Evil Twin Attack - isunet

By: Alec Gordon, Ethan Allgeier, Cody Dunn

1. Introduction and Purpose

1.1 Overview

This report documents the implementation and analysis of an evil twin attack targeting the ISUnet wireless network. An evil twin attack is a type of wireless network exploitation where an attacker creates a fraudulent access point that mimics a legitimate network, deceiving users into connecting and potentially disclosing sensitive credentials.

The primary objectives of this assessment were as follows:

1. Demonstrate vulnerabilities in wireless network authentication
    a. Credential capturing
2. Evaluate the effectiveness of social engineering techniques in credential harvesting
    a. Fake Central Login page prompting for forgotten username & password
3. Assess user susceptibility to visually convincing phishing portals
    a. Central Login isunet page
4. Document technical challenges and limitations in evil twin deployment

This assessment was conducted in a controlled environment for educational purposes to understand network security weaknesses and inform defensive strategies. The implementation of this tool was not conducted while on the ISU campus, in order to avoid accidental connections and the tool from being blocked on the network.

2. Technical Implementation

2.1 Hardware and Software Components

Hardware:

- ESP32 microcontroller with integrated Wi-Fi capabilities
    - Sufficient processing power for simultaneous AP hosting and web server operation

Software:

- ESP32 Marauder firmware
    - Penetration testing platform for ESP32 devices
- Evil Portal module
    - Captive portal framework for credential harvesting
- GitHub Evil Portal workflow
    - Portal customization and deployment

2.2 Attack Architecture

The evil twin attack was implemented using a three-layer architecture:

- Access Point Layer
    - ESP32 broadcasts an SSID identical to isunet
        - Appears as a legitimate network option for nearby devices
- Captive Portal Layer
    - When connected, users are automatically redirected to a fake authentication page
        - Central Login page
- Data Collection Layer
    - Submitted credentials were captured and displayed in the terminal
        - The code can be modified and improved to be able to extract the credentials to the ESP32 hardware

2.3 Portal Development Process

We began by analyzing the HTML code of the legitimate Central Login page in order to mimic the following:

- Visual design elements
    - Logos, color schemes, and overall layout
- Form field structures

    - Username and password inputs
- URL patterns and branding elements

Next, we created a replica login page to mimic the legitimate isunet Central Login portal. To do so, we incorporated the following:

- HTML structure matching the original layout
- CSS styling for visual consistency

Finally, we deployed the assets to the ESP32 platform for demonstration.

2.4 Technical Challenges Encountered

The first challenge we encountered was getting the fake login page to display images during demonstration. This occurred for the reasons listed below:

- Image URLs referenced external microsoft resources
    - Those resources required internet connectivity

Our linked resources, specifically the ISU Central Login 'lock' logo with the Redbird and the Sign-in options 'key' logo returned 404 errors or timed out when viewing the fake login page

In an attempt to resolve this issue, we tried to convert the files to Base64, however the files were too large in size. This will be explained in the next section. Instead, we had to hard-code new image files that can be displayed in the header and footer.

Like mentioned above, our next problem was the Base64 encoding resulting in significantly increased file sizes, that the ESP32 ultimately could not handle.

2.5 Deployment Configuration

The final configuration of our ESP32 portal included the following:

- SSID
    - isunet
- Captive portal trigger
    - Automatic redirection upon connection
- Data logging
    - Credentials output directly to the terminal
        - Further improvements could be extracting the credentials to the ESP32 itself

3. Justification and Analysis

3.1 Attack Vector Effectiveness

Evil twin attacks succeed for many reasons. The first one is brand recognition. Users tend to blindly trust a familiar network name without going through the process of verifying it to ensure legitimacy. Next, is visual legitimacy. When creating an Evil Portal, it is crucial that the appearance matches that of the page you are trying to mimic. This will reduce any suspicion a user may have. Following that is convenience. When using the tool as an open network at a local coffee shop for example, users tend to select the first open passwordless network they can find. Finally, your average user is not technically sound enough to analyze security flaws in networks and rogue access points.

3.2 Technical Vulnerabilities Exploited

The first vulnerability is the limitation of wireless protocols. 802.11 wireless standards lack the mutual authentication that is needed between clients and access points. This makes it so that SSID broadcasting is unauthenticated, allowing arbitrary network name claims. Ultimately, this leaves users unable to verify access point legitimacy without having additional security measures.

Alongside those limitations, we also look at the mechanics behind captive portals. Captive portals work because operating systems automatically detect and display them. For us, users become accustomed to this feature and are essentially trained to expect and comply with login prompts. This allows us to exploit behavioral patterns of users.

3.3 Security Implications

A successful compromise of users through the ESP32 tool can result in the following outcomes.The first is direct harvesting of their usernames and passwords. Along with credential harvesting, you could also potentially see credential reuse across multiple services. For example, a user’s bank login information is the same as their FaceBook login. Lastly, you would potentially have unauthorized access to a large amount of typically secure and personal data.

With captured credentials, attackers could also complete the following attacks. With the credentials used on the fake captive portal, an attacker can authenticate to the legitimate network instead. This would provide them with access to the university’s resources and services. From there, an attacker could also look to pivot to additional internal network targets.

When a user is connected to the Evil Twin, they are exposed to severe privacy violations. This includes network traffic metadata as well as device identification information such as MAC addresses and hostnames.

3.4 Defensive Limitations Exposed

This assessment revealed several defensive gaps. This includes a lack of certificate validation, as your average user rarely verifies SSL/TLS certificates on captive portals. Next is insufficient user training. ISU does not regularly push out security awareness programs, which ultimately leads to users not being able to properly address threats when they are exposed to them. Finally is SSID confusion. This means that there is no visual or technical differentiation between the legitimate isunet, and our Evil Twin.

3.5 Lessons from Implementation Challenges

The technical obstacles encountered provided us with several valuable insights. First would be resource constraints. Attackers face practical limitations such as storage space, processing speed, and power. Next is that highly sophisticated portals may be out of scope when using cheap hardware. In our case, the captive portal was not very complicated, however we did run into the storage space issue when trying to use images.

4. Conclusion

4.1 Findings

Our project successfully demonstrated the viability and effectiveness of evil twin attacks against wireless networks. The ESP32 Marauder platform proved capable of creating convincing rogue access points. Some challenges faced however, were resource limitations (storage space).

Our key findings include that his project had a low technical barrier for entry. Consumer-grade hardware and open-source software enabled us to create a sophisticated wireless attack. This project also has a high probability of success, as average users remain vulnerable to well-executed social engineering via fake portals. We also came across some complexity when trying to implement the image files. Storage constraints make it so that our project required optimization and overall a compromise in our captive portal.

4.2 Recommendations

Implement technical controls such as deploying wireless intrusion detection/prevention systems, implementing 802.1X authentication with certificate validation, using mutual authentication protocols such as EAP-TLS, and monitoring for SSID spoofing and rogue access point signatures, ultimately removing them before users can connect.

It's also important that users are educated. Users should be able to verify network authenticity through official channels (ISU Technology Support Center). Users should be encouraged to make use of ISU's VPN when off campus. Promoting awareness of common phishing indicators should also be a priority for ISU's network security team. Lastly, implement simulated attack exercises could be used to build recognition skills among users.

4.3 Future Improvements

Further improvements to this project could include machine learning application for rogue AP detection. User behavior analysis during captive portal authentication and credential capturing to the ESP32 hardware.

4.4 Ethics

This research was conducted strictly for educational purposes to make recommendations on the improvement of security posture. The techniques described pose serious privacy and security risks when misused. Responsible disclosure and defensive application of this knowledge remain vital to protecting organizational and individual security.