initial part of this attack. They employ the Internet of Things gadget to record and inaudibly jam the commands the user gives the voice assistant. Both the malicious and the voice assistant are active when the user says the wake word. To prevent the voice assistant from understanding the user, the malicious gadget emits an ultrasonic modulated noise. Data retrieval is the next step. Because the malicious device knows what skill the user was attempting to use, it can send the same requests to the echo and determine what data the user was seeking. By echoing the information back to the user, the malicious device can now modify the data and complete the hijacking.

MITIGATION:

Teaching Alexa to distinguish between live and recorded speech is one way to protect against remote voice capturing attempts. Void, a light-weight speech liveness detection system, is an example of this. Multiple deep learning models are used to uncover changes in spectral power (analysis of cumulative power patterns in spectrograms) between live-human sounds and voices replayed through speakers in order to detect voice hacking attacks. The distribution of power into frequency components is referred to as spectral power. When most loudspeakers replay original sounds, they inevitably introduce distortions, causing the overall power distribution over the audible frequency range to show some uniformity and linearity. Speaker-Sonar, on the other hand, is a smart speaker-based sonar-based liveness detecting system.

Sonar is a technology for detecting objects that employs sound transmission. The key concept behind this system was to confirm that the voice command came from the user by tracking user movement via an inaudible stream of ultrasonic sound and comparing the received voice command's direction to the user's direction. This strategy, in particular, created a user experience that was not invasive. However, it only worked reliably in open outside locations, as the Speaker-Sonar system's accuracy was hampered by the highly ornamented interiors of consumer home surroundings.

On the other side, defending against dolphin attacks necessitates a distinct set of strategies. Hardware countermeasures for inaudible attacks go after the source of the problem. Unfortunately, most commercial microphones attached to smart devices such as phones or voice assistants are unable to detect acoustic sounds with frequencies higher than 20 kHz, which is the underlying cause of dolphin attacks and other inaudible voice orders. As a result, microphone tweaks that suppress any acoustic signals in the ultrasonic region would successfully thwart various types of inaudible attacks. In addition, by adding a module to microphones that detects modulated voice instructions within the ultrasonic frequency range, inaudible voice commands can be canceled. The signals would then be demodulated by this module to acquire the baseband.

5. VOICE SERVICE BASED ATTACKS:

These are attacks on the Alexa voice service's common Spoken Language Understanding (SLU) functions, including as Automatic Speech Recognition (ASR), Natural Language Understanding (NLU), and Text To Speech (TTS). It is nearly impossible for software to grasp everything a person says and correctly assess intent in this day and age. Homonyms and homophones are frequently misconstrued even by humans, and computers are currently unable to reliably recognize them in human language. Typically, language processing models aren't precise enough, and they haven't been trained with various languages or accents. This attack surface is designed to exploit these flaws in Alexa's SLU.

Skill Squatting: Skill Squatting is the construction of malevolent skills with invocation and intent names that sound a lot like real skills' invocation and intent names. Skill squatting depends on systematic errors made from word-to-word, like as pauses and mispronunciations, to take advantage of the simple misinterpretation of spoken words. The goal of this attack is to trick Alexa into using the malicious skill instead of the legal one, therefore hijacking the legitimate skill. By utilizing phrases that are only squattable in the demographics of targeted users, this strategy can be focused on specific categories of people.

MITIGATION:

Misinterpretations of Alexa voice services are one of the most commonly exploited attack surfaces. Voice squatting attacks, voice masquerading attacks, and skill squatting assaults are all popular attacks that target this attack surface. A skill-name scanner could be used as a defense against voice squatting and voice masquerade attacks. The scanner would translate a skill's invocation name string into an ARPABET-defined phonetic expression. The phonetic distance between distinct skill names can be calculated using this phonetic expression, and the skill names that the scanner detects as having a subset relation are considered possible voice squatting attacks. Another option is to evaluate the context information, such as the user's speech and the skill's reaction.

Before they can be published to the Alexa skill store for public use, all skills must go through a certification procedure. Attackers must successfully register maliciously squatted skills in order to carry out skill squatting attacks. Skill squatting attacks, in other words, rely on the vulnerabilities in the certification process. As a result, upgrading the certification process by adding extra screens could be a potential defense against skill squatting attacks. For example, using a word-based and phoneme-based analysis of a new skill's invocation name as a screening method to see if it may be confused with other already registered skills would be a good way to prevent skill squatting assaults.

6. VOICE SKILL/APP BASED ATTACKS:

Malicious Skills: Malicious skills are defined as any skill that is designed to act against the user's best interests. This could be a talent that mines user data or steals personal information, or simply a skill that purports to be able to do something it can't. Several tests have been published recently that demonstrate how quickly harmful skills may circumvent Amazon's security and become available for public usage on the official Amazon Alexa skill store. Researchers were able to publish hundreds of policy-breaking talents on the Alexa skill store for customers to access in one study. Many factors contributed to the vulnerability of this attack surface in particular.

After a user individually verifies that their skill complies with Amazon policy, the talent is submitted for review and screening by an Amazon official reviewer under the current Amazon skill publishing method. If the authority determines that the skill adheres to consumer policy, it is made available to the general public. While each skill is given the time and effort of an official evaluation, the present Amazon skill publishing procedure is heavily reliant on screening and places little focus on objective automation. This contradiction arises from the subjective screening procedure. In other words, skill reviewers verify skills based on their own interpretation of Amazon policy, which means that the likelihood of a skill being published is determined on the skill reviewer assigned to its verification rather than the content and intentions of the skill itself.

By delaying the response just long enough to clear the process, attackers were able to post skills with malicious responses in some circumstances. Furthermore, even if developers claimed (via a "developer's form" filled with yes or no questions) that their skill does not collect user information and data, they were allowed to publish it on the Amazon skill store, even if their claim was false, because skill reviewers did not always verify that claim during the official review. This means that, in addition to individual talent reviewers, Amazon relies on the honesty of individual developers. There was also evidence that many assessments were carried out by non-native English speakers who were unfamiliar with local regulations.

Masquerading Attacks: Voice disguising is also a threat to Alexa's voice skill. This attack occurs when a malicious skill is created that imitates the behavior of a valid skill or even the VPA service. These abilities might persuade the user that they are using safe and secure features when, in fact, all of the information they have provided to their voice assistant has been hacked.

MITIGATION:

Amazon and other voice assistant providers have certification processes that do not thoroughly evaluate the skills submitted to their shops. Because developers have the ability to change the functionality of skills after they are certified, skill behavior integrity must be enforced throughout the skill life cycle. When a developer wishes to alter the front-end or back-end, a continual certification/vetting process should be required. Although this may raise the certification process' delay, it will improve its quality and boost the system's dependability.

Another observation of the skill certification process is the potential for human mistake in a screening procedure that relies on human decision-making. An obvious solution would be to use automated skill testing to increase the consistency of the verifications and aid in more thorough testing. To strengthen the certification process even more, voice assistant system providers need access to the skill's back-end code in order to undertake code analyses.

CONCLUSION:

Smart home networks are IOT based systems, to provide ease of access and occasionally comprise home security services. We performed a network security audit on an Amazon echo centered smart home network. First, we compiled the risk impacts, acceptance, and recovery measures; then we classified the vulnerabilities based on impact and liability level to prioritize them; then we identified potential attacks and provided their corresponding preventive measures to mitigate them. And finally, we performed penetration testing to gain a deeper insight of ARP spoofing attack methods and formulated an innovative solution to counter them, fit for a smart home network.