Bachelor thesis
Computer Science

Radboud University

Evil Twin vulnerabilities in Wi-Fi networks

Author:
Matthias Ghering
s4395727

Supervisor/assessor:
Dr.ir. Erik Poll
erikpoll@cs.ru.nl

July 7, 2016

Abstract

Wi-Fi has become extremely popular. You find Wi-Fi networks almost everywhere. These networks can be divided into three types: open Wi-Fi networks also known as public networks, WPA2 personal networks also known as private networks and WPA2-Enterprise networks. With this research we will have a look at the security of these types of Wi-Fi networks, in particular their resistance against Evil Twin attacks.

An Evil Twin attack is an attack that uses a rogue access point (rogue AP) to impersonate another wireless network, in order to trick devices into connecting with the attacker.

We describe multiple Evil Twin attack approaches against public Wi-Fi networks to gain a Man-in-the-Middle position. These attacks are possible because users cannot authenticate public Wi-Fi networks. Most of these attacks also rely on the carelessness of the user. Simply verifying if the available networks should be available, could avert some of these attacks.

Unlike public networks, private networks offer mutual authentication through pre-shared keys, ensuring the user that the AP also knows the key without disclosing the key. An attacker can only attack private networks with an Evil Twin attack based on social engineering.

Social engineering tricks the user instead of his device. In this case the device is tricked to disconnect from the actual AP. The name of the rogue AP might trick the user into connecting with the rogue AP.

WPA2-Enterprise wireless networks such as Eduroam provide server authentication through certificates. However, most Android users do not install these certificates. This causes the device to accept any certificate. Making the authentication server's (AS) certificate practically useless.

The negligence of the user makes it possible for an attacker to set up an rogue AP and trick his victim's device into connecting with the rogue AP. Depending on the users inner authentication settings the user's traffic could be intercepted and his credentials might even be stolen.

In order to measure the susceptibility of Eduroam users to an Evil Twin attack, we created a survey to estimate how many users use vulnerable configured devices.

Acknowledgement

I would like to sincerely thank my thesis supervisor Erik Poll for his time spend on reviewing my drafts and his weekly encouragements to keep me writing.

I am also grateful to my parents, my uncle Jan and my aunt Regina for their support and feedback on my draft version.

I would also like to thank my fellow classmate Abdullah Rasool, who gave me an article about 802.1X in exchange for an acknowledgement in my thesis.

Contents

Chapter 1

Introduction

Wi-Fi enables us to easily connect to the internet without the hassle of cables or the costs of 3 or 4G. These are just some reasons why Wi-Fi became such a widely used technology. The popularity and availability of Wi-Fi makes it even more convenient. There are however some concerns about this wireless technology. Misconfiguration in settings of the client or the network makes the connection vulnerable to attacks. This possibly results in credential theft, leaking of private information, unauthorized use of your bandwidth and financial theft. [9] [8]

In order to investigate the security of wireless networks we must first distinguish between the types of Wi-Fi networks. We can divided them into three groups:

1. The first kind of networks are open Wi-Fi networks also known as public networks. Public networks rarely require authentication and are therefore often unprotected.

2. The second group consists of private networks used by households and small businesses. This kind of network uses a single pre-shared key to secure their authentication. These network use either WEP, WPA or WPA2 Personal.

3. Wireless Enterprise networks form the last group. These networks allows multiple users to authenticate with their own credentials. They are often used by organizations with a large user base, such as Eduroam, govroam, Ziggo WifiSpots. These networks use either WPA or WPA2 Enterprise.

Because each group can be implemented in various ways we decided to choose three case studies to represent each group. Public networks are represented as networks without any authentication. A WPA2-Personal network is used as an example of a private networks, since it is the most recent security

standard for private networks. The last case study investigates Eduroam a WPA2-Enterprise network with RADIUS servers as authentication servers.

We investigate how vulnerable these three kinds of networks are to Evil Twin attacks. An Evil Twin attack is an attack that uses an access point (AP) that pretends to be an already existing AP, hence the name"Evil Twin". These fake APs, also called rogue APs, are used to trick devices and users to connect to them. Connecting to a rogue AP could lead to a Man-in-the-Middle (MitM) reading and altering of your data. The attacker could in some cases even steal your network credentials when you connect to his network.

Chapter 2

The Eduroam architecture

Eduroam (education roaming) was created to enable users to access Wi-Fi services at all the participating educational institutes [2]. At the time of writing 12,000 locations in 76 countries [1] provide Eduroam and this number is still growing [2]. The Eduroam service is limited to academia but the architecture can easily be reused in other environments [21]. An example of such environment is Govroam, which is used by the Dutch government to enable officials to access the Internet (and local network depending on the user's privileges) at different ministries. Eduroam is a WPA2-Enterprise network that uses multiple RADIUS servers to authenticate the users. In the next sections we will explain which protocols are used in Eduroam, how they work and possible vulnerabilities.

2.1 Design Goals

Eduroam's architecture was designed to fulfill some design goals [21] listed below. This research will emphasize on two of the issues, namely security and privacy.

1. Unique identification of users at the edge of the network. Unique identification is needed to authorize access and identify the user in case of abuse.

2. Enable (trusted) guest use. This allows users from other institutes to access the network.

3. Scalable. The infrastructure is designed to allow large numbers of users and institutes.

4. Easy to install and use. If joining or the use of the infrastructure is complicated institutes will not adopt Eduroam.

5. Secure. Eduroam should be designed to prevent credential theft. Eduroam maintains a policy that specifies the minimal requirements. Eduroam's infrastructure should allow an institute to require additional security measures and requirements, without the need to modify the rest of the infrastructure.

6. Privacy preserving. The infrastructure should provide the possibility to hide the user's identity from any third parties, including other institutes. Eduroam protects the user's identity with an anonymous identity. This identity linked to the user's institute, but not to the user himself.

7. Standards based. The infrastructure should use open standards to allow institutes to freely choose the hardware they use.

2.2 Basis of architecture

Eduroam is a WPA2-Enterprise network. This means that it uses an access point (AP) to connect to the user's device (client) and an Authentication server (AS) to authenticate the client.

Now we have discussed the three parties involved (the AP, the client and the AS) we will give an overview on how these parties interact. WPA2-Enterprise uses 802.1X protocol to authenticate the user [10]. The layers of protocols used in 802.1X are shown in Figure 2.1

1. The client connects to the AP using 802.11. The client is then associated to the AP but not authenticated. This mean it is able to communicate with the AP but not with the rest of the network.

2. The client starts the EAPOL protocol. This protocol encapsulates the data between the client and the AP. EAPOL starts by requesting the client's outer identity also known as the anonymous identity. The client gives the user's anonymous identity if it is available otherwise he gives the user's identity.

3. The AP receives the client's identity removes the EAPOL encapsulation and encapsulates the data using RADIUS. RADIUS routes the traffic through the RADIUS hierarchy to the user's home AS.

4. This AS uses the outer authentication phase of the Extensible Authentication Protocol (EAP) to set up a secure TLS tunnel between the client and itself. This tunnel prevents others including the AP from eavesdropping on the inner authentication.

5. The AS authenticate the client using the inner authentication of EAP.

6. When the client is successfully authenticated he will be allowed into the network. The AP uses the four-way handshake of EAPOL to authenticate the client. This authentication is based on the key that the AS and the client computed after the client was successfully authenticated. The AS also provides the key to the AP, since this information is needed to perform the four-way handshake with the client.

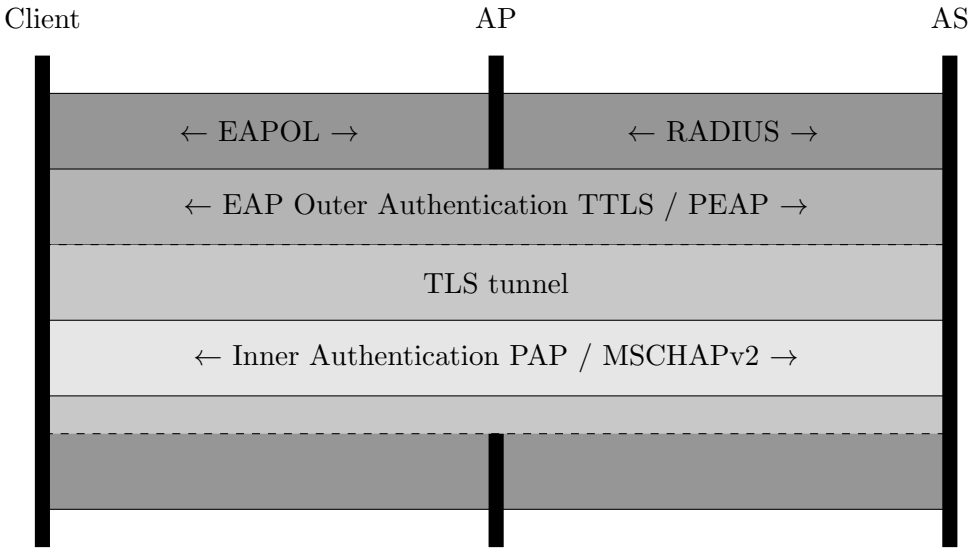

Figure 2.1: Eduroam's 802.1X protocol stack. The outer authentication method established a TLS tunnel in order to protect the inner authentication.

2.3 RADIUS hierarchy

Small WPA2-Enterprise networks often use a single RADIUS server as AS, but Eduroam uses a RADIUS hierarchy to authenticate the clients. Figure 2.3 shows the RADIUS hierarchy used in Eduroam. When a client requests access to the network he will send a message containing his identity (user@realm) such as "bob@institude.home" or an anonymous identity such as "anonymous@institude.home" to the AP. The AP forwards it to the local RADIUS authentication server (AS). The local AS in Figure 2.3 is *institute.visit*. The AS server checks if it is a local client by looking at the realm part of the identity. If the client is local the local AS will authenticate him.

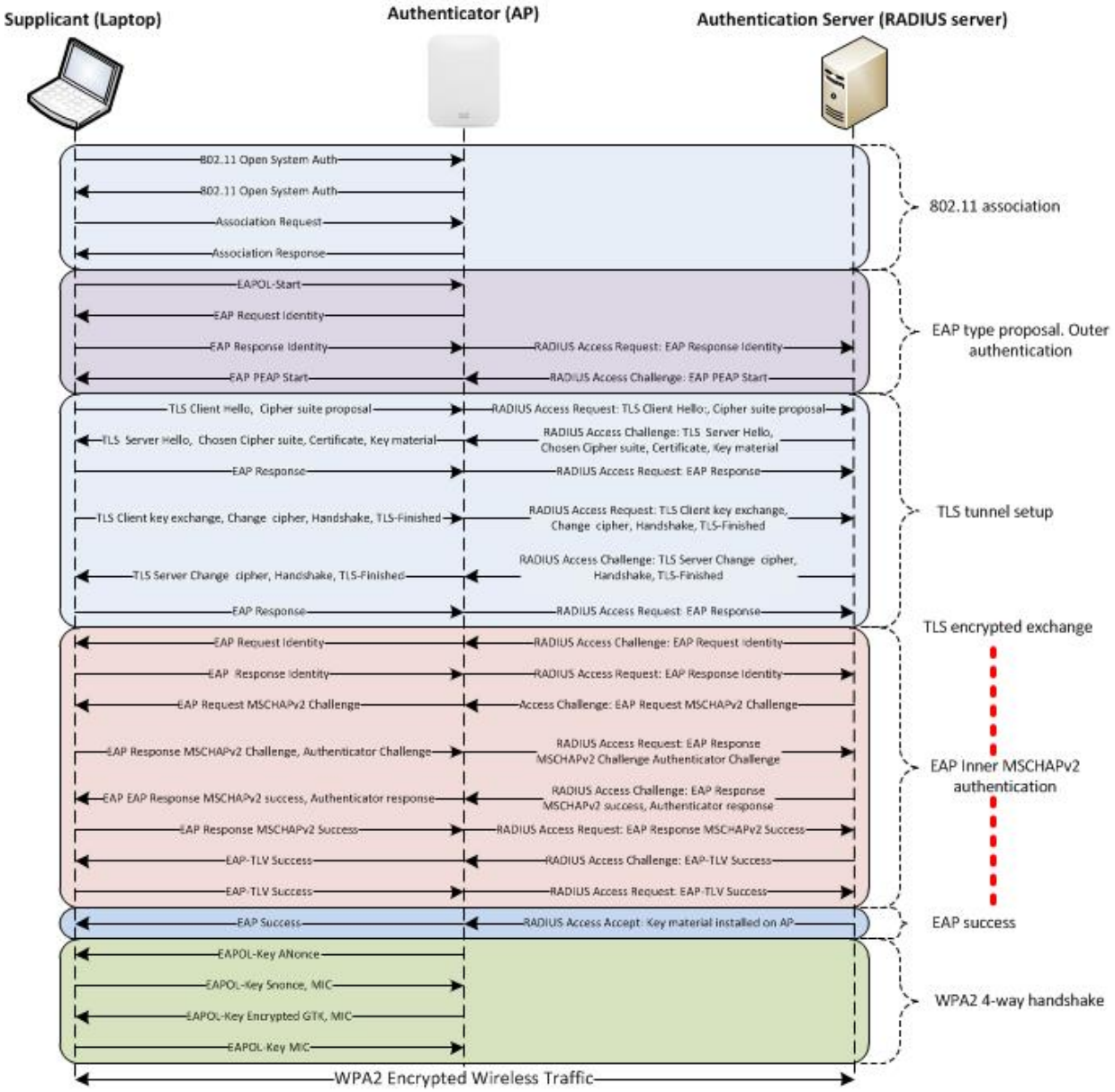

Figure 2.2: An abstract protocol overview [16]

If the client is not local, his request will be forwarded to the *.visit* RADIUS server. The *.visit* RADIUS server inspects the realm part of the identity; since it is not in a *.visit* subdomain, the clients request will be forwarded to the root RADIUS server. This RADIUS server again inspect the users identity and forwards the message to *.home* RADIUS server. This server verifies the client's realm and forwards it to the subdomain institute.home. The home AS (institude.home) receives the client's request, authenticates the client and sends to the *institude.visit* AS a success or fail message to indicate if the client is allowed to join.

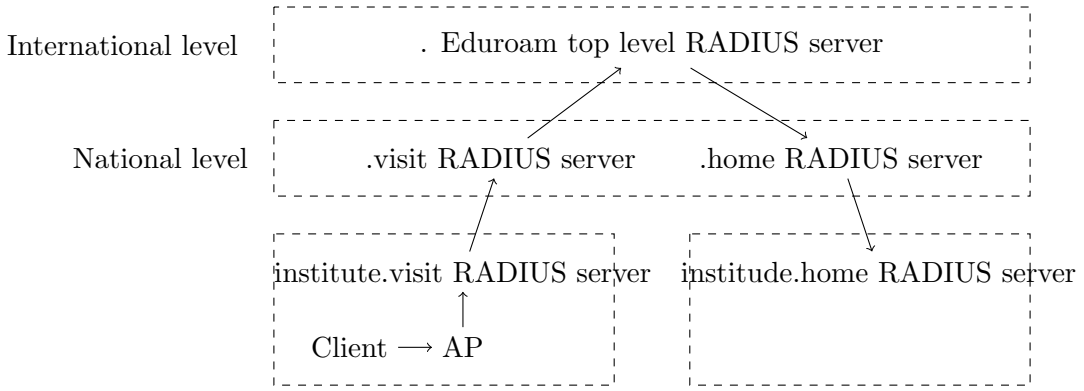

Figure 2.3: RADIUS hierarchy (sending the user's identity).

2.4 Outer authentication protocol

The outer authentication protocol is used to authenticate the AS to the client and create a secure TLS tunnel between the client and the client's home AS. This tunnel is used for the inner authentication. Eduroam uses either PEAP or TTLS as outer authentication protocol. The outer authentication consists of the following steps:

1. The client requests access with user@institution.tld or anonymous@institute.tld if the user configured an anonymous identity.
2. The AP forwards to the AS(Authentication Server). If it is necessary the AS will forward the message to the user's home AS.
3. The user's home AS checks the identity and sends a certificate to start the tunnel.
4. The client validates the certificate and they set up the TLS tunnel.

2.5 Inner authentication protocol

The inner authentication is used to authenticate the client to the AS. Eduroam uses PAP or MS-CHAPv2 as inner authentication protocol. The simplest inner authentication protocol is PAP (Password Authentication Protocol) [13]. This protocol sends the username and password in plain text to the authentication server.

1. The client sends an Authenticate-Request containing his credentials in plain text through the tunnel to his AS.

2. The AS checks the credentials and grants access using a success message or denies it by sending a failure message.

MS-CHAPv2 the successor of Micosofts Challenge-Handshake Authentication Protocol (MS-CHAP), unlike PAP, uses mutual authentication to authenticate both client and authentication server (AS).

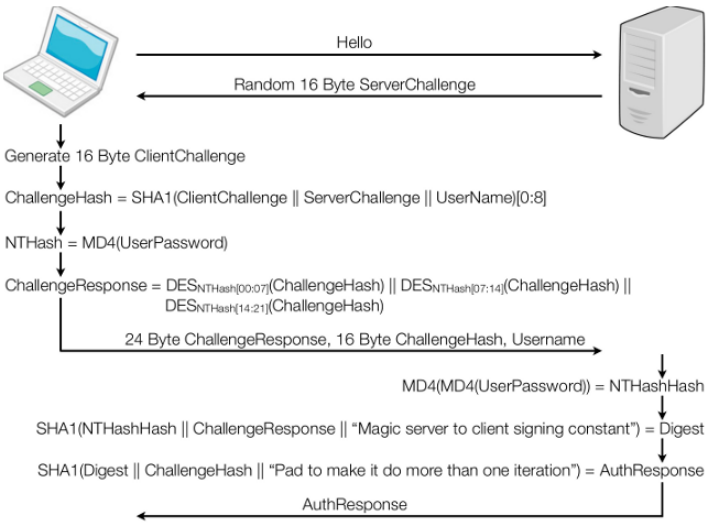

Figure 2.4: MS-CHAPv2-protocol [15]

Figure 2.4 shows the flow of MS-CHAPv2. This protocol might seem a bit "dazzling" as Marlinspike remarks at Defcon 20 [12], but the only variable that the attacker does not know is the MD4 hash of the user's password.

The MS-CHAPv2 protocol consists of the following steps:

1. The client starts the protocol with a "hello" message.

2. The authentication server anwers with a 16 Byte ServerChallenge.

3. The client combines the ServerChallenge, a 16 Byte random ClientChallenge, and the username to create the ChallengeHash.

4. The client then calculates the NTHash which is the MD4 hash of the user's password and pads this with zeros to 21 bytes.

5. Client splits the 21 Byte long NTHash in three 7 Byte long DES keys. A parity bit is added to each 7 bits to create a 8 Byte long DES key.

6. Each key is used to encrypt the ChallengeHash. The concatenation of these three ChallengeHashes forms the ChallengeResponse.

7. The 24 Byte ChallengeResponse, 16 Byte ChallengeHash and the UserName is send to the AS.

8. The AS has to answer with an AuthenticationResponse which uses the md4 hash of the md4 hash of the user's password combined with the ChallengeResponse sent by the user, two constant sentences and the ChallengeHash. This message is sent to verify that the server knows the password as well.

2.5.1 Weaknesses in MS-CHAPv2

There are a few weakness in the MS-CHAPv2 protocol which make it possible to break the authentication. The first weakness is in step 4, in the calculation of the NTHash. This hash is not salted which means that the attacker can reuse this hash. This practically makes the NTHash equal to the password. Not only can the attacker use the NTHash to authenticate as the user, but he can also use it to impersonate the AS and authenticate the user. Hashing the password without a salt also enables the attacker to use rainbow tables.

The second weakness is in the second part of step 4, when the client pads 5 bytes of zeros. The NTHash is then divided into 3 keys each 7 byte long. This means that the last 7 byte key is actually 2 unknown bytes and 5 bytes of zeros. This key with an effective entropy of $2^{16} = 65536$.

The easily computed part of the NTHash can be used to accelerate dictionary attacks with rainbow tables. Because the attacker only needs to check hashes ending on the computed part of the NTHash.

The last weakness is in step 6. This step uses each key to DES encrypt the ChallengeHash separately. This means that the ChallengeHash is copied three times and these copies are each encrypted once and then concatenated. This results in $E_{k1}(ChallengeHash)||E_{k2}(ChallengeHash)||E_{k3}(ChallengeHash)$. This scheme encrypts the ChallengeHash three times separately instead of using the three keys to encrypt the ChallegeHash using tripleDES.

The complexity of a brute force attack on nested encryptions such as triple DES $E_{k3}(E_{k2}(E_{k1}(ChallengeHash)))$ would result in $2^{56} * 2^{56} * 2^{16}$.

Because the complexity of a brute force attack on nested challenge encryption would be the product of the complexity of a brute force attack on each key.

However, the complexity of a brute force attack on separately encrypted challenges is equal to the sum of the complexity of a brute force attack on each key. This makes the complexity of a brute force attack $2^{56} + 2^{56} + 2^{16}$. The complexity is added instead of multiplied because each key can be guessed individually instead of having to guess all the keys at once.

Because the last key was padded, it can easily be found within a couple of seconds. This leaves only two 7 byte long keys. This gives a total complexity of $2^{56} + 2^{56} = 2^{57}$

Since both encryptions use the same ChallengeHash, it is possible to brute force them at the same time. This is possible by iterating through the key space and encrypting the challengeHash with the guessed key and then checking if the encrypted text matches one of the two encrypted texts. Since the DES encryption is a lot more expensive operation than checking if two strings match, it effectively becomes twice as cheap. That means it would have a total complexity of 2^{56} which is the same as a single DES encryption.

Brute forcing the DES key could accelerate with special hardware such as Field Programmable Gate Array (FPGA). FPGAs make it possible to program hardware logic gates (without the cost of custom hardware). These logic gates can be programmed to do a DES encryption in a single cycle [12].

2.6 Eduroam network of the Radboud university

This section contains specific information about the Eduroam network of the Radboud university.

The Radboud university uses PEAP\TTLS as outer authentication and advises to use MS-CHAPv2 as inner authentication [19]. The university also informs it's users that their certificate is signed by the AddTrustExternal Root.

When we verified the certificate of the Radboud University we discovered that the certificate presented by the AS depends on the location of the user. Figure 2.5 shows the certificates and their chain presented by the AS. The AS presents a certificate signed by AddTrust AB (AddTrustExternal Root) when the user connects to an AP located at the campus of the Radboud university, but presents a different certificate signed The UserTrust network (UTN-USERFirst-Hardware).

We do not know if the use of multiple certificates signed by different Root CAs is a common practice at other institutes, but there are no indications that it is required or advised [23] [24].

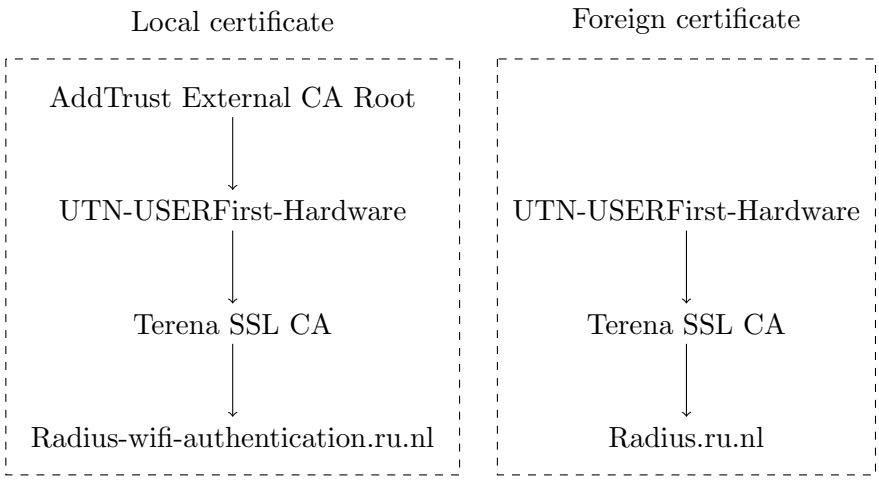

Figure 2.5: Certificate chain presented by the AS of the Radboud university

2.7 Problems

The designers of Eduroam already noticed some of the vulnerabilities to that are exploited in attack 3 in Section 3.5. They remark that their practical experience has shown that many users neglect to configure their devices in a privacy-preserving way or their devices do not support the proper configurations [21]. This remark probably refers to the anonymous identity setting that is an optional setting when connecting to Eduroam. If users do not configure the anonymous identity, the visited network and an attacker masquerading as such a network, are able to see the users identity.

The designers also voice their concerns about users not willing to invest time to inspect the server certificates or install a trusted certificate authority (CA) [21]. This could lead to users connecting with rogue APs, as will be discussed in Section 3.5.3.

Insecure bootstrapping is also one of the concerns. Insecure bootstrapping is simply accepting the incoming server certificate. This server certificate is installed and saved for the next time. If a user initiates with a rogue AP his device will automatically accept the false server certificate. This certificate will be installed and can be used later to connect to the rogue AP.

Insecure bootstrapping is similar to the connection process of IOS explained in Section 4.2.1. However, IOS prompts the user to accept the certificate instead of automatically accepting it as the developers suggest with insecure bootstrapping. The IOS user has the option to decline the certificate based on the information IOS provides about the certificate such

as the common name, the Root CA name, fingerprint and public key. This still presents some problems as previously mentioned users are not willing to invest time to inspect the certificates.

Chapter 3

Evil Twin attacks

We will discuss possible Evil Twin attacks on our three types of networks. We look at possible Evil Twin attacks on public networks in Section 3.3, we discus why private networks are less prone to Evil Twin attacks in Section 3.4 and we look at Evil Twin attacks on WPA2-Enterprise networks in Section 3.5. We also experimented with Attack 1 and Attack 3 to test how effective they are. The experiments and their results can be found in Chapter 4.

3.1 Possible impact of Evil Twin attacks

This section explains the different impacts an Evil Twin attack can have. The impact an attacker has depends on the type of network, the attack he uses and how successful his attack was. These impacts are divided into 3 categories. A successful Evil Twin attack result in an impact in one or more of the categories.

Every successful Evil Twin attack on any network gives the attacker a Man-in-the-Middle (MitM) position. This means that the attacker is able to Eavesdrop and alter the traffic to and from his victim (category A).

Stealing network credentials (category B) is only possible with private and enterprise networks, since public networks do not use network credentials.

An attacker can only identify the user (category C) if the network distinguishes between users. This is only possible with enterprise networks since these networks use a combination of username and password as network credentials. These usernames are needed to identify the user.

A Eavesdrop and alter traffic.

1. Steal credentials from services such as email, socialmedia, paypal, etc. This could be accomplished with tools such as `SSLStripper`.

The attacker could use these credentials to access his victim's accounts. The attacker can steal or alter information, use services, send messages on behalf of his victim, transfer money. The attacker might even sell these account credentials.

2. Collect personal information. This could be used to identify users and potentially blackmail them.
3. Access a client using vulnerabilities in the client's OS.

B Steal network credentials.
These credentials, often a combination of username and password, are used to authenticate the user to the AS. Using the stolen credentials an attacker can impersonate his victim and gain access to the network. This access can be abused to anonymously access the Internet and hide behind his victim identity.

The attacker can also use his victim's credentials to intentionally frame his victim. This would require the network to track the traffic of his users, in order to establish a link between the incriminating evidence and the victim.

In some cases network credentials are re-used for different purposes. For example, the Radboud university's Eduroam credentials are also used for Blackboard, Osiris, email, the schedule site (persoonlijkrooster.ru.nl), Radboud sportcentrum login and perhaps even more functions.

C Identify the user.
If the attacker is able to identify the user he can combine this information with the time and place in order to track the users movement (this can be enhanced with multiple rogue APs). An attacker could also use this capability as an automatic detonator, which may not be a huge risk to students or professor but some politicians using Govroam might see this as a risk.

D Estimating previous movements.
An attacker could eavesdrop on probes sent by the victim's device. These probes reveal the names of previously connected networks. The attacker can query online databases containing names and locations of wireless networks, in order to find the locations possibly visited by his victim. He could also inspect the names to determine if it might be a work or home location.

3.2 Prerequisites and requirements

In order to successfully execute any of the following Evil Twin attacks an attacker and his target must full fill the following requirements:

1. The target connects automatically to known networks.
2. The attacker knows the name of the network he wants to impersonate.
3. The attacker has an AP with a stronger signal than the legitimate AP. This requirement is only needed if a rogue AP impersonates a local AP.
4. The attacker has an Internet connection.

3.3 Attack 1 Evil Twin attack on public Wi-Fi networks

Our first attack concerns public networks. These networks are quite common. You can find them in hotel lobbies, restaurants, bars, public transport, airports and even some stores. Public networks, as the name might suggest, are publicly available and are often unprotected to ensure that visitors and customers can easily connect.
As explained earlier, an Evil Twin attack exploits the fact that the user cannot authenticate the AP. Since public networks don't have any authentication they form the perfect target for an evil twin attack. This section describes some possible Evil Twin attacks involving public networks.

The first attack is used to trick devices connected to public networks that are locally available. The second attack tricks devices or their users when there are no public networks available. The last attack is a complexer but more effective version of the second attack.

We describe an experiment with the second of these attacks, and the results of this experiment in Section 4.1.

3.3.1 Attack 1.1

The attacker needs to find a (popular) public network and create a rogue access point using the same SSID as the public network. This attack only requires the attacker to have a stronger signal than the legitimate access point. This is needed to convince target devices to automatically switch to the evil twin network.

3.3.2 Attack 1.2

Other attacks such as ARP cache poisoning are able to achieve the same results as an Evil Twin attack on a public network, but do not require a rogue AP. However, an advantage of an Evil Twin attacks is the possibility to attack a client without the need of a legitimate public network at the location of the client.

An attacker could use an Evil Twin attack based on social engineering to trick users into connecting with the rogue AP. The attacker simply names his rogue AP "free wifi" and waits until users connect.

But if the attacker happens to know the SSID of a popular public network used by his victims he could create an AP with the same SSID. This approach tricks the device into connecting with the rogue AP instead of tricking the user.

3.3.3 Attack 1.3 Karma

Wireless devices send probe requests to in order to discover Wi-Fi networks [11] [20]. These probe requests can contain a list of known Wi-Fi networks. A Karma attack uses the information from these probe request in combination with Attack 1.2 to create rogue AP that impersonate a network known to the client. This attack does require the client to send probe requests containing a list of known Wi-Fi networks.

The network names (SSIDs) obtained from the client can also be used to estimate the home address of the user. The attacker compares the list of SSIDs with a database of known SSIDs and their location. These databases can be manually build by driving around in an area and use a computer to automatically note the SSID and the GPS location. This method is known as Wardriving. Some of these databases are available online. An attacker could even use tools such as `snoopy` to automate this process [22].

3.3.4 Impact of attack 1

All of the mentioned attacks are used by the attacker to create a MitM position. The impact of Evil Twin attacks on public networks falls in category A described in Section 3.1. Attack 1.3 can also reveal information about the target movements category D described in Section 3.1.

3.3.5 Solutions

1. The owner of a public network could switch to a different system that allows the user to authenticate the network such as Passpoint. These systems need to distribute and install their certificates on the users devices. This might be a hassle for users to configure. The effort needed to configure the user's device could lead to complaining customers. Another downside is the price, since users or public network owners must pay for the services provided by systems such as Passpoint.

2. Users could use a Virtual Private Network (VPN) to create an encrypted tunnel to their home or work network. This network will forward the messages to their destination. This makes sure that even if the traffic is going through a rogue AP it cannot be read or changed

without the user knowing. The downside of VPN is the need for a VPN server. Users could set up their own server at home or buy a service online to provide the server. However, this would require some work to set up, some technical knowledge and money to buy the service or the server. VPN users should make sure to install the server certificate to authenticate the VPN server, otherwise it might be possible for attacker to use the weaknesses in MS-CHAPv2 to derive the user's password or NTHash Section 2.5.

3. Attack 1.2 and 1.3 can be stopped by simply forgetting the public networks if you do not use them anymore. It might surprise you how many you might have. I myself thought that I might have one or two but it turned out to be around thirty. And if you are keen on keeping these networks, you could always disable auto connect. Newer Android versions provide the option to disable auto connect for specific networks, while older versions of Android cannot disable auto connect. IOS does provide the option to disable auto connect, but unfortunately not for specific networks.

   Aside from operating system specific features you can always forget Wi-Fi networks or turn of your Wi-Fi when you're not using it.

4. Attack 1.2 and 1.3 could also be limited by paying attention to which networks you connect. Not using any free Wi-Fi would be a good protection against attack 1.2. Checking if the automatically connected network should be near you is a great way to limit attack 3. You might for example see that you are connected to "Riverwalk Tampa Hotel". If you are not in Tampa Florida US you should probably disconnect. Perhaps you could even notify a security officer or system administrator if you happen to be in an office building.

5. If you are using a public network without a VPN you should always verify if the certificates of the websites are in order and your connection uses https instead of http. This could be a sign that an active attacker is using SSLstripper. This does not work for sites that do not support https. This is of course not a real solution to the problem but might still be helpful. Since most people without a VPN still use public networks if they need Internet access without 3g or 4g.

3.4 Attack 2 Evil Twin attack on private Wi-Fi networks

This section looks at possible Evil Twin attacks on a private network (WPA2-Personal). We did not experiment with private networks because the Evil

Twin attacks we present rely on social engineering or inefficient dictionary attacks.

Private networks are often used in households or small companies. These private networks use pre-shared keys to encrypted their communication. Private networks offer three security access protocols WEP, WPA and WPA2. The encryption of WEP is broken [8] and should not be used anymore. WPA was introduced as an alternative to WEP. It offers more security features, including IV sequencing was enforced (to prevent replay attacks), packet tampering detection and mutual authentication based on shared passphrase [8]. WPA2 also has these features, but uses better encryption than WPA.

However, WPA2 still suffers from dictionary attacks since users often configure easy passwords. The first thing an attacker has to do is to eavesdrop the four-way handshake between his victim's device (client) and the AP. He can then use this information to verify if the passwords form the dictionary match the Message Integrity Check (MIC) from the handshake.

Some researchers from The University of Central Florida [17] remarked that you would need to eavesdrop the four-way handshake of a valid client and the AP. The attacker might not have this handshake, since it requires the attacker to be present when a client successfully connects to the network. In order to still be able to break the password they pretended to be a client trying to authenticate using dictionary passwords. This is known as an online dictionary attack and can be negated by using a maximum numbers of tries before ignoring the client for some time.

Omar Nakhila C.S. presented an attack resembling an Evil Twin attack. This attack uses a single computer pretending to be multiple clients that try to connect to the AP [17]. If this fake client exceeded the maximum number of tries he will be replaced by a new fake client using a different MAC address. This attack looks similar to an Evil Twin attack since it masquerades as someone else. However, it isn't a real Evil Twin attack since it is not a rogue AP masquerading as a known AP in order to let users connect, but simply uses his masquerading ability to pretend to be a fresh client.

There is another attack on private network that is similar to Attack 1.1 in Section 3.3.1 (masquerading as a public network near the actual network). However, this attack needs to rely on social engineering, since it is unable to trick the client into automatically connect to the rogue AP, because the client is able to distinguishes between the networks since one is a private network and the other is a public network. However, it is possible to send deauthentication messages (masquerading as the real AP) to the client to block him from the network. The user notices that his connection is lost and might connect to the rogue AP (public version of his network).

This attack provides the attacker with a Man in the Middle position. The attacker could use this position to send the user a captive web portal.

This could be a false page that notifies the user about updates for the router, prompting the user to fill in the password.

3.4.1 Impact of attack 2

If the attacker does not know the network password, he can only use rogue AP to mimic a public version of the targeted network. If the user connects to the rogue AP, it provides the attacker with a MitM position (category A) in Section 3.1.

This section also describes an social engineering attack using the MitM position and an offline dictionary attack presented by Omar Nakhila C.S. to retrieve the network password (category B) in Section 3.1.

If the attacker knows the network password, he is able to use rogue AP to impersonate the private network. The rogue AP uses the password to complete the mutual authentication with the client. This allows the client to automatically connect to the impersonated network. Resulting in a MitM position without the need for social engineering (category A).

3.5 Attack 3 Evil Twin attack on enterprise Wi-Fi networks

This is an Evil twin attack on a WPA2-Enterprise network that authenticates clients using a RADIUS server as authentication server (AS). As a particular example we looked at the Eduroam network of the Radboud university. This network can be seen as a representative example of an Eduroam network since it accepts and advises the most secure configurations. A lot of other institutes advise the same configurations for their Eduroam network as seen in table 4.4. Table 4.4 is one of the results from the experiments in Chapter 4 regarding Attack 3.

Section 2.7 mentions security and privacy problems with Eduroam. Most of these problems concern insecure configurations caused by laziness, a lack of knowledge or prioritizing reliability over security. Attack 3 abuses these insecure configurations to identify the user, steal network credentials and gain a MitM position.

3.5.1 Prerequisites and requirements

In addition to the mentioned requirements in Section 3.2

1. The target did not configure a certificate.
2. A fake RADIUS server.

3. Computing power and or specialized hardware to brute force two 8 byte DES keys (with a total of entropy 2^{57}).

4. An Internet connection.

3.5.2 Attack flow

This section gives an overview of the attack. If you would like to read a more detailed description of the eduroam flow read Chapter 2.

1. The client connects to the rogue AP.

2. The client sends his anonymous identity or his regular identity if he does not have an anonymous identity.

3. Instead of forwarding to the RADIUS server corresponding to the users identity, the rogue AP pretends to be the RADIUS server and creates a TLS tunnel using his self-signed certificates.

4. The client accepts the self-signed certificate since there is no certificate configured. This step concludes the outer authentication.

   (a) The client will then start the inner authentication to authenticate himself to the fake RADIUS. If the client uses PAP it will simply send his credentials in plaintext.

   (b) The rogue AP will log his credentials, send him a success message and forward the clients traffic to the Internet. The attacker now possesses the clients credentials and a MitM position.

   (a) If the client uses MS-CHAPv2, the rogue AP will start the inner authentication by sending an arbitrary challenge string.

   (b) The client will respond with his username in plaintext, an arbitrary peer challenge string and his response on the server challenge. For the details on MS-CHAPv2 see Section 2.5

   (c) The evil twin logs the clients response (combined with his own challenge). Not knowing the users password or MD4 hash the evil twin is forced to send an eap-success message. Android systems before 5.0 are fooled by an eap-success message [9] resulting in a MitM position for the attacker, but new versions will end the communication.

   (d) The attacker could use specialized hardware and weaknesses in MS-CHAPv2 explained in Section 2.5.1 to find the MD4 hash of the users password. The attacker can use this hash to send a valid response to the clients challenge next time the client connects to the evil twin.

It is also possible to find the password with a dictionary attack. Computing the last part of the NTHash allows the attacker to accelerate the process. Because the attacker only needs to check the rainbow tables or dictionary entries with a NTHash that matches the computed last part of the NTHash.

3.5.3 Impact of attack 3

IOS by default does not use an anonymous identity. It is possible that this can be configured using Wi-Fi profiles but this would require either the institute to give such a profile to its users or the users to make one for themselves.

The anonymous identity is also optional on Android. Configuring the anonymous identity only requires the users to fill in the anonymous identity given by the institute. An example of an anonymous identity could be *anonymous@institute.nl*.

If the user does not configure the anonymous identity, the attacker is able to identify the user even if he has a certificate installed category C in Section 3.1.

If Android users do not install the proper certificate, the attacker is able to identify the user (category C) even if the user installed a anonymous identity. The other impacts of this attack depend on the users configurations and the attackers capabilities. IOS users should not be affected by these attacks since IOS promps the user to accept the certificate chain presented by the AS. However, if the user does not pay attention and simply accepts the certificate it would give the attacker the same possibilities as on users without a certificate.

If the user uses PAP as inner authentication it would allow the attacker to easily see his credentials (category B). PAP also does not require the attacker to authenticate himself. The attacker can complete the inner authentication and acquire a MitM position (category A). This does not affect the majority of IOS users since IOS selects MS-CHAPv2 as inner authentication by default.

If the user configured MS-CHAPv2 it is still possible to derive the users NTHash if the attacker has enough computing power. However, this is not possible to do on the fly unless the attacker is able to quickly derive the NTHash. This is possible with a remote server with enough processing power, but unlikely. Knowing the NTHash enables the attacker to successfully complete the inner authentication and acquire a MitM position (category A).

The attacker could also try to find the users password using rainbow tables and dictionary attacks (category B). These attacks only work on weak or common passwords.

In addition to breaking the password or the NTHash it is also possible to trick Android systems before 5.0 by sending an eap-success message [9] resulting in a MitM position for the attacker (category A).

Android users can only install the CA root [23]. This means that they can still be tricked by a rogue AP using a valid certificate of the same CA root. This gives the attacker the same capabilities as on a user that installed the CA root as on user that did not install a certificate. However, it does require the attacker to have a valid certificate of the same CA root. Since most certificates are linked to a persons identity it would make it easier to identify the attacker when his rogue AP is detected.

IOS requires the user to accept the server certificate and uses by default PEAP, MS-CHAPv2 and no anonymous identity. This makes it impossible for the attacker to trick the client into connecting with him. An attacker can only see the user's identity (category C).

3.5.4 Solutions

1. The first thing the user should do is to verify if he use PEAP/TTLS in combination with MS-CHAPv2.

2. Using strong (preferably random) passwords to counter dictionary attacks and rainbow tables.

3. Configure an anonymous identity to avoid being identified by attackers, this however is still possible using the MAC address of the client.

4. Install the server certificate to authenticate the AS in order to reject attackers that use certificates with the same CA root. This is not possible for some operating systems such as Android. The institute can mitigate this vulnerability by using a self signed CA root specifically used to authenticate the AS. The attacker cannot legitimately obtain a certificate from this CA, and therefore cannot successfully impersonate the network.

5. An institute could use usb Wi-Fi adapters and terminal computers to create (cheap) rogue AP scanners. These scanners could send and warning message to security or the system administrator that a rogue AP has been detected [7].

Chapter 4

Experiments with attacks

This chapter summarizes the result of our experiments with Attack 1 and Attack 3 used to measure the susceptibility of users against Evil twin attacks presented earlier.

We used the same setup to try out attack 1 and attack 3. This setup consists of Ubuntu 14.04 LTE ASUS K53s laptop, wireless usb adapter, `Hostapd-WPE 2.5` [14] [6], `Dnsmasq` and `Iptables`. We use `Hostapd-WPE` to create a rogue AP in order to impersonate our target network. `Dnsmasq` is used as DHCP server which will assign the target with an IP-address. The secondary network adapter is used to provide Internet access to the laptop. Iptables is used to forward the victim's traffic.

We initially used a Raspberry Pi running Rasbian, but it turned out to be quiet slow in use and not very mobile since it requires a constant power source and external screen. The Raspberry Pi might be more suitable for automatic measuring, since it is inconspicuous and stationary.

We used a wireless usb adapter (802.11n) instead of the build in wireless adapter of the ASUS K53s laptop, since the wireless usb adapter's drivers were compatible with `Hostapd-WPE`.

4.1 Experiments with attack 1

To verify that mobile devices can be tricked by rogue APs impersonating a known public network Section 3.3.2 we created our own rogue AP. We choose multiple SSIDs of public networks from the list of known networks on our target device. In an environment without these public networks available we started our rogue AP using `Hostapd-WPE` to impersonate a public network with the given SSID. We then verified if the device a Sony Xperia Z1 with Android 5.1.1 automatically connected. We repeated this experiment with five different SSIDs.

4.1.1 Results of Experiment 1

At first this experiment performed as expected, connecting immediately with the rogue AP. But some of the SSIDs did not automatically connect. We estimated the time when they were connected for the last time and found that all of them were not recently connected.

When we looked at other recently connected SSIDs (approximately a year ago) some of them had the option "automatically connect" disabled. We expect that it is probably related to the time that the network was not used since this feature is not disabled with newer networks. However, this might just be a coincidence since our test set is limited and we do not know the exact date when the device was connected to these SSIDs for the last time.

4.2 Experiments with attack 3

We use the output of `Hostapd-WPE` to view the actions made by the AP such as connecting and verifying users. `Wireshark` is used to capture the traffic between the victim and rogue AP. This traffic reveals among other things if the victim uses an anonymous identity.

4.2.1 Experiment 3.1 determining the susceptibility of Eduroam users

This experiment resembles the experiment described in "A Practical Investigation of Identity Theft Vulnerabilities in Eduroam" [9]. The goal of this experiment is to determine the number of Eduroam users effected by attack 3 as a result of inadequate device configurations.

Pretest

We configured our `Hostapd-WPE` configuration file to use the SSID "eduroam" to create our rogue AP. We tested the attack in an environment without an actual Eduroam network available to ensure Eduroam would not interfere with our rogue AP and our rogue AP would not bother any Eduroam users. We then gathered 10 volunteers whose devices were checked with our setup. The volunteers were asked to answer (under guidance of a researcher) the questions of the survey.

Based on the results and experience of the pretest we concluded that the automatic set-up was not suitable to test multiple volunteers. A couple of factors were taken into account when making this decision; the first was the total time needed to take a single survey including delays such as setting up the equipment. The automatic testing needed to be set up every time a volunteer agreed to participate. This caused a lot of unwanted overhead since a

power source was required, the computer needed to start and `Hostapd-WPE` and `Wireshark` needed to be activated. In addition to the set-up time it often took some time to automatically connect the client to the rogue AP.

The second factor was our test environment which was not possible to create on the campus since Eduroam is present on most locations. If we used the automatic testing we would not be as mobile since our set-up requires a power source.

However, the survey itself was not a problem. Most of the questions required the user to check the settings on their device. This might differ a bit for each OS (version), but this was only an issue with IOS, since IOS selects the options by default and does not show which options were chosen [3] [4]. This is also the reason why only the IOS version, Wi-Fi usage, student or staff and institute is filled in the actual survey.

Survey

The survey was created in order to gather statistical data on the Eduroam configurations of users. This data should enable us to estimate the susceptibility of the users.

Our survey asks about device specifications (Model and OS), if users often enable their Wi-Fi, which inner and outer authentication protocols they selected, if they installed an certificate and if so which certificate, whether they used an anonymous identity, if they are a student or an employee and at which institute.

We conducted the largest part of the survey at Radboud university Nijmegen. This method is technically called cluster sampling, since this location provides us with many possible samples [18]. We asked Eduroam users on the Radboud campus to look at their Eduroam settings and answer the previously mentioned questions.

Results of Experiment 3.1

The results show that none of the Android users installed a certificate. The results also show that a lot of Android users did not select an inner authentication. This however, is not a problem since the pretest showed that Android automatically choses MS-CHAPv2 when PEAP is selected without an inner authentication. It also shows that none of the Android users used an anonymous identity.

IOS, in contrast to Android, automatically chooses most settings as seen in Figure 4.1 unless the users specifically configures otherwise. This automatic bootstrapping of IOS use the certificate provided by the AS which must be verified by the user, PEAP as outer authentication, MS-CHAPv2 as inner authentication and no anonymous identity [3] [4]. An overview of

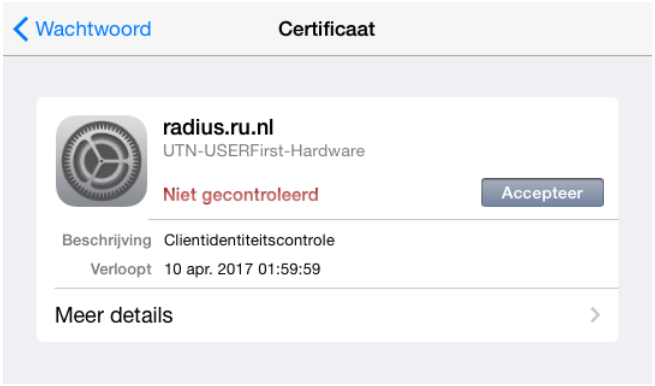

Figure 4.1: IOS bootstrap prompt

Table 4.1: The results of the survey

OS	Certificate installed	PAP selected	Anonymous identity used	Total share
Android	0%	3%	0%	84%
IOS	100%	0%	0%	16%

the survey can be found in Table 4.1. The exact data collected in the survey and the pretests can be found in Table 4.2 and Table 4.3.

During the pretest we noticed that one of the IOS users almost immediately accepted the certificate even though we did not ask him to install it. We could not observe this behavior during the survey, since we did not uses the automated testing.

Table 4.2: Raw data obtained by the pretest

Model & OS	Wi-Fi usage	Outer authentication	Inner authentication	Anonymous identity used	Certificate installed	Institute
Huawei, A5.0.1	Yes	PEAP	MSCHAPv2	No	No	RU
Iphone 5s, IOS 8.1.3	Yes	PEAP	MSCHAPv2	No	Yes	RU
Samsung s3 neo, A4.4.2	Yes	PEAP	MSCHAPv2	No	No	HAN
Iphone 6, IOS 9.3.1	Yes	PEAP	MSCHAPv2	No	Yes	RU
Motorola, A6.0.1	Yes	PEAP	MSCHAPv2	No	No	HAN
Samsung tap 3, A4.4.2	Selective	PEAP	MSCHAPv2	No	No	HAN
Iphone 5s, IOS 9.3.1	Yes	PEAP	MSCHAPv2	No	Yes	HAN
Sony xperia, A4.1.2	Selective	PEAP	MSCHAPv2	No	No	RU
Samsung, A5.0	Selective	PEAP	MSCHAPv2	No	No	HAN
Samsung Galaxy s6, A6.0.1	Yes	TTLS	MSCHAPv2	No	No	RU
Xiaomi mi4 lte, A4.4.4	Yes	PEAP	MSCHAPv2	No	No	RU

Table 4.3: Raw data obtained by the survey

Model & OS	Wi-Fi usage	Outer authentication	Inner authentication	Anonymous identity used	Certificate installed	Institute
Honor Holly, A4.4.2	Selective	PEAP	MSCHAPv2	No	No	RU
Motorola g3, A6.0.1	Yes	PEAP	None	No	No	RU
Iphone 6, IOS 9.3.1	Yes	-	-	-	-	RU
1+1, A6.0.1	Selective	PEAP	None	No	No	RU
1+2, A5.1.1	Yes	PEAP	None	No	No	RU
Sony xperia z5, A6.0	Yes	PEAP	None	No	No	RU
Nexus 5x, A6.0.1	Yes	PEAP	MSCHAPv2	No	No	RU
LG g3, A5.0	Yes	PEAP	None	No	No	RU
1+1, A6.0.1	Selective	PEAP	MSCHAPv2	No	No	RU
Huawei p8 lite, A5.0.1	Yes	PEAP	None	No	No	RU
1+1, A6.0.1	Selective	PEAP	None	No	No	RU
LG Magna, A5.0.1	Yes	PEAP	None	No	No	RU
Iphone 5c, IOS 9.3.2	Yes	-	-	-	-	RU
Jiayu, A4.2.1	Yes	?	?	?	?	RU
Galaxy s6, A6.0.1	Selective	PEAP	None	No	No	RU
Galaxy s6 edge, A6.0.1	Yes	PEAP	MSCHAPv2	No	No	RU
1+1, A6.0.1	No	PEAP	MSCHAPv2	No	No	RU
Motorola g3, A6.0	Selective	PEAP	None	No	No	RU
Samsung s3 mini, A4.2.2	Selective	PEAP	MSCHAPv2	No	No	RU
Nexus 5x, A6.0.1	Selective	PEAP	None	No	No	RU
Moto E, A5.1	Selective	TTLS	MSCHAPv2	No	No	RU
Htc one m8s, A5.0.2	Selective	PEAP	MSCHAPv2	No	No	RU
Samsung s3, A4.3	Selective	PEAP	None	No	No	RU
Iphone 5s,IOS 9.2.1	Yes	-	-	-	-	RU
LG g5, A6.0	Yes	PEAP	None	No	No	RU
Samsung s3 mini, A4.1.2	Yes	PEAP	None	No	No	RU
Iphone 5, IOS 8.0.2	Yes	-	-	-	-	RU
Moto E, A5.0.2	Yes	PEAP	MSCHAPv2	No	No	RU
Galaxy s2 plus, A4.2.2	Yes	PEAP	MSCHAPv2	No	No	RU
HTC one m7, A5.0.2	Yes	PEAP	MSCHAPv2	No	No	RU
Galaxy s2 plus, A4.2.2	Yes	PEAP	MSCHAPv2	No	No	RU
Galaxy s4, A5.0.1	Yes	PEAP	None	No	No	RU
Galaxy s2, A4.2.2	Yes	TTLS	PAP	No	No	VU

4.2.2 Experiment 3.2 using a CA signed certificate

As already mentioned in the pretest Experiment 3.1 in Section 4.2.1. IOS and Android devices use different methods to install certificates. IOS asks the user to install the certificate presented by the AS. While Android users need to ask the administrator which certificate they should install. In case of the Radboud university and many others this is the Root CA.

This experiment will look at the effect of installing a root CA certificate instead of a server certificate on a client device. To view these differences we first create our own self signed CA because we unfortunately were unable to use the same Root CA as the Radboud (which according to their settings is the AddTrustExternal Root [19]).

We then used our CA to create two valid certificates one "real AS" and the other "fake AS". Note that "fake AS" is still a valid certificate but is used to impersonate the AS. Set up the AP using the SSID "Edu" a for our device unknown network and the "real AS" certificate. Keep in mind that normally the AP and AS are not the same device, but in the case of our rogue AP they are combined into one program. The program is also used to create the "real" AP

1. We started the experiment by installing the Root CA on our Android device and connecting to the "real" AP. We verify that the device connects and is set to automatically connect to the "edu" network.

2. We then disconnected the device by stopping the "real" AP.

3. When the "real" AP was deactivated, we started our rogue AP using the "fake AS" certificate. This step indicated if the Android device noticed the difference between a AP using the "real AS" certificate and a rogue AP using the "fake AS" certificate.

The next part of the experiment was to repeat the previous steps but with an IOS device. But instead of installing the Root CA we let the OS automatically install the "real AS" server certificate by connecting it to the "real" AP. After connecting to the AP we moved on to step 2 and so on.

Results of Experiment 3.2

The experiment shows that Android devices cannot distinguish between ASs using different server certificates from the same Root CA when the Root CA is installed. This makes sense since the Root CA was the only certificate installed. This leaves the rest of the chain including the server certificate unknown. This problem has less effect on IOS users since IOS devices automatically install the server certificate after the user has verified the certificate. A watchful user would notice that the common name of the attacker's certificate is different from the name specified by the institute.

Table 4.4: Institute settings

Institute	Outer authentication	Inner authentication	Anonymous identity	Certificate advised on site	Direct link to certificate
University of Twente	TTLS	PAP	No	No	N/A
University of Amsterdam	TTLS	PAP	No	No	N/A
Erasmus University Rotterdam	PEAP	MSCHAPv2	No	Yes	No
Universiteit Leiden	TTLS	PAP	Yes	No	N/A
Universiteit Utrecht	PEAP	MSCHAPv2 & None	No	Yes & No	No
Universiteit van Tilburg	PEAP,TTLS	MSCHAPv2, EAP-MD5	Yes	Yes	No
TU Delft	PEAP, TTLS	PEAP-MSCHAPv2, TTLS-PAP	Yes & No	Yes & No	Yes
Radboud Universiteit Nijmegen	PEAP	MSCHAPv2	No*	No*	N/A*
HAN University of Applied Sciences	PEAP	MSCHAPv2	No	Nokia Only	Yes
Wageningen Universiteit	PEAP	MSCHAPv2	ID	No	N/A

Legend:

"&" means their is contradicting information.

"," means that their are multiple options given.

"ID" means they use the regular identity as anonymous identity

"*" means that the advice has been changed while working on this thesis. The latest version was used in this experiment.

4.2.3 Experiment 3.3

The users configuration dependent on the configurations advised by the users home institute. We looked at the website of the institutes and read the recommended Android configurations in order to verify if institutes advice the most secure configurations.

4.2.4 Results of Experiment 3.3

Of the 10 institutes we looked at we saw that three of them advised PAP and one institute mentions it as an option. Some of these institutes even have automatic configuration files that sets the inner authentication to PAP.

It also shows that only 3 institutes advise the user to install an anonymous identity, although one of them is not consistent with this advice.

Half of them at least mention certificates even though some are not consistent in their advice. We also found that two of the tested institutes allow the user to directly download their Root CA certificate.

Chapter 5

Future work

We investigates the most important aspects Evil Twin attacks. However, there are still some questions that we leave unanswered, but are worth pursuing.

We briefly mention the possibilities of using programs such as CAT and SecureW2 to install certificates and configure the client device. An in depth study on these programs could reveal defects and possible improvements.

The bootstrap of IOS lets the user decide if the presented certificate is correct. It is possible that the user accepts the wrong certificate. It could be interesting to see how susceptible users are to false certificates and perhaps find a way to improve the user's decision making.

Section 2.5.1 describes weaknesses of MS-CHAPv2. Section 2.5.1 also mentions the use of special hardware such as FPGAs to expedite the retrieval of the DES key. It might be interesting to look at possibilities to accelerate the breaking of DES or improve the software running on the hardware.

If an attacker is able to break DES he can break the NTHash and create a MitM position, but he does not get the password of his victim. It might be interesting to look in to the strength of MD4, how susceptible passwords are to dictionary attacks and if it is possible to use hardware such as FPGA to accelerate MD4 hashing.

The Evil Twin attack relies on the client to automatically connect to the rogue AP. It might be interesting to determine the decision making behavior of operating systems when they discover multiple known networks in order to improve Evil Twin attacks or to create new policies that prevent Evil Twin attacks.

Chapter 6

Conclusion

We studied Evil Twin attacks on public, private and WPA2-Enterprise Wi-Fi networks. Subsequently we chose to conducted experiments with Attack1 and Attack3 described in Section 3.3 and Section 3.5 as those are the major threats.

Attack1 uses a rogue AP to masquerade as another public network in order to convince, devices that were previously connected to the original public network, to connect to the rogue AP. Attack3 uses a rogue AP to masquerades as a WPA2-Enterprise to convince device without the proper certificates to connect to the rogue AP.

The results of the experiments with Attack1 show that it is in many cases easy to trick devices into automatically connecting to a rogue AP. The results also indicate that devices could implement a policy to decrease the susceptibility of this attack, such as disabling auto connect with networks after being disconnected for a certain period of time. However, it cannot be concluded that all devices implement these features since the sample size of this experiment was limited, but they do show that Attack1 is still a possible attack.

The survey results of experiment 3.1 show that none of the tested Android users installed a certificate. This is probably caused by laziness, a lack of knowledge or prioritizing reliability over security. These causes were not only mentioned by the developers of Eduroam [21], but also indicated during the survey. While the survey did not ask why the subject chose not to install the root certificate, some subjects mentioned that it seemed like a hassle or that they did not know how to install it.

Since these devices are unable to authenticate the server without the correct certificate, they are all vulnerable to Attack 3 described in Section 3.5. Fortunately, most of them used MS-CHAPv2 as inner authentication which provides some protection. However, MS-CHAPv2 has vulnerabilities that

reduce processing power needed to retrieve the password hash. This hash allows the attacker to complete the Evil Twin attack and secure a MitM position.

Special hardware used by Moxie Marlinspike is able to derive the password hash from the authentication communication in less than a day [12].

The survey and experiments show that IOS users are automatically prompted to install the certificate provided by the AS making it mandatory to access the network. The IOS also uses PEAP and MS-CHAPv2 by default.

IOS just as Android does not required an anonymous identity. Configuring an anonymous identity is even harder when using an IOS device since it does not give the option when a user manually joins the network. It is possible to configure an anonymous identity using a Wi-Fi profile, but it requires either the institute or the user to make one. If users do not use an anonymous identity, attackers will be able to see the users' username even if they have a certificate. This information can be used by the attacker to identify the users and track their movement.

Experiment 3.2 shows that it is essential to verify the common name of the server certificate, because only verifying the Root CA would allow attackers with a valid certificate of the same Root CA to impersonate the AS.

Even though it is a lot more likely for an attacker to acquire a certificate of a commercial Root CA then a certificate signed by the institute's self signed Root CA, it does not mean that institutes are wrong to choose to use a commercial Root CA. Institutes need to be aware of the advantages and disadvantages of both commercial Root CAs and self signed Root CAs, in order to make an informed decision. More information of the Root CA considerations can be found in Section 7.2.

Chapter 7

Recommendation

Our research gave us insight in the vulnerabilities of enterprise networks such as Eduroam. This chapter presents our recommendations to protect against Evil Twin attacks. These recommendations are directed at operating systems, institutes and users.

7.1 Recommendation for operating systems

Android requires the users to find the proper root CA, download it, install it and choose it for the desired network. This process is a big hassle and as the results show, few users take the time to install the root CA.

The bootstrapping method of IOS on the other hand installs the complete certificate chain and is incredibly user-friendly. However, the ease of use allows careless users to install the attacker's certificate without noticing. We do not know how inclined users are to accept the attacker's certificate, but knowing that none of the survey subjects installed a root CA shows the shortcomings of Android's usability.

If Android does not want to implement bootstrapping similar to IOS, we strongly advice to allow users to specify the common name of the server certificate, in order to prevent attackers with a valid certificate of the same root CA to impersonate the AS.

We advice OS' that use automatic bootstrapping such as IOS, to provide more user-friendly ways to verify the Root CA. IOS shows a not verified warning, but this only means that the user did not install this certificate before. We would advice IOS to use their list of known Root CAs to verify the Root CA provided by the AS. This allows the user to only verify the common name of the server certificate.

An improvement for both Android as IOS would be to use the same identity and anonymous identity format for all the institutes such as *user@institute.tld*

and *anonymous@institute.tld*. A uniform format allows the operating system to use a default anonymous identity based on the identity provided by the user.

A long term solution would be to develop a new version of MS-CHAP to patch the vulnerabilities or introduce a new credential based inner authentication for PEAP and TTLS. This would of course take a long time to develop and to be universally accepted.

7.2 Recommendations for institutes

These recommendations aimed at institutes that provide Eduroam using TTLS or PEAP as outer authentication.

Our results show that most users did not use the most secure configurations. We advice institutes to educate their users about the proper configurations. They should advice users to use MS-CHAPv2 instead of PAP, use an anonymous identity, and install the proper certificate and perhaps even a Wi-Fi profile for IOS users. Institutes should also provide the needed information such as the Root CA, Root CA fingerprint, common name of the server certificate and manuals to configure settings and verify certificates.

During our research we noticed that the Radboud university has two server certificates. One is signed by AddTrust and the other is signed by UserTrust. If a Radboud user connects to the Eduroam at the Radboud univeristy, they receive a certificate signed by AddTrust, but if a Radboud user connects to Eduroam at a different institute, they receive a different certificate signed by UserTrust.

IOS supports multiple certificates of multiple Root CAs, but Android users can only configure a single Root CA. Using multiple Root CA's makes securely connecting to Eduroam inconvenient for Android users. These users need to reinstall the Root CA every time they move from their home institute to a different institute and visa versa. It might be possible to use an external application such as CAT in order to accept multiple Root CA, but we have not verified this.

Institutes have to choose between two kinds of Root CAs to sign their server certificate, commercial CAs or self signed CA.

Commercial CAs make sure that their root certificates are installed in devices and browsers. This makes it slightly easier for Android users to acquire the Root CA and for IOS users to verify the presented Root CA.

Unfortunately Android device only verifies the Root CA. This allows an attacker with a valid certificate of the same Root CA to impersonate the

network. Institutes can prevent this attack by using a certificate signed by a self signed Root CA. This self signed Root CA should only be used to sign the AS certificates, in order to limit the chances of an attacker acquiring a valid certificate.

A drawback of this approach is that operating systems such as Windows and Ubuntu, that have a pre-installed list of trusted Root CAs, need to download and install the self signed Root CA. This only needs to be done once, but might require some knowledge or clear instruction.

A downside of using a self signed Root CA for Android users is the caution icon displayed on top of their screen. This icon indicates that the user installed a untrusted Root CA. This warning is displayed as long as the user does not uninstall the Root CA. There exist a few workarounds to prevent this warning, but we have not tested them.

One of these workarounds is the Eduroam's CAT app. There are claims that if the user installs the certificate through the app will not trigger a warning signal [5].

Automatic Wi-Fi configurations apps such as Eduroam's CAT app or secureW2 should allow the user to easily configure their Wi-Fi settings specified by their institute. If institutes provide these apps, we would recommend to verify if the apps are working properly before advising users to install them. During our research we found indication that the secureW2 app advised by the Radboud university does not configure the root CA and the anonymous identity.

We strongly advice institutes to consider purchasing an Evil Twin detection system [7]. These systems do not directly prevent rogue AP or Evil Twin attacks, but they detect the presence of an rogue AP and alert security or an administrator to remove it. The effectiveness of this security measure depends on the coverage of the detection system and the actions taken by the staff.

7.3 Recommendations for users

We advice Android users to install the proper certificates. Users can manually install the Root CA or use an application to configure their Wi-Fi settings if the user's institute provides an app. We also strongly advice Android users to configure MS-CHAPv2 instead of PAP and use strong passwords. This makes it harder for attackers with limited resources to successfully complete the authenticate to the client and become a MitM.

We advice IOS users to verify if the presented certificate is the correct certificate. They need to verify the common name of the server certificate, the common name of the Root CA and the fingerprint of the Root CA. If

the user's institute uses a self signed certificate, the user has to check the "fingerprint" as well.

We recommend users to use an anonymous identity. Since IOS users cannot manually install an anonymous identity, we would advice them to ask the institute for a Wi-Fi profile. These profiles should be able to enable the use of an anonymous identity.

Chapter 8

Terminology and abbreviations

AP Access Point or Wireless Access Point (WAP). This device allows wireless client connect to the network.

AS Authentication server. This server host by the user's home institute authenticates the user and decides if he is allowed in the network.

Client The client is the user's wireless device.

FPGA Field-Programmable Gate Array. FPGAs have configurable logic gates. This feature can be used to rapidly perform a specific set of computations.

Institute In this thesis institute refers to an organization that provides the Eduroam network. An institute is both a Service Provider (SP) and an identity provider (idP). The idP of a user is his home institute while the SP of the user is the visited institute.

Rogue AP This is an access point used by an attacker to impersonate another network.

MS-CHAPv2 Microsoft Challenge Handshake Authentication Protocol version 2. An Authentication Protocol that uses a challenge response to mutually authenticate the client and the AS.

PAP Password Authentication Protocol. A simple Authentication Protocol that sends plain text credentials to authenticate the client to the AS.

PEAP Protected Extensible Authentication Protocol. PEAP is as outer authentication protocol to authenticate the AS to the client.

SSID Service Set Identifier. SSID is the name of the wireless network. SSID is often used instead of the less known ESSID (Extended Service Set Identifier).

TTLS Tunneled Transport Layer Security. TTLS also known as EAP-TTLS (Extensible Authentication Protocol TTLS) is as outer authentication protocol to authenticate the AS to the client.

MitM Man-in-the-Middle. A MitM is an attacker who intercepts and forwards his victim's communication. The attacker is able to read and possibly alter this communication.

Table 8.1: The URLs used in Experiment 3.3 [accessed 6-June-2016]

Institute	URL
University of Twente	https://www.utwente.nl/icts/handleidingen/mobile_devices/android/eduroam_android_nlV2.1/
University of Amsterdam	https://www.kariliq.nl/openbsd/eduroam-uva.html
	https://cloud.securew2.com/public/40655/uva-wireless/
Erasmus University Rotterdam	http://www.eur.nl/campus_faciliteiten/campus/wireless_plug_in_access/
Universiteit Leiden	http://www.issc.leidenuniv.nl/wireless-access/handleidingen-wireless-access.html
	http://media.leidenuniv.nl/legacy/eduroam-androidclient-ics.pdf
Maastricht University	https://cloud.securew2.com/public/13114/eduroam/?device=Android
	https://www.maastrichtuniversity.nl/nl/support/ict-voorzieningen/handleidingen/wifi-en-netwerk/snelstart-eduroam
	https://kb.icts.maastrichtuniversity.nl/display/ISM/ICTS+Servicedesk+Manuals#ICTSServicedeskManuals-CreateaWiFiconnection
Universiteit Utrecht	http://students.uu.nl/wifi-eduroam
	http://students.uu.nl/files/uuitnlwifi150303eduroam-instellen-in-android-samsung-s4pdf
Universiteit van Tilburg	https://www.tilburguniversity.edu/nl/studenten/it/wireless/
	http://drcwww.uvt.nl/its/voorlichting/handleidingen/wireless/Wireless-Windows-7.pdf
TU Delft	https://www.tilburguniversity.edu/nl/studenten/it/wireless/mobile/
	https://intranet.tudelft.nl/services/fmvgict-pdc/netwerk/draadloos-netwerk/eduroam-draadloos-netwerk/
	https://intranet.tudelft.nl/fileadmin/Files/medewerkersportal/ict/Help/Handleidingen/Medewerkers/Draadloos_netwerk/Eduroam/doc/Android_Marshmallow_Eduroam_V2_Different_Warning_Message.pdf
	https://intranet.tudelft.nl/fileadmin/Files/medewerkersportal/ict/Help/Handleidingen/Eduroam/Android_Marshmallow_Eduroam_V31.pdf
	http://servicepunten.tudelft.nl/tuvisitor/files/tudelft/tud-eduroam-android_peap.pdf
	https://intranet.tudelft.nl/services/fmvgict-pdc/netwerk/draadloos-netwerk/eduroam-draadloos-netwerk/eduroam-draadloos-netwerk/
Radboud Universiteit Nijmegen	http://www.ru.nl/isc/studenten/wifi/handmatig-instellen/
	https://cloud.securew2.com/public/01747/eduroam/?device=Android
HAN	http://www.han.nl/start/corporate/contact/draadloos-netwerk/
	https://hanaccount.han.nl/wifi/
Wageningen Universiteit	http://www.wageningenur.nl/en/Expertise-Services/Facilities/Mobile-Device-Support/You-work-or-study-at-an-Eduroam-participant-and-you-bring-your-own-laptop.htm
	https://www.wageningenur.nl/upload_mm/4/2/8/bde90c87-9bcd-42ed-9d9a-9c1aae4c26a7_Connecting%20to%20WUR%20wireless%20network%20with%20Android%20device_UK_v1.2.pdf

Bibliography

[1] www.eduroam.nl. [Online; accessed 30-May-2016].

[2] Why eduroam? www.eduroam.org. [Online; accessed 30-May-2016].

[3] 802.1x authentication. http://training.apple.com/pdf/WP_8021X_Authentication.pdf, 2016. [Online; accessed 27-June-2012].

[4] Does the iphone work with eduroam? www.eduroam.org/faqs/, 2016. [Online; accessed 27-June-2016].

[5] Issue 82036: Self-signed certificates cause "network may be monitored by third party" warning. code.google.com/p/android/issues/detail?id=82036, 2016. [Online; accessed 28-June-2016].

[6] Brad Antoniewicz. hostapd-wpe (wireless pwnage edition). www.github.com/OpenSecurityResearch/hostapd-wpe. [Online; accessed 14-April-2016].

[7] Paramvir Bahl, Ranveer Chandra, Jitendra Padhye, Lenin Ravindranath, Manpreet Singh, Alec Wolman, and Brian Zill. Enhancing the security of corporate Wi-Fi networks using DAIR. In *Proceedings of the 4th international conference on Mobile systems, applications and services*, pages 1–14. ACM, 2006.

[8] Hal Berghel and Jacob Uecker. Wifi attack vectors. *Commun. ACM*, 48(8):21–28, August 2005.

[9] Sebastian Brenza, Andre Pawlowski, and Christina Pöpper. A practical investigation of identity theft vulnerabilities in eduroam. In *Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, page 14. ACM, 2015.

[10] Jyh-Cheng Chen, Ming-Chia Jiang, and Yi-wen Liu. Wireless LAN security and IEEE 802.11 i. *Wireless Communications, IEEE*, 12(1):27–36, 2005.

[11] Dino A Dai Zovi and Shane A Macaulay. Attacking automatic wireless network selection. In *Proceedings from the Sixth Annual IEEE SMC Information Assurance Workshop*, pages 365–372. IEEE, 2005.

[12] David Hulton, Moxie Marlinspike, and Marsh Ray. Defeating PPTP VPNs and WPA2 Enterprise with MS-CHAPv2. `www.defcon.org/html/links/dc-archives/dc-20-archive.html`, 2012. [Online; accessed 24-May-2016].

[13] B. Lloyd and W. Simpson. PPP authentication protocols. `www.tools.ietf.org/html/rfc1334`, 2016. [Online; accessed 6-June-2016].

[14] Jouni Malinen. hostapd: IEEE 802.11 AP, IEEE 802.1X/WPA/WPA2/EAP/RADIUS Authenticator. `www.w1.fi/hostapd/`. [Online; accessed 14-April-2016].

[15] Moxie Marlinspike. Mschapv2 protocol figure. `www.cloudcracker.com/blog/2012/07/29/cracking-ms-chap-v2/andhttp://www.livehacking.com/tag/ms-chapv2/`, 2016. [Online; accessed 24-May-2016].

[16] Cisco Meraki. Configuring RADIUS authentication with WPA2-Enterprise. `documentation.meraki.com/MR/Encryption_and_Authentication/Configuring_RADIUS_Authentication_with_WPA2-Enterprise`, 2016. [Online; accessed 14-May-2016].

[17] Omar Nakhila, Afraa Attiah, Yier Jinz, and Cliff Zoux. Parallel active dictionary attack on WPA2-PSK Wi-Fi networks. In *Military Communications Conference, MILCOM 2015-2015 IEEE*, pages 665–670. IEEE, 2015.

[18] Briony J Oates. *Researching information systems and computing*. Sage, 2012.

[19] Radboud ICT Servicecentrum. Handmatig instellen. `www.ru.nl/isc/studenten/wifi/handmatig-instellen`. [Online; accessed 19-may-2016].

[20] Dominic White and Ian de Villiers. Defeating PPTP VPNs and WPA2 Enterprise with MS-CHAPv2. `www.defcon.org/html/links/dc-archives/dc-22-archive.html`, 2014. [Online; accessed 28-June-2016].

[21] K. Wierenga, S. Winter, and T. Wolniewicz. The eduroam architecture for network roaming. `www.tools.ietf.org/html/rfc7593`. [Online; accessed 30-May-2016].

[22] Glenn Wilkinson. Digital terrestrial tracking: The future of surveillance. defcon-22, 2014. [Online; accessed 28-June-2016].

[23] Stefan Winter and Jákó András. Eap server certificate considerations. wiki.geant.org/display/H2eduroam/EAP+Server+Certificate+considerations, 2016. [Online; accessed 28-June-2016].

[24] Stefan Winter and Laura Durnford. How to deploy eduroam on-site or on campus. https://wiki.geant.org/display/H2eduroam/How+to+deploy+eduroam+on-site+or+on+campus, 2013. [Online; accessed 5-July-2016].