UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

MATEUS RIAD DA CUNHA SOARES

Information Leaks on Wi-Fi Networks

Work presented in partial fulfillment
of the requirements for the degree of
Bachelor in Computer Science

Advisor: Prof. Dr. Raul Weber

Porto Alegre
December 2016

“Even when our eyes are closed,
there’s a whole world out there
that lives outside ourselves and our dreams.”
— Edward Elric

ACKNOWLEDGMENTS

I would like to thank my family for their love, support and continuous encouragement throughout my life. This accomplishment would not have been possible without them.

I would like to express my gratitude to those who have stood by my side during my graduation. Especially to my friends Inatan Hertzog and Cristiano Ruschel who gave me their support when I needed the most.

ABSTRACT

This document presents a study about information leaks on Wi-Fi networks with the final goal of raising security awareness among users. There are many threats that users are exposed to when using Wi-Fi networks, with many of those representing an active risk to the community. Even so, new threats appear every day, and consequently tools that aid users in countering them. These tools cover some of the most common threats but, as demonstrated in this work, some of the most relevant security threats in Wi-Fi networks are not monitored completely by said applications. Thus, a software was developed in the form of an android application with the objective of aiding users in avoiding some of the most common threats that are shown to not be covered by existing software. In the process of analyzing the current situation on Wi-Fi security, it was perceived that awareness must be raised among users of said networks, since no existing application can cover every possible weakness. Hence, an awareness list was created to help users take action to protect themselves from Wi-Fi security threats.

Keywords: Networks. Wi-Fi. Information Security. Awareness.

Vazamento de Informações em redes Wi-Fi

RESUMO

Esse documento apresenta um estudo sobre vazamentos de informações em redes Wi-Fi com o objetivo de aumentar a conscientização de seus usuários sobre sua segurança. Há muitas ameaças às quais usuários estão expostos, sendo muitas dessas um risco ativo à comunidade. Novas ameaças são apresentadas a cada dia, e consequentemente ferramentas que ajudam usuários a combatê-las são desenvolvidas. Essas ferramentas frequentemente abrangem as ameaças mais comuns, entretanto, como demonstrado nesse trabalho, algumas das ameaças de segurança mais relevantes em redes Wi-Fi não são completamente monitoradas por essas aplicações. Para aumentar a segurança de usuários de redes Wi-Fi, um software foi desenvolvido; uma aplicação android com o objetivo de auxiliar usuários a evitar algumas das ameaças mais comuns que não são combatidas pelos softwares existentes. No processo de analisar a situação atual de segurança Wi-Fi, notou-se que a conscientização dos usuários das ditas redes deve ser realizada, considerando-se que nenhuma aplicação existente pode cobrir todas as fraquezas existentes. Consequentemente, uma lista de conscientização foi criada para ajudar usuários a tomar ações para se protegerem de ameaças de segurança Wi-Fi.

Palavras-chave: Redes, Wi-Fi, Segurança da Informação, Conscientização.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

AES	Advanced Encryption Standard
AP	Access Point
ASCII	American Standard Code for Information Interchange
CA	Certificate Authority
CN	Common Name
DES	Data Encryption Standard
FTP	File Transfer Protocol
HSTS	HTTP Strict Transport Security
HTTPS	Hypertext Transfer Protocol Secure
HTTP	Hypertext Transfer Protocol
IMAP	Internet Message Access Protocol
IP	Internet Protocol
LDAP	Lightweight Directory Access Protocol
MITM	Man In The Middle
NAT	Network Address Translation
NTP	Network Time Protocol
OWASP	Open Web Application Security Project
PKI	Public Key Infrastructure
POP	Post Office Protocol
RFC	Request for Comments
SMTP	Simple Mail Transfer Protocol
SSID	Station Set Identifier
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol

TLS	Transport Layer Security
URL	Uniform Resource Locator
VPN	Virtual Private Network
WLAN	Wireless Local Area Network
WPA2	Wi-Fi Protected Access 2
WPE	Wired Equivalent Privacy

CONTENTS

1 INTRODUCTION

Over the past several years, Wi-Fi networks have increased in popularity due to their flexibility and ease of access, driving people in a daily basis through its internet connectivity. Not surprisingly, this connectivity attracts not only users but also attackers, which often have the intention of compromising the user's device and privacy.

Public and Private Wi-Fi networks have been available for quite some time throughout office environments and public areas such as airports, restaurants and shopping malls as a free or commercial service for mobile devices connectivity. However, those environments have not necessarily been well configured or created with the appropriate security policies, enabling attackers to maliciously retrieve private information from the network and the users that are connected to it. Due to the huge acceptance this technology has achieved among users and device manufacturers, vulnerabilities that exist on these networks must be studied and prevented.

Most of the attacks against wireless networks don't target the wireless network itself; rather, attacks such as Karma (ZOVI; MACAULAY, 2005) target client vulnerabilities directly. In this type of attack the device doesn't even need to be connected to a wireless network and it can still cause significant loss to the user security. Nonetheless, should an attack be successful, it can seriously compromise the user's device and privacy.

Techniques and protocols based on the CIA(Confidentiality, Integrity, Availability) principle[1] such as SSL/TLS (MCKINLEY, 2003) and HSTS[2] are used to ensure the user's security and privacy. However, vulnerabilities and bypassing methods are constantly demonstrated by research studies, as well as by proofs-of-concept.

When compromising a network, an attacker can perform attacks such as Man-In-The-Middle (MITM) to degrade the user's communication security. Thus, the attacker would be capable of intercept unencrypted sessions, where he can obtain and modify all the information transmitted.

Due to the attacks' ease of reproducibility and the risks to information and to the devices they can create, users must be aware of the threats they are vulnerable to when using devices with Wi-Fi capabilities.If the user is made aware of the risks associated with Wi-Fi networks, he or she is then able to take actions in order to be safer while accessing them.

Many tools exist and more are constantly being created to help users recognize

[1] http://whatis.techtarget.com/definition/Confidentiality-integrity-and-availability-CIA
[2] https://tools.ietf.org/html/rfc6797

vulnerabilities and mitigate threats regarding Wi-Fi networks, among them Wi-Fi Audit[3], WLANAudit[4], Avast - Mobile Security & Antivirus[5]. Unfortunately, there is no silver bullet in network security; no single application can completely protect a network by itself, and some vulnerabilities are not covered at all by those applications for some reasons. These vulnerabilities may not yet be known by the developers, not yet have a workaround or not affect enough users for it to justify the development effort. Whichever the case, this stresses the point that as much as tools may help users in mitigating security threats they must be aware of the security principles involved, in order to keep them as safe as possible.

This work presents a study of security information threats that are related to Wi-Fi environments even in scenarios in which the user is not connected to the network. In order to help the users to identify security threats on Wi-Fi networks and raise awareness among users, an application for Android systems is developed. Also, based on both research and the analysis of tests performed with the application, an awareness list was created, pertaining the main security issues a user or technician should be concerned with regarding Wi-Fi networks.

The ensuing chapters are organized as follows. Chapter 2 defines some of the significant concepts for understanding the security basis of a secure communication. Chapter 3 shows security threats that can break user's privacy and compromise their devices. Chapter 4 displays the inner workings of the application algorithm. Chapter 5 shows the testing methodology as well as the results obtained. Chapter 6 the awareness list is given and discussed. Chapter 7 outlines the contribution.

[3]https://play.google.com/store/apps/details?id=com.futuremind.wifiaudit
[4]https://play.google.com/store/apps/details?id=es.glasspixel.wlanaudit
[5]https://play.google.com/store/apps/details?id=com.avast.android.mobilesecurity

2 CONCEPTS

This chapter introduces some of the fundamental concepts related to the current methods and techniques used to ensure or compromise confidentiality, integrity and availability of a network.

2.1 CIA Principle

Security in the computer world determines the ability of a system to manage, protect and distribute sensitive information, and has as a fundamental principle the CIA triad: Confidentiality, Integrity and Availability. Also referred as AIC triad (to avoid confusions with the Central Intelligence Agency), the CIA principle is a very fundamental concept in security, in which each key concept is faced as an important step when designing any secure system (STAMP, 2011).

1. Confidentiality: Is the ability to hide information from people unauthorized to access it. Roughly equivalent to privacy, this principle ensures that the information can be viewed only by people with appropriate and correct privileges.

2. Integrity: The ability to ensure the data or the information system has an accurate and unchanged representation of the original secure information. Data must not be changed in transit, and steps must be taken to ensure that the data can not be altered by unauthorized users.

3. Availability: The ability to ensure that the information concerned is readily accessible to the authorized users when required. Systems, access channels, and authentication mechanisms must all be working properly for the information they provide and protect to be available when needed.

The following sections present further details and techniques used to implement characteristics of the security goals mentioned, and some specific vulnerabilities associated with each one. An assessment summary presenting a general vision of each principle is also presented in Table 2.1.

Table 2.1: Security Assessment

CIA	**Risk**	**Controls**	**Primary Focus**
Confidentiality	Information Leak	Cryptography	Concealment of information
Integrity	Fraud	Cryptography	Trustworthiness of information
Availability	Business disruption	Redundancy Fault Tolerance	Access of information

2.2 Cryptography

Cryptography is the art and science of making a cryptosystem that is capable of providing information security. Historically speaking, cryptography was conceived as a way to provide message confidentiality, by converting messages from a comprehensible form into an incomprehensible one. Also referred as encryption, it was primarily used with intent to ensure secrecy in communications from interceptors or eavesdroppers, such as spies and military leaders, without the required secret knowledge to reveal the message. Currently, the field of cryptography has expanded beyond confidentiality concerns, but also including techniques for message integrity checking, identity authentication, digital signatures, and secure computation, among others (MAO, 2003).

The following cryptography-related sections information, such as definitions, are mainly based on (SLONE, 1999).

Cryptography is an accepted and effective way of protecting data in transit. Often used in data and telecommunications, cryptography is used when communicating over any untrusted medium, which includes just about any network, particularly the Internet.

There are five primary functions of cryptography:

1. Commonly, cryptographic protocols make use of sequences of cryptographic primitives and schemes

2. Authentication: The process of proving one's identity.

3. Integrity: Assuring the receiver that the received message has not been altered in any way from the original. Data integrity can not prevent the alteration of data but provides a means for detecting whether data has been manipulated in an unauthorized manner.

4. Non-repudiation: A mechanism to prove that the sender really sent the message.

5. Key exchange: The method by which crypto keys are shared between sender and receiver to communicate securely.

Cryptography can be seen as a procedure, where an unencrypted data, referred to as plaintext, is encrypted into a ciphertext given an encryption method.

The general scheme of encryption and decryption of a data can be formulated as:

$$C = E_k(P)$$
$$P = D_k(C)$$

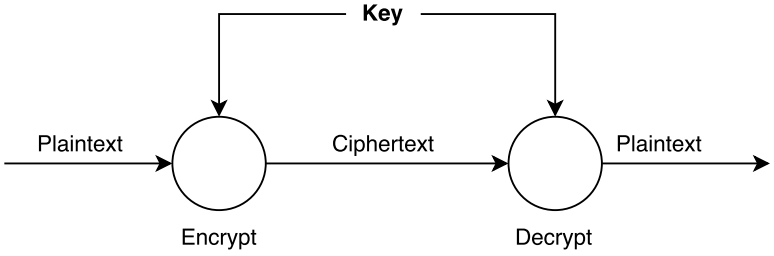
Where P = plaintext, C = ciphertext, E = the encryption method, D = the decryption method, and k = the key.

Regarding encryption concerns, it addresses the problem of how two parties can communicate in secret in the presence of an eavesdropper. Two approaches were designed and develop along the time: Single-key Cryptography and Public-key Cryptography, which are also called Symmetric and Asymmetric Cryptography. The main goals and definition of what each is trying to achieve is presented next.

2.2.1 Single-key Cryptography

Single-key or Symmetric Cryptography uses a single key to encrypt and decrypt a message. The scheme can be seen in Figure 2.1, where a plaintext is encrypted using a key to send to a receiver, thus the receiver uses the same key in order to decrypt the message. In the private-key scheme, all the users involved must know the key, which creates a problem of how to distribute the secret among the parties safely.

Figure 2.1: Symmetric Cryptography Scheme.



Key length impacts how secure the encrypted transmission will be, trying to make it computationally unfeasible to decrypt. The key length may define how well the original message it is protected against exhaustive search attacks such as a brute force attack, in which all possible keys are tried until a match is found to break the ciphertext. Exhaustive attacks are possible because of the increasing computing power, meaning that larger keys are needed for secure use.

Secret based algorithms are generally categorized in two types of cipher: stream cipher or block ciphers.

1. Stream Cipher: Operates a single one bit or byte of plaintext at a time.
2. Block Cipher: Operates a block of data at a time.

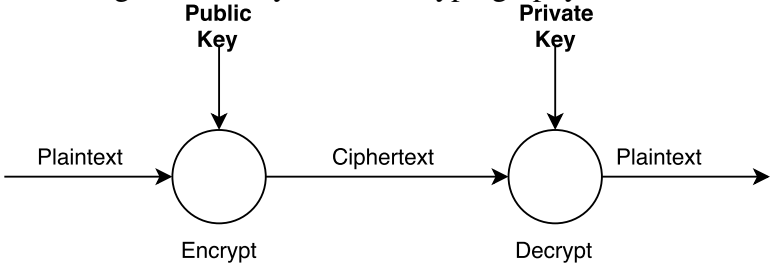
For practical purposes, Stream and Block Ciphers have its advantages and disadvantages. Stream Ciphers have a faster speed of transformation than block based as a result of dealing with just one symbol at a time. Thus, its algorithms are linear in time and constant in space. Accordingly, a block cipher is the slowest of encrypting because an entire block must be accumulated before the encryption/decryption procedure can begin.

Although stream cipher is likely to be faster than a block cipher, the use of block ciphers provide a more secure encryption due to its high diffusion - the information from one plaintext symbol is diffused into several ciphertext symbols. Whereas with a stream cipher, all information of plaintext symbol is contained in a single ciphertext symbol. Thus, most modern symmetric encryption algorithms are block ciphers (e.g., AES).

2.2.2 Public-key Cryptography

Public-key or Asymmetric Cryptography is an encryption scheme that uses two keys: A public key known to everyone and a private or secret key known only by the recipient of the message. Unlike symmetric key algorithms that rely on just one key to both function: encryption and decryption; each key performs a unique function. The public key is used to encrypt and the private key is used to decrypt, as can be seen in Figure 2.2.

Figure 2.2: Asymmetric Cryptography Scheme.



Mathematically proven, it is computationally unfeasible to compute the private key based on the public key in a doable time. Thus, public keys can be freely shared, allowing users an easy and convenient method for encrypting content and verifying digital signatures.

The main uses for public-key cryptography are:

1. *Digital signatures*: content is digitally signed with an individual's private key and is verified by the individual's public key

2. *Encryption*: content is encrypted using an individual's public key and can only be decrypted with the individual's private key.

The use of the Public-key scheme can provide benefits such as Confidentiality, Authentication, Non-repudiation and Integrity.

1. *Confidentiality*: because the content is encrypted with an individual's public key, it can only be decrypted with its private key, ensuring that only the intended recipient can decrypt and view the contents.

2. *Authentication*: since the individual's unique private key was used to apply the signature, recipients can be confident that the individual was the one to actually apply the signature.

3. *Non-repudiation*: since the individual is the only one with access to the private key used to apply the signature, he/she cannot later claim that it wasn't him/her who applied the signature.

4. *Integrity*: when the signature is verified, it checks that the contents of the document or message match what was in there when the signature was applied. Even the slightest change to the original document would cause this check to fail.

2.2.2.1 Public Key Infrastructure (PKI)

A Public Key Infrastructure is a set of rules, policies, and procedures to create, manage, distribute, use, store, and revoke digital certificates (HOUSLEY et al., 2002).

A digital certificate is an electronic identification that allows a person, computer or organization to exchange information securely over the Internet using the public key

infrastructure (PKI). Just like a passport, it provides identifying information, is forgery-resistant and can be verified because it was issued by an official, trusted agency. A digital certificate may also be referred to as a public key certificate.

To provide evidence that a certificate is genuine and valid, it is digitally signed by a root certificate belonging to a trusted certificate authority (CA). Operating systems and browsers maintain lists of trusted CA root certificates so they can easily verify certificates that the CAs have issued and signed. When PKI is deployed internally, digital certificates can be self-signed.

Most of the communications over the internet are safeguarded by certificates, which help authenticate the server and sometimes the client, as well as encrypt the traffic exchanged between them. Digital certificates also play a critical role in signing software, which helps determine the source and authenticity of the program when deciding whether to trust it. Certificates can also be used as the basis for securing VPN and Wi-Fi connections.

2.2.2.2 Certificates Authorities

Durumeric,Kasten,Bailey and Halderman (DURUMERIC et al., 2013) describes *Certificates Authorities* as: "Certificate authorities (CAs) are trusted organizations that issue digital certificates. These organizations are responsible for validating the identity of the websites for which they provide a digital certificate. They cryptographically vouch for the identity of a website by digitally signing the website's leaf certificate using a browser-trusted signing certificate. "

The CA responsibilities are listed below:

- Issuing the public key certificates.
- Distributing the certificates and binding them with the responding entities.
- Renewing public key certificates.
- Revoking public key certificates.

2.3 Cryptographic protocols

A cryptographic protocol is a procedure carried out between two parties, which is used to perform a security-related task. Commonly, cryptographic protocols make use of

sequences of cryptographic primitives and schemes. The primitives and schemes used to design a cryptographic protocol may incorporate aspects, such as:

1. Key agreement or establishment
2. Entity authentication
3. Symmetric encryption and message authentication material construction
4. Secured application-level data transport
5. Non-repudiation methods
6. Secret sharing methods
7. Secure multi-party computation

A practical example of data communication demand requiring the use of a cryptographic protocol would be the transmission of bank credentials, online shopping details, personal or confidential documents from a user to another. Using as an example, a communication from Bob to Alice might proceed within a protocol which typically involves a digital signature scheme(so Bob knows he is communicating to Alice), and a form of encryption(to ensure that Bob's data can not be intercepted, neither modified when in transit).

In order to secure communication on the internet, two main cryptography protocols are used: SSL and TLS.

2.3.1 Secure Sockets Layer (SSL)

Based on (MCKINLEY, 2003) Secure Socket Layer (SSL) is a standard security protocol for establishing a secure link between two communicating applications. This link ensures that all data transmitted remain private and integral. The protocol combines three points to provide connection security. These points are:

- *Privacy*: connection through encryption.
- *Identity authentication*: identification through certificates.
- *Reliability*: dependable maintenance of a secure connection through message integrity checking.

In order to create an SSL connection, a web server makes use of an SSL Certificate.

SSL Certificates use the public-key cryptography exchange model to ensure data encryption. Each certificate typically contains the identity of the server (e.g., website domain), and details of the Certification Authority responsible for the issuance of the certificate. When a browser connects to a secure site it will retrieve the site's SSL Certificate and check if it has not expired, if it has been issued by a Certification Authority the browser trusts, and that it is being used by the website for which it has been issued. If some of the verification steps fails on any one of these checks, the browser will likely display a warning to the end user letting them know that the site is not secured by SSL.

After SSLv3 the protocol was improved and served as basis for TLS 1.0 (Transport Layer Security).

2.3.2 Transport Layer Security (TLS)

Transport Layer Security (TLS) is a new name for SSL. TLS is a cryptographic protocol that is used to secure the web (HTTP/HTTPS) connections, providing privacy and data integrity between two communicating parties (e.g., browser and web service). It has an entity authentication mechanism, based on the X.509 system; a key setup phase, where a symmetric encryption key is formed by employing public-key cryptography; and an application-level data transport function. These three aspects have important interconnections. Standard TLS does not have non-repudiation support.

2.3.3 SSL vs TLS

TLS and SSL are most widely recognized as the protocols that provide secure HTTP (HTTPS) for Internet transactions between Web browsers and Web servers. TLS/SSL can also be used for other application level protocols, such as File Transfer Protocol (FTP), Lightweight Directory Access Protocol (LDAP), and Simple Mail Transfer Protocol (SMTP). TLS/SSL enables server authentication, client authentication, data encryption, and data integrity over networks such as the World Wide Web.

Is important to mention that the complexity of the SSL/TLS protocol and its procedures remain invisible to the user. Web browsers usually provide a key indicator and lock icon to let users know they are currently protected by an SSL/TLS encrypted session.

The main difference between SSL and TLS is how the connection is initialized. There are two distinct ways that an application can initiate a secure connection with a server:

1. By port (explicit) : Is commonly referred to SSL or "explicit", connecting to a specific port means that a secure connection should be used. Ports as 443 for HTTPS (secure web), 993 for secure IMAP, 995 for secure POP, etc. These ports are setup on the server ready to negotiate a secure connection first and do whatever else the application is intended to do after.

2. By protocol (implicit) : These connections first begin with an insecure "hello" to the server and only then switch to secured communications after the handshake between the client and the server is successful. If this handshake fails for any reason, the connection is severed. A good example of this is the command "STARTTLS" used in an outbound email (SMTP) connections.

Use of either could result in a connection encrypted with either SSLv3 or TLS v1.0+, based on what is installed on the server and what is supported by the application.

Both methods (explicit and implicit) ensure that your data is encrypted as it is transmitted across the Internet. Using certificates issued to the application by a trusted third party (e.g., Verisign), each protocol ensures that the application is communicating with the server it is intended to communicate with.

TLS is newer and more secure than SSL. Thus, SSL is directly considered deprecated and the use of TLS is likely encouraged. TLS is currently in the version 1.2 but there is a working draft of the version 1.3.

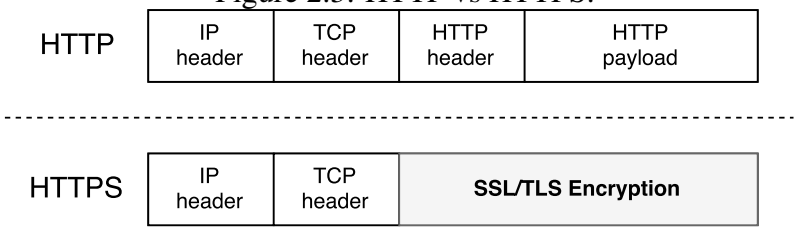
The use of newer protocols, which are vastly tested and accepted by the cryptographyc community is always recommended. Hence, communications methods are updated against newer possible vulnerabilities and attacks discovered along the time.

2.4 HTTP vs HTTPS

Hyper Text Transfer Protocol Secure (HTTPS) is the secure version of HTTP, the protocol over which data is sent between a browser and a website that is connected to. The ’S’ at the end of HTTPS stands for ’Secure’. It means all communications between the browser and the website are encrypted using SSL/TLS. HTTPS is often used to pro-

tect highly confidential online transactions like online banking and online shopping order forms.

Figure 2.3: HTTP vs HTTPS.



All communications sent over regular HTTP connections are in 'plain text' and can be read by any eavesdropper that manages to intercept the connection between the user's browser and the website. This presents a clear danger if the 'communication' is on an order form and includes information such as credit card details or social security number. As can be seen in Figure 2.3, with an HTTPS connection, the entire protocol including the HTTP header(which contains the URL information) is encrypted. This means that even if an eavesdropper manage to get the communication of a connection, he would not be able to read any of the data transmitted.

2.5 HTTP Strict Transport Security (HSTS)

OWASP[1] describes HSTS as: "HTTP Strict Transport Security (HSTS) is an opt-in security enhancement that is specified by a web application through the use of a special response header. Once a supported browser receives this header that browser will prevent any communications from being sent over HTTP to the specified domain and will instead send all communications over HTTPS. It also prevents HTTPS click through prompts on browsers". The specification has been released and published end of 2012 as RFC 6797[2]

HSTS addresses the following threats:

1. User bookmarks or manually types http://example.com and is subject to a man-in-the-middle attack

   Control: HSTS automatically redirects HTTP requests to HTTPS for the target domain

2. Web application that is intended to be purely HTTPS inadvertently contains HTTP links or serves content over HTTP

[1] https://www.owasp.org/index.php/HTTP_Strict_Transport_Security_Cheat_Sheet
[2] https://tools.ietf.org/html/rfc6797

Control: HSTS automatically redirects HTTP requests to HTTPS for the target domain

3. A man-in-the-middle attacker attempts to intercept traffic from a victim user using an invalid certificate and hopes the user will accept the bad certificate.
   Control: HSTS does not allow a user to override the invalid certificate message

The HSTS header can be defined by two main directives: *max-age* and *includeSubDomains*. An example of its usage can be seen below:

```
Strict-Transport-Security: max-age=31536000; includeSubDomains
```

The example above specifies a policy using 31536000 seconds(1 year) max-age, including all present and future subdomains. As result, a browser that receives this policy from a website would stay using HTTPS if the user clicks on HTTP links or even if the user types an HTTP link of the respective domain. The policy would be active for the following 1 year from the last HTTPS connection. After that, the policy outdates and the browser returns to his usual behavior. It is a secure option but will block access to certain pages that can only be served over HTTP.

In addition, an HSTS policy prevents a user from accepting self-signed or abnormally signed certificates, because it remembers the certification authority (CA) that signed the certificate previously seen.

Unfortunately, HSTS is not a security feature that is currently widely deployed on the Internet, since just a few websites use it. However, some reference companies such as Facebook, Twitter, Amazon or Google use this security feature.[3]

[3]https://w3techs.com/technologies/details/ce-hsts/all/all

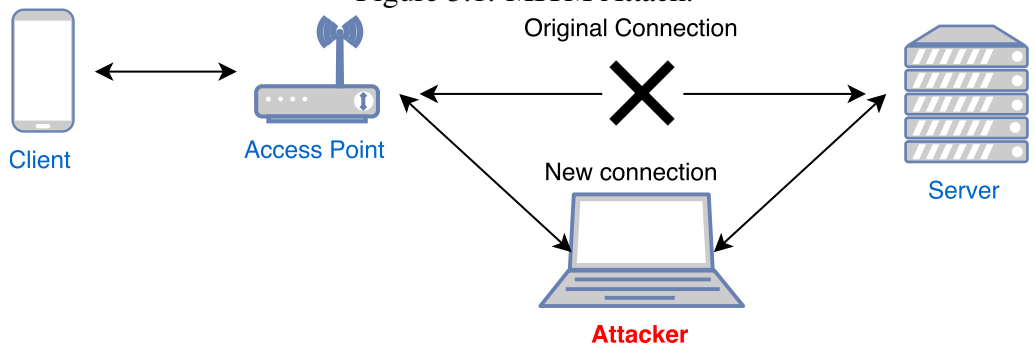
3 SECURITY THREATS

When surrounded by or connected to a Wi-Fi network, a user can be susceptible to a variety range of attacks. Most of the time, crackers are focused on exploiting vulnerabilities of equipment, enabling them to take full control of a system to their desired usage. Therefore, in this work are presented attack vectors focused on information leak, even though they may also be used for another purpose.

3.1 Man In The Middle (MITM)

A Man in the Middle(MITM) attack is when a third party intercepts the connection of a user and most likely alters the communication between user and server. Most of the times the target does not even know that its device is connected to a malicious network or that a reliable network has been compromised (CALLEGATI; CERRONI; RAMILLI, 2009).

Figure 3.1: MITM Attack.



In an HTTP transaction, the target is the TCP connection between client and server. Using different techniques, the attacker splits the original TCP connection into 2 new connections, one between the client and the attacker and the other between the attacker and the server, as shown in Figure 3.1. Once the TCP connection is intercepted, the attacker acts as a proxy, being able to read, insert and modify the data in the intercepted communication.

The MITM attack is very effective due to the nature of the HTTP protocol and data transfer which are all ASCII based. Therefore, it is possible to view and interview within the HTTP protocol and also in the data transferred. Consequently, an attacker can capture information by reading the HTTP header, or change objects inside the application context.

A MITM attack can also be accomplished over an HTTPS connection by using fake certificates. It consists in the establishment of two independent SSL sessions, one over each TCP connection. The browser sets an SSL connection with the attacker, and the attacker establishes another SSL connection with the web server. In general, the browser warns the user that the digital certificate used is not valid, but the user may ignore the warning because he doesn't understand and is not aware of the threat. In some specific contexts, it is possible that the warning doesn't appear, as for example, when the Server certificate is compromised by the attacker or when the attacker certificate is signed by a trusted CA and the CN is the same of the original website.[1]

3.2 Bypassing HSTS

Protections such as HSTS are used to enforce HTTPS connections, thus it has the intent to mitigate attacks where the user is forced to communicate under a not encrypted transmission (HTTP). However, under certain circumstances, an attacker can affect the behavior of the user's device to bypass those protections.

As mentioned before in Section 2.5, there are two main parameters in an HSTS policy. One of them is 'max-age' that represents the amount of seconds that a browser should connect in HTTPS-only mode. Another is the 'IncludeSubdomains', which applies the policy to all the respective subdomains. The attack presented next is based on expiring the max-age functionality.

By default, almost all operating systems automatically synchronize its time with internet servers using the NTP protocol. It is important to mention that NTPv4 supports authentication based on asymmetric cryptography. The server signs NTP messages using his own private key. As a result, clients can verify messages integrity, so MITM techniques shouldn't be possible. However, none operating system uses authentication, so all of them would be vulnerable to MITM attacks.

This inter-operation vulnerability in the HSTS was first presented by Selvi (SELVI, 2014) at the Black Hat Conference. The technique performed and describe by Selvi allows crackers to bypass HSTS protections by performing NTP MITM attack. To easily accomplish that, a tool called 'Delorean' was developed by Jose Selvi. The name, as you probably know, is a reference to the well-known 80's film 'Back to the future' and its time machine.

[1] https://www.owasp.org/index.php/Man-in-the-middle_attack

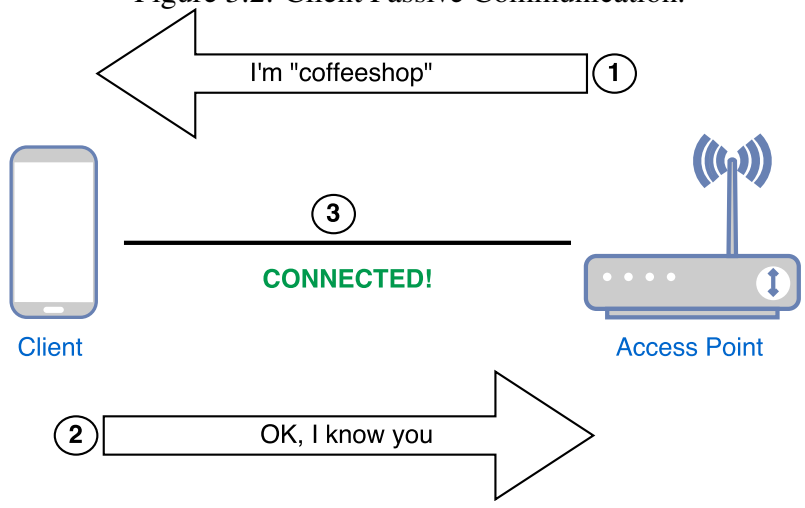
Delorean is an NTP server written in python, open source and available from GitHub[2] that can fake NTP responses. That being said, it can be used to bypass/expire the *max-age* parameter of the HSTS by giving a response to the user's system with a date in the future. Usually, the systems do not display the year parameter to the user, just the current hour. So a user may be deceived and not notice that its system time has been maliciously changed.

This time manipulation attack regards on the user's system policy for time synchronization. Thus, it depends on certain circumstances and configurations in order to work properly.

3.3 Karma Attack

Mobile devices with Wi-Fi connection capability, in order to identify possible and known AP's it starts to passively and actively probe all the networks nearby to its location. In a not malicious environment, an AP will send out a beacon frame that indicates its network SSID, identifying the Wi-Fi network.

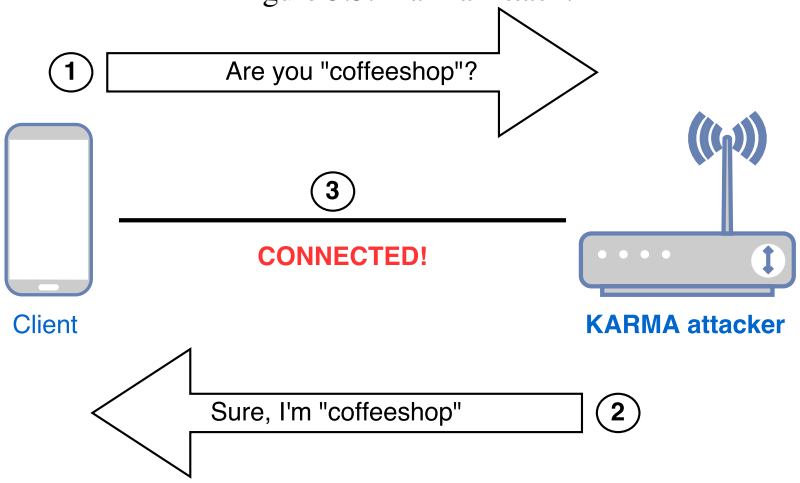
Figure 3.2: Client Passive Communication.



[2]https://github.com/PentesterES/Delorean

Karma attack (ZOVI; MACAULAY, 2005) takes advantage of the active client communication, by passively listening for 802.11 probes requests frames, used to identify known networks, sent from a client and answering as the correspondent SSID requested (hidden or not). Consequently, if configured to automatically connect to known networks, the client will join to the Rogue AP network - a wireless access point that mimics real ones in an attempt to get users to connect to it - to deceive the user's device and have access to its communication. Thus, an attacker can perform MITM to capture credentials or use other attacks, in which he can explore vulnerabilities on the client system.

Figure 3.3: Karma Attack.



3.3.1 Wi-Fi and Authentication

Karma attack can impersonate any network by passively monitoring probe requests from clients. However, when the network authentication is encrypted (e.g., WPA2), the handshake communication between the client and an AP can not be made because the attacker doesn't have the passphrase of the network. Consequently, a Rogue Access point can not impersonate a Wi-Fi network which contains encryption methods, unless the attacker has the correct passphrase.

Is important to mention that if an attacker has the passphrase of the network, he can create a Rogue Access point identical to the legit one. Thus, it is possible to perform deauthentication methods to force clients to reconnect to the network. In case the attacker has a stronger signal, the client's device will likely connect to the attacker's access point.

3.3.2 Portable Devices

Projects such as the Pineapple[3] from Hak5 and Pwn Phone[4] from PWNIE EXPRESS are portable and powerful devices making it incredibly easy to evaluate wired, wireless and Bluetooth networks. Both contain a huge and expansible set of hacking tools. Those devices can be legit used for Wi-Fi auditing by Pentesters or unfortunately with malicious intent by criminals. Its size makes really easy to perform any set of attacks to Wi-Fi networks without being notice for anyone. Being ready for any deployment scenario.

Figure 3.4: Pineapple Device - Nano and Tetra Basic Models.

Source: Pineapple's Website[3]

3.3.3 User Profiling

Data channels are highly correlated. Therefore, the data transmitted from the device has serious implications for privacy (SAPIEZYNSKI et al., 2015).

Wi-Fi information routinely collected by a device can easily be converted to precise information about the user location. Wi-Fi routers reveal where users live, work, and spend their leisure time. Consequently, the use of techniques such as Karma can easily convert WiFi information into geographical position, creating a user profiling and possibly breaking and exposing the user's privacy.

[3]https://www.wifipineapple.com/
[4]https://store.pwnieexpress.com/product/pwn-phone2014b/

4 AUDITING APPLICATION

Due to the vulnerabilities presented in Section 3, an auditing application was developed. As previously declared, it has the aim of increasing awareness of users towards security issues in Wi-Fi networks, especially threats involving malicious networks. As Android system dominates the smartphone market share with a share of 87.6%[1], the auditing application is proposed and developed primary focusing on its users.

In this section, are presented the implemented algorithms to identify evidence of malicious and compromised networks. Techniques such as Karma and MITM mentioned in the previously sections show that danger to the user privacy exists even in scenarios where the device is not connected to a Wi-Fi network. Therefore, two scopes were defined to approach those threats based on the device's network connection state: device disconnected and device connected.

For further purposes and usage, the application developed is named "RWA".

4.1 Device disconnected

In order to identify possible Rogue AP's, a honeypot mechanism was set to detect suspicious networks. The application initially assigns a fake SSID to the client's smartphone network known list and starts to run the main loop while the app is not closed. Consequently, the network scan procedure takes part, probing the bogus network, and in case the correspondent beacons frames are retrieved, they are evaluated and an alert is triggered, logged and displayed to the user.

The core detection procedure is implemented and can run scans in the background, at a determined time interval(configured in seconds), which can be set on the user interface.

When the application is closed, a *destroy* procedure removes the structure used to detect the Karma attack and updates the *shouldRun* to false, then closing the application successfully.

The full procedure can be seen in the Algorithm 4.1.

[1] https://www.idc.com/prodserv/smartphone-os-market-share.jsp

Algorithm 4.1: RWA | KarmaDetector

```
1 shouldRun = true;
2 KarmaDetector(){
3     startWIFIManager();
4     scanInterval = getScanInterval();
5     bogusSSID = getRandomSSID();
6     setDecoyNetwork(bogusSSID);
7     while (shouldRun)do
8         scanNetworks();
9         sleep(scanInterval);
```

Using this approach RWA is able to detect and inform the user of suspicious networks nearby his device. However, in case of the device already being connected to a network, other procedures should be performed to ensure the privacy and confidentiality of the connection.

4.2 Device connected

To identify a possible MITM attack and the use of fake certificates - which are not recognized Root CA's signing certificates and shouldn't be trusted - redirection checking and Certificate Pinning[2] (EVANS; PALMER; SLEEVI, 2015) are used.

Initially, the application maintains a list of a few relevant websites which should provide encrypted connections reliably (e.g., paypal.com, gmail.com, facebook.com). Those websites are used to check request redirection and fake certificates. In this process the application first makes requests using the website stored, retrieving its respective connection instance and parameters. Thus, it is verified if the connection is encrypted or not. In case the request is redirected to an uncrypted session(HTTP), a null certificate will be retrieved. Consequently, the alert procedure is triggered and the user is informed. Otherwise, the Certificate Pinning procedure takes place.

Two approaches were made using Certificate Pinning: CA's Chain of Trust and Self-Signed Certificate Chain.

- CA's Chain of Trust: The application maintain a set of valid certificates and fingerprints to compare with respective retrieved certificates connections using the local trust manager.

[2] https://www.owasp.org/index.php/Pinning_Cheat_Sheet

Problem: The certificates embedded in the app will eventually expire. Thus, an app update plan or a safe method for the application download the new certificate is required.

- Self-Signed Certificate Chain: Only one trusted and controlled certificate is stored. It must use a custom trust manager to instantiate a safe connection. The communication can be used to store and update other certificates through its safe connection.

  Problem: Single point of failure. If the server and the trusted chain is compromised, then the application communication shouldn't be trusted and all the users are affected.

In both methods, if the certificate is not provided or if the fingerprint does not match, the test function will have a false as return. The false result indicates that the user is most likely receiving a fake or null certificate and that he must be alerted about that.

The respective algorithms full procedure for the Certificate Pinning approaches mentioned can be seen in the Algorithm 4.2 and Algorithm 4.3.

Algorithm 4.2: RWA | CheckMITM - CA's Chain of Trust

```
1 CheckCommunication(){
2     domainList = getDomainList();
3     foreach (domain in domainList)do
4         domainCert = getCertificateFromKeyStore(domain);
5         connection = openConnection(domain);
6         connection.connect();
7         if (!FingerprintsEqual(domainCert, connection))then
8             alert();
```

Algorithm 4.3: RWA | CheckMITM - Self-Signed Certificate Chain

```
1  CheckCommunication(){
2      serverDomain = getServerDomain();
3      serverCert = getCertificateFromKeyStore(serverDomain);
4      setupCustomTrustManager(cert);
5      connection = openConnection(serverDomain);
6      connection.connect();
7      if (!FingerprintsEqual(serverCert, connection))then
8          alert();
9      else
10         domainList = getDomainList();
11         foreach (domain in domainList)do
12             domainCert = getCertificateFromServer(domain);
13             connection = openConnection(domain);
14             connection.connect();
15             if (!FingerprintsEqual(domainCert, connection))then
16                 alert();
```

Using the algorithms describe in this section, RWA is able to detect the proposed attacks: Karma, MITM and consequently HSTS bypassing method. Hence, the application alerts the end user accordingly with the threats he is susceptible to.

In order to test the RWA algorithms, a malicious environment was set. The results obtained, as well as the environment setup used can be seen in the next chapter.

5 RESULTS

This chapter presents the results obtained when trying to determine the effectiveness of the algorithms developed, as well as the testing methodology.

5.1 Testing Methodology

The test scenario was chosen to evaluate the application correctness consists of the setup of a malicious environment, where a device is performing the Karma attack and a Rogue access point is available to connect. The same scenario was used to test MITM attacks, by proxying the user connection that is being granted by the rogue access point. To automatize the testing process and setup, one tool were used: MANA Toolkit.

MANA[1] is a toolkit for rogue access point attacks first presented at Defcon 22. The toolkit can be used to perform Karma attacks and to setup rogue access points, essentially providing useful configurations for conducting MITM attacks. It attempts to masquerade as a legitimate and trusted wireless access point to trick unsuspecting users into connecting to the Wi-Fi being broadcasted and using it to access their favorite sites. A MANA evil access point acts as a MITM in all traffic sent through it, and keeps a log of all information being communicated between the devices and the Internet. The traffic logs can then be analyzed to find usernames, email addresses, and passwords in plain text, as well as other information that users send out to websites. This information can then be used by the attacker to gain access to whatever the users accessed while connected to the MANA evil access point.

MANA comes with a set of initiation scripts, which each starts a predefined environment. The different start scripts are listed below and must be edited to point to the right Wi-Fi device (default is wlan0) Those scripts are:

- start-nat-full.sh - Will fire up MANA in NAT mode with MITM components.
- start-nat-simple.sh - Will fire up MANA in NAT mode, but without any of the firelamb, sslstrip, sslsplit, scripts.
- start-noupstream.sh - Will start MANA in a "fake Internet" mode. Useful for places where people leave their Wi-Fi on, but there is no upstream Internet. Also contains a captive portal.

[1] https://github.com/sensepost/mana

- start-noupstream-eap.sh - Will start MANA with the EAP attack and noupstream mode.

The experimentation environment used the operating system Kali Linux 2016.2 64-Bit[2] as the base for the testing procedures. However, any Linux environment can be set and used for the same purpose.

Regarding hardware prerequisites, the only one that is worth noting is a Wi-Fi card that supports "access point"/"master" interface mode.

In the test procedure, two network interfaces were used: one wired (Eth0) and one wireless(WLAN0). The wireless interface was used as downstream, so the Karma attack can be performed and devices can connect themselves to the network. The wired interface is used as upstream, serving as a proxy and providing internet connectivity to the users.

In order to demonstrate the effectiveness of the features presented in the application proposed and developed, a comparison using relevant free apps was made. The applications selected for this purpose were: Wi-Fi Audit[3], WLANAudit[4], Avast - Mobile Security & Antivirus[5]. The choice of these specific applications was mainly based on their respective description and on whether their coverage is supposed to encompass Wi-Fi security concerns or not.

To initialize a simple malicious environment, one can use the script "start-nat-simple.sh" provided in the MANA Toolkit. It has the required commands to configure a NAT network, where clients will likely be connected to, as well with the commands to start scripts such as sslstrip and sslsplit. Those scripts have the main purpose of degrading the user's connection to an unencrypted session (HTTP) and parse the information and parameters transmitted. The "start-nat-simple.sh" script uses as base the configuration written in the file "hostapd-mana.conf". In this file it is possible to configure the name and specifications of a predefined Rogue Access Point, which will likely be probed with the ones requested by client's devices. The Karma behavior and logs can be set and optimized in this file too. A full description of options is available in "hostapd.conf"[6]

[2]https://www.kali.org/downloads/
[3]https://play.google.com/store/apps/details?id=com.futuremind.wifiaudit
[4]https://play.google.com/store/apps/details?id=es.glasspixel.wlanaudit
[5]https://play.google.com/store/apps/details?id=com.avast.android.mobilesecurity
[6]https://github.com/sensepost/hostapd-mana/blob/master/hostapd/hostapd.conf

5.2 Experiments

Each one of the chosen applications was tested against the two main scopes defined for this work: Device Connected and Device Disconnected; each with their respective vulnerabilities, such as Karma attack and MITM.

When running the auditing application and the others in the set environment, the criteria evaluated were whether the specific application has features to detect the Wi-Fi threats and how the results are displayed to the end user.

Table 5.1: Apps Results - Detection.

	Detection	
App	**Karma Attack**	**MITM**
RWA	Yes	Yes
Wi-Fi Audit	No	No
WLANAudit	No	No
Avast - Mobile Security & Antivirus	No	Yes

Initially, the applications were classified according to their threat detection capabilities, as can be seen in Table 5.1. Most of the applications have a naive approach, regarding only the Wi-Fi encryption method used for authentication (e.g., WPE,WPA2), rather than dealing with proper privacy threats. Thus, none of the chosen applications managed to provide any features to detect Karma attacks, and only the Avast Security & Antivirus was able to detect MITM.

Table 5.2: Apps Results - Alert System.

	Alert		
App	**Fake Certificate**	**Redirection**	**Wi-Fi Cryptography**
RWA	Yes	Yes	No
Wi-Fi Audit	No	No	Yes
WLANAudit	No	No	Yes
Avast - Mobile Security & Antivirus	No	No	Yes

When combining the previous results with the Table 5.2, it is possible to notice that the applications showed themselves to only serve a naive and weak purpose. They don't evaluate relevant threats and alert the user correctly, most likely giving a false sense of security to the user.

In order to show how these tools may affect the user population, the number of users of each of the applications is shown below, according to the number of downloads

data retrieved from Google Play[7] website on November 14th, 2016.

Table 5.3: Apps - Number of downloads.

App	**Downloads**
RWA	-
Wi-Fi Audit	1,000 - 5,000
WLANAudit	1,000,000 - 5,000,000
Avast - Mobile Security & Antivirus	100,000,000 - 500,000,000

The previous results show that a considerable number of users are not aware of important threats that they may be dealing with in their daily life, even though they are currently using tools that are supposed to help on Wi-Fi security matters.

Relying on just one way of protection, such as the use of currently available applications, has been demonstrated to not be a considerable option. Hence, alerting users of possible threats becomes an essential task to be accomplished. Therefore, to give some valuable advice and possible actions to mitigate security concerns and raise awareness among users, a set of instructions for users is proposed and thoroughly considered in the next chapter.

[7]https://play.google.com/

6 AWARENESS LIST

Vulnerabilities such as Karma and MITM are not trivial to be identified. There are multiple factors that can affect how vulnerable a system is. Some platforms are inherently more vulnerable than others and auditing applications can be used to alert users of these vulnerabilities. However, a user should not rely only on applications since there is no silver bullet for information security. Therefore, there are several precautions and actions that users can take to limit their exposure when using their devices on Wi-Fi enabled networks.

- **Be Aware**: Public Wi-Fi networks may be insecure, so be cautious when using and keeping them on your device's public network list. Karma attacks can be used to force your device to automatically connect to a malicious network.

- **Treat all Wi-Fi links with suspicion**: Don't just assume that the Wi-Fi link is legitimate. It could be a bogus link (rogue access point) that has been set up by a cybercriminal that's trying to capture valuable, personal information from unsuspecting users.

- **Disable automatic Wi-Fi connecting**: Ensure that your laptop, tablet or smartphone is not configured to automatically connect to open networks within its range.

- **Keep Wi-Fi off when you don't need it**: Even if you haven't actively connected to a network, the Wi-Fi hardware in your computer is still transmitting data to all the networks within range.

- **Disable Active probing**: Wi-Fi scanning means that location and other apps will be able to scan for Wi-Fi networks even when your device's Wi-Fi radio is off. With Android 6.0 Marshmallow, Google also allows apps to use Bluetooth to scan for networks. This means that even if you have Wi-Fi and Bluetooth turned off, your Android device running Marshmallow will still use both Wi-Fi and Bluetooth to scan for networks and improve location accuracy. In order to disable active probing in Android system (Marshmallow), one can accomplish by following: Settings, Location, Scanning; Thus, turning off both features.

- **Try to verify the legitimacy of the wireless connection**: Some bogus links that have been set up by malicious users will have a connection name that's deliberately

similar to the coffee shop, hotel, or venue that's offering free Wi-Fi. If you can speak with an employee at the location that's providing the public Wi-Fi connection, ask for information about their legitimate Wi-Fi access point – such as the connection's name and certificate.

- **Avoid websites which do not provide encrypted sessions (HTTPS)**: It is a good idea to avoid logging or placing information into websites in which there's a chance that cyber criminals could capture your identity, passwords, or personal information. One should pay special attention when accessing social networking sites, online banking services, or any websites that store your user information.

- **Consider using your cell phone mobile data**: If you need to access any websites that store or require the input of any sensitive information – including social networking, online shopping, and online banking sites – it may be worthwhile accessing them via your cell phone network, instead of the public Wi-Fi connection. These networks are in general more secure against MITM and Karma attacks.

- **Choose appropriate passwords**: Simple passwords can easily compromise your access to internet services. Try to always follow the password guidelines for each service you are using and other guidelines for creating secure passwords. Also, avoiding the use of the same password across multiple platforms makes the compromise of a single password less harmful.

- **Enable two-factor authentication**: It is a good practice to enable two-factor authentication on services that support it, such as Gmail, Twitter and Facebook. This way, even if someone does manage to sniff out your password when on public Wi-Fi, you have an added layer of protection.

- **Protect your device against cyber attacks**: Make sure all of your devices are protected by rigorous anti-malware and security solutions such as Antivirus and Auditing applications – and always confirm that the device's software and applications are properly updated.

- **Use a VPN (virtual private network)**: By using a VPN when you connect to a public or unknown Wi-Fi network, you'll effectively be using a 'private tunnel' that encrypts all of your data that passes through the network. This can help to prevent cybercriminals that may be lurking on the network from intercepting your data.

The instructions above have the intention of helping users to mitigate attacks they may be susceptible to when using Wi-Fi networks, henceforth protecting their devices and privacy against unauthorized access or malicious use.

7 CONCLUSIONS

This work presents a study about information leaks on Wi-Fi networks with the final goal of raising security awareness among users. By providing possible attacks as examples of security threats that users may be susceptible to when using devices with Wi-Fi capabilities, respective countermeasures are listed and an android application to aid users is developed. The results achieved with the experiments performed reveal that a considerable number of users can have their privacy compromised when using Wi-Fi devices. Hence, an awareness list was created to help users take action to protect themselves from Wi-Fi security threats.

Wi-Fi networks enable people to live a connected lifestyle due to its flexibility of use and ease of access, but this permanent connectivity comes at a price: a lot of sensitive information is transmitted making use of Wi-Fi networks. Not surprisingly enough, the same features that make Wi-Fi access points desirable for consumers make them desirable for crackers. The use of attacks such as Karma and MITM exploit these features to make it easy to illicitly obtain information from a network and from devices. The usage of techniques and protocols such as SSL/TLS and HSTS is applicable to ensure the user security and privacy, but such is not always the case. Even so, vulnerabilities and bypassing methods are constantly discovered and demonstrated by researchers studies and proofs-of-concept.

Although some users employ the use of Wi-Fi audit applications, the scenarios and applications tested demonstrate how users are unsafe and unaware of security threats. Sometimes said applications even end up giving a false sense of security to the user. Consequently, it is important to note the importance of raising security awareness among users and enabling them to take the proper actions to mitigate the presented - and other - threats.

The current trends in device networking point to a future in which everything will be connected: cars, TVs, fridges, microwaves, and so on. Most of the devices that users routinely use will be linked to the Internet, making security compromises likely much more harmful than today. Thus, the continuous education of users on how to act upon possible threats and protect their privacy, making use of the crescent device connectivity safely must be accomplished.

To sum it all up, this research shows just how easy it is to compromise information from wireless technologies, in specific Wi-Fi networks, and how some of their exploitable

vulnerabilities work. Methods such as the use of audit applications are currently used but should not be the only decision factor on analyzing the safety of Wi-Fi networks. Therefore, the awareness among users must be achieved in order to create a safer environment - and a safer world - of connectivity.

REFERENCES

CALLEGATI, F.; CERRONI, W.; RAMILLI, M. Man-in-the-middle attack to the https protocol. **IEEE Security and Privacy**, IEEE Educational Activities Department, v. 7, n. 1, p. 78–81, 2009.

DURUMERIC, Z. et al. Analysis of the https certificate ecosystem. In: **Proceedings of the 2013 Conference on Internet Measurement Conference**. New York, NY, USA: ACM, 2013. (IMC '13), p. 291–304. ISBN 978-1-4503-1953-9. Available from Internet: <http://doi.acm.org/10.1145/2504730.2504755>.

EVANS, C.; PALMER, C.; SLEEVI, R. **Public key pinning extension for HTTP**. [S.l.], 2015.

HOUSLEY, R. et al. **Internet X. 509 public key infrastructure certificate and certificate revocation list (CRL) profile**. [S.l.], 2002.

MAO, W. **Modern cryptography: theory and practice**. [S.l.]: Prentice Hall Professional Technical Reference, 2003.

MCKINLEY, H. L. Ssl and tls: A beginners guide. **SANS Institute InfoSec Reading Room**, 2003. URL: https://www.sans.org/reading-room/whitepapers/protocols/ssl-tls-beginners-guide-1029.

SAPIEZYNSKI, P. et al. Tracking human mobility using wifi signals. **CoRR**, abs/1505.06311, 2015. Available from Internet: <http://arxiv.org/abs/1505.06311>.

SELVI, J. Bypassing http strict transport security. **Black Hat Europe**, 2014. URL: https://www.blackhat.com/docs/eu-14/materials/eu-14-Selvi-Bypassing-HTTP-Strict-Transport-Security-wp.pdf.

SLONE, J. P. **Local area network handbook**. [S.l.]: CRC Press, 1999.

STAMP, M. **Information security: principles and practice**. [S.l.]: John Wiley & Sons, 2011.

ZOVI, D. A. D.; MACAULAY, S. A. Attacking automatic wireless network selection. In: IEEE. **Proceedings from the Sixth Annual IEEE SMC Information Assurance Workshop**. [S.l.], 2005. p. 365–372.