Binghamton University, Watson School of Engineering

Stage 4:
Deception against Evil Twin Attack on a Wireless Network Using Two Wi-Fi USB Adapters with Atheros AR9271 Chipset

Bhaskara Yashwant Bitra
Science of Cyber Security - CS 559
Prof. Guanhua Yan
Dec 2nd, 2024

Contents

1. Setting up the Environment for Deception

1.1 Recap of the Evil Twin Attack, Prevention, and Detection

In the initial stages of this project, we delved into the vulnerabilities of wireless networks, focusing specifically on the Evil Twin attack:

- **Stage 1: Execution of the Evil Twin Attack**
  - **Attack Overview:** We successfully executed an Evil Twin attack by creating a rogue Wi-Fi Access Point (AP) named "FreeWiFi" using two USB Wi-Fi adapters with the Atheros AR9271 chipset on a Kali Linux machine.
  - **Techniques Used:** Utilizing tools such as airmon-ng, aireplay-ng, and airbase-ng, we performed de-authentication attacks to disconnect legitimate clients from their trusted network and lured them into connecting to our rogue AP.
  - **Data Capture:** A captive portal hosted via a Python script captured sensitive user credentials, including usernames, passwords, and personal information.
- **Stage 2: Implementation of Prevention Techniques**
  - **Theoretical Measures:** We explored hardening wireless configurations, employing strong authentication and encryption protocols like WPA3 and 802.1X authentication, and educating users about the dangers of connecting to unknown networks.
  - **Practical Solution:** Developed a Python script prevent_evil_twin.py that monitored the connected SSID on the Windows victim machine and automatically disconnected from any untrusted networks, effectively preventing connections to rogue APs.
- **Stage 3: Development of Detection Mechanisms**
  - **Theoretical Approaches:** Discussed techniques such as signal strength analysis, SSID and BSSID verification, and the use of Wireless Intrusion Detection Systems (WIDS).
  - **Practical Implementations:** Employed Wireshark on a Kali Linux machine to detect de-authentication frames indicative of an Evil Twin attack. Additionally, implemented a Python script on the Windows victim machine to monitor for duplicate SSIDs with different BSSIDs, alerting the user when a potential rogue AP was detected.

These stages highlighted the evolving nature of wireless security threats and the importance of a multifaceted approach encompassing attack execution, prevention, and detection strategies.

1.2 Deception Objective

Building upon the insights gained from the previous stages, the objective of Stage 4 is to implement deception techniques to gather information about potential attackers proactively. Instead of solely focusing on defending against attacks, we aim to:

- **Engage Attackers in a Controlled Environment**: By deploying deceptive elements such as honeypot access points and fake vulnerabilities, we can lure attackers into interacting with our controlled setup.
- **Collect Intelligence on Attacker Methods**: Monitoring attacker behaviors allows us to gather valuable data on their tools, techniques, and procedures (TTPs).
- **Enhance Threat Awareness**: Understanding attacker intentions and strategies contributes to a deeper knowledge of the threat landscape, enabling the development of more robust defense mechanisms.
- **Strengthen Security Posture**: The insights gained can be used to improve existing prevention and detection measures, making our wireless networks more resilient against sophisticated attacks.

This approach not only helps fortify the security of wireless networks but also transforms the network into a proactive defense mechanism that anticipates and analyzes attacker strategies.

1.3 Host Machine Setup (Attacker and Victim)

Attacker's Machine (Kali Linux):

- **Operating System**: Kali Linux
- **Hardware:** Two USB Wi-Fi adapters with the Atheros AR9271 chipset
  - wlan1: Configured in monitor mode for packet sniffing and potential de-authentication attacks.
  - wlan2: Set up in master mode to create the honeypot access point.
- **Tools and Utilities**:
  - airmon-ng, aireplay-ng, airbase-ng for wireless operations.
  - dnsmasq for DHCP and DNS services.
  - Python for hosting the fake captive portal and logging scripts.
  - tcpdump and Wireshark for network traffic analysis.

Victim (Decoy) Machine (Windows 11):

- **Operating System**: Windows 11
- **Hardware**: Standard Wi-Fi network interface card.
- **Security Tools:**
  - Custom Python scripts for monitoring and logging.
  - Network monitoring tools (if compatible).
- **Network Configuration**:
  - Configured to connect to the honeypot access point for monitoring purposes.
  - Equipped with previous prevention and detection scripts for integration with deception strategies.

Network Setup:

- **Proximity**: All devices are within the same wireless network range to ensure seamless interaction and data capture.
- **Honeypot Deployment:** The Kali Linux or Windows machine will simulate a vulnerable network to attract attackers while monitoring and logging all interactions.
- **Decoy Role:** The Windows machine serves as a passive decoy, representing potential targets within the network without exposing real assets.

By carefully configuring the environment, we aim to create a realistic scenario where attackers are drawn to our deceptive setup, allowing us to observe and analyze their behavior.

2. Theoretical Deception Techniques

2.1 Deploying Honeypot Access Points to Attract Attackers

Honeypot access points are decoy wireless networks intentionally set up to entice attackers into interacting with them. By creating a network that appears vulnerable or valuable, security professionals can monitor and analyze attacker behaviors in a controlled environment. This strategy allows for the collection of data on attack methods and tools without risking actual network assets. Deploying honeypot access points helps organizations understand potential threats better and enhance their overall security posture by learning from the attackers' techniques.

2.2 Creating Decoy SSIDs Mimicking Vulnerable Networks

By broadcasting decoy SSIDs that mimic common or vulnerable network names, such as "Free_Public_WiFi" or "Guest_Network," defenders can lure attackers searching for easy targets. These decoy networks appear attractive to malicious actors who may attempt to exploit them. Monitoring interactions with these SSIDs enables security teams to gather intelligence on attacker behaviors and identify potential threats before they impact legitimate networks.

2.3 Implementing Fake Captive Portals with Logging Capabilities

Fake captive portals are simulated login pages presented to users when they connect to a network. Implementing such portals with logging capabilities allows defenders to capture input from attackers attempting to exploit the network. This can include credentials, commands, or other data entered by the attacker. Analyzing this information provides valuable insights into the attacker's methods and objectives, which can be used to strengthen security measures and prevent future attacks.

2.4 Beacon Frame Manipulation to Lure Attackers

Beacon frames are packets broadcasted by access points to announce the presence of a wireless network. Manipulating these frames to include enticing information such as indicating weak security protocols or suggesting the availability of valuable resources can attract attackers. By altering beacon frames, defenders create the illusion of vulnerable networks, encouraging attackers to engage. Monitoring these engagements helps in identifying attack patterns and understanding the tools and techniques used by malicious actors.

2.5 Using Honeytokens and Decoy Data to Gather Intelligence

Honeytokens are pieces of decoy data, like fake credentials or files, planted within a network to detect unauthorized access. When an attacker interacts with these honeytokens, it triggers alerts and logging mechanisms. This strategy allows defenders to monitor the attacker's activities and gather information such as IP addresses, tools used, and actions taken. Using Honeytokens helps in the early detection of breaches and provides actionable intelligence to improve security defenses.

2.6 Monitoring De-authentication Frames for Attack Patterns

Attackers often use de-authentication frames to disconnect legitimate users from a network, facilitating attacks like the Evil Twin. By continuously monitoring for unusual patterns or frequencies of de-authentication frames, defenders can detect potential attacks in progress. Analyzing these patterns helps in identifying the source of the attack and enables a timely response to mitigate its impact. This proactive monitoring is crucial for maintaining network integrity and user connectivity.

2.7 Legal and Ethical Considerations in Deception Strategies

While deception techniques are valuable for security, they raise important legal and ethical considerations. Organizations must ensure that their use of honeypots, fake portals, and data collection complies with laws and regulations concerning privacy and unauthorized access. Ethical considerations include respecting user privacy, avoiding entrapment, and ensuring that deception does not cause harm to legitimate users. Careful planning and legal consultation are essential to implement deception strategies responsibly.

2.8 Integrating Deception with Existing Prevention and Detection Measures

Deception techniques are most effective when integrated with existing security frameworks. Combining deception with traditional prevention and detection measures creates a layered defense strategy. Insights gained from deception can inform and enhance intrusion detection systems, firewall rules, and user education programs. This integration ensures that deception contributes to the overall security posture, enabling organizations to respond more effectively to threats and reduce the risk of successful attacks.

Special Note - In section 3, I will try to implement some of these methods using code, shell scripting, software, or a combination of all or some of them, or I will implement other practical methods for Deception not mentioned in section 2.

3. Evil Twin Attack Deception Implementation (Practical)

3.1 Overview of Practical Deception Methods

Deception in the context of an Evil Twin attack focuses on gathering information about the attacker while misleading them. The goal is to extract details such as the attacker's IP address, MAC address, and network setup by interacting with their rogue access point (AP). This involves:

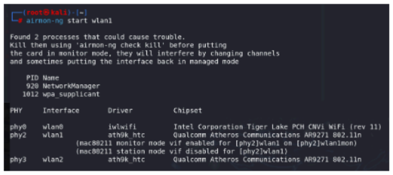
- Connecting to the rogue AP (e.g., "FreeWiFi").
- Submitting decoy credentials to waste the attacker's resources.
- Monitoring network traffic for signs of data exfiltration or system configuration.

Implementing such deception techniques requires certain tools and configurations on the victim machine.

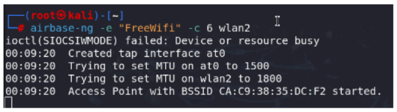
3.2 Recap of Evil Twin Attack

To effectively implement and test the Deception mechanisms against the Evil Twin attack, it is necessary to recreate the attack scenario from Stage 1. By launching the attack, we can observe how the detection methods respond in a realistic environment where the victim machine might fall into the trap. The attack involves several key steps: enabling monitor mode on wlan1 to intercept packets and perform the de-authentication attack on the wireless network “Ron”, verifying the monitor mode status, scanning for available networks to identify the target, performing a de-authentication attack to disconnect legitimate clients from their current network (Ron). I created a rogue access point named "FreeWiFi" using wlan2; and assigned an IP address to the rogue AP enabling IP forwarding to route traffic between the rogue AP and the internet. I dealt with configuring iptables and NAT for proper traffic forwarding, setting up DHCP and DNS services using dnsmasq, creating and hosting a captive portal that mimics a login page to capture sensitive information; and finally, capturing the credentials submitted by the victim. Reiterating these steps is essential because, to proceed with decepting an attacker performing the Evil Twin attack, the attack must be actively running. This allows us to test the Deception tools and scripts in a real-world scenario, ensuring that the Deception mechanisms are effective when the victim machine is exposed to the attack.

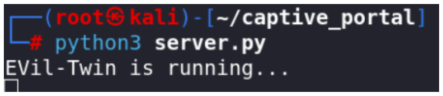
To prevent the attack from the victim's perspective, repeat the attack from the attacker's machine.



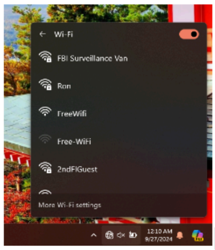
```
┌──(root㉿kali)-[~]
└─# airmon-ng start wlan1

Found 2 processes that could cause trouble.
Kill them using 'airmon-ng check kill' before putting
the card in monitor mode, they will interfere by changing channels
and sometimes putting the interface back in managed mode

    PID Name
    920 NetworkManager
   1012 wpa_supplicant

PHY     Interface       Driver          Chipset

phy0    wlan0           iwlwifi         Intel Corporation Tiger Lake PCH CNVi WiFi (rev 11)
phy2    wlan1           ath9k_htc       Qualcomm Atheros Communications AR9271 802.11n
                (mac80211 monitor mode vif enabled for [phy2]wlan1 on [phy2]wlan1mon)
                (mac80211 station mode vif disabled for [phy2]wlan1)
phy3    wlan2           ath9k_htc       Qualcomm Atheros Communications AR9271 802.11n
```



```
┌──(root㉿kali)-[~]
└─# airbase-ng -e "FreeWifi" -c 6 wlan2
ioctl(SIOCSIWMODE) failed: Device or resource busy
00:09:20  Created tap interface at0
00:09:20  Trying to set MTU on at0 to 1500
00:09:20  Trying to set MTU on wlan2 to 1800
00:09:20  Access Point with BSSID CA:C9:38:35:DC:F2 started.
```



```
┌──(root㉿kali)-[~/captive_portal]
└─# python3 server.py
EVil-Twin is running...
```



3.3 Implementation of Deception Scripts/Tools

Challenges and Limitations

Despite planning to implement deception techniques, the following challenges prevented a successful practical demonstration:

1. **Windows as the Victim Machine**:
   - The Windows system was not configured for cybersecurity tasks like analyzing rogue APs or gathering detailed attacker information.
   - Tools like Wireshark and Nmap were available but needed more specialized configurations for advanced deception.
2. **Lack of USB Wi-Fi Adapters for Deception**:
   - Only two Wi-Fi adapters were available, both used to execute the Evil Twin attack.
   - The USB Wi-Fi adapter intended for the virtualized Kali Linux instance was not recognized by the system, preventing its use for deception.
3. **Virtualized Kali Linux**:
   - The Kali Linux instance on VirtualBox did not have access to a compatible USB Wi-Fi adapter, which limited its ability to perform necessary deception techniques like packet injection, scanning, and interaction with the rogue AP.

Although I could not implement these methods, the following section provides a detailed step-by-step guide of how I could have performed deception techniques to gather information about the attacker.

Step-by-Step Guide for Deception Techniques

1. The victim was connected to the Rogue AP on purpose

- From the victim machine (e.g., Windows or Kali Linux):
    1. Open Wi-Fi settings and connect to the rogue AP named "FreeWiFi."
    2. Confirm connection using the ipconfig (Windows) or ifconfig (Linux) command:
        - Identify the **Default Gateway IP** (e.g., 10.0.0.1), which corresponds to the attacker's rogue AP.

2. Identify the Attacker's IP and MAC Address

- Use the arp command to find the attacker's MAC address:
    - **Windows**:
      arp -a
        - Look for the **Default Gateway** in the output and note its MAC address.
    - **Linux**:
      arp -n

3. Submit Honey Credentials

- Open a browser and navigate to any website (e.g., http://example.com). The captive portal should redirect to a phishing page.
- Submit fake details (honey credentials):
    - Username: decoy_user
    - Password: Trap123!
    - Email: decoy@example.com
    - Phone: 5555555555
- Observe the portal's response to validate its logging mechanism.

4. Scan the Rogue AP for Open Ports and Services

- Use **Nmap** to probe the rogue AP for open ports and services:
    - Windows (with Nmap installed): **nmap -A 10.0.0.1**
    - Linux: **sudo nmap -A 10.0.0.1**
    - This identifies running services like DNS (port 53), HTTP (port 80), and potential back-end systems.

5. Monitor Network Traffic

- Install and use **Wireshark** on the victim machine to monitor traffic:
    1. Start capturing packets on the victim's wireless interface.
    2. Apply a filter to focus on traffic involving the rogue AP:

        ```
        ip.addr == 10.0.0.1
        ```

    3. Look for signs of DNS spoofing, HTTP redirects, or credential exfiltration.

6. Send Probing Requests

- Use PowerShell or a terminal to send crafted requests to the rogue AP:
    - For example, send HTTP requests with large payloads:

        ```
        Invoke-WebRequest -Uri http://10.0.0.1/test -Method POST -Body "Random Test Data"
        ```

    - This could overload the attacker's logging system or provide insight into their setup.

7. Observe DNS and IP Forwarding

- Use nslookup to test DNS responses from the rogue AP:

    ```
    nslookup example.com
    ```

- Check if the rogue AP forwards requests to external servers or provides fake DNS responses.

8. Log and Document Findings

- Record details about the rogue AP, such as:
    - **SSID**: "FreeWiFi"
    - **IP Address**: `10.0.0.1`
    - **MAC Address**: From `arp`
    - **Observed Behavior**: Phishing attempts, DNS spoofing, NAT configurations.

3.4 Practical Deception Method Results

While I could not implement these methods due to hardware and configuration limitations, this guide provides a comprehensive approach to deception in an Evil Twin attack scenario. Properly configured victim machines (with compatible tools and hardware) can gather valuable information about the attacker's infrastructure, including IP addresses, MAC addresses, running services, and behavioral patterns. This can lead to stronger defense mechanisms and insights into attacker techniques.

4. Conclusion and Future Work

This project highlighted the potential of using deception as a proactive strategy against the Evil Twin attack. Rather than just defending against threats, deception techniques create an opportunity to actively learn about attackers' methods and tools in a controlled environment. Through strategies like honeypots, fake captive portals, and decoy SSIDs, defenders can gather valuable intelligence while misleading attackers.

Although practical implementation faced challenges, including hardware limitations and system configurations, the theoretical groundwork and detailed guidelines provided a strong framework for future efforts. These approaches not only help detect and deter attackers but also offer a way to strengthen the overall security posture of wireless networks.

Future Work

Looking ahead, several areas can further enhance the application of deception in wireless security:

1. Upgrading Hardware: Investing in additional USB Wi-Fi adapters and ensuring compatibility with tools and virtual environments will enable more seamless and sophisticated implementation of deception techniques.
2. Automating Processes: Developing scripts and tools to automate honeypot creation, beacon frame manipulation, and attacker interaction logging will make the system more efficient and less reliant on manual input.
3. Leveraging Machine Learning: Using machine learning to analyze traffic patterns and detect attacker behavior in real-time could make deception techniques more adaptive and effective.
4. Data Analysis for Threat Profiling: Using the data gathered during deception to build profiles of attackers will not only enhance current defenses but also provide insights for anticipating future threats.

5. Bibliography

1. Atheros AR9271 Chipset - https://techinfodepot.shoutwiki.com/wiki/Atheros_AR9271
2. Airmon-ng Documentation - https://www.aircrack-ng.org/doku.php?id=airmon-ng
3. Aireplay-ng Documentation - https://www.aircrack-ng.org/doku.php?id=aireplay-ng
4. Airbase-ng Documentation - https://www.aircrack-ng.org/doku.php?id=airbase-ng
5. Aircrack-ng Suite Documentation - https://www.aircrack-ng.org/documentation.html
6. dnsmasq Man Page - https://thekelleys.org.uk/dnsmasq/doc.html
7. Python HTTPServer Documentation - https://docs.python.org/3/library/http.server.html
8. De-authentication Attack - https://en.wikipedia.org/wiki/Wi-Fi_deauthentication_attack
9. Rogue Access Point - https://en.wikipedia.org/wiki/Rogue_access_point
10. Monitor Mode Verification - https://linux.die.net/man/8/iwconfig
11. Linux Steps - https://linux.die.net/man/8/
12. iptables - https://en.wikipedia.org/wiki/Iptables#:~:text=6%20External%20links-,Overview,traversing%20the%20rules%20in%20chains.
13. index.html - https://developer.mozilla.org/en-US/docs/Web/HTML
14. Airodump-ng Documentation - https://www.aircrack-ng.org/doku.php?id=airodump-ng
15. Wireshark Documentation - https://www.wireshark.org/docs/
16. PyWiFi Documentation - https://pypi.org/project/pywifi/