Secure the wireless network at the packet level

In this article, I will show you some useful techniques for troubleshooting wireless network security issues at the packet level.

In this article, I will show you some useful techniques for troubleshooting wireless network security issues at the packet level.

The problem with wireless networks is that you can't see what you're dealing with. In a wireless network, establishing a connection is not as simple as in a wired network (just plug the cable), physical security is not simply a matter of hindering or completely preventing network access of individuals are not licensed, and troubleshooting even simple problems sometimes has its own difficulties associated with access points. That said, securing wireless networks will continue to be a challenge for the foreseeable future.

In this article, I will show you some useful techniques for troubleshooting wireless problems at the packet level. The article will begin by introducing methods to properly collect data packets in the wireless network. After collecting, we will discuss analytical techniques, including analyzing WEP / WPA authentication, filtering encrypted traffic and finding fake access points.

Capture wireless data packets

Packet levels in wireless networks and wired networks have some similarities. Wireless networks still use TCP / IP for data communication and comply with all connection rules of wired hosts. The differences between these two connection platforms are found at lower levels in the OSI reference model. Wireless networks carry out communications by sending data over the air, completely different from sending data by signal wires. The space that wireless data is communicated on is a shared environment, which is why a special consideration is needed in the data link layer and the physical layer to ensure that there is no data. Any data is conflicting and data is distributed reliably.

This is equivalent to troubleshooting a wireless network because we still need a certain number of attempts to capture the second layer of 802.11 data packets needed for troubleshooting in one way. satisfactory. To do this, you must be able to put your wireless network interface card (WNIC) into special mode called Monitor Mode. Monitor mode is a special driver setup, which limits the ability of WNICs to send data and only allows passive listening on the selected channel.

In Linux-based operating systems, you can easily change the WNIC to monitor mode, but most Windows drivers do not allow this function. As a result, we need to use a special piece of hardware to help it work. This piece of hardware is called AirPcap and is manufactured by CACE Technologies. AirPcap device is basically a WNIC designed for use in test mode with Windows and Wireshark packet capture utility. Using this device, you can capture the second layer of 802.11 data packets from the wireless channel listening.

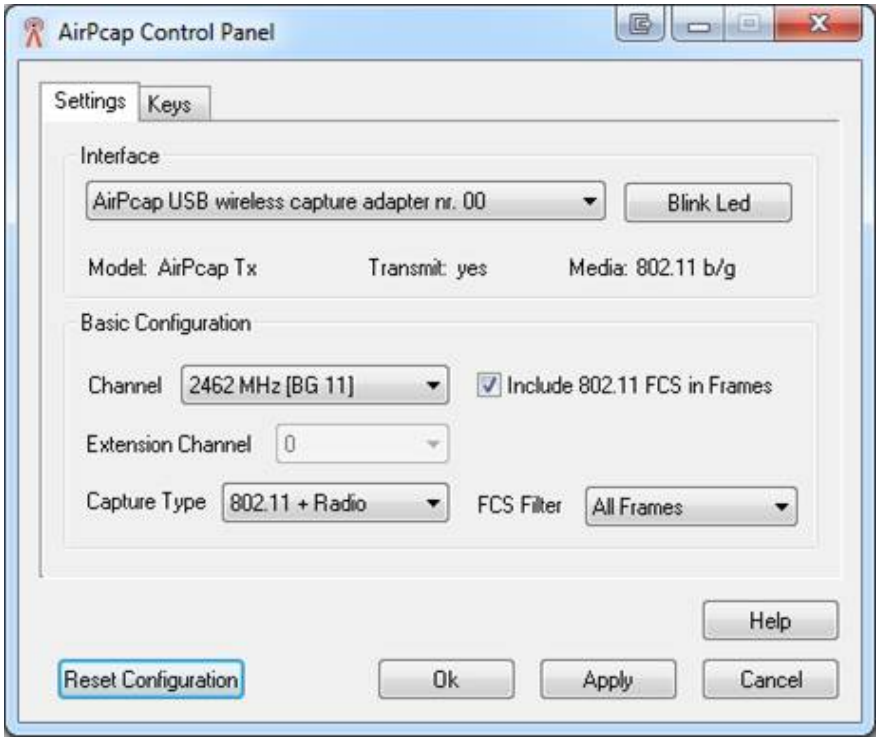

Figure 1: The AirPcap configuration screen allows you to configure the channel you are listening to

802.11 data packet structure

The main difference between wireless and wired network packets is the header. This is a two-layer header containing extended information about the package and the environment it is conveying. There are three types of packages here: data packets, management packs, and control packs.

- **Management package** - These packages are used to establish connections between hosts at layer two. Some important types in these managed data packages are: authentication, linking and signaling.
- **Control** package - **Control** packets allow distribution of management and data packages and are related to congestion management. Types of control packages include: Request-to-Send and Clear-to-Send.
- **Data** packets - These packages contain real data and are packet types that are forwarded from the wireless network to a wired network.

In this article we will not introduce these types of 802.11 packets, but focus on some areas related to security.

Find fake access points

Physical security for IT assets is the most frequently considered security domain. One of the most common oversights in this area is the addition of unauthorized devices in the network. In the world of wired networks, an unauthorized router can cause a denial of service. Even so, a fake WAP wireless access point is always a bigger concern because it can allow someone from outside to increase access to the network as if they had suddenly stolen and stolen a laptop. their with the network jack on the wall.

Luckily for us, detecting a WAP can be done quite easily. To do this, you must start by capturing wireless traffic from some areas within the broadcast range of networks. When this is done, there will be several different filters used to detect whether there are fake access points and whether the clients are involved in them.

One of the simplest solutions to do that is to know the MAC address of the legitimate WAP. By using this information, you can perform filtering **! Wlan.bssid == 00: 11: 88: 6b: 68: 30** , replace your WAP MAC in the MAC address location we provide. This will show you all wireless traffic that is rotating in and out of WAP. If there are multiple WAPs in the area, you can combine these filters with the OR operator (||). In that case, you can use **! Wlan.bssid == 00: 11: 88: 6b: 68: 30 || ! wlan.bssid == 00: 11: ff: a1: a4: 22** to filter out two known valid access points.

That method can help you detect access points in general, but what if you want to go further and want to find out if your mobile stations are connecting to fake WAPs? or not? One way to do this is to filter link requests. To do this, you can combine one of the above filters with filters **wlac.fc.type_subtype eq 0** and **wlac.fc.type_subtype eq 2** . The first filter will show all connection requests and the second will show reconnection requests. When necessary, you can combine one of these filters with the above filters using the AND operator (**&&**).

Finally, you can go a step further by determining whether any real data is being transferred between the fake mobile clients and WAP. It can be done by filtering all data packets that are being communicated with illegal access points by using the filter **wlan.fc.type eq 2** in combination with the previous filters, except for valid WAPs. known.

Filter unencrypted traffic

The only hope in preventing data packets from being eavesdropped when they are played in the air is to use encryption. This encryption is done by executing WPA or WPA2 in modern systems. That said, we need to authenticate our wireless networks more often to ensure that no wireless clients transmit data unencrypted.

Finding unencrypted data in the wireless network needs to use another filter. In this case, we can find all unencrypted data using the **wlan.fc.protected == 0.** filter. **Now if you use it immediately, you will see some results returned. not as expected.** 802.11 **control and management frames** are not encrypted but only perform administrative functions for WAPs and wireless clients. With this problem, we have to extend the filter by appending to **wlan.fc.type eq 2** . This will ensure that the filter displays only unencrypted packets. The last filter will be **wlan.fc.protected == 0 && wlan.fc.type eq 2** .

Technical analysis of WEP and WPA authentication

The previous method used for securing data when playing in wireless networks is Wired Equivalent Privacy (WEP). WEP was quite successful in the previous years until it discovered some weaknesses in its encryption key management problem. As a result, new standards have been introduced, including Wi-Fi Protected Access (WPA) and WPA2. Although WPA and WPA2 are not really safe, it is considered to be more secure than WEP.

It is quite useful to be able to distinguish WEP and WPA authentication on wired networks. If you can do this, you can remove the WEP authentication type on your network and switch to WPA. With that, you need to be able to analyze failed attempts to realize them when they happen.

WEP authentication

WEP authentication works by using a request and response mechanism. When the client tries to connect to WAP, WAP will issue the requested text. This request is answered and then the client will use this text, decrypt it with the WEP key provided by the client and generate the aggregate string back to WAP.

When the WAP verifies that the response text is necessary, it will issue a message back to the client to inform that the authentication process was successful. The filter for successful response responses is *wlan_mgt.fixed.status_code == 0x0000* .

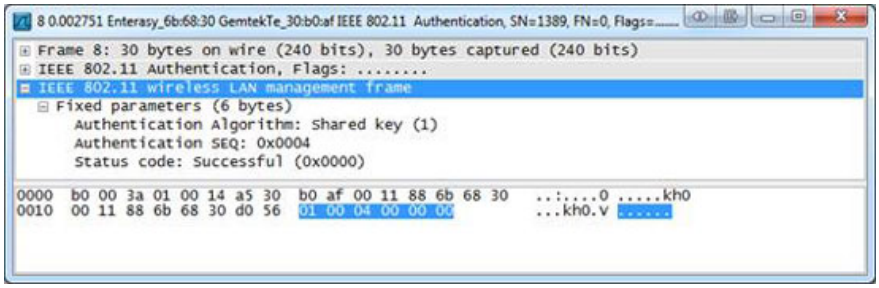

Figure 2: WAP alerts the client that the authentication process has been successful

In the case of unsuccessful authentication, WAP will issue a message stating that it has failed to authenticate.

Figure 3: WAP alerts the client that the authentication process has failed

The filter to find the failure notification package is **wlan_mgt.fixed.status_code == 0x000e** .

WPA authentication

WPA authentication also uses a request and response mechanism, but it works in a different way. At the packet level, WPA authentication uses EAPOL to perform its request and response. You can find these packets using the simple **EAPOL** filter **. During the successful authentication process, you will see four** EAPOL **packages** corresponding to the two requests and two responses. Each request and response can be paired using the Replay Counter value within the package.

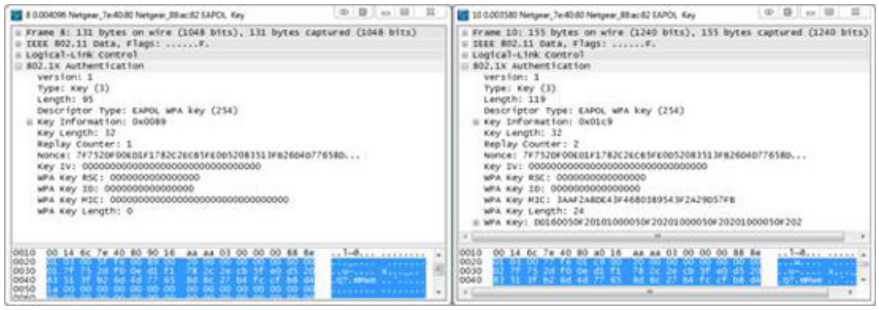

Figure 4: Replay Counter field is used to fix some problems and responses

In a situation where the WPA authentication process fails, you will use some EAPOL filters where the request and response are attempted multiple times. When this process really fails, you will see an authentication package.

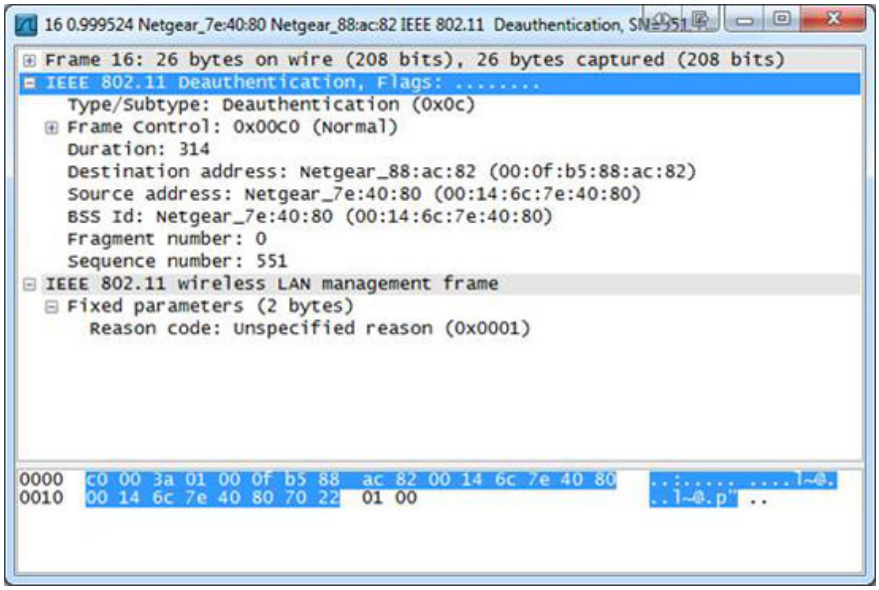

Figure 5: After the WPA handshake fails, the client will perform the authentication again

There are two ways to filter failed WPA authentication. The first way is to use the EAPOL filter and count the number of packets transferred between the WAP and the wireless client. The other way is to filter the real deal with the **wlan.fc.type_subtype == 0x0c** filter. This filter will return some additional results unrelated to the failed authentication process so that in order to be able to verify that the packages are related to this problem, you need to dig deeper and create a Other filters contain all data packets between the WAP and the wireless client mentioned.

Conclude

In this article, I have shown you some basic details about capturing network packets and some important applications for packet analysis from a wireless security standpoint. Wireless connectivity, wireless security, packet analysis, all are broad topics. This article cannot be presented to you in detail but only arouses further research into these areas. If you are particularly concerned about analyzing data packets, download Wireshark or another data capture utility and start your packet analysis.