CCTV Interception

Made by: Noodle Hackerspace.
Published by: AT Products LLC.
Published on: May 17th, 2023.

Prenote

Needed Items

A computer with Wireshark and AirCrack, a Wi-Fi adapter that supports monitoring, and some time.

Keywords

`{ }` means commands for the terminal. `( )` means general terms.

To make this thing work, we will need to use Wireshark to sniff out traffic between our hosts. The goal is to capture unencrypted HTTP traffic to the hosts' computer.

First things first, we have to break the encryption of the network. Even if you have the password, do not connect quite yet, it will bring attention to us and open up more risk of detection. Instead, we can add some Wi-Fi keys we know to Wireshark. To decrypt the data, we sniff without connecting to the network. This means our attack will be mostly passive, leaving minimal opportunity for us to be detected.

One serious thing we'll need that is not passive is a Wi-Fi handshake to see the ongoing traffic. Because since Wireshark needs to observe a Wi-Fi handshake to decrypt subsequent traffic, the computer we're interested in with an awesome filter, captures a four-way WPA handshake and then decrypts the data with the password we know.

Conditions must be favorable for this attack to have a chance of succeeding. In particular, if the camera does not use an insecure interface, then the data will be encrypted, and we will not be able to see it.

If no one is watching the camera feed, or it's not displayed on a monitor, there will be no insecure traffic to intercept, so we will not see anything. If we cannot kick a client off the network to generate a four-way handshake, then knowing the password won't do us any good. And finally, if we're out of range of the network, we won't be able to intercept traffic we can't clear.

While this may seem like a lot of requirements, it's fairly common to be able to do this. If the target has a Wi-Fi security camera and keeps a monitor viewing the display, the Wi-Fi password should be all you need, aside from a Wi-Fi adapter.

You should be ready once you're in range and have any operating system loaded with the required tools. Plug in your adapter, and make sure you have Wireshark installed to start. If you don't have Wireshark, `sudo apt install Wireshark -y` or `sudo pacman -S Wireshark -y`.

Step 1

To start, access the built-in interface on whatever webcam or Wi-Fi security camera you want to intercept. In a browser window on your "host" computer, navigate to the HTTP interface, enter any password required, and begin viewing the live webcam view.

If you need to find your camera on the network, you can run a Nmap scan to discover different devices on the network running insecure HTTP ports.

For this command, you'll need to know the network range. you can find this by typing `ifconfig` and copying down the IP address assigned to your computer. Then, you can type `ipcalc` and your IP address to calculate the network range. It should be something like `192.168.0.0/24`. Run the following command but replace "IP" for your network range. `sudo nmap -p 80,81,8080, 8081 IP`

This will look for devices with these ports "open", and when you find one, you can navigate it by typing the IP address and then `:81` on that IP address. If you want to navigate to port `80801` type `192.168.0.1:8081` to your browser window.

Step 2

Plug in your Wi-Fi adapter (preferred "Alfa AWUSO36NHA"). Two things are required before starting up Wireshark, the first being putting the card into wireless monitor mode, and the second being identifying the channel the router you're targetting is broadcasting on.

To put your card into wireless monitor mode, identify the name of your card by running `ifconfig` in a terminal window. It should be named something like wlan0 or wlan0mon.

Once you've found the name of your wireless card, we'll need to put it into monitor mode. Run the following command in a terminal window, with the name of your card substituted for wlan0.

```
airmon-ng start wlan0
```

```
airodump-ng start wlan0mon
```

This will put your card in the wireless monitor more, changing the name of the card to add "mon" at the end. It will also start Airodump-ng, which will start scanning for nearby wireless networks.

Please be sure to look for the Wi-Fi network that you're looking to sniff, and note the channel that it's on. We'll need to switch our card to that channel to intercept the images in Wireshark. The following is a sample output. Some of the output has been converted to a table. It is still a part of the console output.

```
CH  4 ][ Elapsed: 0 s ][ 2018-12-24 02:42
```

BSSID	PWR	Beacons	#Data,	#/s	CH	MB	ENC	CIPHER	AUTH	ESSID
C0:8A:DE:39:CD:D9	-46	2	0	0	1	130	WPA2	CCMP	MGT	TWCWiFi-Passpoint
C0:8A:DE:F9:CD:D8	-47	2	0	0	1	130	OPN			TWCWiFi

```
C0:8A:DE:B9:CD:D8  -46  2  0  0  1  130  OPN                    SpectrumWiFi
C0:8A:DE:39:CD:D8  -47  2  0  0  1  130  OPN                    CableWiFi
78:96:84:00:B5:B0  -42  2  0  0  1  130  WPA2 CCMP     PSK       The Daily Planet
00:9C:02:D2:5E:B9  -60  3  0  0  1  54e.  WPA2 CCMP     PSK       HP-Print-B9-Officejet Pro 8600
20:10:7A:92:76:43  -51  2  0  0  1  130  WPA2 CCMP     PSK       SBG6580E8
DE:F2:86:EC:CA:A0  -45  1  0  0  11 195  WPA2 CCMP     PSK       Bourgeois Pig Guest
D6:04:CD:BD:33:A1  -55  1  0  0  11 130  WPA2 CCMP     PSK       DirtyLittleBirdyFeet

BSSID              STATION            PWR   Rate    Frames  Lost    CCMP   Probe

root@kali:~/Desktop#
```

If our target is on channel 11, we'll run the following command to set our card to channel 11.

```
airmon-ng start wlan0mon 11
```

Step 3

Now that our wireless network adapter is listening on the same channel as the traffic we want to intercept, it's time to start Wireshark. When Wireshark opens, double-click the card you put in monitor mode to start the capture. Our card should now be scanning on the correct channel, but without the network password, we won't be able to see anything. To solve that, we'll need to add some encryption keys to Wireshark.

Step 4

To add the encryption keys to Wireshark, click on "Edit" in the menu bar, then "Preferences" to show the preferences menu. Next, select "Protocols" from the sidebar to see a list of protocols that Wireshark can translate.

In the Protocols drop-down menu you just opened, select `IEEE 802.11` to show options for decrypting Wi-Fi. Make sure that the "Enable Decryption" box is checked, and then click the "Edit" button next to "Decryption keys" to open the list of keys Wireshark will try to use to decrypt traffic.

Once the WEP and WPA decryption key menu is open, click on the field to the left and select "pa-psw" to add. While we can also add a `wpa-psk` here, we would have to calculate it ourselves, which is more complicated than simply entering the password.

For the decryption to work, you must add the key by clicking on the plus `(+)` icon, and then enter the key in the format `(password:networkname)` to add to the list.

Click "OK" to save the key, and now we should be able to decrypt traffic from his network if we can grab a four-way Wi-Fi handshake.

Step 5

In our Wireshark capture, we're sure to be seeing a lot of traffic. While we can't yet decrypt it because we don't have a handshake, we can build a filter to make sure we're only seeing traffic to the device we're sniffing.

The best way to do this over a Wi-Fi network is to find a piece of traffic to the computer we're looking for, and then make a display filter to show only packets heading to that MAC address. That means that any traffic directed to the target computer will be displayed, and any other network traffic will be ignored.

Looking under the packet information, right-click the "Receiver address" for a packet being sent to the target device, select "Apply as Filter", and then "Selected". Now, we should see only packets to the target.

Step 6

Now that we've isolated the traffic from our target device, we need to generate a four-way handshake by kicking the target computer off the network momentarily while Wireshark is listening. To do this, we can use a tool named MDK3, which can kick any devices connected to Wi-Fi off and generate a handshake. Highly suggest looking into using MDK3 for Advanced Wi-Fi jamming.

Because we already know the channel our Wi-Fi network is on, we can use MDK3 to take out any device operating on that channel. You should not need long to generate a WPA handshake. With "wlan0mon" swapped for the name of your wireless card, and "11" swapped for the channel you're attacking, run the following command in a terminal window to start jamming the network.

```
{ mdk3 wlan0mon d -c 11 }
```

After a few moments, nearby devices on the network should automatically reconnect, allowing you to intercept the WPA four-way handshake. If you want to make sure you have it, you can open a new terminal and run Airodump-ng to see when you get a WPA handshake. To do so, type:

```
{

airodump-ng wlan0mon 11

}
```

Substituting "wlan0mon" and "11" for your actuals to watch for WPA handshakes while you run MDK3.

Once you see the result above, you've captured a WPA four-way handshake! Make sure to match the MAC address shown with the wireless network you're targeting to avoid a handshake for the wrong network.

Now that we have a four-way handshake and have entered the network key, we should have full access to data flowing over the network. While HTTPS is still on the table, we should be able to see raw HTTP just fine.

Step 7

While we've gained access to the network traffic and narrowed it down to the target computer, there may be other traffic that's unrelated and makes it difficult to focus on what we're looking for. To cut through this, we'll add another network filter to show only HTTP traffic flowing through the network.

In the Wireshark main view, type HTTP into the display filter bar.

Step 8

Now that we can see the HTTP traffic from the web app, we'll need to select the encoded JPEG files to turn them into something we can work with. Stop the capture, and then click on "File", then "Export Objects." We'll be exploring the HTTP objects we've found, so click on "HTTP" to open the object list.

In the HTTP object list, we'll see a list of HTTP objects we've intercepted. Here we can see the JPEG images we want to decode. You can select one or all of them, and then click "Save" or "Save All" and pick a location to export the files to.

Click "Close", and then navigate to the folder you exported the images to. You should see a list of files that Wireshark exported from our capture. This will be more or less depending on how long you ran the capture.

Finally, click on one of the images to see the image that was intercepted on the way to the target computer. You should see a frame from the video feed!