# Silent Mode: Disabling All Devices For Privacy

Sayam Dhingra

## Abstract

In today's world, the "Internet of Things" (IoT) has become extremely widespread and essential. It has become an integral part of our daily lives, making it nearly impossible to imagine a life without it. However, along with its benefits, there are also drawbacks. One significant concern is the misuse of IoT devices to invade people's privacy. Disturbingly, there have been cases where AirBnB homeowners installed hidden cameras and microphones in their properties to spy on their tenants. In this study, we aim to explore effective methods of disabling these invasive devices and protecting your privacy.


## Introduction

The advent of the "Internet of Things" (IoT) has revolutionized our modern era, permeating every aspect of our lives. Its pervasive presence has become the new norm, rendering life without it virtually unimaginable. While IoT technology has undeniably become an indispensable part of our lifestyles, it is not without its drawbacks. One particularly concerning issue is the potential misuse of IoT devices, which can infringe upon individuals' privacy and security.

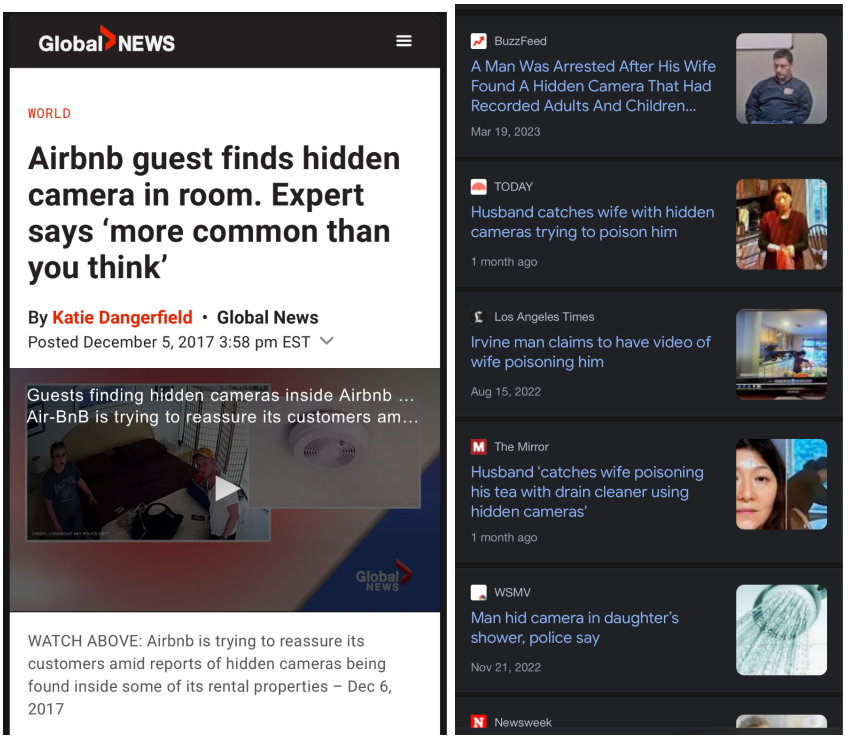
One alarming instance of this misuse involves the illicit surveillance carried out by some AirBnB homeowners. Reports have surfaced regarding hidden cameras and microphones being surreptitiously placed within rented accommodations, enabling unscrupulous hosts to spy on unsuspecting tenants. These revelations have heightened concerns about the invasion of privacy and have prompted a need for solutions to combat such breaches. Another instance is where IoT has been reported in cases of interpersonal abuse [1]. Where the victim was spied upon, tracked and also harassed using IoT devices.

In response to this pressing issue, we have undertaken a comprehensive study to explore two potential strategies for disabling these invasive IoT devices and safeguarding personal privacy. Our objective is to devise effective methods that empower individuals to protect themselves from privacy breaches while utilizing IoT technology to its fullest potential.

In this work we focus on two major contributions

- Using esp32 board to deauthenticate all devices on the selected access point (AP)
- Using a raspberry pi zero w to attempt ARP Spoof on all devices to ensure network traffic passes through our device.

Later on in this blog we continue to discuss “Related Work” in the next section. Followed by the method we used and a detailed explanation of the techniques and code explanation. Then in the next section we will discuss the results discovered and the challenges faced. Finally we will end this blog with the conclusion and future work possibilities.




## Related Work

In this section we discuss the previous works done by other people based on this topic and how they helped us achieve the goals we set in our study. In a recent paper published by Stephenson, S et al. [1], they discuss and frame abuse vectors to help understand IoT enabled interpersonal abuse. This research addresses the issue of tech-enabled interpersonal abuse (IPA) where abusers exploit smart devices to surveil and harass their victims. The study conducts a large-scale analysis of smart devices used in IPA by crawling Google Search results for web pages discussing their misuse.

In another study by Apthorpe N et al. [5] investigate the impact of internet-connected consumer devices on interpersonal relationships within multi-occupant households. The study identifies several themes that shed light on the widespread interpersonal costs and benefits associated with these devices. The findings provide valuable exploratory data that can guide future research and inform the design of IoT technologies, with the goal of enhancing positive impacts and minimizing negative effects on interpersonal dynamics.

Similarly, anonymous authors published a study [6] exploring the privacy needs and negotiation intentions of bystanders in smart homes, specifically focusing on the conflicting privacy concerns between users and bystanders. The researchers conducted a vignette study involving 867 participants in the context of Airbnb, varying factors such as device types, device location, and duration of stay. The study examined participants' preferences regarding the negotiation of privacy, including whom to negotiate with, when to negotiate, how to negotiate, and why.

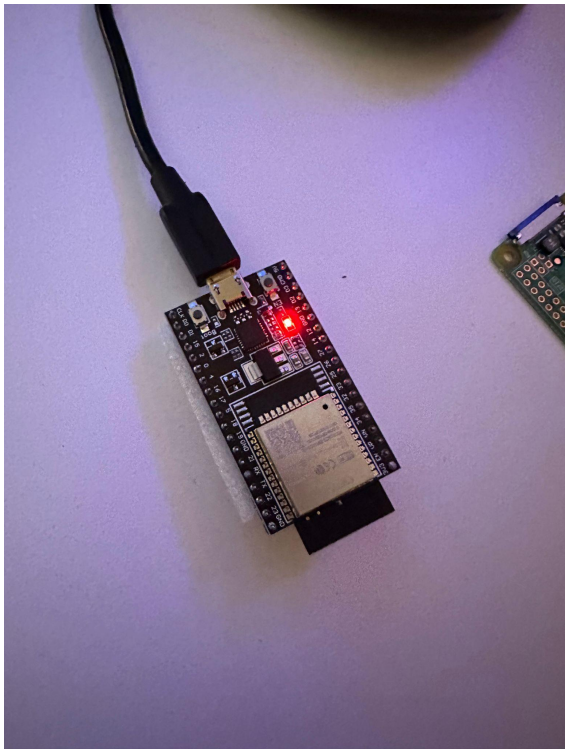
These papers have similar motivations which coincide with our own for this paper. Their research and results helped us understand the necessity for security and the requirement for devices that we created.

## Methodology

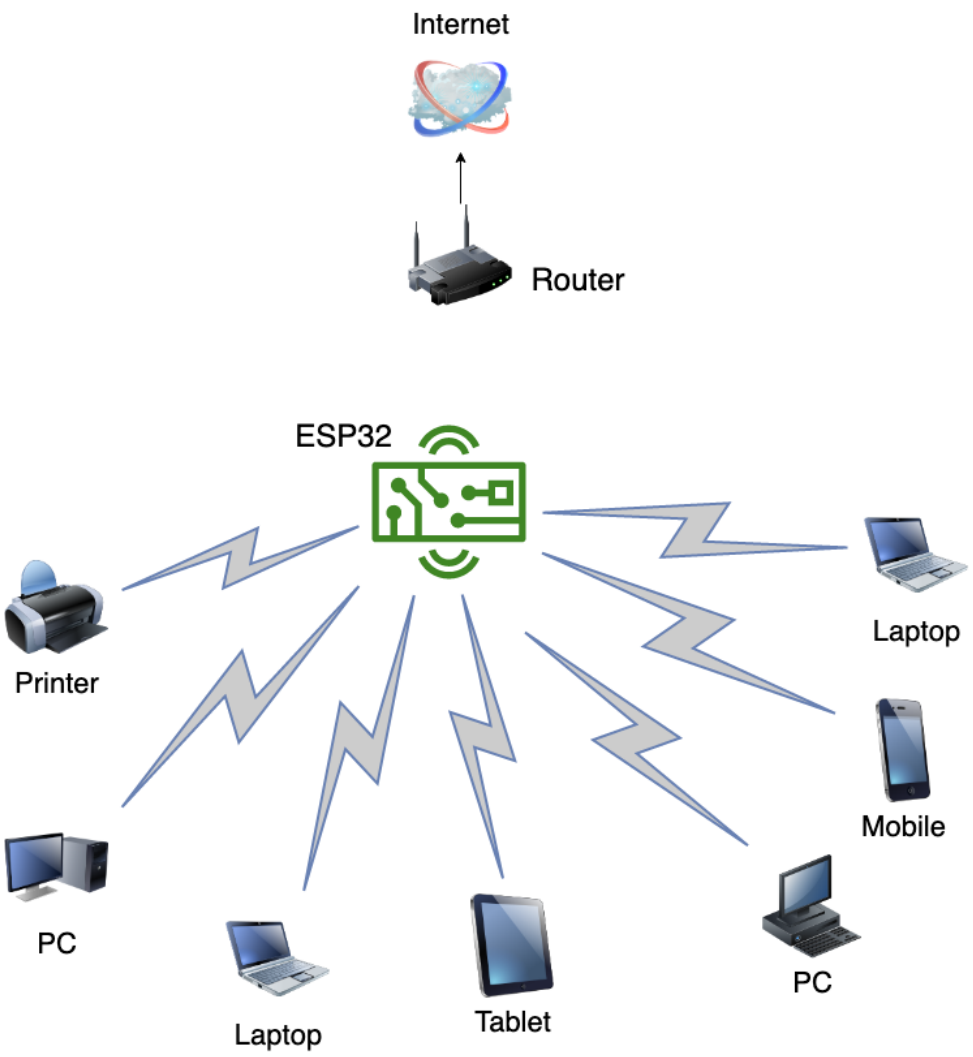
This section will be divided into two parts where in each part we discuss the two components. The first being the esp32 device used to attempt deauthentication and the second part focusing on the second device, raspberry pi zero w.

**>> Esp32**

The ESP32 is a powerful, low-cost, and versatile microcontroller chip that belongs to the Espressif Systems' ESP32 series. It is widely used in various IoT (Internet of Things) applications due to its robust features and capabilities.



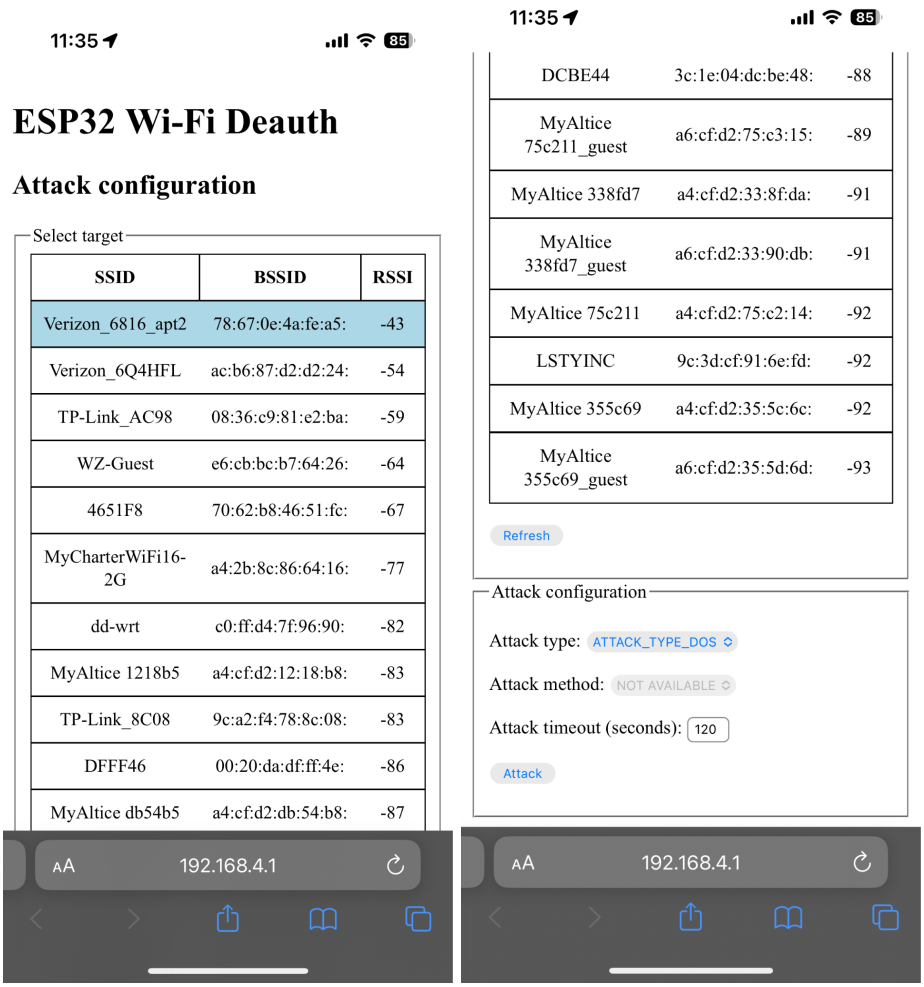
One key feature of the esp32 board is its ability to send raw 802.11 data packets. This allows us to engineer a unique type of 802.11 packet called the deauthentication packet. The purpose of this packet is to inform devices connected to an access point that their connection is no longer valid and has expired. This prompts the devices to disconnect and attempt reconnection. But since we are constantly bombarding all devices connected to the AP, the devices are never able to communicate with the AP. Thereby denying service to the internet and thus the malicious host of the device.




The way we achieved this was a bit tricky. Though the standard esp32 allows us to send any raw 802.11 packet, it does check for malicious packets such as the deauthentication packet and does not allow us to send it. However, using an exploit discovered by user GANESH ICMC/USP on github [7] we were able to bypass the check function by declaring a duplicate of our own and forcing the compiler to use our definition over the standard one. This allowed us to bombard devices with deauthentication packets.

When the esp32 device is powered on, it broadcasts an access point of its own. This allows the user to connect to the device. The access point requires a password to connect to it, thereby ensuring integrity of the device. The device also hosts a web server which the user connect to with their browser on their device which then shows them the list of available AP's. Here the user has to choose their AP and press the attack button. This will initiate the attack for the decided period of time.

This tool allows the users to disable Wi-Fi based IoT devices. However a drawback of this is that the user cannot connect to the AP either. A future prospect of this is to include exceptions based on the MAC address of devices connected to the AP. So, the user's devices are left unharmed.


11:35
85
ESP32 Wi-Fi Deauth
Attack configuration
Select target
SSID
BSSID
RSSI
Verizon_6816_apt2
78:67:0e:4a:fe:a5:
-43
Verizon_6Q4HFL
ac:b6:87:d2:d2:24:
-54
TP-Link_AC98
08:36:c9:81:e2:ba:
-59
WZ-Guest
e6:cb:bc:b7:64:26:
-64
4651F8
70:62:b8:46:51:fc:
-67
MyCharterWiFi16-2G
a4:2b:8c:86:64:16:
-77
dd-wrt
c0:ff:d4:7f:96:90:
-82
MyAltice 1218b5
a4:cf:d2:12:18:b8:
-83
TP-Link_8C08
9c:a2:f4:78:8c:08:
-83
DFFF46
00:20:da:df:ff:4e:
-86
MyAltice db54b5
a4:cf:d2:db:54:b8:
-87
AA
192.168.4.1
11:35
85
DCBE44
3c:1e:04:dc:be:48:
-88
MyAltice 75c211_guest
a6:cf:d2:75:c3:15:
-89
MyAltice 338fd7
a4:cf:d2:33:8f:da:
-91
MyAltice 338fd7_guest
a6:cf:d2:33:90:db:
-91
MyAltice 75c211
a4:cf:d2:75:c2:14:
-92
LSTYINC
9c:3d:cf:91:6e:fd:
-92
MyAltice 355c69
a4:cf:d2:35:5c:6c:
-92
MyAltice 355c69_guest
a6:cf:d2:35:5d:6d:
-93
Refresh
Attack configuration
Attack type:
ATTACK_TYPE_DOS
Attack method:
NOT AVAILABLE
Attack timeout (seconds):
120
Attack
AA
192.168.4.1


**>> Raspberry Pi Zero W**

The Raspberry Pi Zero W is a compact and affordable single-board computer developed by the Raspberry Pi Foundation. It is a variant of the Raspberry Pi Zero, with the added feature of built-in wireless connectivity.

The Raspberry Pi Zero W supports the same operating systems as other Raspberry Pi models, including various Linux distributions (Raspbian, Ubuntu, etc.), and it can run a wide range of applications and software designed for the Raspberry Pi ecosystem. This makes it a very powerful board with capabilities far beyond the esp32. This allows us to send ARP packets which are not possible to send on the former. ARP.

ARP (Address Resolution Protocol) packets are network protocol packets used to map an IP address to a physical (MAC) address on a local network. When a device wants to send data to another device on the same network, it needs to know the MAC address of the destination device. The ARP protocol helps in this process by sending out an ARP request packet, asking "Who has this IP address?" The request is broadcasted to all devices on the network, and the device with the matching IP address responds with an ARP reply packet, providing its MAC address.

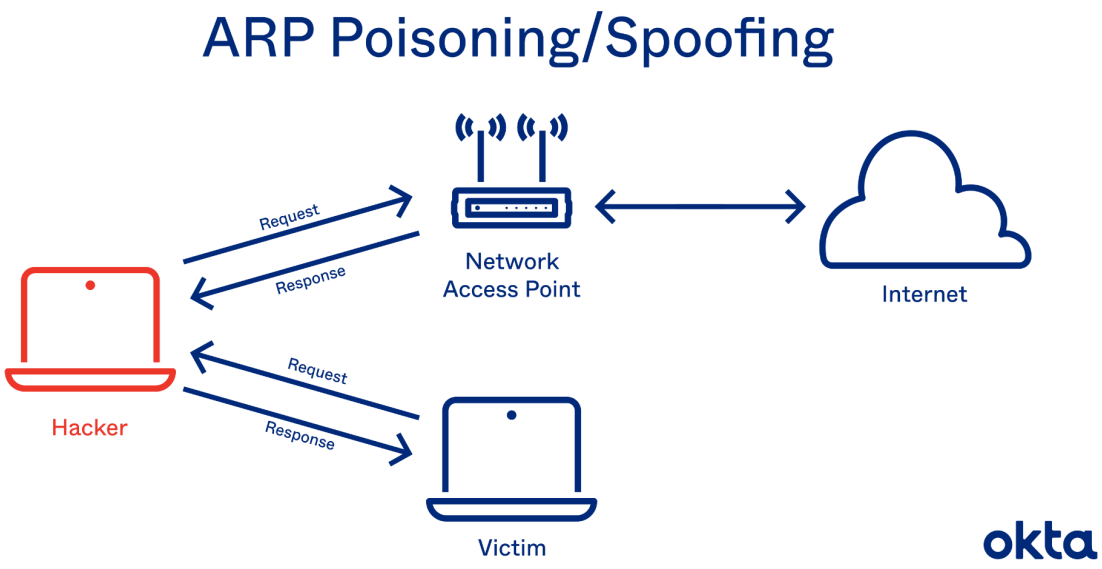
We run a python script on this device so that we can transfer network traffic through this device. This allows us to control the flow of traffic and thereby disabling all network traffic by dropping packets for devices that we target, or even all devices. We achieve this by writing a python script to perform the ARP spoof using the scapy library. Another capability which was also in the esp32 was to monitor traffic and storing network packets in a pcap file for later analysis.

```
Interface: 192.168.1.155 --- 0x16
  Internet Address      Physical Address      Type
  192.168.1.1           78-67-0e-4a-fe-a4     dynamic
  192.168.1.212         c8-89-f3-b4-dc-9b     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
  224.0.0.252           01-00-5e-00-00-fc     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\Cheesy Fries>arp -a

Interface: 192.168.1.155 --- 0x16
  Internet Address      Physical Address      Type
  192.168.1.1           c8-89-f3-b4-dc-9b     dynamic
  192.168.1.212         c8-89-f3-b4-dc-9b     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
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

A common disadvantage of both of these devices is that they are two separate devices. The esp32 is unable to send ARP packets and the raspberry pi is unable to send raw 802.11 deauth packets. A future prospect would be to possibly work on a device that has both capabilities. Therefore instead of two devices we can work with just one device.




## Results, Conclusions and Future Prospects

In conclusion, my journey to address privacy and security concerns in the IoT domain has been an enlightening and empowering experience. Through extensive research and dedication, I successfully developed two devices that disable intrusive IoT devices on a wireless network.

For concrete results we would need to make these devices available to the public beyond making the source code publicly available with proper steps to set this up. We would then also have to survey them to ensure that these devices have helped them secure their privacy and alleviate their concerns.

Looking ahead, I am excited to explore further advancements in this field. One future prospect that holds immense potential is developing a single device capable of both disabling IoT devices on wireless networks and effectively working with devices on both Ethernet and Wi-Fi connections. This integration would offer users a comprehensive solution, allowing them to protect their privacy across various network configurations. Also, I believe that a single device capable of doing both the arp spoof and wifi deauth would result in a more convenient and useful user experience.

Additionally, I envision a wireless device that goes beyond merely disabling intrusive IoT devices. By leveraging proximity detection technology, similar to an airtag, this device would be able to identify and alert users to the presence of IoT devices in their surroundings. This innovation would not only enhance security but also raise awareness about potential privacy breaches, empowering individuals to take proactive measures to safeguard their personal information.

As technology continues to evolve and the IoT landscape expands, it is crucial that we remain proactive in addressing privacy and security concerns. Through continued research, development, and collaboration, we can pave the way for a future where individuals have full control over their IoT devices, ensuring a safe and secure digital environment for all. Together, let us strive for a world where privacy and security are paramount in the IoT domain.

## References


1. Stephenson, S., Almansoori, M., Emami-Naeini, P., Huang, D.Y. and Chatterjee, R., Abuse Vectors: A Framework for Conceptualizing IoT-Enabled Interpersonal Abuse.
2. Guests Across America Find Cameras in Their Airbnb Rentals, https://www.newspressnow.com/guests-across-america-find-cameras-in-their-airbnb-rentals/article_822ee0ac-f433-11ed-a378-6f1df8052a1e.html
3. Horror As Women Discover Hidden Camera in Airbnb Bathroom: 'Freaking out' https://www.newsweek.com/horror-women-discover-hidden-camera-airbnb-bathroom-canada-1796840
4. A group of friends say they found hidden cameras in the bathrooms of their Airbnb rental while on holiday https://www.businessinsider.com/airbnb-hidden-cameras-bathroom-viral-tiktok-video-2023-5
5. Noah Apthorpe, Pardis Emami-Naeini, Arunesh Mathur, Marshini Chetty, and Nick Feamster. 2022. You, Me, and IoT: How Internet-connected Consumer Devices Affect Interpersonal Relationships. ACM Trans. Internet Things 3, 4, Article 25 (November 2022), 29 pages. https://doi.org/10.1145/3539737

6. Exploring Tenants' Preferences of Privacy Negotiation in Airbnb
7. https://github.com/GANESH-ICMC