Available online at: https://ijact.in

Date of Submission	10/12/2018
Date of Acceptance	23/02/2019
Date of Publication	30/03/2019
Page numbers	3075-3078 (4 Pages)

ISSN:2320-0790

COMPUSOFT
An International Journal of Advanced Computer Technology

IMPROVING WI-FI SECURITY AGAINST EVIL TWIN ATTACK USING LIGHT WEIGHT MACHINE LEARNING APPLICATION

Dr. Harsha S [1], Dr. Khalid Nazim S A[2], Dr. S Balaji[3],Vallabh V Rao[4]
[1]Associate Professor, Department of IS&E, Jyothy Institute of Technology, Bengaluru, Karnataka, India,
harsha.s@jyothyit.ac.in
[2]Assistant Professor, Department of CSI, College of Science, Majmaah University, Al- Majmaah 11952, Saudi Arabia,
k.sattar@mu.edu.sa,
[3]Professor, Computer Science & engineering, CIIRC, Jyothy Institute of Technology, Bengaluru, Karnataka, India,
[4] Department of IS&E, Jyothy Institute of Technology, Bengaluru, Karnataka, India

Abstract**:** In the current world, all the devices are aiming to be or already are wireless and mobile. The trend is building smarter devices that offer the users all their required services with minimal human intervention. Due to this all manufacturers design their devices to be signal hungry as that is the only requirement that users feel. Since this has been the motto of all brands of wireless telecommunication devices to better the user experience, all devices attempt to automatically latch on to the network that has the highest signal strength and that is easily available. However, this design makes the devices vulnerable to a classic "MalNet[1]" attack called "Evil Twin[1]". Most devices suffer data loss or bandwidth loss regularly to this attack. There are also cases of financial losses [3] suffered by the users because of the said attack. In this paper we have attempted to use Android API to build a simple light weight security system that can prevent the evil twin attack.

Keywords: Wi-FiSecurity, Machine Learning, Evil Twin, MalNet, Bayesian Classification.

I. INTRODUCTION

Irrespective of brands, service providers' names and claims, all smart devices have one basic requirement and that is to stay connected always. This need causes the systems to go beyond the most important requirement; that is security. This leads to the innumerable attacks on devices when they attempt to connect to a usually unknown Wi-Fi network at a public place or try to latch on to some open network in a neighborhood. This attack is termed "Evil Twin" [1] attack. This attack consists of mimicking the Service Set Identifier (SSID) [4] and using an Application Programming Interface (API) [5], to convince the device attempting to connect that the connection is real. The scenario looks as shown in the below figure 1.

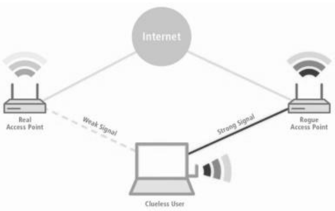


Figure 1: Evil Twin attack scenario [10]

The rogue access point is the evil twin. It has the same SSID as the real access point. But it generates a proxy script that causes the user's machine to trust that its connectivity to the internet is better in terms of signal strength, bandwidth as well as sometimes appearing open (without security). Even when it appears to have security measures such as WPA/PSK or WPA2/PSK [5], the security pass phrase/word entered would just be recorded so that the evil twin can later attack the real access point and has no purpose in establishing the connection. Thus, an unknowing user transfers all their data through a malicious access point which can be harmful in more than one way, as listed below

i. Threat to privacy.
ii. Identity theft.
iii. Financial losses.
iv. Data theft /hijacking.
v. Bandwidth loss.

II. Literature Review

The identification of the problem alone came after studying the nature of MalNets. Kevin Bauer et.al. [9] have shown how malicious network access points work to steal bandwidth, data or even identity of unknowing users as early as 2008.

Google ™ has published several white papers discussing Android security and attacks. [8]. Since the Android revolution began, the number of large-scale attacks on device integrity and application security has been on the rise consistently. Android systems are being designed with higher consideration for connectivity and faster data transfer even now. This causes a major flaw in their security feature, which we have tried to solve.

William Encket.al. [7] have presented an extensive study of android application security architectures and their shortcomings. All the lacunae identified circle around the fact that devices tend to connect automatically to any and every access point available that promises better bandwidth irrespective of security.

Bahman Rashidi et.al. [6] have published a critical review of several security issues in android devices and methods to defend against the same. They have also presented the weaknesses of such security measures and how they falter in time. This led us to consider large companies that design, develop and market security products and services for computers worldwide.

Kaspersky ® has published a public document [5] that lists a host of ways with which a user can enhance security to his/her device and data. Although, the list and descriptions are elaborate, they centre on user's actions rather than the device behavior. All the methods listed require the users to be vigilant and thorough with their connection mechanisms.

CISCO® has presented a white paper [4] in which they show how device designs can be improved to aid the devices, users as well as service providers in this continuing battle against security threats.

Kavita Sharma et.al.[3] have presented a multi layered machine learning approach to defend against Malicious application software (Malware) attacks on wireless devices, which prompted us to explore the possibility of using machine learning to solve the problem with higher efficiency and accuracy.

CandaceBerretta et.al.[2]have presented a Bayesian classification of spatial data which shows a unique and highly useful method to classify multi dimensional data into binary classes. The method we have presented in this paper resembles this to a minimal extent. The methodology is presented in the next section.

Christian Szongott et.al. [1], have discussed the different ways by which data theft is done using Evil Twins or Malicious Network Access points. The article also, shows many methods used to defend against such attacks. However, the defense is rendered weak because of the systemic feature of android devices and laptops that makes them signal hungry. Our method relies on altering this behavior to ensure the security first motto.

III. METHODOLOGY

Our proposed method uses a simple API [7] for smart phones to ensure the available network's security. The algorithm used is described below. For ease of understanding and as the available APIs are used for the known connection procedure, the steps are listed from the commencement of the evil twin [1] attack. Our system depends on the storage of known SSID and the MAC address of access points.

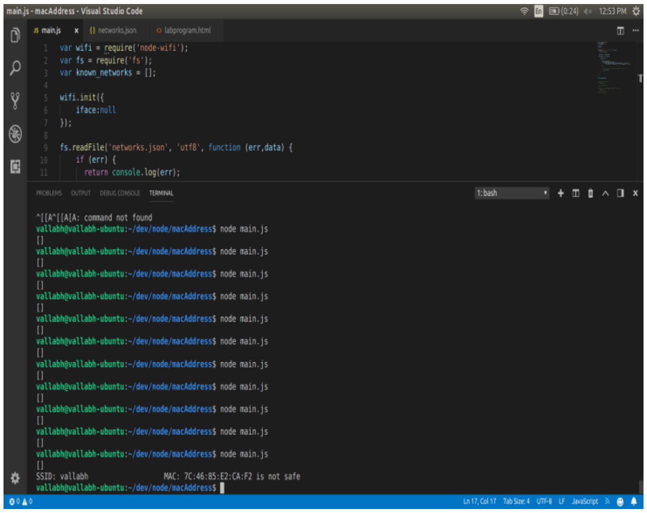
1. Check module
    a. On availability of networks with known SSID
    b. Call Learning module
    c. Check the look up table for corresponding MAC address
    d. Request MAC address of the access point
    e. On reception and comparison yields "TRUE"
    f. Call 3 -party Authentication module
    g. On result from 3 -party authentication module and learning module "TRUE"
    h. Store access point data in the "SAFE" list
    i. Else exit connection and list New SSID and MAC address as "UNSAFE"
2. Learning module
    a. Add each new SSID and corresponding MAC address to training set
    b. On reception of result from Check module
    c. Store consequent values
    d. On each new SSID sent to 3-party authentication module
    e. Add SSID and corresponding MAC address to test set

   f. Run Bayesian Classification [2] to predict the probability of it being "UNSAFE"
   g. On probability greater than 0.75 (The threshold set at 0.75 for experimentation), deem access point to be "Evil Twin" and append to "UNSAFE" list
   h. Else, append the access point to "SAFE" list

3. 3 -party authentication module
   a. Start arbitration and wait for request
   b. On request send authentication string generated using access point's SSID, MAC address and signature to device and vice versa to the access point
   c. On reception of responses, compare
   d. ON result = "TRUE"
   e. Send "SAFE" signal to device and learning module
   f. Else Send "UNSAFE" signal to both and exit

IV. EXPERIMENTAL DATA

The algorithm is implemented as a program in JAVA ™ for proof of concept. The screen shot of the program and its output are shown in Figure 2.



```
var wifi = require('node-wifi');
var fs = require('fs');
var known_networks = [];

wifi.init({
    iface:null
});

fs.readFile('networks.json', 'utf8', function (err,data) {
    if (err) {
      return console.log(err);
```

```
^[[A^[[A[A: command not found
vallabh@vallabh-ubuntu:~/dev/node/macAddress$ node main.js
[]
vallabh@vallabh-ubuntu:~/dev/node/macAddress$ node main.js
[]
vallabh@vallabh-ubuntu:~/dev/node/macAddress$ node main.js
[]
vallabh@vallabh-ubuntu:~/dev/node/macAddress$ node main.js
[]
vallabh@vallabh-ubuntu:~/dev/node/macAddress$ node main.js
[]
vallabh@vallabh-ubuntu:~/dev/node/macAddress$ node main.js
[]
vallabh@vallabh-ubuntu:~/dev/node/macAddress$ node main.js
[]
vallabh@vallabh-ubuntu:~/dev/node/macAddress$ node main.js
[]
vallabh@vallabh-ubuntu:~/dev/node/macAddress$ node main.js
[]
vallabh@vallabh-ubuntu:~/dev/node/macAddress$ node main.js
[]
vallabh@vallabh-ubuntu:~/dev/node/macAddress$ node main.js
[]
SSID: vallabh               MAC: 7C:46:85:E2:CA:F2 is not safe
vallabh@vallabh-ubuntu:~/dev/node/macAddress$
```

Figure 2: Screen shot of the Java program and its output

As described in methodology, the program, runs on an Android simulator [3]. It requires the access point to be classified as "safe" before the device can send data packets through the access point. The steps involved are:

1. The device sends a handshake packet to identify the access point
2. The access point sends its SSID as a response (Normally, this is enough for the device to begin the establishment of connection)
3. The script sends a second packet to the AP requesting for the MAC address.
4. The AP has to send the data to proceed further.
5. On reception the program verifies if the known SSID and MAC addresses match the current AP.
6. If they do not match the conection will be terminated.
7. If they do, then the program requests the AP to authenticate itself using a three way handshake.

Thus, the program, while being light weight ensures that an "Evil Twin" or a "MalNet" cannot dupe the device into sharing data. Also, the training and test set split is not defined due to the inherent nature of the system. All, known access points are by default added to the training set irrespective of their consequent value. This feature enhances the learning capability of our proposed system and hence is a great advantage against "Evil Twin" attack. The data fields captured are represented in the experimental data as they are captured with their consequent values. The experimental results for a sample progressive training set are as shown in Table 1.

Table 1: Experimental data -Training set

SSID	MAC Address	Security	security_flags__wpa	Consequent
Vallabh	E8:CC:18:9 D:90:B4	WPA1 WPA2	pair_ccmpgroup_tkipp sk	SAFE
w4w	80:26:89:02: B5:29	WPA1 WPA2	pair_tkippair_ccmpgro up_tkippsk	SAFE
w4w	98:DE:D0:A 2:E9:4E	WPA1 WPA2	pair_ccmpgroup_ccm ppsk	UNSAFE
D-Link_DIR-816_5G	80:26:89:02: B5:2A		(none)	UNSAFE
STHITIGAR	00:26:15:66: 0D:8A	WPA1	pair_tkipgroup_tkipps k	SAFE
VIRUS_FOUND	E8:CC:18:9 D:90:B5	WPA1	pair_tkipgroup_tkipps k	SAFE
!!!HighRisk!!!	80:26:89:02: B5:30	WPA1 WPA2	pair_tkippair_ccmpgro up_tkippsk	SAFE
No_Name	98:DE:D0:A 2:E9:4E	WPA1 WPA2	pair_ccmpgroup_ccm ppsk	UNSAFE
FreeWiFi	9A:26:8E:02 :95:2B		(none)	UNSAFE
OpEnNeT	00:2A:15:B6 :5D:8A	WPA1 WPA2	pair_ccmpgroup_ccm ppsk	SAFE
IHACKYOU	C8:C2:F1:E 4:93:B7	WPA1	pair_tkipgroup_tkipps k	SAFE
KillerWiFi	80:26:89:02: B5:31	WPA1 WPA2	pair_tkippair_ccmpgro up_tkippsk	SAFE
D-Link_DIR-816_5G	9E:57:99:0A :5C:8B	WPA1WPA2	pair_tkippair_ccmpgro up_tkippsk	SAFE

The table shows that as and when new access points are available to be connected through, our system; treats them as test set members, checks and verifies their authenticity. Once this task is complete, they become the part of the training set. The result of the assessment is shown as a graph in Figure 3. The status '1' indicates that the access point is safe where as a '0' indicates that the access point is either an "Evil Twin" or a "MalNet". The addition of each test data into training set leads to a Sharp rise and quick stabilization to the learning curve as shown in figure 4.

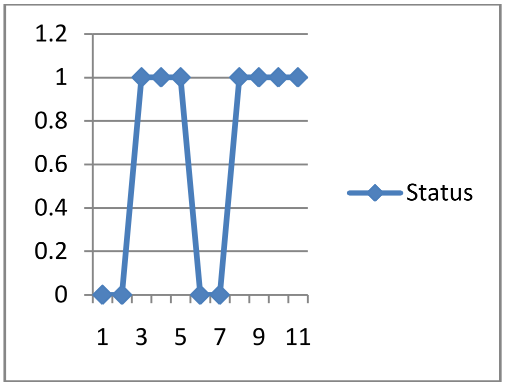


Figure 3: Assessment Result

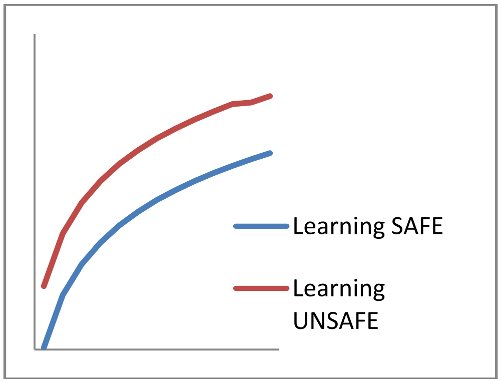


Figure 4: Learning curves for the Training Set

V. Results and Analysis

It is evident from the learning curve shown in Figure 3, that since we add each test data to the training set after the verification, the accuracy of our proposed system is very high. Each new access point added to the system leads to better reinforcement of learning and hence improves the performance. The system stabilizes learning within a limited number of samples. Also, it is clearly shown that identification of "MalNet" or "Evil twin" attackers is efficiently performed by our method. The down side of our method is that, the device must store every available access point data. Even though, the type of the data makes the required storage space negligible, it may pose a challenge in later stages for users who are mostly travelling long distances regularly. However, we have not tried to analyze the data storage space requirement and its impacts on the device performance as the application is light and the task is not in the scope of the current research.

VI. Conclusion

To provide security to one's identity, data and privacy is an exponentially challenging task in today's world. Our proposed system achieves this by altering the basic nature of devices; by making them safety hungry instead of signal hungry. The machine learning feature enhances its performance over time and always provides better results to the user. Thus, our system may be a hope for a heightened level of security for mobile and smart devices in the future.

VII. References

[1]. Szongott C., Henne B., Smith M. (2012) Mobile Evil Twin Malnets – The Worst of Both Worlds. In: Pieprzyk J., Sadeghi AR., Manulis M. (eds) Cryptology and Network Security. CANS 2012. Lecture Notes in Computer Science, vol 7712. Springer, Berlin, Heidelberg

[2]. Candace Berrett, Catherine A. Calder, “Bayesian spatial binary classification”, Spatial Statistics, Volume 16, 2016, Pages 72-102, ISSN 2211-6753, https://doi.org/10.1016/j.spasta.2016.01.004 (http://www.sciencedirect.com/science/article/pii/ S2211675316000063)

[3]. Kavita Sharma, B.B. Gupta, Multi-layer Defense Against Malware Attacks on Smartphone Wi-Fi Access Channel, Procedia Computer Science, Volume 78, 2016, Pages 19-25, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2016.02.005. (http://www.sciencedirect.com/science/article/pii/ S1877050916000077)

[4]. https://www.cisco.com/c/dam/en/us/products/collateral/wireless/aironet-3600-series/white-paper-c11-713103.pdf, 2015.

[5]. https://usa.kaspersky.com/resource-center/preemptive-safety/public-wifi, 2015.

[6]. A Survey of Android Security Threats and Defenses, Bahman Rashidi, Carol Fung, Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, volume: 6, number: 3, pp. 3-35, 2015.

[7]. William Enck, Damien Octeau, Patrick McDaniel, and Swarat Chaudhuri. 2011. A study of android application security. In Proceedings of the 20th USENIX conference on Security (SEC'11). USENIX Association, Berkeley, CA, USA, 21-21.

[8]. Google work for Android Security White paper, 2015.

[9]. K. Bauer, H. Gonzales and D. McCoy, "Mitigating Evil Twin Attacks in 802.11," 2008 IEEE International Performance, Computing and Communications Conference, Austin, Texas, 2008, pp. 513-516.
doi: 10.1109/PCCC.2008.4745081

[10]. Vibhawari V. Nanavare, Prof. Dr. V. R. Ghorpade, Robust and Effective Evil Twin Access Point Detection Technique at End User Side, International Journal of Innovative Research in Science, Engineering and Technology, ISSN online: 2319-8753, ISSN Print: 2347-6710.

[11]. Dr. Adiline Macriga G., Prevention Technique for Creating Fake Profiles and Accounts on Websites, COMPUSOFT, An International Journal of Advanced Computer Technology, 2018, ISSN: 2320-0790, pp. 2826-2830.