The 7th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications
12-14 September 2013, Berlin, Germany

Malware Injection in Wireless Networks

Matej Kacic[1] , Petr Hanacek[2] , Martin Henzl[3] , Peter Jurnecka[4]
[1] Brno University of Technology, FIT, Bozetechova 2, 612 66 Brno, Czech Rebuplic
{[1] ikacic, [2] hanacek, [3] ihenzl, [4] ijurnecka}@fit.vutbr.cz

***Abstract* – Nowadays we use the 802.11i standard, which is considered safe for now. In 2010, the Hole 196 vulnerability of Group Transient Key was discovered, which allows an undetectable insider ARP poisoning attack. On this ground we have founded the possibility of malware injection into wireless communication with purpose of avoiding intrusion detection system. This paper describes creation and injection of valid 802.11i frames with malware payload. We also discuss its impact on users at home and wide corporate wireless network.**

***Keywords* – Wireless network security, Malware injection, Group key vulnerability**

I. Introduction

In the last decade Wireless networks have become more and more popular in corporate and also home environment. Many devices like laptops, tablets, smartphones, even kitchen's devices have wireless network cards installed. Although several security-defence systems have been developed such as firewalls, encryption, authentication, and VPNs, most of the wireless systems are still susceptible to attacks. One of many possible type of attacks is an attack from inside of network. This paper shows how it is possible to do an attack from inside environment of network and inject some kind of malware to victim in order to abuse buffer overflow vulnerability of remote network service to gain, for example, a privileged access. First, we have to understand the key hierarchy and mainly group transient key vulnerability described in the following part.

A. Keys hierarchy

Wireless networks secured by 802.11i standard use several levels of cryptographic keys for different type of frames [1]. There are two possible top level keys that are used to generate the rest of the keys in the hierarchy. Those keys depend on chosen type of authentication of wifi network. The first key, pre-shared key, is used in home networks or in small business networks. On the other hand, huge corporate network use 802.1x [2], [3] authentication. Both keys serve to derive other key called Pairwise Master Key (PMK), which is then used for derivation of another key, Pairwise Transient Key (PTK). This key is unique for every connected client of wireless network and access point use this key for encrypting of communication between access point and wireless client.

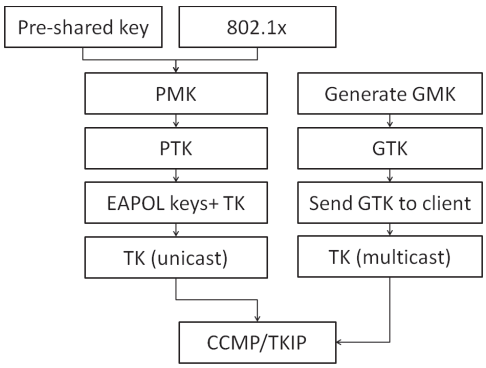

Figure 1. 802.11i keys hierarchy

Next keys in the hierarchy are EAPOL keys (KCK, KEK) that are used to establish secure communication and to update expired keys. Those keys are independent of temporal key (TK) used to encrypting data traffic between station and access point.

The problem with maintaining an individual key with each client becomes apparent when dealing with multicast and broadcast traffic. If N clients are associated, the AP will have to retransmit the frame N times, encrypting it with a different key each time. To avoid this, the AP generates a random group master key (GMK). Every time a client associates or disassociates, the AP derives a new group transient key (GTK) from the GMK. This GTK is delivered to each of the clients to be used to encrypt and they decrypt multicast and broadcast traffic.

When stations have connected to wireless network and all cryptographic keys have seen established, the encryption of traffic can begin. Standard 802.11 allows two encryption algorithms:

1) Temporal Key Integrity Protocol (TKIP) [4] was designed to replace WEP in existing wireless hardware.
2) Counter Mode with Cipher Block Chaining Message Code Protocol (CCMP) uses AES block cipher and provides high level of confidentiality, integrity and replay protection.

II. GTK key vulnerability

The purpose of GTK key is encrypting broadcast and multicast frames on access point side and decrypting

frames on authorized station side. Normally, all communication runs exactly as the standard defines, authorized wireless client never uses GTK key to encrypt broadcast or multicast frames and all traffic from client is sent directly to access point. For this part the driver of wireless network card is responsible, which doesn't allow changes in transmitting frames.

It could happen that authorized client starts to break the rules of standard and thus becomes an attacker. The main consequence of this vulnerability is that all authorized client successfully receive and accept this malicious frames encrypt with GTK key with assumption that they originate from access point. This vulnerability was published in 2010 in BlackHat conference [5], [6].

By this way an attacker can inject any type of broadcast and multicast frames into valid traffic and all authorized wireless clients connected to network accept this frame in faith that the frame came from access point and they consider this malicious frame as valid. There are many opportunities how this vulnerability could be used. The original paper of this vulnerability describes ARP poisoning attack, DNS manipulation and DoS attack as an example how to abuse this vulnerability.

It should be noted where the position of GTK key is in key hierarchy of 802.11i standard [7]. GTK key doesn't depend on authentication type and encryption algorithm, which implies that both authentications are vulnerable, with pre-shared key and even authentication via 802.1x eg. enterprise mode, and even both encryption modes (TKIP, CCMP) are vulnerable.

Our idea is to use this vulnerability to inject malware into wireless network without its detection by standard intrusion detection systems, usually deployed in wired segment of network and even without its detection by wireless intrusion detection systems based on signature detection like Kismet [8].

III. Frame crafting

First, we have to describe in short 802.11 frame format and frame crafting for better understanding of the idea of malware injection in wireless networks and, finally, the last part of this section discusses certain conditions for malware injection attack.

A. 802.11 frame

Frame header, defined in 802.11 standard [7], consists of fields such as control field, duration, sequence number, three or four address fields, payload (data), frame checksum. Basic form of 802.11 frame is shown in table I.

Table I. 802.11 Frame

Field	Control	Duration	Address1	Address2
Lenght	2B	2B	6B	6B
Address3	Sequence	Address4	Payload	FCS
6B	2B	6B	0B - 2312B	4B

The frame control field contains various flags. The most significant flags for us are:

1) Type/Subtype - Beacon, Data, etc.
2) ToDS - to the distribution system
3) FromDS - exit from the Distribution System.

Author of "Hole 196" vulnerability have implemented example attack with modifying linux driver. This approach is very simple, but it has many disadvantages:

1) Driver dependency
2) Platform dependency
3) Unable to create custom frame

We have chosen completely different approach based on custom frame crafting. First, we create MAC 802.11 frame, where we can simply set any field in frame header. For successful implementing of GTK key vulnerability we have to set FromDS flag, Address 1 field set to broadcast MAC address, Address 2 field set to BSSID MAC address, Addres 3 field set to attacker's MAC address.

We have known that every MAC frame from a node comes with a unique sequence number, which the node increments every time a frame is sent out. The purpose of the sequence number is to use to re-assemble fragments of frame [9] and intrusion detection system keeps track of the latest sequence number of each node. If the sequence number of injected frame is equal to or smaller than the current sequence number of corresponding node, the injected frame is considered as retransmitted frame eg. duplicated frame, hence it has to discard. If a frame sequence number is greater than current sequence number of corresponding node and smaller than window size, the system usually accepts the frame.

Very similar behaviour can be observed in sequence numbers used by TKIP or AES-CCMP, where integrity of frame is checked. The correctness of these sequence numbers is very important, because if sequence number checking fails, the frame is automatically dropped before its processing and the countermeasures are triggered. Therefore, the success of injection of frame depends on choosing the right value of sequence number.

It is obvious that frame must contain correct frame checksum and must be also encrypted with correct key, thus the next important step is the key extraction from driver. The GTK key can be easily extracted from driver of wireless nic by command tool called iwlist [10]. Here is an example of using this tool: *iwlist wlan1 enc*.

B. The Transport Layer

The transport layer exits on top of the network layer and it provides communication between processes. It extends IP address with collection of ports, each of which is capable of being the source or destination port address for communication between hosts. [11]

Two protocols operate at this layer: the User Datagram Protocol (UDP) or the Transmission Control Protocol (TCP). The main purpose of TCP is providing a reliable

stream between machines. It uses a three-way handshake to establish a reliable connection stream between two processes. First, a client sends a packet to the destination with SYN flag. This packet includes a random initialization for a sequence number. In response, the server replies with SYN-ACK packet, indicating that the server wishes to accept the connection. Finally, the client sends a final packet ACK with acknowledgement number.

TCP also incorporates a cumulative acknowledge scheme. Both communicating sides send data via their established TCP connection. After the sender sends the data to listening side, the receiver will confirm that it has received data by sending ACK packet.

On the other hand, UDP provides best-effort communication channel between two ports. In contrast to TCP, the UDP protocol does not guarantee reliable connections ei. does not guarantee correctness of packet delivery. It has no initial handshake to establish connection, but rather sends datagram immediately.

We have described the Transport layer, because the way to inject malware is different when we use TCP and UDP protocol. By using TCP protocol to inject malware we have to make a three-way handshake to establish reliable connection, send data and wait for ACK packet from server side. We can also ignore ACK packet sent by receiver to confirm that data has arrived, because it is not relevant for malware injection. Figure 2 shows in detail all necesary steps to achieve our goal. First, we make TCP three-way handshake with a victim by traditional way ie. an attacker and the victim communicate according to standard. After that we are ready to send malicious packet to the victim, but now it is the time to use GTK key vulnerability. We send this malicious packet directly to the victim - without access point.

C. Malware injection

This part describes our idea of how to inject malware to specific wireless client with purpose, for example, buffer overflow insertion attack to specific network application. We know that buffer overflow conditions exist when program allows to put into a buffer more data than it can hold. Attacker can use this vulnerability and insert malicios code into the memory of process and start execution. This can lead to gaining control of privileged application.

We also know ISO/OSI model of network communication, where each layer is responsible for certain function and send data to upper or lower layer. There is no checking of correctness of frame type between these layers. We can use this behaviour to create a special type of frame, where layer 2 will use GTK key vulnerability, thus it will be a broadcast frame encrypted by GTK key. The payload of layer 2 frame will be an IP packet (network layer) with victim destination address and malware in its payload. Simple example of custom made wireless frame with malware payload shows figure 3.

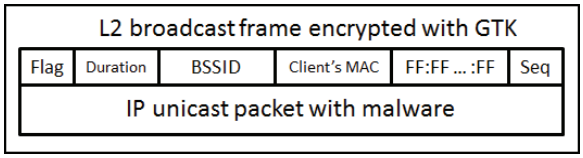

Figure 3. Crafted frame

Payload in IP packet highly depends on a network service on victim side. First, we have to find suitable type of malware with certain conditions. Very important condition is payload size, thus packet size must be smaller than 2312 bytes and vulnerable network service has to be explotaible by one IP packet, because we are not able to create valid communication between the attacker and the victim. We can only send one packet with malware payload. Next section shows two examples of using this type of attack.

IV. Results

For purpose of proof of our idea we create a simple network service with specific functionality. The main function of this service is to listen on some port and wait for IP packet with specific payload from network. If that packet arrives, program will write a message to terminal.

Attacker's station has one wireless card with two virtual interfaces. The first interface is in standard infrastructure mode (STA) and the second interface is in monitor mode (MON) with capability of frame injection. We assume that first interface is successfully authetificated in wireless network with pre-shared password or some corporate methods as described in introduction.

Implementation of application to create malware injection is realized by several important steps. First we have to extract GTK key from interface connected to wireless network. It is obvious that the key is necessary to encrypt the frame. After that we create a program loop (figure 4), where the program waits for broadcast frame sent by access point. We set a capture filter to find a data frame sent by AP (FromDS flag) and with address1 field set to broadcast MAC address, address2 field is set to BSSID address and third address field is set to sender MAC address.

Immediately after frame was captured, the program sends a custom frame with sequence number and initialization vector increased by one. This guarantees to us that frame will be accepted by victim and intrusion detection system would not detect that type of frame.

The idea of malware injection was successfully tested by above mentioned procedure. Now we can show an example attack from exploit-db database [12] using a vulnerability in ftp server with goal of establishing a remote shell on the target system that we can command.

We have chosen a vulnerable version of VSFTPD ftp server version 2.3.4. The remote FTP server that is

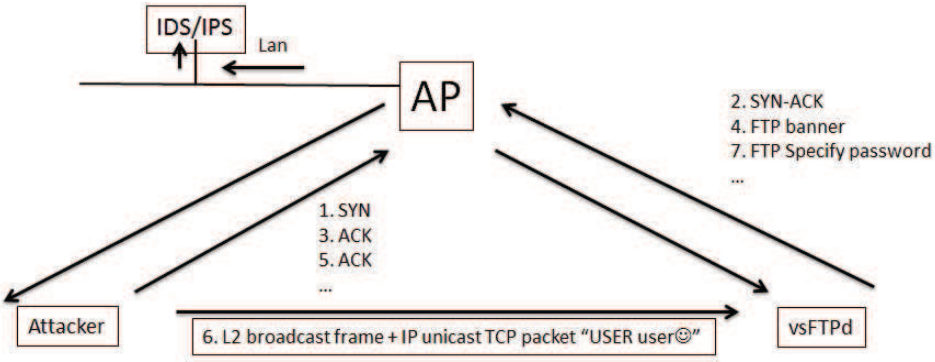

Figure 2. Crafted frame sent directly to victim

```
for (;;) {
capturedFrame=capture('type data and
wlan addr1 ff:ff:ff:ff:ff:ff and
wlan addr2 00:21:91:71:54:f2 and not
wlan addr3 70:f1:a1:59:19:0e and
dir fromds');

iv=getIV(capturedFrame);
newframe.initVector=iv+1;
key=getGTK();
newframe.key=key;

// malware encapsulated by upper
// TCP/IP layers
newframe.payload=malwarepayload;

send(newframe);
}
```

Figure 4. Program loop

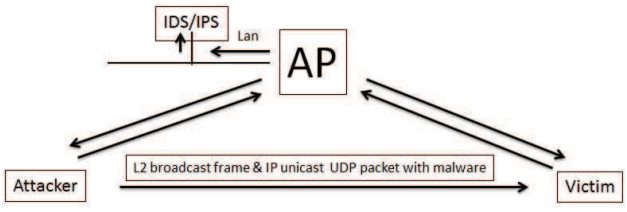

Figure 5. Crafted frame

running on our target host was compiled with a back-door. When attempting to login to the server with ":)" (a smiley face) as a username, the backdoor is triggered and listens for a connection on port 6200/TCP. After the client connects and then disconnects from the back-door, the shell will stop listening for a connection. [13]

As we have mentioned above, we have to handle three-way handshake with FTP server. After that we simply send a random username with smiley, read the response and send a random password. Now the ftp server have opened the 6200 port and we can connect back to attacker and spawn a command shell, for example we can use shell from metasploit called shell_reverse_tcp. It is obvious that we send username, password and shell-code directly (with GTK key vulnerability) to avoid detection on access point or to avoid detection by network IDS system.

V. Conclusion

Every day the number of mobile devices using wireless system is increasing. Laptops, tablets and even smart-phones are becoming part of our daily routine and we are securing them mostly by the newest security standards. On the other side, we are forgetting to ensure security from internal environment. Many corporates have extensive wireless networks with huge number of users. In many cases, we are not aware of importance of security from the inside of network.

We did a research on vulnerability called "Hole 196" dealing with issues of security inside wireless networks. In detail, we describes abusing of the vulnerability of the encryption key used for broadcast communication and consequences of this vulnerability are shown in malware injection attack. Impact of this kind of attack on user is major, because with an increasing number of users a risk of abuse of user privileges has increased. The result of this security incident is to compromise the target client by attacker. The proposed attack is an insider type of attack abusing user privileges and this kind of attack is able to bypass any traditional network intrusion detection system.

In accordance with security issues in this paper, our future research will be mainly focused on research in protection against insider attacks in wireless networks. We are planning to do some testing environment for evaluating and comparing wireless intrusion detection systems.

Acknowledgment

This work was supported by the European Regional Development Fund in the IT4Innovations Centre of Ex-

cellence project (CZ.1.05/1.1.00/02.0070), by the project CEZ MSM0021630528 Security-Oriented Research in Information Technology and by project FIT-S-11-1 Advanced Secured, Reliable and Adaptive IT.

REFERENCES

[1] Kevin Benton, *The Evolution of 802.11 Wireless Security*. UNLV Informatics, 2010-04-18 [cit. 2012-01-06].

[2] A. Earle, *Wireless security handbook*. Auerbach, 2005. [Online]. Available: http://books.google.cz/books?id=DojR6q1E5ZUC

[3] EAP Working Group, "Protected eap protocol (peap) version 2," http://tools.ietf.org/html/draft-josefsson-pppext-eap-tls-eap-10, 2004.

[4] Martin Beck, Erik Tews, "Practical attacks against wep and wpa," http://dl.aircrack-ng.org/breakingwepandwpa.pdf, 2008-11-08.

[5] AirTight Networks, "Wpa2 hole196 vulnerability: Exploits and remediation strategies," http://www.defcon.org/, 2010.

[6] AHMAD, Md, "Wpa too!" http://www.defcon.org/images/defcon-18/dc-18-presentations/Ahmad/DEFCON-18-Ahmad-WPA-Too-WP.pdf, 2010.

[7] *IEEE Std 802.11i-2004*. IEEE, 2004, iSBN 0-7381-4074-0.

[8] "Kismet [online]," http://www.kismetwireless.net, 2010 [cit. 2011-03-03].

[9] F. Guo and T.-c. Chiueh, "Sequence number-based mac address spoof detection," in *Proceedings of the 8th international conference on Recent Advances in Intrusion Detection*, ser. RAID'05. Berlin, Heidelberg: Springer-Verlag, 2006, pp. 309–329. [Online]. Available: http://dx.doi.org/10.1007/11663812_16

[10] "Linux wireless [online]," http://linuxwireless.org/en/users/Documentation/iw, 2013 [cit. 2013-01-20].

[11] M. Goodrich and R. Tamassia, *Introduction to Computer Security*. USA: Addison-Wesley Publishing Company, 2010.

[12] "Exploit-DB [online]," http://www.exploit-db.com/, 2013 [cit. 2013-01-20].

[13] "Metasploit - Penetration framework [online]," http://www.metasploit.com/, 2013 [cit. 2013-06-01].