Continue

Kali Linux is a Debian-based distribution specifically tailored to penetration testing, digital forensics, and security auditing. Security professionals and researchers widely use it for penetration testing, network analysis, and ethical hacking.AI-GeneratedWith the help of Airodump-ng, you can capture packets from multiple wireless networks simultaneously and filter them based on different criteria, such as the BSSID, the channel, or the packet type.Airodump-ng is a powerful tool for capturing wireless packets and is widely used by security professionals and researchers for wireless network analysis, penetration testing, and ethical hacking.This blog post will take a closer look at Airodump-ng and its capabilities. We'll cover the installation process, explain how to use Airodump-ng to capture wireless packets, and discuss some of its key features. By the end of this post, you will have a good understanding of how Airodump-ng works and how you can use it to improve your wireless security testing workflow.Getting started with Airodump-ngAirodump-ng is a wireless penetration testing tool that comes pre-installed with Kali Linux. However, in some cases, you may need to install it manually. In this section, we will go over the steps for installing Airodump-ng on Kali Linux.PrerequisitesA wireless card that supports monitor modeKali Linux operating systemOpen a terminal in Kali Linux and typesudo apt install aircrack-ngThis command will install the Airodump-ng and other tools in the Aircrack-ng suite. You can now use Airodump-ng by typing 'airodump-ng' in the terminal.Please note that if you are facing issues while installing Airodump-ng, you might want to check if your wireless card supports monitor mode. Also, ensure you have the latest version of Kali Linux installed. Sometimes, you may need to update the package list using the command 'apt-get update' before installing Airodump-ng.Also, please note that using Airodump-ng or other wireless penetration testing tools may be illegal in some countries and regions. Before using Airodump-ng, please make sure to check your local laws and regulations and obtain the necessary permissions.UsageAirodump-ng is used to capture wireless packets and save them to a file for later analysis. This section will explain how to use Airodump-ng to capture wireless packets and provide examples of common use cases.To start capturing wireless packets, you will need to use the following command:sudo airodump-ng Where an interface is the name of the wireless card you want to use to capture packets; for example, if your wireless card is named wlan0, you would use the following command:sudo airodump-ng wlan0You can also specify the channel and the BSSID of the wireless network you want to capture packets from by using the -c and -b flags, respectively. For example, to capture packets from the network with BSSID 00:11:22:33:44:55 on channel 1, you would use the following command:sudo airodump-ng -c 1 -b 00:11:22:33:44:55 wlan0You can also use Airodump-ng to save the captured packets to a file by using the -w flag. For example, to save the captured packets to a file named capture.pcap, you would use the following command:sudo airodump-ng -w capture wlan0Here are some common use cases for Airodump-ng:Wireless network analysis: Airodump-ng can analyze captured packets to gain insights into the network's behavior and security.Wireless penetration testing: It can be used to identify vulnerabilities in the network.Wireless audit: It can also be used to identify security issues and compliance with industry standards.WPA/WPA2 cracking: We can use it to crack the WPA/WPA2 key.Advanced Features of Airodump-ngIn this section, we will take a closer look at some of the key features of Airodump-ng and provide code samples to demonstrate how they can be used.Supports multiple output formatsAirodump-ng supports multiple output formats such as pcap, pcap-ng, and ivs. This is useful for compatibility with other tools or for specific use cases. For example, the following command saves the captured packets in the pcap-ng format:airodump-ng -w capture --output-format pcapng wlan0These are just a few examples of the many features that Airodump-ng offers. As a security professional or researcher, you can use Airodump-ng in many different ways to improve your wireless security testing workflow and achieve your goals. The ability to capture packets from multiple wireless networks simultaneously, filter them based on different criteria, and save them to a file for later analysis are some of the key features that make Airodump-ng a powerful tool for wireless network analysis, wireless penetration testing, and ethical hacking.Real-time monitoringAirodump-ng allows you to monitor wireless networks in real time, providing you with a clear view of the wireless traffic, clients, and devices connected to the network and the type of encryption used. This feature is useful for identifying potential vulnerabilities and analyzing the behavior of the network.Client trackingAirodump-ng allows you to track clients connected to a wireless network. This feature is useful for identifying connected devices and analyzing their behavior, and it can also be used to track rogue devices connected to a network.Deauthentication and Disassociation attackAirodump-ng allows you to perform de-authentication and disassociation attacks. This feature is useful for testing the robustness of wireless networks and identifying vulnerabilities, and it can also be used to disconnect clients from a wireless network.ConclusionIf you're interested in learning more about Airodump-ng and its capabilities, here are some additional resources that you may find useful:Please note that the usage and the information provided in this post may change according to the version or the operating system you are using. Also, depending on the nature of the task you are trying to do with Airodump-ng, you may need to install additional libraries and packages. This script can be used to enable monitor mode on wireless interfaces. It may also be used to kill network managers, or go back from monitor mode to managed mode. Entering the airmon-ng command without parameters will show the interfaces status. usage: airmon-ng [channel] or airmon-ng Where: indicates if you wish to start or stop the interface. (Mandatory) specifies the interface. (Mandatory) [channel] optionally set the card to a specific channel. "check" will show any processes that might interfere with the aircrack-ng suite. It is strongly recommended that these processes be eliminated prior to using the aircrack-ng suite. "check kill" will check and kill off processes that might interfere with the aircrack-ng suite. For "check kill" see ~# airmon-ng PHY Interface Driver Chipset phy0 wlan0 ath9k_htc Atheros Communications, Inc. AR9271 802.11n When putting a card into monitor mode, it will automatically check for interfering processes. It can also be done manually by running the following command: ~# airmon-ng check Found 5 processes that could cause trouble. If airodump-ng, aireplay-ng or airtun-ng stops working after a short period of time, you may want to kill (some of) them! PID Name 718 NetworkManager 870 dhclient 1104 avahi-daemon 1105 avahi-daemon 1115 wpa_supplicant This command stops network managers then kill interfering processes left: ~# airmon-ng check kill Killing these processes: PID Name 870 dhclient 1115 wpa_supplicant Note: It is very important to kill the network managers before putting a card in monitor mode! ~# airmon-ng start wlan0 Found 5 processes that could cause trouble. If airodump-ng, aireplay-ng or airtun-ng stops working after a short period of time, you may want to kill (some of) them! PID Name 718 NetworkManager 870 dhclient 1104 avahi-daemon 1105 avahi-daemon 1115 wpa_supplicant PHY Interface Driver Chipset phy0 wlan0 ath9k_htc Atheros Communications, Inc. AR9271 802.11n (mac80211 monitor mode vif enabled for [phy0]wlan0 on [phy0]wlan0mon) (mac80211 station mode vif disabled for [phy0]wlan0) As you can see, it created a monitor mode interface called wlan0mon and it notified there are a few process that will interfere with the tools. ~# airmon-ng stop wlan0mon PHY Interface Driver Chipset phy0 wlan0mon ath9k_htc Atheros Communications, Inc. AR9271 802.11n (mac80211 station mode vif enabled on [phy0]wlan0) (mac80211 monitor mode vif disabled for [phy0]wlan0mon) Don't forget to restart the network manager. It is usually done with the following command: service network-manager start This describes how to put your interface into monitor mode. After starting your computer, enter "iwconfig" to show you the current status of the wireless interfaces. It likely looks similar the following output. Enter "iwconfig": lo no wireless extensions. eth0 no wireless extensions. wifi0 no wireless extensions. ath0 IEEE 802.11b ESSID:"" Nickname:"" Mode:Managed Channel:0 Access Point: Not-Associated Bit Rate:0 kb/s Tx-Power:0 dBm Sensitivity=0/3 Retry:off RTS thr:off Fragment thr:off Encryption key:off Power Management:off Link Quality:0 Signal level:0 Noise level:0 Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0 Tx excessive retries:0 Invalid misc:0 Missed beacon:0 If you want to use ath0 (which is already used): airmon-ng stop ath0 And the system will respond: Interface Chipset Driver wifi0 Atheros madwifi-ng ath0 Atheros madwifi-ng VAP (parent: wifi0) (VAP destroyed) Now, if you do "iwconfig": System responds: lo no wireless extensions. eth0 no wireless extensions. wifi0 no wireless extensions. You can see ath0 is gone. To put wifi0 in monitor mode: airmon-ng start wifi0 System responds: Interface Chipset Driver wifi0 Atheros madwifi-ng ath0 Atheros madwifi-ng VAP (parent: wifi0) (monitor mode enabled) Now enter "iwconfig" System responds: lo no wireless extensions. eth0 no wireless extensions. wifi0 no wireless extensions. ath0 IEEE 802.11g ESSID:"" Mode:Monitor Frequency:2.452 GHz Access Point: 00:0F:B5:88:AC:82 Bit Rate=2 Mb/s Tx-Power:18 dBm Sensitivity=0/3 Retry:off RTS thr:off Fragment thr:off Encryption key:off Power Management:off Link Quality=0/94 Signal level=-96 dBm Noise level=-96 dBm Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0 Tx excessive retries:0 Invalid misc:0 Missed beacon:0 You can see ath0 is in monitor mode. Also make sure the essid, nickname and encryption have not been set. The access point shows the MAC address of the card. The MAC address of the card is only shown when using the madwifi-ng driver. Other drivers do not show the MAC address of the card. If ath1/ath2 etc. is running then stop them first prior to all the commands above: airmon-ng stop ath1 You can set the channel number by adding it to the end: airmon-ng start wifi0 9 To confirm that the card is in monitor mode, run the command "iwconfig". You can then confirm the mode is "monitor" and the interface name. For the madwifi-ng driver, the access point field from iwconfig shows your the MAC address of the wireless card. To determine the current channel, enter "iwlist channel". If you will be working with a specific access point, then the current channel of the card should match that of the AP. In this case, it is a good idea to include the channel number when running the initial airmon-ng command. It depends on which driver you are using. For all drivers except madwifi-ng: airmon-ng stop For madwifi-ng, first stop ALL interfaces: airmon-ng stop athX Where X is 0, 1, 2 etc. Do a stop for each interface that iwconfig lists. Then: wlanconfig ath create wlandev wifi0 wlanmode sta See madwifi-ng site documentation. For mac80211 drivers, nothing has to be done, as airmon-ng keeps the managed interface alongside the monitor mode one (mac80211 uses interface types rather than modes of operation). If you no longer need the monitor interface and want to remove it, use the following: airmon-ng stop monX X is the monitor interface number - 0 unless you run multiple monitoring interfaces simultaneously. airmon-ng has two options to show more information, which can be useful when reporting or debugging issues. It gives information about the system as well as details about the wireless card. root@kali:~# airmon-ng --verbose No LSB modules are available. Distributor ID: Kali Description: Kali GNU/Linux Rolling Release: 2019.1 Codename: n/a Linux kali 4.19.0-kali4-amd64 #1 SMP Debian 4.19.28-2kali1 (2019-03-18) x86_64 GNU/Linux Detected VM using lspci This appears to be a VMware Virtual Machine If your system supports VT-d, it may be possible to use PCI devices If your system does not support VT-d, you can only use USB wifi cards K indicates driver is from 4.19.0-kali4-amd64 V indicates driver comes directly from the vendor, almost certainly a bad thing S indicates driver comes from the staging tree, these drivers are meant for reference not actual use, BEWARE ? indicates we do not know where the driver comes from... report this X[PHY]Interface Driver[Stack]-FirmwareRev Chipset Extended Info K[phy1]wlan0 ath9k_htc[mac80211]-1.4 Qualcomm Atheros Communications AR9271 802.11n mode managed In this case, the following additional information can be seen: Detailed information about the Linux distribution as well as kernel version System is a virtual machine (and detailed information about supported features) Detailed driver information (kernel, vendor driver, staging or unknown source), wireless stack, current operating mode and firmware version It will give the same information as verbose and add more details: root@kali:~# airmon-ng --debug /bin/sh -> /usr/bin/dash SHELL is GNU bash, version 5.0.3(1)-release (x86_64-pc-linux-gnu) Copyright (C) 2019 Free Software Foundation, Inc. License GPLv3+: GNU GPL version 3 or later < This is free software; you are free to change and redistribute it. There is NO WARRANTY, to the extent permitted by law. No LSB modules are available. Distributor ID: Kali Description: Kali GNU/Linux Rolling Release: 2019.1 Codename: n/a Linux kali 4.19.0-kali4-amd64 #1 SMP Debian 4.19.28-2kali1 (2019-03-18) x86_64 GNU/Linux Detected VM using lspci This appears to be a VMware Virtual Machine If your system supports VT-d, it may be possible to use PCI devices If your system does not support VT-d, you can only use USB wifi cards K indicates driver is from 4.19.0-kali4-amd64 V indicates driver comes directly from the vendor, almost certainly a bad thing S indicates driver comes from the staging tree, these drivers are meant for reference not actual use, BEWARE ? indicates we do not know where the driver comes from... report this X[PHY]Interface Driver[Stack]-FirmwareRev Chipset Extended Info getStack mac80211 getBus usb getdriver() ath9k_htc getchipset() Qualcomm Atheros Communications AR9271 802.11n BUS = usb BUSINFO = 0CF3:9271 DEVICEID = getFrom() K getFirmware 1.4 K[phy1]wlan0 ath9k_htc[mac80211]-1.4 Qualcomm Atheros Communications AR9271 802.11n mode managed Additional information: Debug information regarding the wireless adapter and loaded driver Quite often, the standard scripts on a linux distribution will setup ath0 and or additional athX interfaces. These must all be removed first per the instructions above. Another problem is that the script set fields such as essid, nickname and encryptions. Be sure these are all cleared. ~# airmon-ng stop wlan0mon PHY Interface Driver Chipset phy0 wlan0mon ath9k_htc Atheros Communications, Inc. AR9271 802.11n You are trying to stop a device that isn't in monitor mode. Doing so is a terrible idea, if you really want to do it then you need to type 'iw wlan2mon del' yourself since it is a terrible idea. Most likely you want to remove an interface called wlan[0-9]mon If you feel you have reached this warning in error, please report it. It most likely mean the interface mode was changed from monitor to managed mode by a network manager. In this case, when stopping monitor mode, this is not a problem. It usually means the interface was put in monitor mode prior to killing network managers. And the network manager put the card back in managed mode. Refer to the documentation above to kill network managers and put it back into monitor mode. The original problem description and solution can be found in this forum thread. Problem: Every time the command "airmon-ng start wifi0 x" is run, a new interface is created as it should, but there where two problems. The first is that for each time airmon-ng is run on wifi0 the interface number on ath increases: the first time is ath1, the second ath2, the third ath3, and and so on. And this continues so in a short period of time it is up to ath56 and continuing to climb. Unloading the madwifi-ng driver, or rebooting the system has no effect, and the number of the interface created by airmon-ng continues to increase. The second problem is that if you run airmon-ng on wifi0 the athXX created does not show as being shown as in Monitor mode, even though it is. This can be confirmed via iwconfig. All these problem related to how udev assigns interface names. The answer is in this ticket: 12 Thanks to lucida. The source of the problem comes from the udev persistent net rules generator. Each distro is different... So here is a solution specifically for Gentoo. You should be able to adapt this solution to your particular distribution. Gentoo 2.6.20-r4 Udev 104-r12 Madwifi 0.9.3-r2 Aircrack-ng 0.7-r2 Solution: Change the file /etc/udev/rules.d/75-persistent-net-generator.rules From: KERNEL=="eth*|ath*|wlan*|ra*|sta*…….. To: KERNEL=="eth*|Ath*|wlan*|ra*|sta*……. In other words, you just capitalize the a. ath* becomes Ath*. Save the file. Now delete the file /etc/udev/rules.d/70-persistent-net.rules. Remove the driver and insert back. Removing ath also works: KERNEL=="eth*|wlan*|ra*|sta*…. This is also on Gentoo, both 2.6.19-gentoo-r5 and 2.6.20-gentoo-r6 For Ubuntu, see this Forum posting. The modified version of /etc/udev/rules.d/75-persistent-net-generator.rules is: # these rules generate rules for persistent network device naming ACTION=="add", SUBSYSTEM=="net", KERNEL=="eth*|Ath*|wlan*|ra*|sta*" \ NAME!="?*", DRIVERS=="?*", GOTO="persistent_net_generator_do" GOTO="persistent_net_generator_end" LABEL="persistent_net_generator_do" # build device description string to add a comment the generated rule SUBSYSTEMS=="pci", ENV{COMMENT}="PCI device attr{vendor}:$attr{device}($attr{driver})" SUBSYSTEMS=="usb", ENV{COMMENT}="USB device 0x$attr{idVendor}:0x$attr{idProduct}($attr{driver})" SUBSYSTEMS=="ieee1394", ENV{COMMENT}="Firewire device $attr{host_id})" SUBSYSTEMS=="xen", ENV{COMMENT}="Xen virtual device" ENV{COMMENT}=="", ENV{COMMENT}="$env{SUBSYSTEM} device ($attr{driver})" IMPORT{program}="write_net_rules $attr{address}" ENV{INTERFACE_NEW}=="?*", NAME="$env{INTERFACE_NEW}" LABEL="persistent_net_generator_end" This troubleshooting tip applies to madwifi-ng drivers. First try stopping each VAP interface that is running ("airmon-ng stop IFACE" where IFACE is the VAP name). You can obtain the list from iwconfig. Then do "airmon-ng start wifi0". If this does not resolve the problem then follow the advice in this thread. If you get error messages similar to: Error message: "SIOCSIFFLAGS : No such file or directory" Error message: "ioctl(SIOCGIFINDEX) failed: No such device" Then See this FAQ entry. If you receive "wlanconfig: command not found" or similar then the wlanconfig command is missing from your system or is not in the path. Use locate or find to determine if it is on your system and which directory it is in. If it is missing from your system then make sure you have done a "make install" after compiling the madwifi-ng drivers. On Ubuntu, do "apt-get install madwifi-tools". If it is not in a directory in your path then move it there or add the directory to your path. See this entry under installing the RT73 driver. You receive an error similar to: Interface Chipset Driver wlan0 iwl4965 - [phy0]/usr/sbin/airmon-ng: line 338: /sys/class/ieee80211/phy0/add_iface: Permission denied mon0: unknown interface: No matching device found (monitor mode enabled on mon0) or similar to this: wlan0 iwlagn - [phy0]/usr/local/sbin/airmon-ng: 856: cannot create /sys/class/ieee80211/phy0/add_iface: Directory nonexistent Error for wireless request "Set Mode" (8B06) : SET failed on device mon0 ; No such device. mon0: ERROR while getting interface flags: No such device This means you have an old version of airmon-ng installed. Upgrade to at least v1.0-rc1. Preferably you should upgrade to the current version. See the installation page for more details. Also, don't forget you need to be root to use airmon-ng (or use sudo). Distros from now on are going to adopt 'upstart' which is going to replace the /sbin/init daemon which manages services and tasks during boot. Basically do: service network-manager stop service avahi-daemon stop service upstart-udev-bridge stop and then proceed with greping and killing the pids of dhclient and wpa_supplicant. This is the only way to kill ALL of the potentially problematic pids for aireplay-ng permanently. The trick is the kill the daemons first and then terminate the 'tasks'. Source thread: and If you have an output similar to: # airmon-ng start wlan0 Interface Chipset Driver wlan0 Broadcom b43 - [phy0]SIOCSIFFLAGS: Unknown error 132 (monitor mode enabled on mon0) It indicates that RF are blocked. It needs to be enabled by using the switch on your laptop and/or using the following command: rfkill unblock all See also It is known to happen on the Raspberry Pi, when using airmon-ng. When that happens, the following can be seen in dmesg: brcmfmac: brcmf_vif_add_validate: Attempt to add a MONITOR interface... brcmfmac: brcmf_vif_add_validate: ... there is already a monitor interface, returning EOPNOTSUPP brcmfmac: brcmf_cfg80211_add_iface: iface validation failed: err=-95 There may be instances of the following in dmesg as well prior to the above output: brcmfmac: brcmf_vif_add_validate: Attempt to add a MONITOR interface... brcmfmac: brcmf_mon_add_vif: brcmf_mon_add_vif called brcmfmac: brcmf_mon_add_vif: Adding vif "wlan0mon" brcmfmac: brcmf_cfg80211_get_channel: chanspec failed (-52) Even though dmesg says the interface is already in monitor mode and "iw dev wlan0 info" confirms it is, airodump-ng will fail and report the interface data linktype is Ethernet. This is a bug in the driver and/or firmware, and the workaround is to reboot the system or to reload the driver: rmmod brcmfmac modprobe brcmfmac In this article, we are going to use Airodump-ng and Aireplay-ng to get the Handshake address passed between the router and the client. Before starting with the actual process, it is important to first understand how a connection initialization in a WiFi router works and how clients are authenticated to get connected to the router. Working on WiFi Authentication and ConnectionTypically, connection between a wireless router and client device works as follows: The client device searches for all the available networks nearby and displays their Service Set Identifier (SSID) which can be possibly connected by the device.Then the user chooses a wireless network to connect to.The client device requests the router for connection which further follows a 4-way handshake.After the router acknowledges the request, a connection gets established between the client and the wireless network.Now before moving forward, it is essential to understand all the terminologies: Service Set Identifier (SSID)A Service Set Identifier (SSID) is a distinct label assigned to a wireless router, serving to distinguish and identify it amidst the presence of multiple nearby Wi-Fi networks. NonceNonce is a pseudo-random number generated by devices during the authentication process for smooth communication and verification, which can only be used once so even if this nonce is captured in the middle by conventional intercepting methods, it cannot be reused to establish the same connection again. Group Temporal Key (GTK)Group Temporal Key (GTK) is used to encrypt all the traffic to and fro between the wireless router and the client devices connected to it. All the client devices connected has this GTK and this is unique so all the access point would have a different GTK. Pairwise Transient Key (PTK)Pairwise Transient Key (PTK) is a unique key generated by combining nonces during the authentication process between the two devices used for the encryption of all the data between the router and the client device. 4-Way HandshakeModern days wireless networks and providers follow a 4-way handshake protocol which includes the following steps: The client device sends a request to the router to allow the connection along with some information about the device and a nonce generated by the device.Once the router receives the initial request from the client, it also generates a new nonce and sends it along with some WiFi network information and GTK, which is encrypted with the password of the Wi-Fi.As the client device receives the response from the router, it then decrypts the data packets sent by the router using the password for connecting to Wifi. After decryption, the client device combines the nonce that it generated and the one it got in response from the router hence creating a new PTK. It then encrypts this information using the Wi-Fi password and sends it to the router.Upon response from the client device, the router decrypts the information using Wifi password and thereby matching the PTK it got from the client and the one which it evaluated. If the PTK matches, then it authenticates the connection request and allows the device to connect to the wireless network.Now, it is important to understand more about Airodump-ng and Aireplay-ng packages. Airodump-ngAirodump-ng is a command line tool which is under the suite of Aircrack-ng which is used to assess Wifi network security. This tool is specifically developed and designed to monitoring and intercept the wireless network traffic, including but not limited to Wifi Access points. A few key distuinguishing features of Airodump-ng are: Network ScanningIt provides functionality for scanning WiFi networks available nearby and displays information about networks like its MAC Address, ESSID, the channel on which Access Point is operating and the encryption type of the network. Detecting Connected ClientIt can be used to detect the number of devices connected to the wireless router network along with their MAC Adresses for further analysis. Intercepting the Data Packets This also provides functionality to intercept the transmission of Data packets between the client device and the wireless router. Aireplay-ngSimilar to Airodump-ng, Aireply-ng is a command line tool available under Aircrack-ng suite for various purposes like packet injecting network packets, deauthentication attacks and testing vulnerabilities on the network. The tool is valuable for evaluating the security of wireless networks and enhancing penetration testing capabilities. A few characteristics of Aireplay-ng are: Packet InjectionIt is designed to send or inject specially designed custom crafted packets which may include but not limited to deauthentication and disassociation, making client devices to disconnect and hence revealing potential vulnerabilities. Deauthentication AttacksThe most popular use of Aireplay-ng is to perform Deauthentication Attacks which includes sending deauth packets to the router to disconnect a client with specific MAC Address or every client connected to the network ApproachCapturing Handshake Address includes various steps to be followed. First, we will setup our network adapter to monitor mode to analyse different networks around, monitor and fetch information related to the routers, then we will deauthenticate all/specific clients from the WiFi Network and setup Airodump-ng to intercept any Handshake Addresses transmitted between the client device and the router. Now, after client device is deauthenticated from the WiFi network, the device will automatically try to reconnect to the same network, by attempting to initiate the 4-way Handshake process, during which Airodump-ng tool will intercept the Handshake Address transmitted between and save it in a file. InstallationThough, aircrack-ng and all its command line tools come by default, it can be downloaded using the following commands: Updating package repository: sudo apt updateInstalling aircrack-ng: sudo apt install aircrack-ngCapturing Handshake AddressSetting up Network Adapter in Monitor ModeFirst, using the following command check for all available interfaces and name of our Network Adapter iwconfigExplanation: 'iwconfig' command lists all the available network interfaces in the operating system with some of their basic information As we can see the name of our network adapter is wlan0, and the adapter is in Managed Mode, to monitor and inject deauth packets we have to put the adapter in monitor mode using the following command: sudo airmon-ng start wlan0Explanation: 'sudo' : gives higher level priviledges to perform some actions related to configuration of system settings 'airmon-ng' : it is the script that is used to enable/disable monitor mode for network adapters 'start' : the argument instructs airmon-ng to start monitor mode 'wlan0' : it is the default name of the network adapter to be used for the attack Now, we have to start monitoring all nearby WiFi networks for information about them. We can monitor all available networks using: sudo airodump-ng wlan0Explanation: 'sudo' : gives higher level priviledges to perform some actions related to configuration of system settings 'airodump-ng' : it is the tool to monitor wifi networks 'wlan0' : the default name of the network adapter to be used for the attack As we can see all nearby networks are listed with their BSSID (MAC), Channel they are operating on (CH), Encryption Type (ENC) etc. Here, our target is the third network in the list HARSH JIO 4G . So after setting the target and copying the BSSID we will specifically only monitor that network while also constantly trying to intercept Handshake Address if any using the following command: sudo airodump-ng --bssid -c -w psk wlan0Explanation: Here, we are giving '--bssid' argument to pass the BSSID of our target, '-c' argument to specifying the channel our target is operating on '-w' to specify the prefix of the output file, which means any output files generated having network information or handshakes will be saved in a file with the specified prefix The monitoring and checking for handshakes has been started on the target and now we will deauthenticate all the client devices from the router, so they automatically try to reconnect, and airodump-ng captures the handshake address. To deauthenticate using aireplay-ng, we can use following command: sudo aireplay-ng -0 -a wlan0Explanation: '-0 argument' : specifies the number of deauthentication packets to send to the router and passing 0 means to send the packets continuously until interrupted. '-a argument' : sets the BSSID/MAC of the target to deauthenticate clients from '-c (optional) argument' : is used to deauthenticate a specific client from the network by their MAC address and if not specified it disassociates all clients from the network As the attack starts, it disassociates all the clients connected to the network and as a result the devices will automatically try to reconnect to the network and this attempt to reconnect with the handshake address will be captured by the 'airodump-ng' script running. As it can be seen at the top right of the screen, airodump-ng fetched the WPA Handshake for the Access Point with that specific BSSID.We can see all the captured Handshake files in the current directory using: ls Now, these are all the handshake files captured during the transmission of data packets between router and the client device. We can look through the handshake file for information using: aircrack-ng .cap Q1. Are there alternatives to Airodump-ng and Aireplay-ng for capturing handshake addresses? Answer: Yes there multiple alternatives to Airodump-ng and Aireplay-ng available in the market, two prominent among them are Wireshark and Airgeddon Q2. How to decrypt the password from the Handshake Address? Answer: This Handshake Address can be decrypted into usable password by employing Dictionary Attack where the address can be checked against all the possible passwords in a dictionary or Bruteforce attack where the address is compared against all possible permutations of characters to get password. Q3.Can it be used to capture Handshake address for any WiFi Network? Answer: Yes it can be used to intercept Handshake address for any WiFi Network, but it is illegal to use them against any WiFi network without permissions. It is necessary to always have proper authorization before using these methods. Q4.How to secure Network? Answer: While it is not completely possible to avoid these types of attacks, but to make the network secure, it is advisable to use latest encryption type like WPA3 and using a Strong and lengthy password, so even if attacker gains Handshake Address, it is very difficult to crack the address to get the password. Q5.What to do if WiFi network security has been compromised? Answer: The First and Foremost thing to do is to change the password, check for any unauthorized change on the Admin page of Router and consult a Network Security professional. Conclusion:Capturing the Handshake Address is an essential step in assessing the security of a Wi-Fi network. By using already available tools like Airodump-ng and Aireplay-ng, one can monitor and intercept the Network Authentication process between a client device and a Wi-Fi router to gain information about the network. This information is very important for network administrators, as it can help them identify potential vulnerabilities and weaknesses in the network's security. But on the other hand, this information can also be used by attackers with malicious intent to break into someone's private network by exploiting these vulnerabilities. By understanding this techniques and tools involved in capturing handshake addresses and how an attacker can use it to evade someone's privacy, network administrators and users can take necessary steps to strengthen their network security to prevent any unauthorised person to access their private network. Airodump-ng is used for packet capture, capturing raw 802.11 frames. It is particularly suitable for collecting WEP IVs (Initialization Vector) or WPA handshakes for the intent of using them with aircrack-ng. If you have a GPS receiver connected to the computer, airodump-ng is capable of logging the coordinates of the found access points. Additionally, airodump-ng writes out several files containing the details of all access points and clients seen, which can be used for scripting, or creating custom tools Before running airodump-ng, you may start the airmon-ng script to list the detected wireless interfaces. It is possible, but not recommended, to run Kismet and airodump-ng at the same time. usage: airodump-ng [,,...] Options: --ivs : Save only captured IVs --gpsd : Use GPSd --write : Dump file prefix -w : same as --write --beacons : Record all beacons in dump file --update : Display update delay in seconds --showack : Prints ack/cts/rts statistics -h : Hides known stations for --showack -f : Time in ms between hopping channels --berlin : Time before removing the AP/client from the screen when no more packets are received (Default: 120 seconds) -r : Read packets from that file -T : While reading packets from a file, simulate the arrival rate of them as if they were "live". -x : Active Scanning Simulation --manufacturer : Display manufacturer from IEEE OUI list --uptime : Display AP Uptime from Beacon Timestamp --wps : Display WPS information (if any) --output-format : Output format. Possible values: pcap, ivs, csv, gps, kismet, netxml, logcsv --ignore-negative-one : Removes the message that says fixed channel : -1 --write-interval : Output file(s) write interval in seconds --background : Override background detection. -n : Minimum AP packets recv'd before for displaying it Filter options: --encrypt : Filter APs by cipher suite --netmask : Filter APs by mask --bssid : Filter APs by BSSID --essid : Filter APs by ESSID --essid-regex : Filter APs by ESSID using a regular expression -a : Filter unassociated clients By default, airodump-ng hop on 2.4GHz channels. You can make it capture on other/specific channel(s) by using: --ht20 : Set channel to HT20 (802.11n) --ht40- : Set channel to HT40- (802.11n) --ht40+ : Set channel to HT40+ (802.11n) --channel : Capture on specific channels --band : Band on which airodump-ng should hop -C : Uses these frequencies in MHz to hop --cswitch : Set channel switching method 0 : FIFO (default) 1 : Round Robin 2 : Hop on last -s : same as --cswitch --help : Displays this usage screen You can convert .cap / .dump file to .ivs format or merge them. airodump-ng will display a list of detected access points, and also a list of connected clients ("stations"). Here's an example screenshot: CH 9][Elapsed: 1 min][2007-04-26 17:41][WPA handshake: 00:14:6C:7E:40:80 BSSID PWR RXQ Beacons #Data, #/s CH MB ENC CIPHER AUTH ESSID 00:09:5B:1C:AA:1D 11 16 10 0 0 11 54. OPN NETGEAR 00:14:6C:7A:41:81 34 100 57 14 1 9 11e WEP WEP bigbear 00:14:6C:7E:40:80 32 100 752 73 2 9 54 WPA TKIP PSK teddy BSSID STATION PWR Rate Lost Packets Notes Probes 00:14:6C:7A:41:81 00:0F:B5:32:31:31 51 36-24 2 14 (not associated) 00:14:A4:3F:8D:13 19 0-0 0 4 mossy 00:14:6C:7A:41:81 00:0C:41:52:D1:D1 -1 36-36 0 5 00:14:6C:7E:40:80 00:0F:B5:FD:FB:C2 35 54-54 0 99 teddy The first line shows the current channel, elapsed running time, current date and optionally if a WPA/WPA2 handshake was detected. In the example above, "WPA handshake: 00:14:6C:7E:40:80" indicates that a WPA/WPA2 handshake was successfully captured for the BSSID. In the example above the client rate of "36-24" means: The first number is the last data rate from the AP (BSSID) to the Client (STATION). In this case 36 megabits per second. The second number is the last data rate from Client (STATION) to the AP (BSSID). In this case 24 megabits per second. These rates may potentially change on each packet transmission. It is simply the last speed seen. These rates are only displayed when locked to a single channel, the AP/client transmission speeds are displayed as part of the clients listed at the bottom. NOTE: APs need more then one packet to appear on the screen. APs with a single packet are not displayed. FieldDescription BSSIDMAC address of the access point. In the Client section, a BSSID of "(not associated)" means that the client is not associated with any AP. In this unassociated state, it is searching for an AP to connect with. PWRSignal level reported by the Wi-Fi adapter. Its signification depends on the driver, but as you get closer to the AP or the station, the signal gets higher. It usually is the RSSI. If the BSSID PWR is -1, then the driver doesn't support signal level reporting. If PWR is -1 for some access points, it means the access point is out of range, however airodump-ng got at least a frame sent to it. If the PWR is -1 for a limited number of stations then this is for a packet which came from the AP to the client but the client transmissions are out of range for your Wi-Fi adapter. Meaning you are hearing only 1/2 of the communication. If all clients have PWR as -1 then it is likely that the driver doesn't support signal level reporting. A strong signal is around -40. An average one is around -55, and a weak one starts around -70. Wi-Fi adapters lower limit (aka receive sensitivity) is often around -80/-90. RXQReceive Quality as measured by the percentage of packets (management and data frames) successfully received over the last 10 seconds. See note below for a more detailed explanation. BeaconsNumber of announcements packets sent by the AP. Each access point sends about ten beacons per second at the lowest rate (1M), so they can usually be picked up from very far. # DataNumber of captured data packets (if WEP, unique IV count), including data broadcast packets. #/sNumber of data packets per second measure over the last 10 seconds. CHChannel number (taken from beacon packets). Note: sometimes packets from other channels are captured even if airodump-ng is not hopping, because of radio interference or overlapping channels. MBMaximum speed supported by the AP. If MB = 11, it's 802.11b, if MB = 22 it's 802.11b+ and up to 54 are 802.11g. Anything higher is 802.11n or 802.11ac. The dot (after 54 above) indicates short preamble is supported. Displays "e" following the MB speed value if the network has QoS enabled. ENCEncryption algorithm in use. OPN = no encryption,"WEP?" = WEP or higher (not enough data to choose between WEP and WPA/WPA2), WEP (without the question mark) indicates static or dynamic WEP, and WPA, WPA2 or WPA3 if TKIP or CCMP is present (WPA3 with TKIP allows WPA or WPA2 association, pure WPA3 only allows CCMP). OWE is for Opportunistic Wireless Encryption, aka Enhanced Open. CIPHERThe cipher detected. One of CCMP, WRAP, TKIP, WEP, WEP40, or WEP104. Not mandatory, but TKIP is typically used with WPA and CCMP is typically used with WPA2. WEP40 is displayed when the key index is greater then 0. The standard states that the index can be 0-3 for 40bit and should be 0 for 104 bit. AUTHThe authentication protocol used. One of MGT (WPA/WPA2 using a separate authentication server), SKA (shared key for WEP), PSK (pre-shared key for WPA/WPA2), or OPN (open for WEP). ESSIDShows the wireless network name. The so-called "SSID", which can be empty if SSID hiding is activated. In this case, airodump-ng will try to recover the SSID from probe responses and association requests. See this section for more information concerning hidden ESSIDs. STATIONMAC address of each associated station or stations searching for an AP to connect with. Clients not currently associated with an AP have a BSSID of "(not associated)". Rate Station's receive rate, followed by transmit rate. Displays "e" following each rate if the network has QoS enabled. LostThe number of data packets lost over the last 10 seconds based on the sequence number. See note below for a more detailed explanation. PacketsThe number of data packets sent by the client. NotesAdditional information about the client, such as captured EAPOL or PMKID. ProbesThe ESSIDs probed by the client. These are the networks the client is trying to connect to if it is not currently connected. NOTES: RXQ expanded: Its measured over all management and data frames. The received frames contain a sequence number which is added by the sending access point. RXQ = 100 means that all packets were received from the access point in numerical sequence and none were missing. That's the clue, this allows you to read more things out of this value. Lets say you got 100 percent RXQ and all 10 (or whatever the rate) beacons per second coming in. Now all of a sudden the RXQ drops below 90, but you still capture all sent beacons. Thus you know that the AP is sending frames to a client but you can't hear the client nor the AP sending to the client (need to get closer). Another thing would be, that you got a 11MB card to monitor and capture frames (say a prism2.5) and you have a very good position to the AP. The AP is set to 54MBit and then again the RXQ drops, so you know that there is at least one 54MBit client connected to the AP. N.B.: RXQ column will only be shown if you are locked on a single channel, not channel hopping. Lost expanded: It means lost packets coming from the client. To determine the number of packets lost, there is a sequence field on every non-control frame, so you can subtract the second last sequence number from the last sequence number and you know how many packets you have lost. Possible reasons for lost packets: You cannot send (in case you are sending) and listen at the same time, so every time you send something you can't hear the packets being transmitted in that interval. You are maybe losing packets due too high transmit power (you may be too close to the AP). There is too much noise on the current channel (other APs, microwave oven, bluetooth…) To minimize the number of lost packets, vary your physical position, type of antenna used, channel, data rate and/or injection rate. To speed up the cracking process, run aircrack-ng while you are running airodump-ng. You can capture and crack at the same time. Aircrack-ng will periodically reread the captured data so it is always working with all the available IVs. To limit the data capture to a single AP you are interested in, include the "- -bssid" option and specify the AP MAC address. For example: "airodump-ng -c 8 - -bssid 00:14:6C:7A:41:20 -w capture ath0". To minimize disk space used by the capture, include the "- -ivs" option. For example: "airodump-ng -c 8 - -bssid 00:14:6C:7A:41:20 -w capture - -ivs ath0". This only stores the initialization vectors and not the full packet. This cannot be used if you are trying to capture the WPA/WPA2 handshake or if you want to use PTW attack on WEP. Lets say, for example, you wish to capture packets for all Cisco-Linksys APs where the BSSID starts with "00:1C:10". You specify that starting bytes you wish to match with the "-d" / "–bssid" option and pad with zeroes to a full MAC. Then use "-m" / "–netmask" option to specify which part of the BSSID you wish to match via "F"s and pad with zeroes to a full MAC. So since you want to match "00:1C:10", you use "FF:FF:FF". airodump-ng -d 00:1C:10:00:00:00 -m FF:FF:FF:00:00:00 wlan0 The "–channel" (-c) option allows a single or specific channels to be selected. Example of a single channel: airodump-ng -c 11 wlan0 For cards which needs to be reset when on a single channel: airodump-ng -c 11,11 wlan0 Example of selected channels: airodump-ng -c 1,6,11 wlan0 Each time airodump-ng is run with the option to write IVs or full packets, a few text files are also generated and written to disk. They have the same name and a suffix of ".csv" (CSV file), ".kismet.csv" (Kismet CSV file) and ".kismet.netxml" (Kismet newcore netxml file). The CSV file contains the details of all access points and clients seen. See kismet documentation for more details about the kismet CSV and netxml. Here is an example: BSSID, First time seen, Last time seen, channel, Speed, Privacy, Cipher, Authentication, Power, # beacons, # IV , LAN IP, ID-length, ESSID, Key 00:1C:10:26:22:41, 2007-10-07 12:48:58, 2007-10-07 12:49:44, 6, 48, WEP , WEP, , 171, 301, 0, 0. 0. 0. 0, 5, zwang, 00:1A:70:51:B5:71, 2007-10-07 12:48:58, 2007-10-07 12:49:44, 6, 48, WEP , WEP, , 175, 257, 1, 0. 0. 0. 0, 9, brucey123, 00:09:5B:7C:AA:CA, 2007-10-07 12:48:58, 2007-10-07 12:49:44, 11, 54, OPN , , , 189, 212, 0, 0. 0. 0. 0, 7, NETGEAR, Station MAC, First time seen, Last time seen, Power, # packets, BSSID, Probed ESSIDs 00:1B:77:7F:67:94, 2007-10-07 12:49:43, 2007-10-07 12:49:43, 178, 3, (not associated) , If you have a laptop with a builtin wireless card, ensure it is "turned on / enabled" in the bios Does your card works in managed mode? If not, the problem is not with airodump-ng. You need to get this working first. See if this madwifi-ng web page has information that may be helpful. Although it is not very "scientific", sometimes simply unloading then reloading the driver will get it working. This is done with the rmmod and modprobe commands. Also see the next troubleshooting tip. Make sure you used the "-c" or "- -channel" option to specify a single channel. Otherwise, by default, airodump-ng will hop between channels. You might need to be physically closer to the AP to get a quality signal. Make sure you have started your card in monitor mode with airmon-ng (Linux only). Make sure there are no other VAPs running. There can be issues when creating a new VAP in monitor mode and there was an existing VAP in managed mode. You should first stop ath0 then start wifi0: airmon-ng stop ath0 airmon-ng start wifi0 or wlanconfig ath0 destroy wlanconfig ath create wlandev wifi0 wlanmode monitor This is happening because your driver doesn't discard corrupted packets (that have an invalid CRC). If it's a ipw2100 (Centrino b), it just can't be helped; go buy a better card. If it's a Prism2, try upgrading the firmware. The most common cause is that a connection manager is running on your system and takes the card out of monitor mode. Be sure to stop all connection managers prior to using the aircrack-ng suite. In general, disabling "Wireless" in your network manager should be enough but sometimes you have to stop them completely. It can be done with airmon-ng: airmon-ng check kill Recent linux distributions use upstart; it automatically restarts the network manager. In order to stop it, see the following entry. As well, make sure that wpa_supplicant is not running. Another potential cause is the PC going to sleep due to power saving options. Check your power saving options. The madwifi-ng driver for the atheros chipset contains a bug in releases up to r2830 which causes airodump-ng in channel hopping mode to stop capturing data after a few minutes. The fix is to use r2834 or above of the madwifi-ng drivers. See also this entry for recent You will sometimes see "" as the SSID on the airodump-ng display. This means the SSID is hidden. The "?" is normally the length of the SSID. For example, if the SSID was "test123" then it would show up as "" where 7 is the number of characters. When the length is 0 or 1, it means the AP does not reveal the actual length and the real length could be any value. To obtain the hidden SSID there are a few options: Wait for a wireless client to associate with the AP. When this happens, airodump-ng will capture and display the SSID. Deauthenticate an existing wireless client to force it to associate again. The point above will apply. Use a tool like mdk3 to bruteforce the SSID. You can use Wireshark combined with one or more of these filters to review data capture files. The SSID is included within these packets for the AP. wlan.fc.type_subtype == 0 (association request) wlan.fc.type_subtype == 4 (probe request) wlan.fc.type_subtype == 5 (probe response) There are two workarounds: Change the rate before using airodump-ng If the top of your airodump screen looks something like: CH 6][Elapsed: 28 s][2008-09-21 10:39][fixed channel ath0: 1 Then this means you started started airodump-ng with a fixed channel parameter (-c / –channel) but some other process is changing the channel. "CH 6" on the left is the channel that was specified when airodump-ng was started. "fixed channel ath0: 1" on the right indicates that ath0 was used when airodump-ng was started but the interface is currently on channel 1 (instead of channel 6). You might also see this channel number changing indicating that channel scanning is taking place. It is critical that the root cause of the problem be eliminated and then airodump-ng restarted again. Here are some possible reasons and how to correct them: There is one or more interfaces in "managed mode" and these are are scanning for an AP to connect to. Do not use any command, process or program to connect to APs at the same time as you use the aircrack-ng suite. Other processes are changing the channel. A common problem are network managers. You can also use "airmon-ng check" on current versions of the aircrack-ng suite to identify problem processes. Then use "kill" or "killall" to destroy the problem processes. For example, use "killall NetworkManager && killall NetworkManagerDispatcher" to eliminate network managers. If you are using the madwifi-ng driver and have more then the ath0 interface created, the driver may be automatically scanning on the other interfaces. To resolve this, stop all interfaces except ath0. You have wpa_supplicant running at the same time. Stop wpa_supplicant. You run airmon-ng to set the channel while airodump-ng is running. Do not do this. You run another instance of airodump-ng in scanning mode or set to another channel. Stop airodump-ng and do not do this. It can also means that you cannot use this channel (and airodump-ng failed to set the channel). Eg: using channel 13 with a card that only supports channels from 1 to 11. Where did my output files go? You ran airodump-ng and now cannot find the output files. First, make sure you ran airodump-ng with the option to create output files. You must include -w or –write plus the file name prefix. If you fail to do this then no output files are created. By default, the output files are placed in the directory where you start airodump-ng. Before starting airodump-ng, use "pwd" to display the current directory. Make a note of this directory so your return to it a later time. To return to this directory, simply type "cd ". To output the files to a specific directly, add the full path to the file prefix name. For example, lets say you want to output all your files to "/aircrack-ng/captures". First, create /aircrack-ng/captures if it does not already exist. Then include "-w /aircrack-ng/captures/" on your airodump-ng command line. To access your files later when running aircrack-ng, either change to the directory where the files are located or prefix the file name with the full path. Windows specific The adapter is not detected Make sure the special driver is installed. Read Driver installing page for a guide on installing such driver. If the special driver is installed but it still isn't detected, try another version of the driver (older or newer). The application has failed to start because MSVCR70.dll was not found The application freezes under Microsoft Windows Ensure you are using the correct drivers for your particular wireless card. Plus the correct Wildpackets driver. Failure to do so may result in your PC freezing when running airodump-ng. The powersaver option on the card can also cause the application to freeze or crash. Try disabling this option via the "Properties" section of your card. Another kludge is to keep moving your mouse every few minutes to eliminate the powersaver option from kicking in. How to get airodump-ng to work under Windows Vista? The following fix has reportedly worked for some people: What you have to do is right click on airodump-ng.exe, select properties, compatibility, and check run in compatibility mode for Windows XP. Also, check the box at the bottom that says to run as administrator. peek.sys file is zero bytes! Peek.sys being zero bytes is normal. You can proceed to use airodump-ng. This file is created by airodump-ng to prevent the driver dialog box from being shown each time the program is run. error: "Failed to download Peek files" You may have a DNS problem or there is an Internet connectivity problem. Manually download the following files and place them in the same directory as the airodump-ng.exe file. Various errors referencing peek.dll If you receive one or more of these errors: Dialog Box Error: "The application or DLL C:\????\bin\Peek.dll is not a valid Windows image. Please check this against your installation diskette." GUI Screen Error: "LoadLibrary (Peek.dll) failed, make sure this file is present in the current directory. Press Ctrl-c to exit." This means the peek.dll and/or peek5.sys file are missing from the directory which contains the airodump-ng.exe file or are corrupted. See the previous troubleshooting entry for instructions on how to download the files. No data is captured under Windows Using the Windows network connections manager, ensure the wireless device is enabled. Ensure that your Windows wireless configuration manager is enabled and the configuration manager that comes with your card is disabled. Do not run any wireless configuration manager while trying to use the aircrack-ng suite. Do not run any wireless program such as monitor mode checkers while trying to use the aircracck-ng suite. Check the "Driver Provider" name for the driver being used for your wireless device via properties to ensure it says Wildpackets. Also confirm the driver version is what you expect. Using a command prompt, change to the directory where airodump-ng.exe is located. Confirm that peek.dll and peek.sys exist in this directory. Using the command prompt and while still in the directory containing airodump-ng, try starting airodump-ng. It should not ask you about downloading Wildpackets or peek files. If it does, you do not have everything installed correctly. Redo the installation instructions. Review all your steps If airodump-ng is not functioning, it cannot detect your card or you get the blue screen of death, review the instructions for installing the software and drivers. If you cannot identify the problem, redo everything from scratch. Also check the this tutorial for ideas. Airodump-ng Bluescreen Airodump-ng or any "user space" program cannot produce a bluescreen, it is the driver which is the root cause. In most cases, these bluescreen failures cannot be resolved since these drivers are closed source. Interaction Since revision r1648, airodump-ng can receive and interpret key strokes while running. The following list describes the currently assigned keys and supposed actions. [a]: Select active areas by cycling through these display options: AP+STA; AP+STA+ACK; AP only; STA only [d]: Reset sorting to defaults (Power) [i]: Invert sorting algorithm [m]: Mark the selected AP or cycle through different colors if the selected AP is already marked [r]: (De-)Activate realtime sorting - applies sorting algorithm everytime the display will be redrawn [s]: Change column to sort by, which currently includes: First seen; BSSID; PWR level; Beacons; Data packets; Packet rate; Channel; Max. data rate; Encryption; Strongest Ciphersuite; Strongest Authentication; ESSID [SPACE]: Pause display redrawing/ Resume redrawing [TAB]: Enable/Disable scrolling through AP list [UP]: Select the AP prior to the currently marked AP in the displayed list if available [DOWN]: Select the AP after the currently marked AP if available If an AP is selected or marked, all the connected stations will also be selected or marked with the same color as the corresponding Access Point. Back to Lab Listing Lab Objective: Learn how to discover nearby Wi-Fi networks with Airodump-ng. Lab Purpose: Airodump-ng is a part of the Aircrack-ng set of utilities for analysing Wi-Fi networks for weaknesses. It is mainly used for Wi-Fi discovery. Lab Tool: Kali Linux Lab Topology: You can use Kali Linux in a VM for this lab. Lab Walkthrough: Task 1: Airodump-ng comes pre-installed on Kali. You will need a wireless card which is capable of being put into "monitor mode" to complete this lab. In this lab, we will use an Alfa network card for this purpose. There are numerous Wi-Fi adapters on market which are supports Wi-Fi hacking. In this page, you can find some of them: We will begin this lab by first connecting our wireless network card to our Kali machine. Once the network card is connected, we can use airmon-ng to show us the available network cards which will work with the Aircrack-ng tools by using this command: sudo su – airmon-ng The next step is to then place this card into monitor mode using the following command: airmon-ng start wlan0 This will allow our network card to intercept nearby Wi-Fi packets. We can confirm that our network card is in monitor mode by executing the following command: ifconfig You will note that the interface now has a "mon" after its name; "wlan0mon" in this instance. Task 2: Once the above task is done, we can then start airodump-ng and discover nearby Wi-Fi networks using the following command: airodump-ng wlan0mon This will start the airodump-ng tool and it will begin searching for nearby Wi-Fi networks. 1) Shows us MAC address of detected Access Points. 2) Signal power level also tells target device distance from our Wi-Fi antenna. Higher numbers indicate better signal. 3) Channel number on which target APs are running. 4) The encryption methods that targets are using. 5) If the target AP advertises themselves with a name, we can see it in this section. 6) MAC address of connected clients to various AP stations. MAC address of various client devices that are connected to APs around. Task 3: Once your target network has been found, we can stop the search by hitting ctrl + c on the keyboard. We will now run airodump-ng again, but only for the target network. This way, we can capture more information about our target network and the clients connected to the network. To do this, we will use the following command: airodump-ng -c 9 –bssid 94:e4:ba:8b:81:ab -w /root/Desktop wlanmon When you execute this command, you will then see that airodump is now only capturing information about your target network. "-bssid" is used to specify the BSSID MAC address of a target machine. "-w" is used to specify the location where our files are going to be written. "-c" is used to specify the target channel number. We can see that, on the network above, there is one client connected to the network and the number of frames that client is communicating through the network. Keep this running state, as we will need it for the next lab where we capture the WPA handshake file. Stay Informed, Stay Inspired: Subscribe for Cutting-Edge IT-Certification Insights