METDS - A Self-contained, Context-Based Detection System for Evil Twin Access Points

Christian Szongott[1], Michael Brenner[1], and Matthew Smith[2]

[1] Distributed Computing & Security Group, Leibniz Universität Hannover, Germany
`{szongott,brenner}@dcsec.uni-hannover.de`
[2] Department of Computer Science,
Rheinische Friedrich-Wilhelms-Universität Bonn, Germany
`smith@cs.uni-bonn.de`

Abstract. Mobile Evil Twin attacks stem from the missing authentication of open WiFi access points. Attackers can trick users into connecting to their malicious networks and thereby gain the capability to mount further attacks. Although some recognition and prevention techniques have been proposed, they have been impractical and thus have not seen any adoption. To quantify the scale of the threat of evil twin attacks we performed a field study with 92 participants to collect their WiFi usage patterns. With this data we show how many of our participants are potentially open to the evil twin attack. We also used the data to develop and optimize a context-based recognition algorithm, that can help mitigate such attacks. While it cannot prevent the attacks entirely it gives users the chance to detect them, raises the amount of effort for the attacker to execute such attacks and also significantly reduces the amount of vulnerable users which can be targeted by a single attack. Using simulations on real-world data, we evaluate our proposed recognition system and measure the impact on both users and attackers. Unlike most other approaches to counter evil twin attacks our system can be deployed autonomously and does not require any infrastructure changes and offers the full benefit of the system to early adopters.

Keywords: Mobile device security, Evil twin access points, Attack detection, 802.11

1 Introduction

The growing power as well as the proliferation of mobile devices is changing the way we use the Internet. While a decade ago wired Internet was the norm, now wireless Internet is used on a daily basis. Coffee shops, fast food restaurants, mobile carriers, public transport, and many other entities offer access points for both free and paywalled Internet access. While the risks of open access points (i.e. non-WPA encrypted) have received a fair amount of coverage, they are still in widespread use and the proliferation of open access points seems to be on the rise. If users connect to these open access points their devices usually store the network identifiers (SSID) and will automatically connect to all future networks with the same name. Critically this can happen silently without the user even noticing it. When a device is in range of a known access point and requires a connection to the Internet, it will associate with the access point automatically.

Unfortunately the SSID on which this mechanism is based is completely unprotected and access points do not need to authenticate themselves to the user. This opens up the door to a very easy and effective attack, in which the attacker spoofs the name of open access points to capture users devices. This kind of attack is called the evil twin attack [11].

Once the attacker has devices connected to the evil twin access point, there are diverse ways to attack them. For instance with a man-in-the-middle attack an attacker can eavesdrop and modify all unencrypted information sent and received by the user's device. Recent research has shown that weaknesses in SSL encryption of mobile devices are wide spread [2] (and are not being fixed by the developers [3]) and thus encrypted connections are threatened as well. Another possible attack is the distribution of platform specific malware by injecting malicious content into web pages or emails.

Even users who try to stem the dangers of public and unencrypted WiFi networks by activating a VPN before doing sensitive tasks are still at risk. In the study we present in this paper, we found out that more than 78% of all connections to open access points are established automatically by the device without any user interaction. Thus, users often do not even have the chance to activate their VPN to protect themselves. Unfortunately VPN-based security measures are not widespread for private smartphone usage as well.

In order to mitigate the threats from these evil twin access points we developed a Mobile Evil Twin Detection System (METDS). METDS is a self-contained and context-based detection system, which uses as much environmental data of smartphones as possible during the association process to help decide, if the access point is legitimate or the user needs to be warned of a potential attack. To quantify the scale of the threat of evil twin attacks we performed a field study with 92 participants over a period of $2,5$ months to collect their WiFi usage patterns. We evaluated the data, implemented a prototypical detection system and optimized the relevant parameters through multiple simulations on this real-world data. With our results we show that the approach of current mobile operating systems, only checking the SSID of access points is not sufficient and leaves users at risk of being attacked. Additional environmental data such as the current location of the user, nearby wireless networks and other information is needed to mitigate this threat.

The rest of the paper is organized as follows: In section 2 related work in the field of evil twin detection is presented. In section 3 we describe different types of attacks in an evil twin scenario and how they can be detected with the help of network information and sensors of current mobile devices. Section 4 presents the user study we performed to gain real-world WiFi usage data. In section 5 we describe our proposed detection algorithm, which is evaluated and optimized with the help of simulations in section 6. Section 7 concludes the paper and gives an overview of possible future work in this research area.

2 Related Work

Surprisingly not much research has been done in the area of of evil twin detection and the protection of users against this kind of attacks. Bauer et al. [1] showed in their work how users can be tricked into associating with evil twin access points with common SSIDs and present a detection strategy that is based solely on the SSIDs of nearby access points. The authors did not evaluate their algorithms and methods with real-world data and we will show in our work that their approach is insufficient to protect against evil twin attacks. In [4] Gonzales et al. extended their work by additionally verifying the signal strength of access points. Again no proper evaluation with real world data was conducted and our experiments show that the signal strength fluctuates too much to improve the detection accuracy.

Roth et al. [8] propose an authentication method for access points that uses light color sequences, displayed to the user on devices that have to be mounted near access points. Kindberg et al. [5] also propose a system that uses physical evidence and include an adaption of the Interlock protocol into the wireless association process. They also use public displays near the access point for their authentication and key agreement protocol. These solutions have two major disadvantages. First the access point has to be visible to the users. This is quite often not the case and it is also doubtful that users would be willing to invest the effort to go and look in any case. The second disadvantage of these solutions is the requirement for additional hardware and configuration. Many providers operate thousands of public access points. Most of them would shy away from the costs for hardware and its setup for each single access point.

Song et al. [9] follow two different protection approaches. The first one uses a network sniffer to detect interpacket arrival times (IAT) and compares them to previously learned statistics of the access point. The learning process is needed since the signal strength and the saturation of the wireless network directly influences the IAT. Another approach additionally involves the IAT between the client and a remote server. Since their system needs to be trained and can not detect evil twin access points before user data is being transmitted it is not feasible for mobile devices which constantly exchange sensitive data in many different locations. Also the lack of APIs to build a network sniffer and battery consumption issues make this solution suboptimal for mobile devices. In the approach of Mónica et al. [7] watermarked packets are sent out. It constantly scans other WiFi channels to detect the packet and thereby recognize an evil twin. Similar to above this method has severe drawbacks for smartphones and similar devices. The approach of Lanze et al. [6] uses clock skew to passively detect faked access points through device fingerprinting. Unfortunately their approach also requires additional infrastructure and does not raise alarms if access points are replaced entirely, since their system is focused on attacks, that spoof the BSSID of legitimate access points.

3 Types of mobile evil twin attacks

There are several ways an attacker can use evil twin attacks to gain access to the users' device. In the following we describe such attacks and discuss which technical skills are necessary to carry them out. We also show how these attacks can be circumvented by using our Mobile Evil Twin Detection System (METDS). In the following scenarios the attacker's goal always is to fake a legitimate access point as unobtrusive as possible. Due to the lack of information provided by common mobile operating systems, the user will not be able to distinguish between a legitimate and a malicious access point.

3.1 Faking an access point's SSID (Type A)

This first attack scenario (called type A) only requires a very basic setup. The attacker sets up an access point with a commonly used SSID like *BTOpenzone* or *tmobile* and waits for users who previously connected to these networks to come along. This approach can be optimized significantly by obtaining SSIDs from probe request frames. Many mobile device send out probe request frames to find their favorite networks as fast as possible. These probe requests contain the SSID the device is searching for. An attacker can easily receive them with common WiFi sniffer software and thus tune the attack to the current devices in the neighborhood. Since mobile operating systems only verify the SSID to check, if the present network is known, the attack is not limited to a specific location. Mobile devices will connect to this kind of access point no matter where it is located.

While this trivial attack is currently very effective, it is also easy to counter. METDS saves the MAC-address (BSSID) of the connected access point. During subsequent connections the BSSID will be verified and simply faking the SSID will throw a warning.

3.2 Faking an access point's BSSID (Type B)

Assuming BSSID checks become common attackers can step up their game to circumvent these checks, since they can also fake the BSSID. Just like spoofing the SSID, spoofing a BSSID can be done with freely available tools and it does not need any additional technical skills. We call that type of attack type B. However, this attack is already more limited than attack type A. If there are two Starbucks WiFi users who have connected to different Starbucks access points, they can no longer be targeted with the same attack, since the BSSID needs to match the exact access point. However, targeted attacks are still possible.

To further degrade the attackers capability, METDS will employ additional environmental parameters to detect evil twin access points. METDS collects the surrounding visible access points and saves SSIDs, BSSIDs and supported encryption schemes and uses a combination of measures to deal with the noisy environment - as will be described in section 5. With this countermeasure in place the attacker is limited to mounting the evil twin attack at the same location as the original access point or faking at least part of the original network environment as will be described in the next subsection.

3.3 Faking a network environment (Type C)

An attacker who is aware of context-based detection could try to simulate the whole network environment. Therefore the attacker might take a snapshot of the target wireless network environment and imitates them with additional equipment. For instance, modified versions of the open source router software DD-WRT[3] and its virtual interfaces could be used in combination with the appropriate hardware to achieve this setup. This form of attack is possible and cannot always be prevented by METDS, however since additional equipment, technical skills and prior preparations are necessary for this kind of attack, the effort is much higher than before.

To counter this attack METDS additionally takes the device's location into account when connecting to a access point. In case the distance between the current and the saved one of the access point exceeds a specific threshold, the METDS raises a warning and interrupts the connection process.[4]

3.4 Faking the entire environment (Type D)

The final stage of the cat and mouse game would force the attacker to fake the entire environment (including cell towers) or execute the attack very close to the legitimate access point.

Mobile devices connect to the access point with the highest signal strength. If the attacker installs a malicious access point that has a higher signal strength than the legitimate one, the user's devices will automatically connect to his access point. In this scenario the attacker has to hazard the consequences of operating two access points that open up the same network.

Attacks of this type can not be detected by the METDS. However, the attacker is forced to execute a much more targeted and sophisticated attack than it is currently necessary. The attacker also runs the risk of the legitimate access point provider detecting the attack.

---

[3] http://www.dd-wrt.com

[4] This is not a foolproof method, since GPS location is not always available and WiFi based positioning can be fooled by an attack of type C. However cell tower ID based positioning works in many cases and raises the bar for the attacker.

4 Field Study

To gain an overview over WiFi connections in the everyday life of smartphone users and how vulnerable they are to potential evil twin attacks, we conducted a field study. To the best of our knowledge this is the first study to gather real-world data on this type of attack potential. We recruited 92 participants through a study mailing list and social media. They installed an Android App on their own smartphones. The app collected their WiFi usage statistics for up to 74 days with an average participation time of 46.75 days. 83 of our participants additionally completed a questionnaire with demographic questions and provided details about their network usage habits. The Android app includes a background service that records data about each connection to a WiFi network. More than 220,000 connections to WiFi networks were recorded together with additional data about the network environments and other meta data.

4.1 Survey

To conclude the study and to participate in a competition, the participants had to fill out a questionnaire. The participants were at the age of 18 to 56 with an average of 28.24 years. 13 out of the 83 participants are female. With 55% most of the participants were students followed by 32% full-time employees. The remaining 13% consist of pupils, part-time workers, self-employed and unemployed persons.

60 participants stated that they use their smartphones only for private use. 22 participants use their smartphone for private use as well as professional and only one participant stated that he uses his smartphone solely for his job. Half of our participants have an IT-related background.

We also asked the participants how they assess their own IT-related knowledge. We asked them how often they ask friends and how often they are asked by friends in case of IT-related problems. The combined result show that the vast majority of the participants believe to be well grounded in IT-related topics. This means our participants are skewed towards the more tech-savvy end of the population, which to a certain extent is to be expected since we could only recruit Android users. For the purpose of this study we believe this is not much of a problem, since we had enough non-tech-savvy users to study both types of participants and we found no difference between them.

4.2 Connection statistics

We gathered data from 223,877 connections that were established to access points during the study. As a starting point, we analyzed how big the threat of open WiFi access points is in the wild. As shown in prior research work[10] the automated connection establishment to previously configured networks is an increasingly serious threat for today's smartphone users. Therefore we collected anonymous statistics about all configured networks on the participants devices. In figure 1 one can see the amount of configured wireless networks per user.

They are differentiated by unencrypted networks like public and open access points and encrypted networks, that use encryption schemes like WPA2-PSK or WPA-Enterprise.

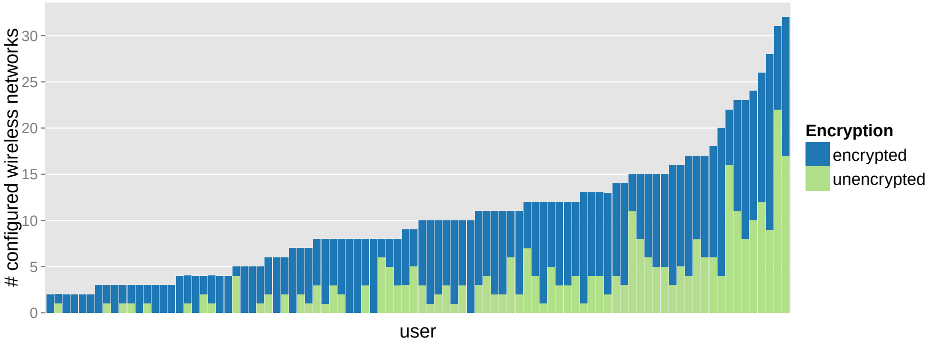

Fig. 1. Number of configured WiFi networks on participants devices, divided into unencrypted (green) and encrypted (blue) networks

One can see that the participants on average have more than 10 configured networks. On the right hand side of the diagram power users can be found having more than 20 and up to 32 different configured networks. From the view point of evil twin attack it is critical that more than 75% of the participants have at least one but up to 22 open WiFis configured.

Besides the number of configured networks we also collected SSIDs and encryption schemes of each configured network. Since we assume that potentially dangerous WiFi networks will have SSIDs of commonly-used public access points we analyzed which of these networks have been configured on most of the devices. We also compared the participant's believed open access point usage patterns to their actual configuration. Less than 4% of the participants that filled out the questionnaire stated, that they use open access points more than once in a month. However more than 49% stated that they use open access points rarely and the remaining 47% of the participants do not use them at all.

In total we gathered data about 239 unencrypted WiFi networks from all of the 92 participant's devices. Only two of the participants stated they use open access points on a daily basis and one participant declared he uses open access points several times a month. These participants might be aware that configured unencrypted networks on their devices and that their smartphones will automatically connect to them. Of more interest are the participants who reported that they never use open WiFi access points, but have several configured on their devices. 37 participants stated not to use any open access points, but only 37.8% of these participants did not have a single one configured on their device. The remaining 62.2% had at least one, but up to 20 different configured open access points. With an average of 3.65 they do not have a significantly lower number of configured open WiFis than the users that stated a daily usage.

From the remaining 41 participants that stated that they are using open access points rarely, only 5 had none configured. In this group the participants had an average of 4.89 configured open access points.

Another interesting piece of data collected for our proposed detection system is the device's location during the connection establishment. During the study we collected 171.532 locations which corresponds to about 79% of all recorded connections. To determine the current position of the device we primarily use the location API of the Google Play Services if they are available on the device. The fallback to this method is the native Android location API, allowing the manual determination of the current location by cell tower triangulation and GPS. Although a GPS location has a higher accuracy the amount of time needed to obtain it is much higher than the cell tower triangulation. In our study most of the participants had Google Play Services enabled on their devices. Thereby 99.4% of all locations have been determined through its API, which also seemed to be the most stable and reliable source for our purpose. The average accuracy of the collected locations was 115.06m, which also seems to be sufficient for our algorithm.

As discussed above the METDS needs more environmental parameters to serve its purpose. Thus, we focused our analysis on WiFi-related environment parameters. As a first step we collected information about each WiFi network that is in communication range during the connection process to a specific network. To utilize this data for the METDS we did not only collect the SSIDs of nearby networks, but also their BSSIDs, the signal strengths, encryption schemes and the frequencies on which they operate. In our study we could observe on average 10.26 nearby WiFi networks during a connection.

To demonstrate the dangers of automatic re-connections to allegedly known networks we also investigated how often smartphones connect to WiFi networks without any kind of user interaction. For this we chose one of the major hotspot providers, T-Mobile. In our study 8 different users established a total of 476 connections to open access points of this provider. 374 of these connections have been established without any user interaction. We can rule out user interactions, since we additionally gathered data of the device's display status and lock state during the connection. Since the display was switched off we assume, that more than 78% of these connections have been initiated by the device itself without the user even noticing it.

To gain a more detailed overview of the distribution of configured open access points we analyzed the gathered SSIDs. From a total of 238 unencrypted configured networks, we could identify 107 networks as known public hotspot networks. 30 could be assigned to personal networks and we are uncertain about the remaining 101 networks.

The above data clearly shows that evil twin attacks can affect a large number of users and that the majority of connections to popular open access points are done automatically by mobile phones without the user being aware of the connection and thus also not aware of the location of the connection.

5 METDS: Mobile Evil Twin Detection System

A detection system always has to struggle with the limited set of data that can be used for a reliable decision. The detection system we developed uses as much data from the surrounding environment as possible. As described in section 3 our algorithm utilizes more parameters than any of the detection system discussed in the related work section. In the following we describe which parameters are used and how they are combined and analyzed to reach the best detection accuracy.

We start with a user's device which is in the communication range of an access point, it is about to connect to. Here our algorithm has to face two main initial situations. In the first one the user wants to connect to a wireless network for the first time and the device has never been connected to it before. In this case the METDS does not have any further knowledge of the network and can not assist the user. This is the same Trust On First Use (TOFU) decision the user always has to face in this situation. If the user has been connected to the access point before the METDS can react differently. In this case the METDS already has an appropriate dataset, that can be used to verify the current environment.

To detect malicious access points the underlying algorithm of the METDS utilizes the following parameters to describe and verify an access point's environment. Primarily the **SSID** of a wireless network is used to recognize known networks, like all current mobile operating systems do. Additionally the algorithm takes the **BSSID** of previously connected access points into account. Since a BSSID can be faked as well, the algorithm records and compares the **wireless network environment**. To characterize the environment as accurately as possible we do not only store the SSIDs of surrounding networks, but also the BSSIDs of the access points as well as their supported authentication, key management, and encryption schemes. In the course of the development of METDS we also investigated if and how the inclusion of signal strengths and frequencies could help to improve the accuracy of the detection algorithm. The signal strength is strongly dependent on the device's position and orientation, which makes it an unreliable source for the decision process. The operated frequency of the access point does not improve our results as well, since modern access points often use auto-tuning algorithms to change between WiFi channels depending on the degree of capacity utilization. Therefore both of these additional parameters lead to a higher false positive rate of the algorithm and have been excluded from the decision process. Since most of the modern smartphones and tablets are connected to a mobile communication network we additionally use **cell tower information** as an environmental parameter. As a last parameter the algorithm takes the device's **location** into account, which is determined either through the Google Play services API if present or directly through the native Android location API.

By collecting and evaluating the mentioned parameters the detection system reaches diverse states which have to be treated differently to warn the user of potentially dangerous connections to access points. However the false positive rate also needs to be kept small enough. Otherwise the acceptance and effectiveness of the system will be adversely affected.

5.1 Unknown SSID

As mentioned above, the connection to a new wireless network for which no information is stored in the METDS database will be accepted by the algorithm, since the user actively has to choose it. In this case no warning is displayed to the user, but information about the access point and the surrounding environment is stored for a later recognition of this context.

5.2 Unknown BSSID

If the algorithm detects a known SSID, but has no corresponding BSSID in its database a warning message is shown to the user, making him aware of the potentially dangerous situation. In this case the user gets two options. If the user knows that a legitimate access point is present and thereby trusts it, an access point profile is created and stored to the database. If the user is not willing to connect to it, the connection process will be canceled and no further information will be collected by the system. This basically extends the TOFU principle to new access points. This has the usability impact of asking users whether to trust a new access point of a chain of access points. For example METDS will ask the user to confirm each new Starbucks WiFi instead of blindly trusting any network with the Starbucks SSID. In our field study the average number of times a user would have to accept a new BSSID is only 8.14. This only happens when new locations are visited. Thus we believe this is an acceptable trade-off, since it significantly reduces the attack capabilities of evil twin networks.

5.3 Unknown environment

If the tuple of SSID and BSSID can be found in the database the network environment will be verified to detect attacks in unknown environments. For this, a WiFi scan is started in the background. The result of this scan is compared to existing access point profiles of this wireless network. An access point profile consists of basic information like the BSSID and a set of different environments. These different environments are needed to consider multiple WiFi signal propagations, for instance, if an access point can be received in more than one room with highly variable sets of other receivable access points in the environment. To compare the environments the METDS calculates the Jaccard index for each combination of sets. If a combination has an Jaccard index higher than a specific threshold, the according profile is accepted as a known network environment. In multiple simulations and parameter studies we found out that a Jaccard index of 0.7 seems to be the best trade-off between the false positive and the false positive rate. However, this is a parameter which can be tuned by the user to fit their desired level of protection and their tolerance of warning messages.

Another advantage of our proposed algorithm is its ability to adapt to changing network environments. In case a known network environment is recognized, the learning algorithm adds an access point to the access point's profile, if it is found in the vicinity multiple times. The deletion of previously recorded access points is done in the same way. Using this learning algorithm METDS is able to adapt itself to environments that change over time but still detects malicious environments in unknown environments.

If the network environment of an access point is not known, cell tower information is being checked as well. The algorithm determines the location area code (LAC) and the cell ID of the current mobile network connection. These parameters are stored, verified and learned similarly to the surrounding access points. We implemented the learning here as well to take overlapping cell sites into account. If a suiting tuple of cell tower information can not be found either, a warning message is displayed and the connection process is interrupted. In our current configuration if only the cell tower is recognized but not the wireless network environment no warning message is shown, since we believe faking cell IDs is beyond most attackers. However, this is a configuration parameter and warnings can of course be shown.

5.4 Unknown location

Since the determination of the user's current GPS position is not only time consuming but also consumes a lot of energy, this last step is only performed if the network environment as well as the cell tower information is not being recognized or if it is not available during the connection. To save energy the algorithm checks if the Google Play services are available on the device. If present, the current position can be retrieved in under 1 second and is accurate enough to meet our requirements. If they are not available, we determine the position through the native Android Location API. Fortunately more than 99% of the users of our study that allowed us, to retrieve their location had the Google Play services installed. Hence, the usage of the native Location API can be seen as fallback solution, which will only be used very rarely.

Once the position is determined it is compared to the saved position of the stored access point. If the determined position is within a specific radius of the stored location, the position of the access point is accepted and the connection will be allowed without warnings. In a parameter study we found out, that a radius smaller than 100m leads to many false positive warnings due to the accuracy of the determined positions. To get the most conservative detection system we chose a radius of 100m for our simulations, we present in section 6.

With the algorithm described above we try to use as much environmental data as possible to support the decision, if a malicious access point might be present. All the parameters are requested from sensors and APIs of the Android operating system. The values and profiles are stored in an on-device database. Thereby the METDS is self-contained and does not need any further communication to backend servers or additional infrastructure to work. We also take energy-efficiency seriously by preferring methods and API calls that are less power consuming.

6 Evaluation

To test our system we implemented the algorithm and simulated our detection system on the real-world data we gathered during the study.

As with all heuristic based detection systems the goal is to create a system that protects the user as much as possible while not creating too many false warnings. There will always be an area of conflict between security and usability. One caveat of our simulation and the data is that we have no way of knowing if our real world data contained evil twin attacks. However, since currently evil twin attacks would fall in category A or B with a very high probability, for the purpose of the simulation we assume the connections we gathered within our study are connections to legitimate access points. Should there have been evil twin attacks of type A or B against our subjects during the study period this does not impact our simulation results, since the evil twin would simply be viewed as another legitimate access point. This might seem unintuitive at first, however since we are only interested in finding a set of environmental parameters which are as sensitive as possible to changes in the measured environment while keeping the number of false positives as low as possible, the potential evil twin access point of type A or B do not pose a problem. We think the probability of an attack of type C or D is next to 0 since there is no motivation for attackers to invest the effort since even attack types A and B are currently undetectable to the vast majority of the population.

The simulator we developed helped us to adjust the parameters for the detection system. For a later adoption to the Android platform, we developed the detection system as well as the simulator together with all of its interfaces and libraries in Java. Thus the detection algorithm can be migrated to Android to run real-world on-device studies. The basic architecture of the simulation framework and the simulator is shown in figure 2.

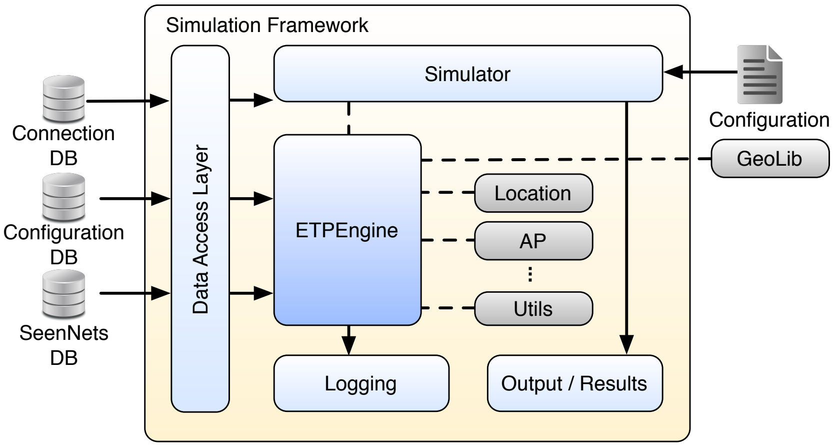

Fig. 2. Architecture of the simulation framework and its interfaces

The simulator works with data from different databases. The connection database contains all information about the connections, that have been recorded, stored and submitted to our servers during the user study. The configuration database contains all device-specific configurations, such as configured wireless networks and other meta data. In the third database all wireless networks are stored, that have been registered during the WiFi scans within the study. The data sources are connected to the simulation framework via interfaces to be easily exchangeable. The simulator itself reads input data, orchestrates components and delivers the results to the output class. The simulator component also reads the configuration for simulations and parameter studies. The centerpiece is the *ETPEngine*. Here the algorithm and the logic of our detection system is implemented. It utilizes several classes that assist the ETPEngine with storing and analyzing information about the wireless environments. While the logging component may help future developers to setup simulations correctly, the output component collects all results from ETPEngine and the simulator and exports them in a form that can be further processed by other programs.

6.1 Results

We simulated all participants of our study that took part in the study for 10 days or more and who connected to open access points. We left out users, that participated for a shorter period of time since the learning process of METDS needs at least a couple of days of activity before it can reliably predict the environment of access points during new connection attempts. We only simulated connections to open access points, since these are the prime targets for evil twin attacks.[5] This left us with 43 users to simulate. We configured the METDS used by the simulator with the parameters that have been discussed in section 5.

In figure 3 one can see how many users saw how many warning messages per day. The number of users who saw only one warning are shown in dark green and the scale goes up to the number of users who saw 11 or more warning who are shown in red. For clarity we did not plot the number of users who did not see any warnings since these would have scaled the graph to such an extent that the different warning levels would not have been distinguishable. The number users with 0 warnings can be calculated by subtracting the number of users shown in the diagram from 43, the total number of users in the simulation. All different types of warnings from section 5 are included in this plot. As one can see the majority of users do not see any warnings and of those who do only see a small number per day and a downwards trend is visible as METDS learns. While this number may still seem high, since new BSSID warnings are included, we believe these warnings to be acceptable. The new BSSID warnings have a 0% false positive rate and thus definitively present a new access point. We believe users should be asked before connecting to such a new network, just like when connecting to a new SSID. While our intuition says that this will be acceptable to

---

[5] While it is also possible to mount evil twin attacks against Enterprise WPA networks, these would go beyond the scope of this paper.

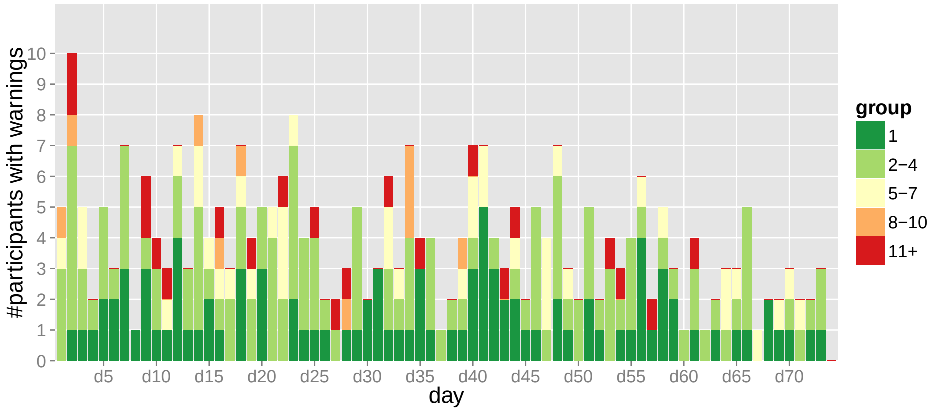

Fig. 3. Amount of BSSID and environmental warnings users saw per day. Users are grouped by how many warnings they saw per day. The huge group of users who saw no warnings is omitted for clarity.

users, we will need to deploy METDS in a full field study to test users reactions to confirm this in future work.

In figure 4 we removed the new BSSID warnings. The remaining warnings represent probably false positive warnings. Since our assumption is that our ground truth data contains no attacks these warnings are shown to users because the network environment changed to such an extent that the METDS decided to show it. Using our simulator we ran parameter studies to find a good configuration of METDS. The goal was to find a set of parameters that is as sensitive to potential evil twin attacks of Type B, C or D as possible, but that creates as few false positives as possible. There is no objective way to measure this since there currently are no attacks of this type. However, as soon as METDS is deployed attackers could and would upgrade their attacks from type A to type B through D. So we create a first test configuration of METDS based on a best effort estimation of parameters that would make attacks of types B through D as difficult as possible without burdening users with too many warnings. Table 1 in the appendix shows this configuration. These parameters are of course up for debate and can actually be configured on a per user basis. So if users feel they are seeing to many warnings they can tune the system down at the cost of making it easier for an attacker to fake an environment in which an evil twin attack can be mounted. This will be down to user preferences. For the rest of the paper we selected a set of parameters which in our opinion would give users a good level of protection while not burdening most users with any warnings at all and only a few with some. As can be seen in figure 4 the number of these undesirable warnings is fairly low, with the vast majority of users seeing no warnings at all and only a single user seeing a high number of warnings.

To get an idea how well the algorithm performs overall we calculated the percentage of users that receive at least one warning on a specific day. On average

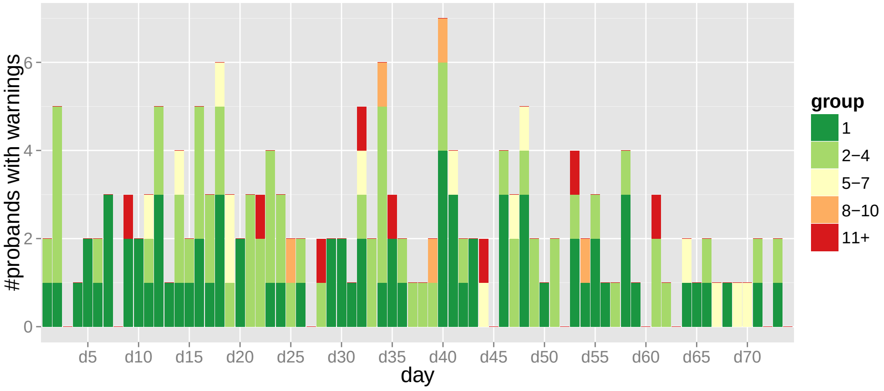

Fig. 4. All simulated warnings except new BSSID warnings during the complete study, grouped by the number of warnings. The group with 0 warnings is omitted for clarity.

only 5.81% ($\sigma = 4.4$) of the considered users have to react to warnings that have been raised by METDS. In appendix B the users perspective is shown.

7 Conclusion & Outlook

In this paper we present the first study of real-world WiFi usage with respect to the susceptibility to evil twin access point attacks. We carried out a study with more than 90 participants that collected real-world WiFi usage and environmental data for more than two months. We showed that 43 of our users are susceptible to evil twin attacks since they use open access points. We introduce three types of evil twin attacks which go beyond the simple evil twin attack against current mobile devices. Furthermore we developed METDS, a prototypical detection system, that protects against the simple evil twin attacks and utilizes environmental data to mitigate the more sophisticated evil twin attacks introduced in this paper. To evaluate our detection system, we set up a variety of simulations based on the real-world data we collected from the participants of our study. Our results show, that many connections to unknown access points currently go undetected and METDS would allow users to decide whether to trust new access points or not. We also created a first test configuration of METDS to detect future sophisticated evil twin attacks and show how many false positive warnings this would entail for users. The next steps in our research are to deploy METDS and conduct a field study to both search for evil twin attacks in the wild and to evaluate the current METDS configuration with real users in their everyday life.

References

1. Bauer, K., Gonzales, H., McCoy, D.: Mitigating Evil Twin Attacks in 802.11. In: 2008 IEEE International Performance, Computing and Communications Conference. pp. 513–516 (Dec 2008)
2. Fahl, S., Harbach, M., Muders, T., Baumgärtner, L., Freisleben, B., Smith, M.: Why eve and mallory love android: An analysis of android ssl (in)security. In: Proceedings of the 2012 ACM Conference on Computer and Communications Security. pp. 50–61. ACM (2012)
3. Fahl, S., Harbach, M., Perl, H., Koetter, M., Smith, M.: Rethinking ssl development in an appified world. In: Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security. pp. 49–60. ACM (2013)
4. Gonzales, H., Bauer, K., Lindqvist, J., McCoy, D., Sicker, D.: Practical defenses for evil twin attacks in 802.11. In: 2010 IEEE Global Telecommunications Conference. pp. 1 –6 (Dec 2010)
5. Kindberg, T., Mitchell, J., Grimmett, J., Bevan, C., O'Neill, E.: Authenticating public wireless networks with physical evidence. In: Wireless and Mobile Computing, Networking and Communications, 2009. WIMOB 2009. IEEE International Conference on. pp. 394–399 (Oct 2009)
6. Lanze, F., Panchenko, A., Braatz, B., Engel, T.: Letting the puss in boots sweat: Detecting fake access points using dependency of clock skews on temperature. In: Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security. pp. 3–14. ACM (2014)
7. Mónica, D., Ribeiro, C.: Wifihop - mitigating the evil twin attack through multi-hop detection. In: Proceedings of the 16th European Conference on Research in Computer Security. pp. 21–39. ESORICS'11, Berlin, Heidelberg (2011)
8. Roth, V., Polak, W., Rieffel, E., Turner, T.: Simple and effective defense against evil twin access points. In: Proceedings of the first ACM conference on Wireless network security - WiSec '08. p. 220 (Mar 2008)
9. Song, Y., Yang, C., Gu, G.: Who is peeping at your passwords at starbucks? - to catch an evil twin access point. In: DSN'10. pp. 323–332 (2010)
10. Szongott, C., Henne, B., Smith, M.: Evaluating the threat of epidemic mobile malware. In: WiMob. pp. 443–450. IEEE Computer Society (2012)
11. Szongott, C., Henne, B., Smith, M.: Mobile evil twin malnets–the worst of both worlds. In: Cryptology and Network Security, pp. 126–141. Springer (2012)

A METDS Sample Configuration

In table 1 the most important configuration parameters for our sample configuration are shown. These values have been used for the mentioned simulations from section 6. As one can see the algorithm only reacts to connections to unencrypted networks. For future research other encryption schemes can be enabled to analyze similar attacks on encrypted wireless networks. The BSSID thresholds define, how often an access point needs to be detected or missed, until it is added to or removed from the according access point profile. The maximum distance threshold defines how close to each other two locations have to be at least, until the algorithm regards them as equal. The length of the learning period of an access point is defined by the next parameter. Within this period the algorithm learns the access points environment and adapts itself. The current value represents one week. The last two parameter enable the Jaccard index comparison of network environments and set the threshold to 0.7 as discussed in section 5.

Table 1. Configuration parameters of the METDS

Configuration item	**value**
ACCOUNT_UNENCRYPTED	TRUE
ACCOUNT_WPA_PSK	FALSE
ACCOUNT_WPA_ENTERPRISE	FALSE
BSSID_DELETION_THRESHOLD	-3
BSSID_ADDITION_THRESHOLD	3
MAXIMUM_DISTANCE_THRESHOLD	100.0
LEARNING_PHASE_NEW_AP_LENGTH	604,800,000
USE_JACCARD_ALGORITHM	true
JACCARD_ENVIRONMENT_OK	0.7

B User Perspective

Figure 5 shows the user's perspective. Both diagrams show the number of warnings each user (along the Y-axis) sees on average per 100 connections. In the first graph only warning messages for unknown BSSIDs are shown. As stated above we believe these warnings are necessary since they definitively present an unknown access point and a connection should not be established without the users consent. The second graph only shows the false-positives at our current configuration of METDS. Also as stated above the number of warnings shown here is configurable and is down to user preferences.

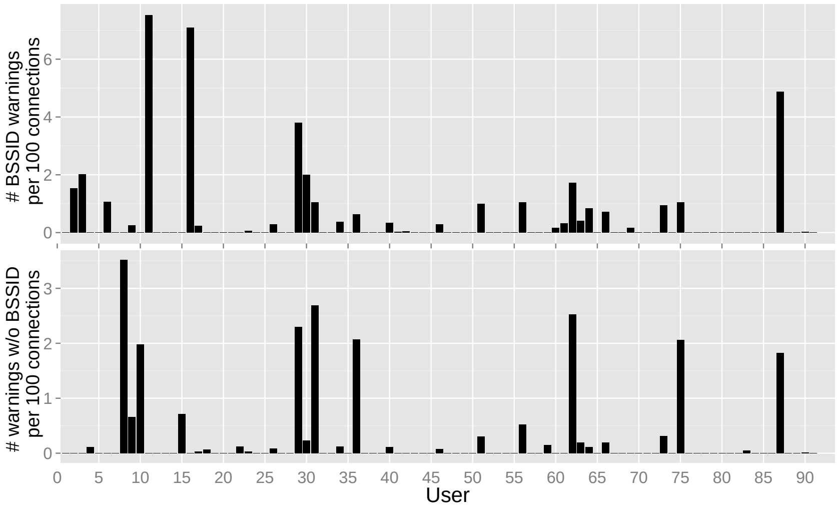

Fig. 5. Amount of warnings a user would receive per 100 connections. In the first diagram only BSSID warning are shown, in the second all remaining warnings have been plotted.