International Journal of Innovations in Engineering and Technology (IJIET)
http://dx.doi.org/10.21172/ijiet.133.21

Implementation and Analysis of Key Reinstallation Attack

Saba Khanum[1], Ishita kalra[2]
[1]*Department of Information Technology, MSIT, Janakpuri, New Delhi, India*
[2]*Department of Computer Science and Engineering, MSIT, Janakpuri, New Delhi, India*

Abstract- The objective of the paper is to implement and analyzed the impact of Key Reinstallation Attack (popularly dubbed as KRACK) on debian based machines. The paper elucidates on the capture of packets through the attack without being a part of the network and affecting the target machines with the help of an attack machine placed inside the network. It basically exploits the nonce of the network which ultimately paves way to the execution of the attack. The issue tends to gather more eyeballs as it affects all devices using Wi-Fi through WPA2 protocol. Hence, the catastrophe complimented along the attack is severe. The analysis of the impact is carried on by analyzing the type of packets visible as well as captured during the course of the implementation. Here, we have created a python script which identifies whether the targeted machine is vulnerable to KRACK or not and corresponding to that the packet capture starts and ultimately, the impact is measured.
Keywords – KRACK, weakness, WPA2, attack, security

I. INTRODUCTION

The presence of the bug has been detected in the cryptographic nonce of the WPA2 and can be used to clone a connected party to reinstall a used key. The presence of the nonce is specifically intended to prevent reuse, but in this particular case, it gives malicious users the opportunity to replay, decrypt, or forge packets, ultimately enabling them to access all previously considered encrypted information without actually being part of the network. This could be done by manipulating and replaying cryptographic handshake messages and attack can be achieved. The presence of vulnerability across all platforms and devices using WPA2 protocol makes this attack prone to most of the current generation devices. The impact of the risk can be divided into the two categories. The impact is in the form that foremost affects the supplicant that is the wireless endpoints. In simple words, the devices like Windows, Linux that we use. The second form it is expected to take is the form of an authenticator that basically means the infrastructure devices (mainly wireless) we use in daily lives.

Limitations of the papers proposing the attack

No in-depth implementation of the attack across systems is reflected on. The analysis of a variety of packets used for the 4-way cryptographic handshakes specifically Extensible Authentication Protocol (EAP) over LAN (EAPoL)

This papers mainly focuses on a comprehensive implementation of KRACK and how the cryptographic nonce can make the 4-way WPA2 handshake more than ninety percent of today's devices prone to such a disaster. The process is carried forward using Python 2.7 to detect the vulnerable machine and then analysis of the packets used in the communication with the help of EOPL. Our methodology can be divided into three major categories that is; configuration, set-up, configuration, implementation and then the analysis. We lastly present a graph that provides the suggested analysis with the help of a bar graph.

II. LITERATURE SURVEY

As per our study, most of the fellow researchers have encountered a common barriers while implementing such a vast attack, hence the proof of concept in the paper[1] only highlights the packet capture part of the attack using wireshark and the similar constraints were established as multiple patches of the same were established as soon as the presence of the attacks were detected. Therefore, we here for this study will make use of older versions of Kali Linux and Ubuntu virtual machines. A variety of researchers have attempted to break WPA2 through Man in the Middle attack but it the attempts of implementation of this exploit has not been explored properly. All the implementations are currently carried on virtual images of the platforms connect in the same network.

The 802.11i Amendment

IEEE 802.11i, popularly known as 802.11i, is an extension of the original IEEE 802.11. It's implementation is called Wi-Fi Protected Access II (WPA2). This standard provides a variety of security mechanisms for wireless networks, hence, replacing the insufficient privacy clause of the previous standards with a thorough Authentication and Security clause.

The 4-way Handshake

The four-way handshake is designed to allow the access point (or authenticator) and wireless client (or applicant) to prove to each other independently that they know the PSK / PMK without ever disclosing the key. Instead of

disclosing the key, the access point (AP) and the client encrypt messages between themselves — which can only be decrypted using the PMK they already share. if the messages were decrypted successfully, this proves the PMK's knowledge. The four-way handshake is critical to protecting the PMK from malicious access points — such as an attacker's SSID that impersonates a real access point — so that the client never has to tell their PMK to the access point.

The Group Temporal Key

It may need to be updated in the network due to the expiry of a preset timer. The GTK also needs to be updated when a device leaves the network. This is to prevent the device from receiving more messages from the access point, either multicast or broadcast.

802.11i defines a Group Key Handshake consisting of a two-way handshake to handle the update: the AP sends the new GTK to each networked STA. Using the KEK assigned to that STA, the GTK is encrypted and by using a MIC protects the data from tampering. The STA recognizes the new GTK and answers the AP.

III. METHODOLOGY

3.1 Basic Overview initial setting

Here we describe the basic overview of the pipeline used in our method. Our pipeline basically consists of the following major steps:

Set-up

Configuration

Development of python script to assist the intended procedure

Implementation across network using virtual machines

Analysis of the impact

Steps “a” and “b” need to carried on every time the practical impact of the flaw has to be calculated whereas step “c” does not need such frequent upgradations.

3.2. Preprocessing Steps

There were majorly two important aspects that comprise of the pre-processing phrase. The setup phrase and the configuration phrase. The first phrase involves:

Updating & installing Packages

Configuring Wi-Fi adapters

Setting up the virtual machine

Disabling all kinds of hardware encryption

Study the current monitor modes of the wireless device being used.

In contrast to which the second phrase titled configuration the arrangement of all host-apd files, network.conf to work along with the established monitor modes.

IV. IMPLEMENTATION PHASE

The experiments we performed and the results that were obtained are described here.

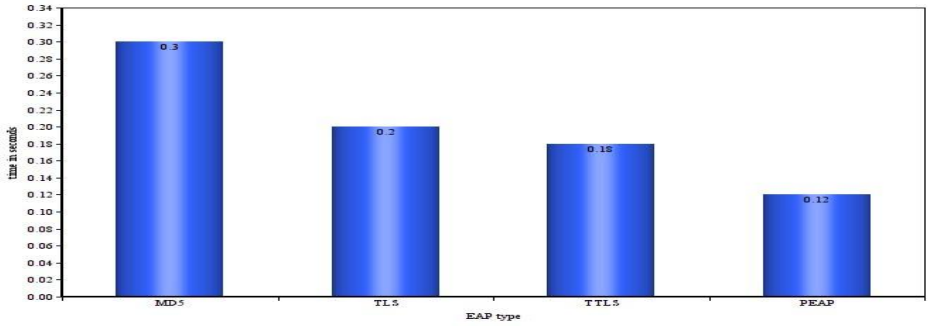
a) Development of the python script

The python script was developed using python 2.7 and used across Ubuntu 2016 LTS systems on Kali Linux 2017.1.

b) Implementation of the flaw

First, the network manager's Wi-Fi is disabled. Followed by removing unused virtual interfaces to start with a clean slate. After the interface monitor mode configuration, the patched host-apd instance carrying the test is opened and then ran. Close the host-apd file and clean up to handle scapy DHCP requests. Additionally, configure the IP gateway as a response to requests for ARP and ping. Manipulate the handshake messages after the 4-way handshake test.

c) Analysis of the study- The study is analyzed using the following graph:



This graph shows the time change for different types of EAPOL packets. Time is plotted in milliseconds inx-axis and EAPOL packet types are plotted in y-axis. The results obtained clearly decrypt all the information that had to be encrypted during the transfer of the packet.

V. CONCLUSION

We described in this paper our basic ideology to tackle the problem understanding the practical influence if the flaw across Linux systems using the iso images of the platforms. Such flaws clearly state that the protocols ' security properties should not be taken as formal proofs, thus highlighting the limitations of the models used by such protocols. The models in particular do not specify when the data confidentiality protocol should install a key for use. In the end, it allows an opponent to replay broadcast and multicast frames. When attacking the handshake of the 4-way or fast BSS transition, the precise impact depends on using the data confidentiality protocol. However, in all cases, frames can be decrypted and therefore TCP connections can be hijacked.

This allows data to be injected into non-encrypted HTTP connections. In addition, it is possible to trigger the installation of an all-zero key against Linux systems by implementing the flaw, completely eliminating any security guarantees. Even if certain handshake messages are lost due to background noise, our key reinstallation attack occurs spontaneously. This means that implementations reuse nuances under certain conditions without the presence of an adversary.

VI. REFFERENCES

[1] B. Vanhoef, Mathy, and Frank Piessens. "Key reinstallation attacks: Forcing nonce reuse in WPA2." Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2017.
[2] Abo-Soliman, Mohamed A., and Marianne A. Azer. "A study in WPA2 enterprise recent attacks." 2017 13th International Computer Engineering Conference (ICENCO). IEEE, 2017.
[3] Vanhoef, Mathy. "Key Reinstallation Attacks: Breaking the WPA2 Protocol." London: Black Hat Briefings Europe (2017).
[4] Luo, Qian, and Jiajia Liu. "Wireless Telematics Systems in Emerging Intelligent and
[5] Connected Vehicles: T hreats and Solutions." IEEE Wireless Communications 99 (2018): 1-7.
[6] Franks, John, et al. HTTP authentication: Basic and digest access authentication. No. RFC 2617. 1999.
[7] Langner, Ralph. "Stuxnet: Dissecting a cyberwarfare weapon." IEEE Security & Privacy 9.3 (2011): 49-51.
[8] Aboba, Bernard, et al. Extensible authentication protocol (EAP). No. RFC 3748. 2004.
[9] Dantu, Ram, Gabriel Clothier, and Anuj Atri. "EAP methods for wireless networks." Computer Standards & Interfaces 29.3 (2007): 289-301.