SCREENSHOTS

MODULES USED IN BETTER CAP

[1] **any.proxy ->** A firewall redirection to any custom proxy.

- any.proxy on : Start the custom proxy redirection.
- any.proxy off : Stop the custom proxy redirection.

Parameters

- any.proxy.dst_address : Address where the proxy is listening. (default=<interface address>)
- any.proxy.dst_port : Port where the proxy is listening. (default=8080)
- any.proxy.iface : Interface to redirect packets from. (default=<interface name>)
- any.proxy.protocol : Proxy protocol. (default=TCP)
- any.proxy.src_address : Leave empty to intercept any source address. (default=)
- any.proxy.src_port : Remote port to redirect when the module is activated, also supported a comma separated list of ports and/or port-ranges. (default=80)

[2] api.rest -> Expose a RESTful API.

- api.rest on : Start REST API server.
- api.rest off : Stop REST API server.
- api.rest.record off : Stop recording the session.
- api.rest.record FILENAME : Start polling the rest API periodically recording each sample in a compressed file that can be later replayed.
- api.rest.replay off : Stop replaying the recorded session.
- api.rest.replay FILENAME : Start the rest API module in replay mode using FILENAME as the recorded session file, will revert to normal mode once the replay is over.

Parameters

- api.rest.address : Address to bind the API REST server to. (default=127.0.0.1)
- api.rest.alloworigin : Value of the Access-Control-Allow-Origin header of the API server. (default=*)
- api.rest.certificate : API TLS certificate. (default=)
- api.rest.certificate.bits : Number of bits of the RSA private key of the generated HTTPS certificate. (default=4096)
- api.rest.certificate.commonname : Common Name field of the generated HTTPS certificate. (default=bettercap)
- api.rest.certificate.country : Country field of the generated HTTPS certificate. (default=US)
- api.rest.certificate.locality : Locality field of the generated HTTPS certificate. (default=)
- api.rest.certificate.organization : Organization field of the generated HTTPS certificate. (default=bettercap devteam)
- api.rest.certificate.organizationalunit : Organizational Unit field of the generated HTTPS certificate. (default=https://bettercap.org/)
- api.rest.key : API TLS key (default=)
- api.rest.password : API authentication password. (default=)
- api.rest.port : Port to bind the API REST server to. (default=8081)
- api.rest.record.clock : Number of seconds to wait while recording with api.rest.record between one sample and the next one. (default=1)
- api.rest.username : API authentication username. (default=)
- api.rest.websocket : If true the /api/events route will be available as a websocket endpoint instead of HTTPS. (default=false)

[3] arp.spoof -> Keep spoofing selected hosts on the network.

- arp.spoof on : Start ARP spoofer.
- arp.ban on : Start ARP spoofer in ban mode, meaning the target(s) connectivity will not work.
- arp.spoof off : Stop ARP spoofer.
- arp.ban off : Stop ARP spoofer.

Parameters

- arp.spoof.fullduplex : If true, both the targets and the gateway will be attacked, otherwise only the target (if the router has ARP spoofing protections in place this will make the attack fail). (default=false)
- arp.spoof.internal : If true, local connections among computers of the network will be spoofed, otherwise only connections going to and coming from the external network. (default=false)
- arp.spoof.skip_restore : If set to true, targets arp cache won't be restored when spoofing is stopped. (default=false)
- arp.spoof.targets : Comma separated list of IP addresses, MAC addresses or aliases to spoof, also supports nmap style IP ranges. (default=<entire subnet>)
- arp.spoof.whitelist : Comma separated list of IP addresses, MAC addresses or aliases to skip while spoofing. (default=)

[4] ble.recon -> Bluetooth Low Energy devices discovery.

- ble.recon on : Start Bluetooth Low Energy devices discovery.
- ble.recon off : Stop Bluetooth Low Energy devices discovery.
- ble.clear : Clear all devices collected by the BLE discovery module.
- ble.show : Show discovered Bluetooth Low Energy devices.
- ble.enum MAC : Enumerate services and characteristics for the given BLE device.
- ble.write MAC UUID HEX_DATA : Write the HEX_DATA buffer to the BLE device with the specified MAC address, to the characteristics with the given UUID.

Parameters

- ble.device : Index of the HCI device to use, -1 to autodetect. (default=-1)
- ble.show.filter : Defines a regular expression filter for ble.show (default=)
- ble.show.limit : Defines limit for ble.show (default=0)
- ble.show.sort : Defines sorting field (rssi, mac, seen) and direction (asc or desc) for ble.show (default=rssi asc)
- ble.timeout : Connection timeout in seconds. (default=5)
- ble.ttl : Seconds of inactivity for a device to be pruned. (default=30)

[5] c2 -> A CnC module that connects to an IRC server for reporting and commands.

- c2 on : Start the C2 module.
- c2 off : Stop the C2 module.
- c2.channel.set EVENT_TYPE CHANNEL : Set a specific channel to report events of this type.
- c2.channel.clear EVENT_TYPE : Clear the channel to use for a specific event type.
- c2.template.set EVENT_TYPE TEMPLATE : Set the reporting template to use for a specific event type.
- c2.template.clear EVENT_TYPE : Clear the reporting template to use for a specific event type.

Parameters

- c2.channel.control : IRC channel to receive commands from. (default=#events)
- c2.channel.events : IRC channel to send events to. (default=#events)
- c2.channel.output : IRC channel to send commands output to. (default=#events)
- c2.nick : IRC nickname. (default=bettercap)
- c2.operator : IRC nickname of the user allowed to run commands. (default=admin)
- c2.password : IRC server password. (default=password)
- c2.sasl.password : IRC server SASL password. (default=)
- c2.sasl.username : IRC SASL username. (default=)
- c2.server : IRC server address and port. (default=localhost:6697)
- c2.server.tls : Enable TLS. (default=true)
- c2.server.tls.verify : Enable TLS certificate validation. (default=false)
- c2.username : IRC username. (default=bettercap)

[6] caplets -> A module to list and update caplets.

- caplets.show : Show a list of installed caplets.
- caplets.paths : Show a list caplet search paths.
- caplets.update : Install/updates the caplets.

[7] dhcp6.spoof -> Replies to DHCPv6 messages, providing victims with a link-local IPv6 address and setting the attackers host as default DNS server (https://github.com/fox-it/mitm6/).

- dhcp6.spoof on : Start the DHCPv6 spoofer in the background.
- dhcp6.spoof off : Stop the DHCPv6 spoofer in the background.

Parameters

- dhcp6.spoof.domains : Comma separated values of domain names to spoof. (default=microsoft.com, google.com, facebook.com, apple.com, twitter.com)

[8] dns.spoof -> Replies to DNS messages with spoofed responses.

- dns.spoof on : Start the DNS spoofer in the background.
- dns.spoof off : Stop the DNS spoofer in the background.

Parameters

- dns.spoof.address : IP address to map the domains to. (default=<interface address>)
- dns.spoof.all : If true the module will reply to every DNS request, otherwise it will only reply to the one targeting the local pc. (default=false)
- dns.spoof.domains : Comma separated values of domain names to spoof. (default=)
- dns.spoof.hosts : If not empty, this hosts file will be used to map domains to IP addresses. (default=)
- dns.spoof.ttl : TTL of spoofed DNS replies. (default=1024)

[9] events.stream -> Print events as a continuous stream.

- events.stream on : Start events stream.
- events.stream off : Stop events stream.
- events.show LIMIT? : Show events stream.
- events.on TAG COMMANDS : Run COMMANDS when an event with the specified TAG is triggered.
- events.triggers : Show the list of event triggers created by the events.on command.
- events.trigger.delete TRIGGER_ID : Remove an event trigger given its TRIGGER_ID (use events.triggers to see the list of triggers).
- events.triggers.clear : Remove all event triggers (use events.triggers to see the list of triggers).
- events.waitfor TAG TIMEOUT? : Wait for an event with the given tag either forever or for a timeout in seconds.
- events.ignore FILTER : Events with an identifier matching this filter will not be shown (use multiple times to add more filters).
- events.include FILTER : Used to remove filters passed with the events.ignore command.
- events.filters : Print the list of filters used to ignore events.
- events.filters.clear : Clear the list of filters passed with the events.ignore command.
- events.clear : Clear events stream.

Parameters

- events.stream.http.format.hex : If true dumped HTTP bodies will be in hexadecimal format. (default=true)
- events.stream.http.request.dump : If true all HTTP requests will be dumped. (default=false)
- events.stream.http.response.dump : If true all HTTP responses will be dumped. (default=false)
- events.stream.output : If not empty, events will be written to this file instead of the standard output. (default=)
- events.stream.output.rotate : If true will enable log rotation. (default=true)
- events.stream.output.rotate.compress : If true will enable log rotation compression. (default=true)
- events.stream.output.rotate.format : Datetime format to use for log rotation file names. (default=2006-01-02 15:04:05)
- events.stream.output.rotate.how : Rotate by 'size' or 'time'. (default=size)
- events.stream.output.rotate.when : File size (in MB) or time duration (in seconds) for log rotation. (default=10)
- events.stream.time.format : Date and time format to use for events reporting. (default=15:04:05)

[10] gps -> A module talking with GPS hardware on a serial interface or via GPSD.

- gps on : Start acquiring from the GPS hardware.
- gps off : Stop acquiring from the GPS hardware.
- gps.show : Show the last coordinates returned by the GPS hardware.

Parameters

- gps.baudrate : Baud rate of the GPS serial device. (default=4800)
- gps.device : Serial device of the GPS hardware or hostname:port for a GPSD instance. (default=/dev/ttyUSB0)

[11] hid -> A scanner and frames injection module for HID devices on the 2.4Ghz spectrum, using Nordic Semiconductor nRF24LU1+ based USB dongles and Bastille Research RFStorm firmware.

- hid.recon on : Start scanning for HID devices on the 2.4Ghz spectrum.
- hid.recon off : Stop scanning for HID devices on the 2.4Ghz spectrum.
- hid.clear : Clear all devices collected by the HID discovery module.
- hid.sniff ADDRESS : Start sniffing a specific ADDRESS in order to collect payloads, use 'clear' to stop collecting.
- hid.show : Show a list of detected HID devices on the 2.4Ghz spectrum.
- hid.inject ADDRESS LAYOUT FILENAME : Parse the duckyscript FILENAME and inject it as HID frames spoofing the device ADDRESS, using the LAYOUT keyboard mapping.

Parameters

- hid.force.type : If the device is not visible or its type has not being detected, force the device type to this value. Accepted values: logitech, amazon, microsoft (default=logitech)
- hid.hop.period : Time in milliseconds to stay on each channel before hopping to the next one. (default=100)
- hid.lna : If true, enable the LNA power amplifier for CrazyRadio devices. (default=true)
- hid.ping.period : Time in milliseconds to attempt to ping a device on a given channel while in sniffer mode. (default=100)
- hid.show.filter : Defines a regular expression filter for hid.show (default=)
- hid.show.limit : Defines limit for hid.show (default=0)
- hid.show.sort : Defines sorting field (mac, seen) and direction (asc or desc) for hid.show (default=mac desc)
- hid.sniff.period : Time in milliseconds to automatically sniff payloads from a device, once it's detected, in order to determine its type. (default=500)
- hid.ttl : Seconds of inactivity to consider a device as not in range. (default=1200)

[12] http.proxy -> A full featured HTTP proxy that can be used to inject malicious contents into webpages, all HTTP traffic will be redirected to it.

- http.proxy on : Start HTTP proxy.
- http.proxy off : Stop HTTP proxy.

Parameters

- http.port : HTTP port to redirect when the proxy is activated. (default=80)
- http.proxy.address : Address to bind the HTTP proxy to. (default=<interface address>)
- http.proxy.blacklist : Comma separated list of hostnames to skip while proxying (wildcard expressions can be used). (default=)
- http.proxy.injectjs : URL, path or javascript code to inject into every HTML page. (default=)
- http.proxy.port : Port to bind the HTTP proxy to. (default=8080)
- http.proxy.redirect : Enable or disable port redirection with iptables. (default=true)
- http.proxy.script : Path of a proxy JS script. (default=)
- http.proxy.sslstrip : Enable or disable SSL stripping. (default=false)
- http.proxy.whitelist : Comma separated list of hostnames to proxy if the blacklist is used (wildcard expressions can be used). (default=)

[13] http.server -> A simple HTTP server, to be used to serve files and scripts across the network.

- http.server on : Start httpd server.
- http.server off : Stop httpd server.

Parameters

- http.server.address : Address to bind the http server to. (default=<interface address>)
- http.server.path : Server folder. (default=.)
- http.server.port : Port to bind the http server to. (default=80)

[14] https.proxy -> A full featured HTTPS proxy that can be used to inject malicious contents into webpages, all HTTPS traffic will be redirected to it.

- https.proxy on : Start HTTPS proxy.
- https.proxy off : Stop HTTPS proxy.

Parameters

- https.port : HTTPS port to redirect when the proxy is activated. (default=443)
- https.proxy.address : Address to bind the HTTPS proxy to. (default=<interface address>)
- https.proxy.blacklist : Comma separated list of hostnames to skip while proxying (wildcard expressions can be used). (default=)
- https.proxy.certificate : HTTPS proxy certification authority TLS certificate file. (default=~/.bettercap-ca.cert.pem)
- https.proxy.certificate.bits : Number of bits of the RSA private key of the generated HTTPS certificate. (default=4096)
- https.proxy.certificate.commonname : Common Name field of the generated HTTPS certificate. (default=Go Daddy Secure Certificate Authority - G2)
- https.proxy.certificate.country : Country field of the generated HTTPS certificate. (default=US)
- https.proxy.certificate.locality : Locality field of the generated HTTPS certificate. (default=Scottsdale)
- https.proxy.certificate.organization : Organization field of the generated HTTPS certificate. (default=GoDaddy.com, Inc.)
- https.proxy.certificate.organizationalunit : Organizational Unit field of the generated HTTPS certificate. (default=https://certs.godaddy.com/repository/)
- https.proxy.injectjs : URL, path or javascript code to inject into every HTML page. (default=)
- https.proxy.key : HTTPS proxy certification authority TLS key file. (default=~/.bettercap-ca.key.pem)
- https.proxy.port : Port to bind the HTTPS proxy to. (default=8083)
- https.proxy.redirect : Enable or disable port redirection with iptables. (default=true)
- https.proxy.script : Path of a proxy JS script. (default=)
- https.proxy.sslstrip : Enable or disable SSL stripping. (default=false)
- https.proxy.whitelist : Comma separated list of hostnames to proxy if the blacklist is used (wildcard expressions can be used). (default=)

[15] https.server -> A simple HTTPS server, to be used to serve files and scripts across the network.

- https.server on : Start https server.
- https.server off : Stop https server.

Parameters

- https.server.address : Address to bind the http server to. (default=<interface address>)
- https.server.certificate : TLS certificate file (will be auto generated if filled but not existing). (default=~/.bettercap-httpd.cert.pem)
- https.server.certificate.bits : Number of bits of the RSA private key of the generated HTTPS certificate. (default=4096)
- https.server.certificate.commonname : Common Name field of the generated HTTPS certificate. (default=bettercap)
- https.server.certificate.country : Country field of the generated HTTPS certificate. (default=US)
- https.server.certificate.locality : Locality field of the generated HTTPS certificate. (default=)
- https.server.certificate.organization : Organization field of the generated HTTPS certificate. (default=bettercap devteam)
- https.server.certificate.organizationalunit : Organizational Unit field of the generated HTTPS certificate. (default=https://bettercap.org/)
- https.server.key : TLS key file (will be auto generated if filled but not existing). (default=~/.bettercap-httpd.key.pem)
- https.server.path : Server folder. (default=.)
- https.server.port : Port to bind the http server to. (default=443)

[16] mac.changer -> Change active interface mac address.

- mac.changer on : Start mac changer module.
- mac.changer off : Stop mac changer module and restore original mac address.

Parameters

- mac.changer.address : Hardware address to apply to the interface. (default=<random mac>)
- mac.changer.iface : Name of the interface to use. (default=<interface name>)

[17] mdns.server -> A mDNS server module to create multicast services or spoof existing ones.

- mdns.server on : Start mDNS server.
- mdns.server off : Stop mDNS server.

Parameters

- mdns.server.address : IPv4 address of the mDNS service. (default=<interface address>)
- mdns.server.address6 : IPv6 address of the mDNS service. (default=<interface address6>)
- mdns.server.domain : mDNS domain. (default=local.)
- mdns.server.host : mDNS hostname to advertise on the network. (default=kali.)
- mdns.server.info : Comma separated list of informative TXT records for the mDNS server. (default=rpBA=DE:AD:BE:EF:CA:FE, rpAD=abf99d4ff73f, rpHI=ec5fb3caf528, rpHN=20f8fb46e2eb, rpVr=164.16, rpHA=7406bd0eff69)
- mdns.server.port : Port of the mDNS service. (default=52377)
- mdns.server.service : mDNS service name to advertise on the network. (default=_companion-link._tcp.)

[18] mysql.server -> A simple Rogue MySQL server, to be used to exploit LOCAL INFILE and read arbitrary files from the client.

- mysql.server on : Start mysql server.
- mysql.server off : Stop mysql server.

Parameters

- mysql.server.address : Address to bind the mysql server to. (default=<interface address>)
- mysql.server.infile : File you want to read. UNC paths are also supported. (default=/etc/passwd)
- mysql.server.outfile : If filled, the INFILE buffer will be saved to this path instead of being logged. (default=)
- mysql.server.port : Port to bind the mysql server to. (default=3306)

[19] ndp.spoof -> Keep spoofing selected hosts on the network by sending spoofed NDP router advertisements.

- ndp.spoof on : Start NDP spoofer.
- ndp.spoof off : Stop NDP spoofer.

Parameters

- ndp.spoof.neighbour : Neighbour IPv6 address to spoof, clear to disable NA. (default=fe80::1)
- ndp.spoof.prefix : IPv6 prefix for router advertisements spoofing, clear to disable RA. (default=d00d::)
- ndp.spoof.prefix.length : IPv6 prefix length for router advertisements. (default=64)
- ndp.spoof.targets : Comma separated list of IPv6 victim addresses. (default=)

[20] net.probe -> Keep probing for new hosts on the network by sending dummy UDP packets to every possible IP on the subnet.

- net.probe on : Start network hosts probing in background.
- net.probe off : Stop network hosts probing in background.

Parameters

- net.probe.mdns : Enable mDNS discovery probes. (default=true)
- net.probe.nbns : Enable NetBIOS name service discovery probes. (default=true)
- net.probe.throttle : If greater than 0, probe packets will be throttled by this value in milliseconds. (default=10)
- net.probe.upnp : Enable UPNP discovery probes. (default=true)
- net.probe.wsd : Enable WSD discovery probes. (default=true)

[21] net.recon -> Read periodically the ARP cache in order to monitor for new hosts on the network.

- net.recon on : Start network hosts discovery.
- net.recon off : Stop network hosts discovery.
- net.clear : Clear all endpoints collected by the hosts discovery module.
- net.show : Show cache hosts list (default sorting by ip).
- net.show ADDRESS1, ADDRESS2 : Show information about a specific comma separated list of addresses (by IP or MAC).
- net.show.meta ADDRESS1, ADDRESS2 : Show meta information about a specific comma separated list of addresses (by IP or MAC).

Parameters

- net.show.filter : Defines a regular expression filter for net.show (default=)
- net.show.limit : Defines limit for net.show (default=0)
- net.show.meta : If true, the net.show command will show all metadata collected about each endpoint. (default=false)
- net.show.sort : Defines sorting field (ip, mac, seen, sent, rcvd) and direction (asc or desc) for net.show (default=ip asc)

[22] net.sniff -> Sniff packets from the network.

- net.sniff stats : Print sniffer session configuration and statistics.
- net.sniff on : Start network sniffer in background.
- net.sniff off : Stop network sniffer in background.
- net.fuzz on : Enable fuzzing for every sniffed packet containing the specified layers.
- net.fuzz off : Disable fuzzing

Parameters

- net.fuzz.layers : Types of layer to fuzz. (default=Payload)
- net.fuzz.rate : Rate in the [0.0,1.0] interval of packets to fuzz. (default=1.0)
- net.fuzz.ratio : Rate in the [0.0,1.0] interval of bytes to fuzz for each packet. (default=0.4)
- net.fuzz.silent : If true it will not report fuzzed packets. (default=false)
- net.sniff.filter : BPF filter for the sniffer. (default=not arp)
- net.sniff.local : If true it will consider packets from/to this computer, otherwise it will skip them. (default=false)
- net.sniff.output : If set, the sniffer will write captured packets to this file. (default=)
- net.sniff.regexp : If set, only packets matching this regular expression will be considered. (default=)
- net.sniff.source : If set, the sniffer will read from this pcap file instead of the current interface. (default=)
- net.sniff.verbose : If true, every captured and parsed packet will be sent to the events.stream for displaying, otherwise only the ones parsed at the application layer (sni, http, etc). (default=false)

[23] packet.proxy -> A Linux only module that relies on NFQUEUEs in order to filter packets.

- packet.proxy on : Start the NFQUEUE based packet proxy.
- packet.proxy off : Stop the NFQUEUE based packet proxy.

Parameters

- packet.proxy.chain : Chain name of the iptables rule. (default=OUTPUT)
- packet.proxy.plugin : Go plugin file to load and call for every packet. (default=)
- packet.proxy.queue.num : NFQUEUE number to bind to. (default=0)
- packet.proxy.rule : Any additional iptables rule to make the queue more selective (ex. --destination 8.8.8.8). (default=)

[24] syn.scan -> A module to perform SYN port scanning.

- syn.scan stop : Stop the current syn scanning session.
- syn.scan IP-RANGE START-PORT END-PORT : Perform a syn port scanning against an IP address within the provided ports range.
- syn.scan.progress : Print progress of the current syn scanning session.

Parameters

- syn.scan.show-progress-every : Period in seconds for the scanning progress reporting. (default=1)

[25] tcp.proxy -> A full featured TCP proxy and tunnel, all TCP traffic to a given remote address and port will be redirected to it.

- tcp.proxy on : Start TCP proxy.
- tcp.proxy off : Stop TCP proxy.

Parameters

- tcp.address : Remote address of the TCP proxy. (default=)
- tcp.port : Remote port to redirect when the TCP proxy is activated. (default=443)
- tcp.proxy.address : Address to bind the TCP proxy to. (default=<interface address>)
- tcp.proxy.port : Port to bind the TCP proxy to. (default=8443)
- tcp.proxy.script : Path of a TCP proxy JS script. (default=)
- tcp.tunnel.address : Address to redirect the TCP tunnel to (optional). (default=)
- tcp.tunnel.port : Port to redirect the TCP tunnel to (optional). (default=0)

[26] ticker -> A module to execute one or more commands every given amount of seconds.

- ticker on : Start the ticker.
- ticker off : Stop the ticker.

Parameters

- ticker.commands : List of commands separated by a ; (default=clear; net.show; events.show 20)
- ticker.period : Ticker period in seconds (default=1)

[27] ui -> A module to manage bettercap's UI updates and installed version.

- ui.version : Print the currently installed UI version.
- ui.update : Download the latest available version of the UI and install it.

Parameters

- ui.basepath : UI base installation path. (default=/usr/local/share/bettercap/)
- ui.tmpfile : Temporary file to use while downloading UI updates. (default=/tmp/ui.zip)

[28] update -> A module to check for bettercap's updates.

- update.check on : Check latest available stable version and compare it with the one being used.

[29] wifi -> A module to monitor and perform wireless attacks on 802.11.

- wifi.recon on : Start 802.11 wireless base stations discovery and channel hopping.
- wifi.recon off : Stop 802.11 wireless base stations discovery and channel hopping.
- wifi.clear : Clear all access points collected by the WiFi discovery module.
- wifi.recon MAC : Set 802.11 base station address to filter for.
- wifi.recon clear : Remove the 802.11 base station filter.
- wifi.client.probe.sta.filter FILTER : Use this regular expression on the station address to filter client probes, 'clear' to reset the filter.
- wifi.client.probe.ap.filter FILTER : Use this regular expression on the access point name to filter client probes, 'clear' to reset the filter.
- wifi.deauth BSSID : Start a 802.11 deauth attack, if an access point BSSID is provided, every client will be deauthenticated, otherwise only the selected client. Use 'all', '*' or a broadcast BSSID (ff:ff:ff:ff:ff:ff) to iterate every access point with at least one client and start a deauth attack for each one.
- wifi.probe BSSID ESSID : Sends a fake client probe with the given station BSSID, searching for ESSID.
- wifi.channel_switch_announce bssid channel : Start a 802.11 channel hop attack, all client will be force to change the channel lead to connection down.
- wifi.fake_auth bssid client : send an fake authentication with client mac to ap lead to client disconnect
- wifi.assoc BSSID : Send an association request to the selected BSSID in order to receive a RSN PMKID key. Use 'all', '*' or a broadcast BSSID (ff:ff:ff:ff:ff:ff) to iterate for every access point.
- wifi.ap : Inject fake management beacons in order to create a rogue access point.
- wifi.show.wps BSSID : Show WPS information about a given station (use 'all', '*' or a broadcast BSSID for all).
- wifi.show : Show current wireless stations list (default sorting by essid).
- wifi.recon.channel CHANNEL : WiFi channels (comma separated) or 'clear' for channel hopping.

Parameters

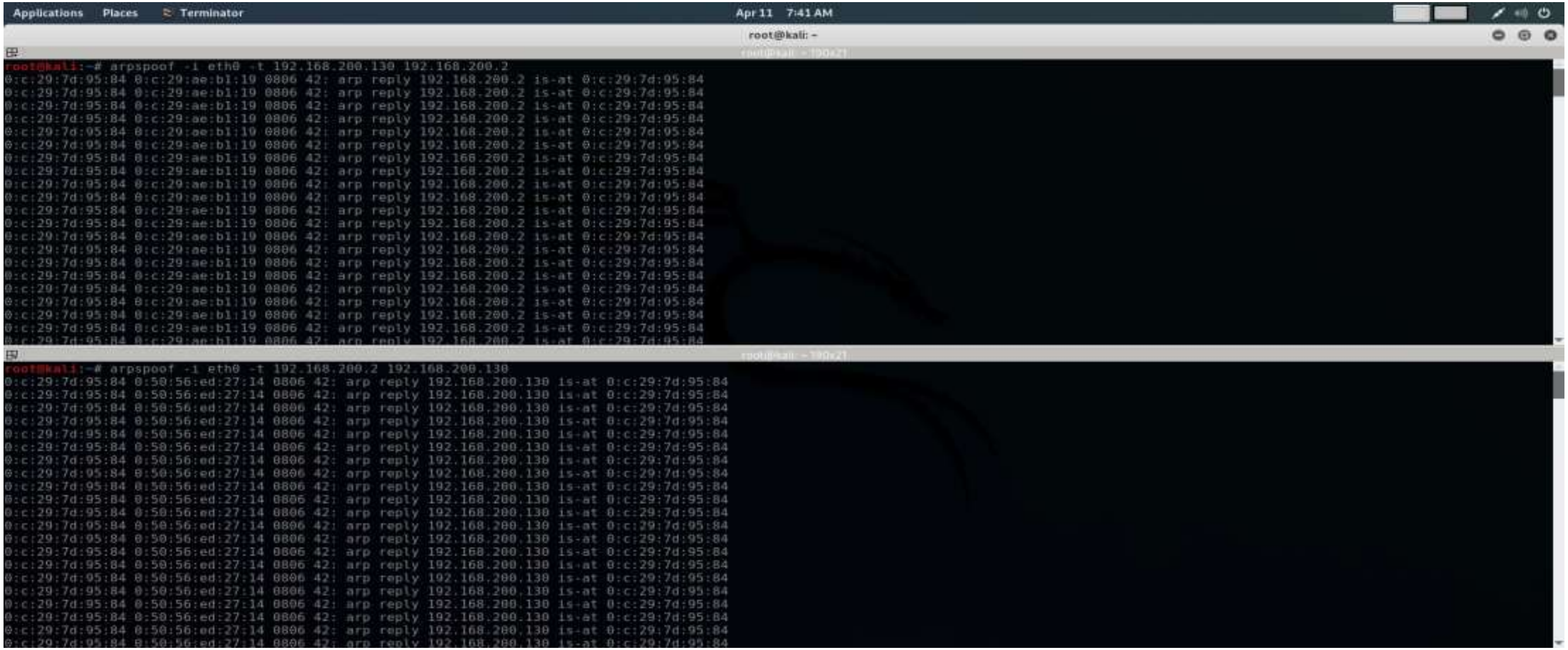
- wifi.ap.bssid : BSSID of the fake access point. (default=<random mac>)
- wifi.ap.channel : Channel of the fake access point. (default=1)
- wifi.ap.encryption : If true, the fake access point will use WPA2, otherwise it'll result as an open AP. (default=true)
- wifi.ap.ssid : SSID of the fake access point. (default=FreeWiFi)
- wifi.ap.ttl : Seconds of inactivity for an access points to be considered not in range anymore. (default=300)
- wifi.assoc.acquired : Send association to AP's for which key material was already acquired. (default=false)
- wifi.assoc.open : Send association requests to open networks. (default=false)
- wifi.assoc.silent : If true, messages from wifi.assoc will be suppressed. (default=false)
- wifi.assoc.skip : Comma separated list of BSSID to skip while sending association requests. (default=)
- wifi.channel_switch_announce.silent : If true, messages from wifi.channel_switch_announce will be suppressed. (default=false)
- wifi.deauth.acquired : Send wifi deauth packets from AP's for which key material was already acquired. (default=false)
- wifi.deauth.open : Send wifi deauth packets to open networks. (default=true)
- wifi.deauth.silent : If true, messages from wifi.deauth will be suppressed. (default=false)
- wifi.deauth.skip : Comma separated list of BSSID to skip while sending deauth packets. (default=)

- wifi.fake_auth.silent : If true, messages from wifi.fake_auth will be suppressed. (default=false)
- wifi.handshakes.aggregate : If true, all handshakes will be saved inside a single file, otherwise a folder with per-network pcap files will be created. (default=true)
- wifi.handshakes.file : File path of the pcap file to save handshakes to. (default=~/bettercap-wifi-handshakes.pcap)
- wifi.hop.period : If channel hopping is enabled (empty wifi.recon.channel), this is the time in milliseconds the algorithm will hop on every channel (it'll be doubled if both 2.4 and 5.0 bands are available). (default=250)
- wifi.interface : If filled, will use this interface name instead of the one provided by the -iface argument or detected automatically. (default=)
- wifi.region : Set the WiFi region to this value before activating the interface. (default=)
- wifi.rssi.min : Minimum WiFi signal strength in dBm. (default=-200)
- wifi.show.filter : Defines a regular expression filter for wifi.show (default=)
- wifi.show.limit : Defines limit for wifi.show (default=0)
- wifi.show.manufacturer : If true, wifi.show will also show the devices manufacturers. (default=false)
- wifi.show.sort : Defines sorting field (rssi, bssid, essid, channel, encryption, clients, seen, sent, rcvd) and direction (asc or desc) for wifi.show (default=rssi asc)
- wifi.skip-broken : If true, dot11 packets with an invalid checksum will be skipped. (default=true)
- wifi.source.file : If set, the wifi module will read from this pcap file instead of the hardware interface. (default=)
- wifi.sta.ttl : Seconds of inactivity for a client station to be considered not in range or not connected to its access point anymore. (default=300)
- wifi.txpower : Set WiFi transmission power to this value before activating the interface. (default=30)

[30] wol -> A module to send Wake On LAN packets in broadcast or to a specific MAC.

- wol.eth MAC : Send a WOL as a raw ethernet packet of type 0x0847 (if no MAC is specified, ff:ff:ff:ff:ff:ff will be used).
- wol.udp MAC : Send a WOL as an IPv4 broadcast packet to UDP port 9 (if no MAC is specified, ff:ff:ff:ff:ff:ff will be used).

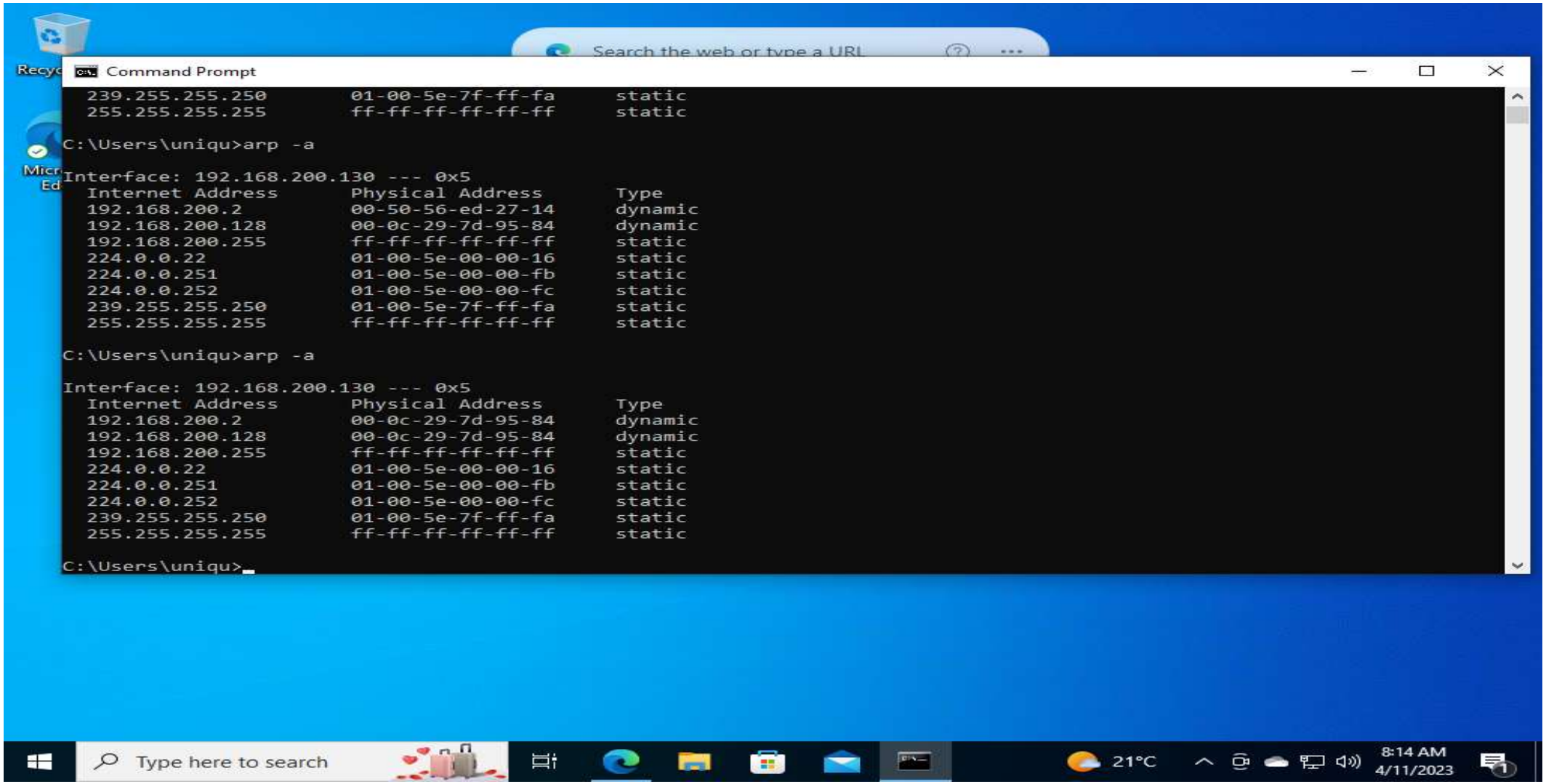
Spoofing Using arp.spoof



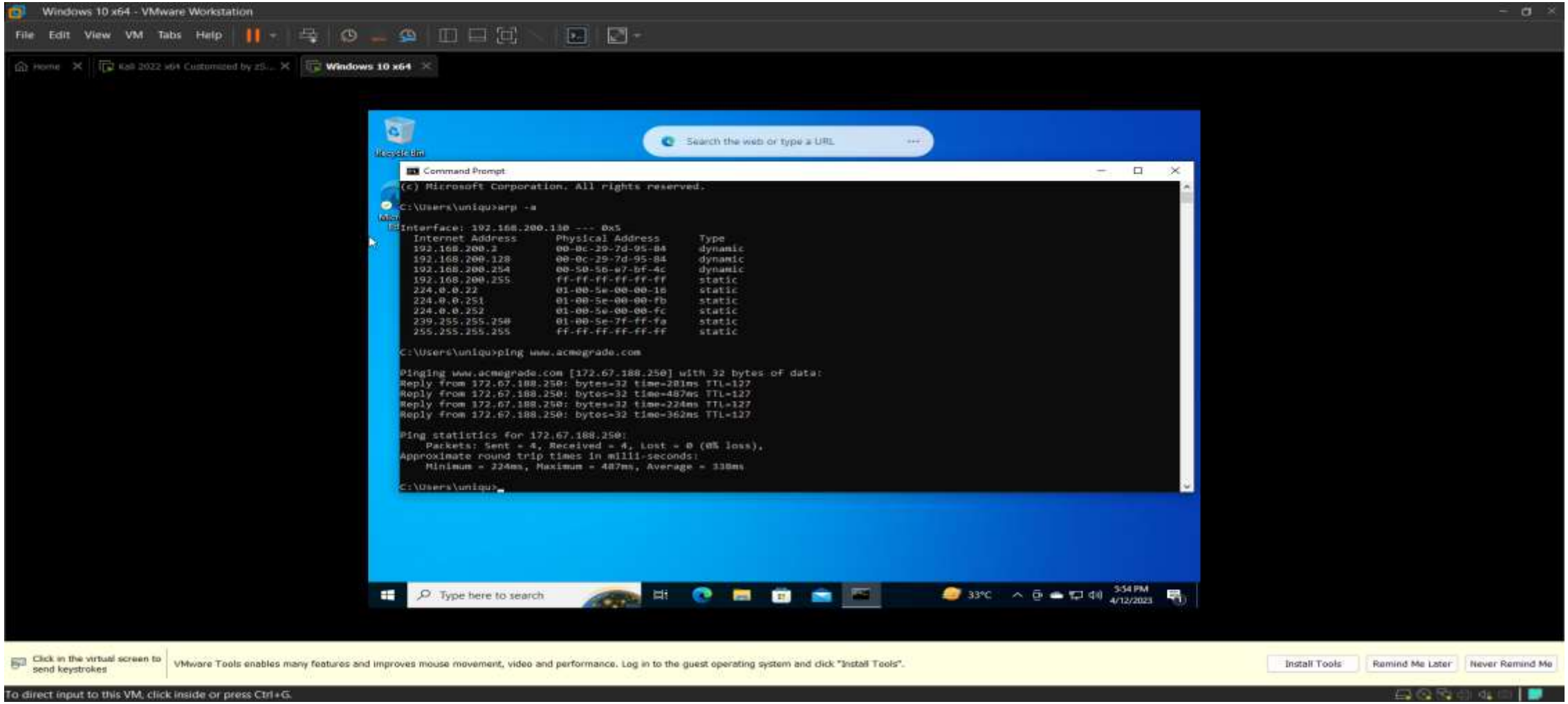
```
root@kali:~# arpspoof -i eth0 -t 192.168.200.130 192.168.200.2
0:c:29:7d:95:84 0:c:29:ae:b1:19 0806 42: arp reply 192.168.200.2 is-at 0:c:29:7d:95:84
0:c:29:7d:95:84 0:c:29:ae:b1:19 0806 42: arp reply 192.168.200.2 is-at 0:c:29:7d:95:84
0:c:29:7d:95:84 0:c:29:ae:b1:19 0806 42: arp reply 192.168.200.2 is-at 0:c:29:7d:95:84
0:c:29:7d:95:84 0:c:29:ae:b1:19 0806 42: arp reply 192.168.200.2 is-at 0:c:29:7d:95:84
0:c:29:7d:95:84 0:c:29:ae:b1:19 0806 42: arp reply 192.168.200.2 is-at 0:c:29:7d:95:84
0:c:29:7d:95:84 0:c:29:ae:b1:19 0806 42: arp reply 192.168.200.2 is-at 0:c:29:7d:95:84
0:c:29:7d:95:84 0:c:29:ae:b1:19 0806 42: arp reply 192.168.200.2 is-at 0:c:29:7d:95:84
0:c:29:7d:95:84 0:c:29:ae:b1:19 0806 42: arp reply 192.168.200.2 is-at 0:c:29:7d:95:84
0:c:29:7d:95:84 0:c:29:ae:b1:19 0806 42: arp reply 192.168.200.2 is-at 0:c:29:7d:95:84
0:c:29:7d:95:84 0:c:29:ae:b1:19 0806 42: arp reply 192.168.200.2 is-at 0:c:29:7d:95:84
0:c:29:7d:95:84 0:c:29:ae:b1:19 0806 42: arp reply 192.168.200.2 is-at 0:c:29:7d:95:84
0:c:29:7d:95:84 0:c:29:ae:b1:19 0806 42: arp reply 192.168.200.2 is-at 0:c:29:7d:95:84
0:c:29:7d:95:84 0:c:29:ae:b1:19 0806 42: arp reply 192.168.200.2 is-at 0:c:29:7d:95:84
0:c:29:7d:95:84 0:c:29:ae:b1:19 0806 42: arp reply 192.168.200.2 is-at 0:c:29:7d:95:84
0:c:29:7d:95:84 0:c:29:ae:b1:19 0806 42: arp reply 192.168.200.2 is-at 0:c:29:7d:95:84
0:c:29:7d:95:84 0:c:29:ae:b1:19 0806 42: arp reply 192.168.200.2 is-at 0:c:29:7d:95:84
0:c:29:7d:95:84 0:c:29:ae:b1:19 0806 42: arp reply 192.168.200.2 is-at 0:c:29:7d:95:84
0:c:29:7d:95:84 0:c:29:ae:b1:19 0806 42: arp reply 192.168.200.2 is-at 0:c:29:7d:95:84
0:c:29:7d:95:84 0:c:29:ae:b1:19 0806 42: arp reply 192.168.200.2 is-at 0:c:29:7d:95:84
0:c:29:7d:95:84 0:c:29:ae:b1:19 0806 42: arp reply 192.168.200.2 is-at 0:c:29:7d:95:84
```

```
root@kali:~# arpspoof -i eth0 -t 192.168.200.2 192.168.200.130
0:c:29:7d:95:84 0:50:56:ed:27:14 0806 42: arp reply 192.168.200.130 is-at 0:c:29:7d:95:84
0:c:29:7d:95:84 0:50:56:ed:27:14 0806 42: arp reply 192.168.200.130 is-at 0:c:29:7d:95:84
0:c:29:7d:95:84 0:50:56:ed:27:14 0806 42: arp reply 192.168.200.130 is-at 0:c:29:7d:95:84
0:c:29:7d:95:84 0:50:56:ed:27:14 0806 42: arp reply 192.168.200.130 is-at 0:c:29:7d:95:84
0:c:29:7d:95:84 0:50:56:ed:27:14 0806 42: arp reply 192.168.200.130 is-at 0:c:29:7d:95:84
0:c:29:7d:95:84 0:50:56:ed:27:14 0806 42: arp reply 192.168.200.130 is-at 0:c:29:7d:95:84
0:c:29:7d:95:84 0:50:56:ed:27:14 0806 42: arp reply 192.168.200.130 is-at 0:c:29:7d:95:84
0:c:29:7d:95:84 0:50:56:ed:27:14 0806 42: arp reply 192.168.200.130 is-at 0:c:29:7d:95:84
0:c:29:7d:95:84 0:50:56:ed:27:14 0806 42: arp reply 192.168.200.130 is-at 0:c:29:7d:95:84
0:c:29:7d:95:84 0:50:56:ed:27:14 0806 42: arp reply 192.168.200.130 is-at 0:c:29:7d:95:84
0:c:29:7d:95:84 0:50:56:ed:27:14 0806 42: arp reply 192.168.200.130 is-at 0:c:29:7d:95:84
0:c:29:7d:95:84 0:50:56:ed:27:14 0806 42: arp reply 192.168.200.130 is-at 0:c:29:7d:95:84
0:c:29:7d:95:84 0:50:56:ed:27:14 0806 42: arp reply 192.168.200.130 is-at 0:c:29:7d:95:84
0:c:29:7d:95:84 0:50:56:ed:27:14 0806 42: arp reply 192.168.200.130 is-at 0:c:29:7d:95:84
0:c:29:7d:95:84 0:50:56:ed:27:14 0806 42: arp reply 192.168.200.130 is-at 0:c:29:7d:95:84
0:c:29:7d:95:84 0:50:56:ed:27:14 0806 42: arp reply 192.168.200.130 is-at 0:c:29:7d:95:84
0:c:29:7d:95:84 0:50:56:ed:27:14 0806 42: arp reply 192.168.200.130 is-at 0:c:29:7d:95:84
0:c:29:7d:95:84 0:50:56:ed:27:14 0806 42: arp reply 192.168.200.130 is-at 0:c:29:7d:95:84
0:c:29:7d:95:84 0:50:56:ed:27:14 0806 42: arp reply 192.168.200.130 is-at 0:c:29:7d:95:84
0:c:29:7d:95:84 0:50:56:ed:27:14 0806 42: arp reply 192.168.200.130 is-at 0:c:29:7d:95:84
```

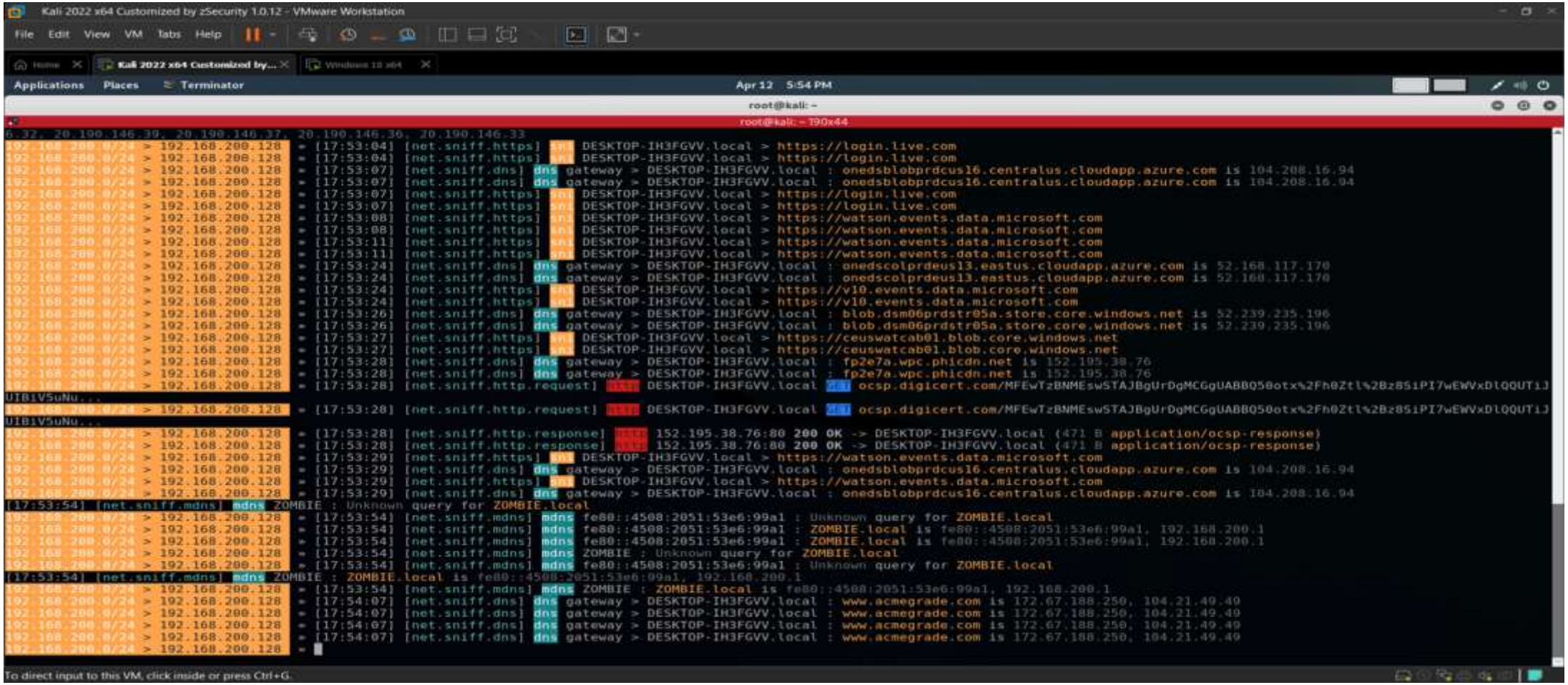
Linking IP address with mac address



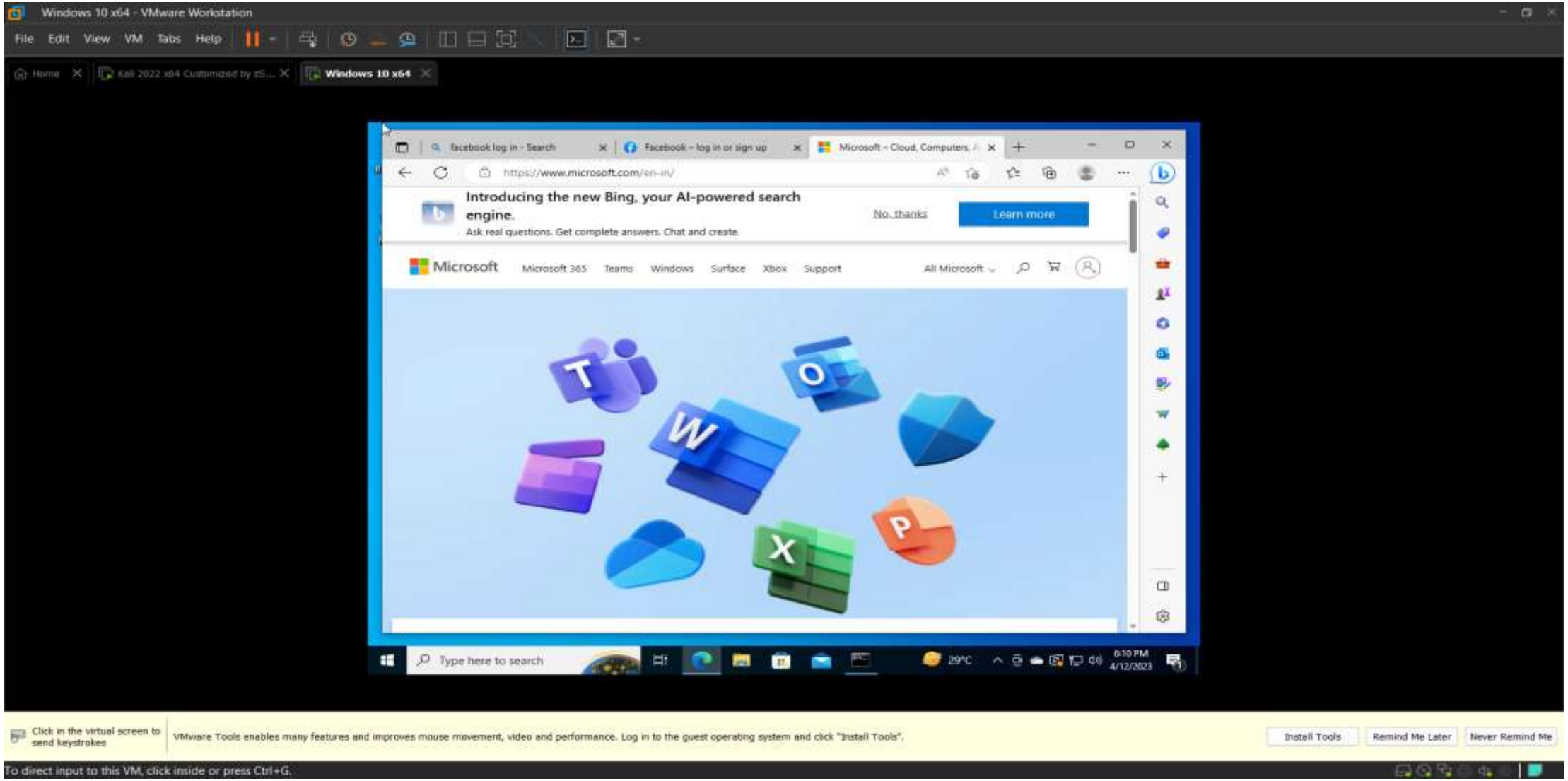
Sniffing using net.sniff



As I ping to acmegrade in my target machine in the previous slide so now it can be seen in the bottom most in the below picture in my hacking machine.



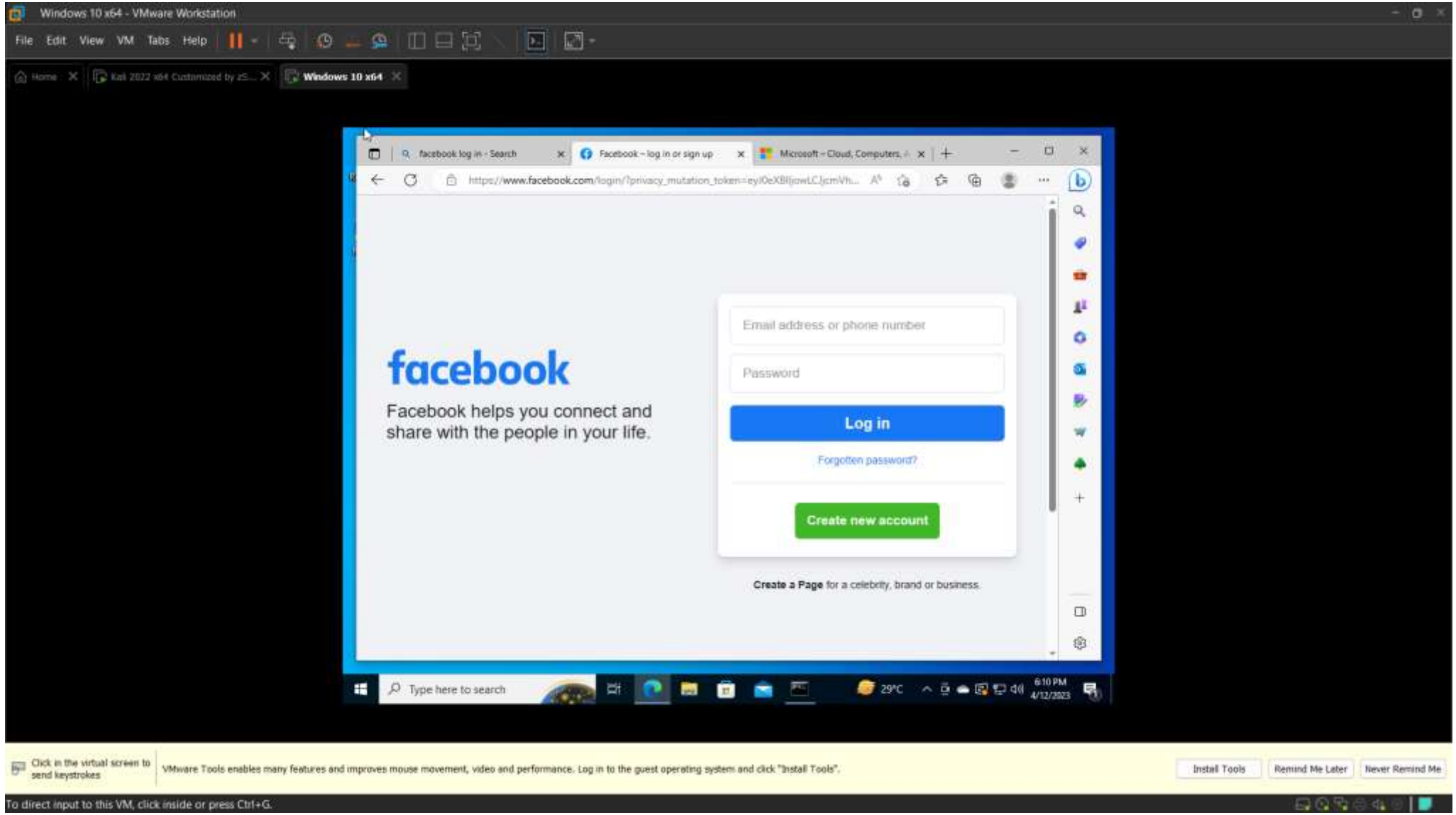
Searched for Microsoft in bing (target mahine)



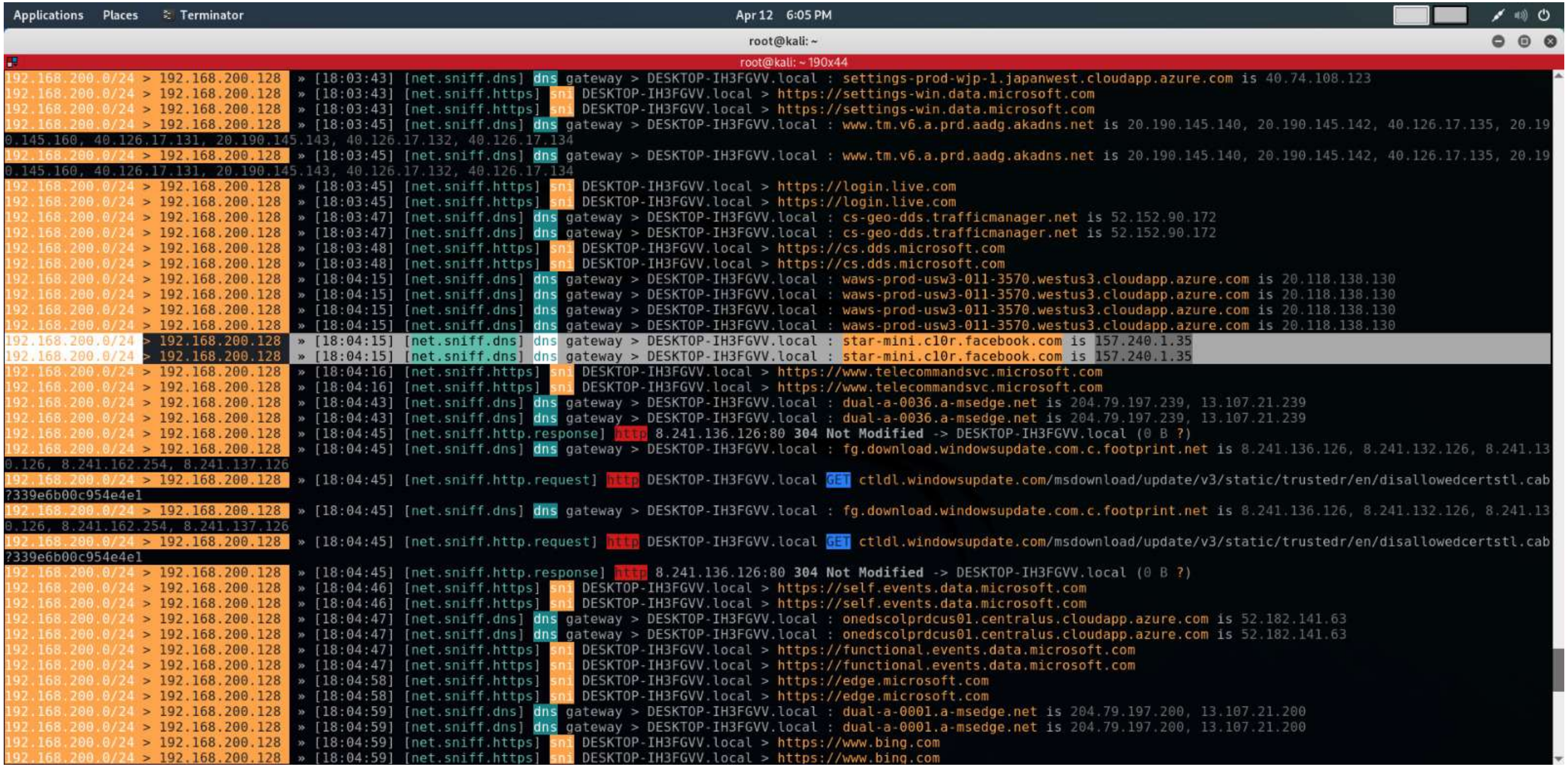
And the results getting displayed in hacking machine.

```
Applications  Places  Terminator                                   Apr 12  6:13 PM
                                                                   root@kali: ~
192.168.200.0/24 > 192.168.200.128  » [18:08:29] [net.sniff.https] sni DESKTOP-IH3FGVV.local > https://browser.events.data.microsoft.com
192.168.200.0/24 > 192.168.200.128  » [18:08:52] [net.sniff.dns] dns gateway > DESKTOP-IH3FGVV.local : dual-a-0001.a-msedge.net is 204.79.197.200, 13.107.21.200
192.168.200.0/24 > 192.168.200.128  » [18:08:52] [net.sniff.dns] dns gateway > DESKTOP-IH3FGVV.local : dual-a-0001.a-msedge.net is 204.79.197.200, 13.107.21.200
192.168.200.0/24 > 192.168.200.128  » [18:09:00] [net.sniff.https] sni DESKTOP-IH3FGVV.local > https://q.clarity.ms
192.168.200.0/24 > 192.168.200.128  » [18:09:00] [net.sniff.https] sni DESKTOP-IH3FGVV.local > https://q.clarity.ms
192.168.200.0/24 > 192.168.200.128  » [18:09:00] [net.sniff.https] sni DESKTOP-IH3FGVV.local > https://q.clarity.ms
192.168.200.0/24 > 192.168.200.128  » [18:09:00] [net.sniff.https] sni DESKTOP-IH3FGVV.local > https://q.clarity.ms
192.168.200.0/24 > 192.168.200.128  » [18:09:01] [net.sniff.mdns] mdns ZOMBIE : PTR query for _spotify-connect._tcp.local
192.168.200.0/24 > 192.168.200.128  » [18:09:01] [net.sniff.mdns] mdns fe80::4508:2051:53e6:99a1 : PTR query for _spotify-connect._tcp.local
192.168.200.0/24 > 192.168.200.128  » [18:09:01] [net.sniff.mdns] mdns fe80::4508:2051:53e6:99a1 : PTR query for _spotify-connect._tcp.local
192.168.200.0/24 > 192.168.200.128  » [18:09:01] [net.sniff.mdns] mdns fe80::4508:2051:53e6:99a1 : PTR query for _spotify-connect._tcp.local
192.168.200.0/24 > 192.168.200.128  » [18:09:15] [net.sniff.https] sni DESKTOP-IH3FGVV.local > https://cdn-dynmedia-1.microsoft.com
192.168.200.0/24 > 192.168.200.128  » [18:09:15] [net.sniff.https] sni DESKTOP-IH3FGVV.local > https://cdn-dynmedia-1.microsoft.com
192.168.200.0/24 > 192.168.200.128  » [18:09:15] [net.sniff.https] sni DESKTOP-IH3FGVV.local > https://statics-marketingsites-eas-ms-com.akamaized.net
192.168.200.0/24 > 192.168.200.128  » [18:09:15] [net.sniff.https] sni DESKTOP-IH3FGVV.local > https://statics-marketingsites-eas-ms-com.akamaized.net
192.168.200.0/24 > 192.168.200.128  » [18:09:15] [net.sniff.https] sni DESKTOP-IH3FGVV.local > https://www.microsoft.com
192.168.200.0/24 > 192.168.200.128  » [18:09:15] [net.sniff.https] sni DESKTOP-IH3FGVV.local > https://img-prod-cms-rt-microsoft-com.akamaized.net
192.168.200.0/24 > 192.168.200.128  » [18:09:15] [net.sniff.https] sni DESKTOP-IH3FGVV.local > https://img-prod-cms-rt-microsoft-com.akamaized.net
192.168.200.0/24 > 192.168.200.128  » [18:09:15] [net.sniff.https] sni DESKTOP-IH3FGVV.local > https://www.microsoft.com
[18:09:15] [net.sniff.https] sni DESKTOP-IH3FGVV.local > https://via.placeholder.com
192.168.200.0/24 > 192.168.200.128  » [18:09:15] [net.sniff.https] sni DESKTOP-IH3FGVV.local > https://via.placeholder.com
192.168.200.0/24 > 192.168.200.128  » [18:09:16] [net.sniff.dns] dns gateway > DESKTOP-IH3FGVV.local : k.bf.contentsquare.net is 3.223.120.108, 52.200.137.54, 34.234.114.51, 54.224.59.87, 54
.163.164.32, 52.205.78.47, 3.216.0.125, 34.196.23.220
192.168.200.0/24 > 192.168.200.128  » [18:09:16] [net.sniff.dns] dns gateway > DESKTOP-IH3FGVV.local : k.bf.contentsquare.net is 3.223.120.108, 52.200.137.54, 34.234.114.51, 54.224.59.87, 54
.163.164.32, 52.205.78.47, 3.216.0.125, 34.196.23.220
192.168.200.0/24 > 192.168.200.128  » [18:09:17] [net.sniff.https] sni DESKTOP-IH3FGVV.local > https://k-aus1.clicktale.net
192.168.200.0/24 > 192.168.200.128  » [18:09:17] [net.sniff.https] sni DESKTOP-IH3FGVV.local > https://k-aus1.clicktale.net
192.168.200.0/24 > 192.168.200.128  » [18:09:17] [net.sniff.https] sni DESKTOP-IH3FGVV.local > https://k-aus1.clicktale.net
192.168.200.0/24 > 192.168.200.128  » [18:09:17] [net.sniff.https] sni DESKTOP-IH3FGVV.local > https://k-aus1.clicktale.net
192.168.200.0/24 > 192.168.200.128  » [18:09:26] [net.sniff.dns] dns gateway > DESKTOP-IH3FGVV.local : clarity-ingest-eus-b-sc.eastus.cloudapp.azure.com is 20.231.53.73
192.168.200.0/24 > 192.168.200.128  » [18:09:26] [net.sniff.dns] dns gateway > DESKTOP-IH3FGVV.local : clarity-ingest-eus-b-sc.eastus.cloudapp.azure.com is 20.231.53.73
192.168.200.0/24 > 192.168.200.128  » [18:09:27] [net.sniff.https] sni DESKTOP-IH3FGVV.local > https://q.clarity.ms
192.168.200.0/24 > 192.168.200.128  » [18:09:27] [net.sniff.https] sni DESKTOP-IH3FGVV.local > https://q.clarity.ms
192.168.200.0/24 > 192.168.200.128  » [18:09:30] [net.sniff.dns] dns gateway > DESKTOP-IH3FGVV.local : onedscolprdweu03.westeurope.cloudapp.azure.com is 13.69.109.131
192.168.200.0/24 > 192.168.200.128  » [18:09:30] [net.sniff.dns] dns gateway > DESKTOP-IH3FGVV.local : onedscolprdweu03.westeurope.cloudapp.azure.com is 13.69.109.131
192.168.200.0/24 > 192.168.200.128  » [18:09:31] [net.sniff.https] sni DESKTOP-IH3FGVV.local > https://browser.events.data.microsoft.com
192.168.200.0/24 > 192.168.200.128  » [18:09:31] [net.sniff.https] sni DESKTOP-IH3FGVV.local > https://browser.events.data.microsoft.com
192.168.200.0/24 > 192.168.200.128  » [18:10:02] [net.sniff.https] sni DESKTOP-IH3FGVV.local > https://q.clarity.ms
192.168.200.0/24 > 192.168.200.128  » [18:10:02] [net.sniff.https] sni DESKTOP-IH3FGVV.local > https://q.clarity.ms
192.168.200.0/24 > 192.168.200.128  » [18:10:15] [net.sniff.dns] dns gateway > DESKTOP-IH3FGVV.local : a1339.g2.akamai.net is 23.76.156.18, 23.76.156.8
192.168.200.0/24 > 192.168.200.128  » [18:10:15] [net.sniff.dns] dns gateway > DESKTOP-IH3FGVV.local : e13678.dscb.akamaiedge.net is 104.91.65.176
192.168.200.0/24 > 192.168.200.128  » [18:10:15] [net.sniff.dns] dns gateway > DESKTOP-IH3FGVV.local : e13678.dscb.akamaiedge.net is 104.91.65.176
192.168.200.0/24 > 192.168.200.128  » [18:10:15] [net.sniff.dns] dns gateway > DESKTOP-IH3FGVV.local : a1339.g2.akamai.net is 23.76.156.18, 23.76.156.8
192.168.200.0/24 > 192.168.200.128  » [18:10:15] [net.sniff.dns] dns gateway > DESKTOP-IH3FGVV.local : a1449.dscg2.akamai.net is 23.76.156.58, 23.76.156.48
192.168.200.0/24 > 192.168.200.128  » [18:10:15] [net.sniff.dns] dns gateway > DESKTOP-IH3FGVV.local : a1449.dscg2.akamai.net is 23.76.156.58, 23.76.156.48
```

Searched for facebook in bing (target machine)



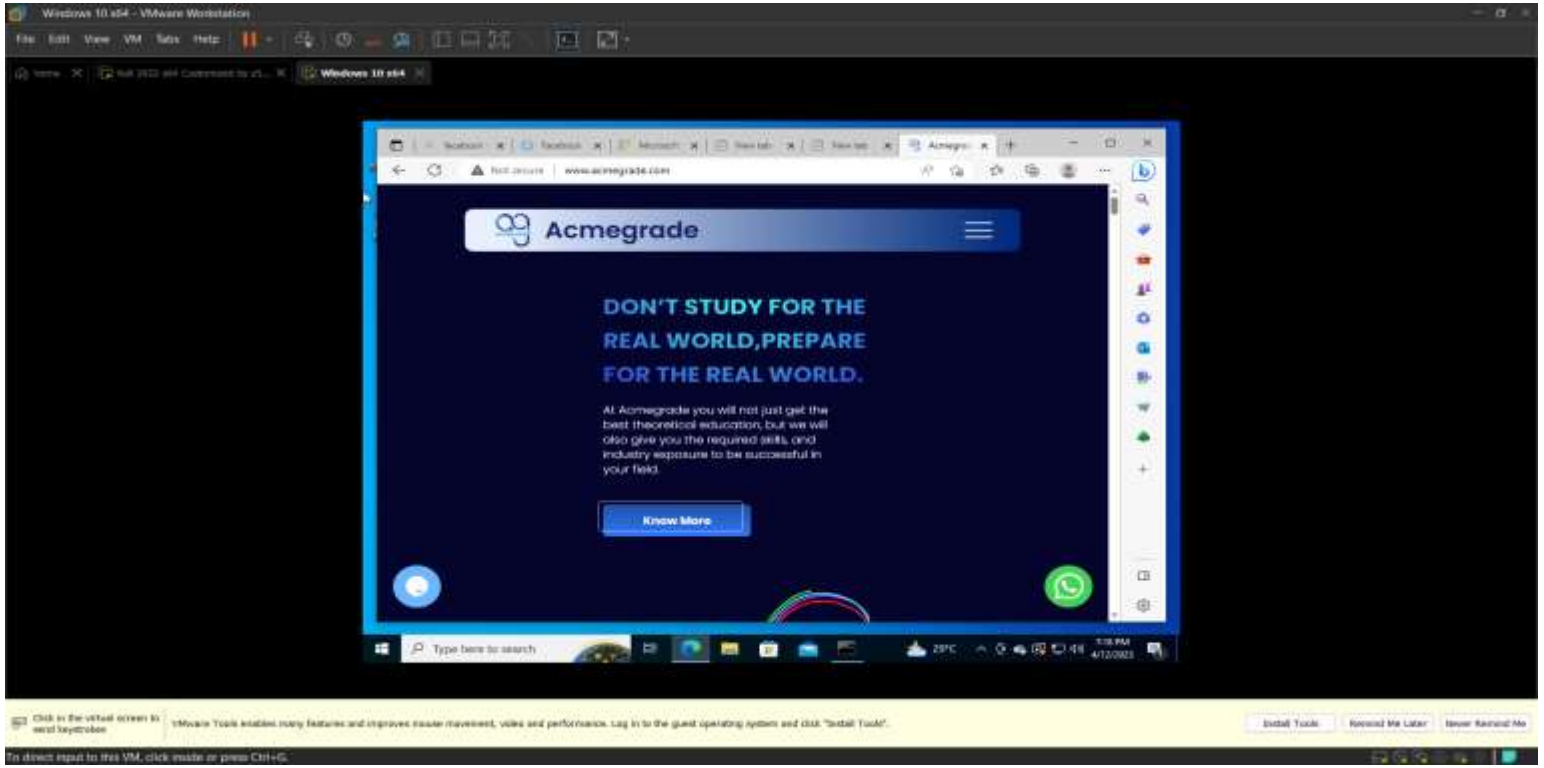
Results getting displayed in my hacking machine.



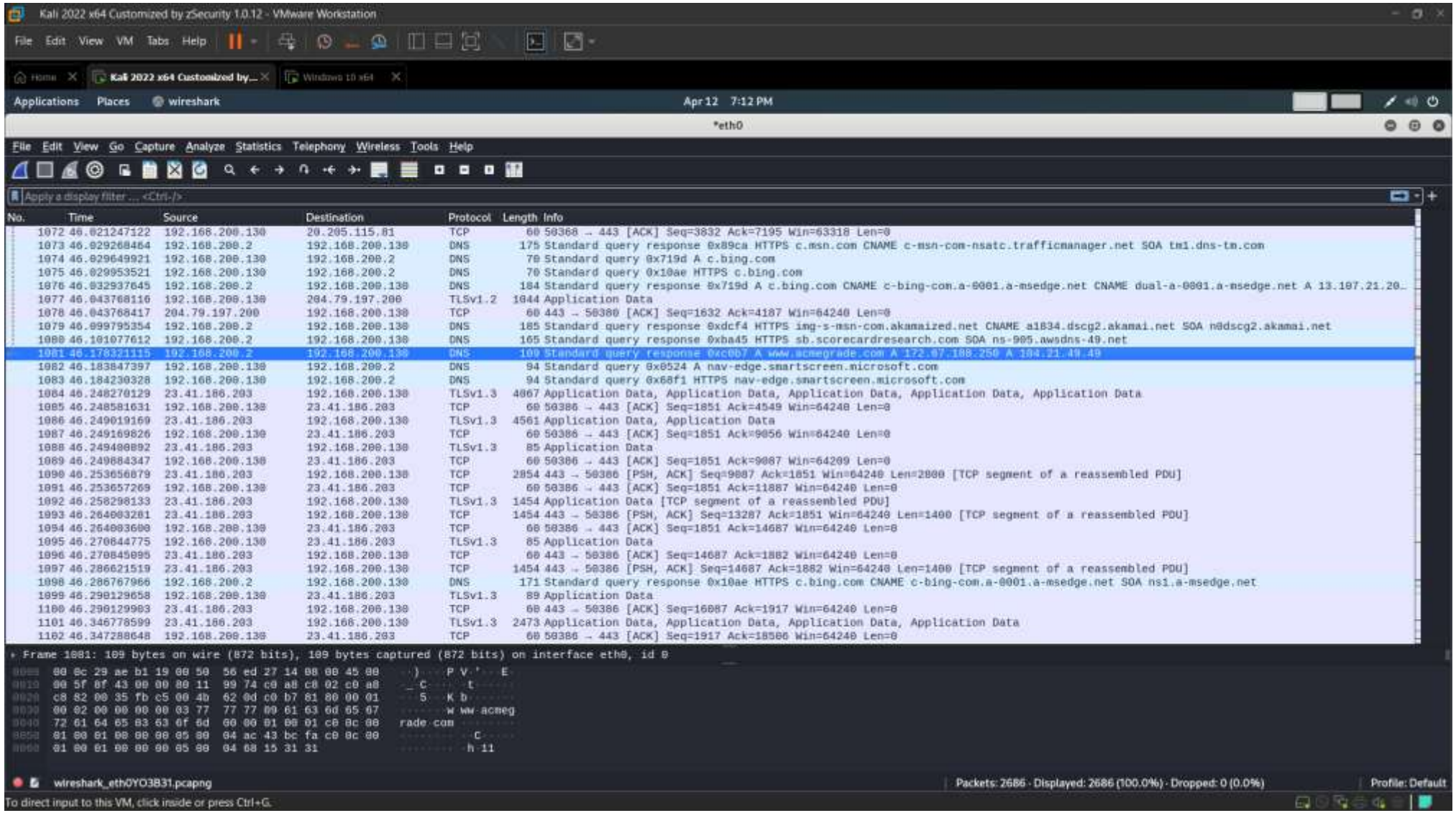
```
192.168.200.0/24 > 192.168.200.128  » [18:03:43] [net.sniff.dns] dns gateway > DESKTOP-IH3FGVV.local : settings-prod-wjp-1.japanwest.cloudapp.azure.com is 40.74.108.123
192.168.200.0/24 > 192.168.200.128  » [18:03:43] [net.sniff.https] sni DESKTOP-IH3FGVV.local > https://settings-win.data.microsoft.com
192.168.200.0/24 > 192.168.200.128  » [18:03:43] [net.sniff.https] sni DESKTOP-IH3FGVV.local > https://settings-win.data.microsoft.com
192.168.200.0/24 > 192.168.200.128  » [18:03:45] [net.sniff.dns] dns gateway > DESKTOP-IH3FGVV.local : www.tm.v6.a.prd.aadg.akadns.net is 20.190.145.140, 20.190.145.142, 40.126.17.135, 20.19
0.145.160, 40.126.17.131, 20.190.145.143, 40.126.17.132, 40.126.17.134
192.168.200.0/24 > 192.168.200.128  » [18:03:45] [net.sniff.dns] dns gateway > DESKTOP-IH3FGVV.local : www.tm.v6.a.prd.aadg.akadns.net is 20.190.145.140, 20.190.145.142, 40.126.17.135, 20.19
0.145.160, 40.126.17.131, 20.190.145.143, 40.126.17.132, 40.126.17.134
192.168.200.0/24 > 192.168.200.128  » [18:03:45] [net.sniff.https] sni DESKTOP-IH3FGVV.local > https://login.live.com
192.168.200.0/24 > 192.168.200.128  » [18:03:45] [net.sniff.https] sni DESKTOP-IH3FGVV.local > https://login.live.com
192.168.200.0/24 > 192.168.200.128  » [18:03:47] [net.sniff.dns] dns gateway > DESKTOP-IH3FGVV.local : cs-geo-dds.trafficmanager.net is 52.152.90.172
192.168.200.0/24 > 192.168.200.128  » [18:03:47] [net.sniff.dns] dns gateway > DESKTOP-IH3FGVV.local : cs-geo-dds.trafficmanager.net is 52.152.90.172
192.168.200.0/24 > 192.168.200.128  » [18:03:48] [net.sniff.https] sni DESKTOP-IH3FGVV.local > https://cs.dds.microsoft.com
192.168.200.0/24 > 192.168.200.128  » [18:03:48] [net.sniff.https] sni DESKTOP-IH3FGVV.local > https://cs.dds.microsoft.com
192.168.200.0/24 > 192.168.200.128  » [18:04:15] [net.sniff.dns] dns gateway > DESKTOP-IH3FGVV.local : waws-prod-usw3-011-3570.westus3.cloudapp.azure.com is 20.118.138.130
192.168.200.0/24 > 192.168.200.128  » [18:04:15] [net.sniff.dns] dns gateway > DESKTOP-IH3FGVV.local : waws-prod-usw3-011-3570.westus3.cloudapp.azure.com is 20.118.138.130
192.168.200.0/24 > 192.168.200.128  » [18:04:15] [net.sniff.dns] dns gateway > DESKTOP-IH3FGVV.local : waws-prod-usw3-011-3570.westus3.cloudapp.azure.com is 20.118.138.130
192.168.200.0/24 > 192.168.200.128  » [18:04:15] [net.sniff.dns] dns gateway > DESKTOP-IH3FGVV.local : waws-prod-usw3-011-3570.westus3.cloudapp.azure.com is 20.118.138.130
192.168.200.0/24 > 192.168.200.128  » [18:04:15] [net.sniff.dns] dns gateway > DESKTOP-IH3FGVV.local : star-mini.c10r.facebook.com is 157.240.1.35
192.168.200.0/24 > 192.168.200.128  » [18:04:15] [net.sniff.dns] dns gateway > DESKTOP-IH3FGVV.local : star-mini.c10r.facebook.com is 157.240.1.35
192.168.200.0/24 > 192.168.200.128  » [18:04:16] [net.sniff.https] sni DESKTOP-IH3FGVV.local > https://www.telecommandsvc.microsoft.com
192.168.200.0/24 > 192.168.200.128  » [18:04:16] [net.sniff.https] sni DESKTOP-IH3FGVV.local > https://www.telecommandsvc.microsoft.com
192.168.200.0/24 > 192.168.200.128  » [18:04:43] [net.sniff.dns] dns gateway > DESKTOP-IH3FGVV.local : dual-a-0036.a-msedge.net is 204.79.197.239, 13.107.21.239
192.168.200.0/24 > 192.168.200.128  » [18:04:43] [net.sniff.dns] dns gateway > DESKTOP-IH3FGVV.local : dual-a-0036.a-msedge.net is 204.79.197.239, 13.107.21.239
192.168.200.0/24 > 192.168.200.128  » [18:04:45] [net.sniff.http.response] http 8.241.136.126:80 304 Not Modified -> DESKTOP-IH3FGVV.local (0 B ?)
192.168.200.0/24 > 192.168.200.128  » [18:04:45] [net.sniff.dns] dns gateway > DESKTOP-IH3FGVV.local : fg.download.windowsupdate.com.c.footprint.net is 8.241.136.126, 8.241.132.126, 8.241.13
0.126, 8.241.162.254, 8.241.137.126
192.168.200.0/24 > 192.168.200.128  » [18:04:45] [net.sniff.http.request] http DESKTOP-IH3FGVV.local GET ctldl.windowsupdate.com/msdownload/update/v3/static/trustedr/en/disallowedcertstl.cab
?339e6b00c954e4e1
192.168.200.0/24 > 192.168.200.128  » [18:04:45] [net.sniff.dns] dns gateway > DESKTOP-IH3FGVV.local : fg.download.windowsupdate.com.c.footprint.net is 8.241.136.126, 8.241.132.126, 8.241.13
0.126, 8.241.162.254, 8.241.137.126
192.168.200.0/24 > 192.168.200.128  » [18:04:45] [net.sniff.http.request] http DESKTOP-IH3FGVV.local GET ctldl.windowsupdate.com/msdownload/update/v3/static/trustedr/en/disallowedcertstl.cab
?339e6b00c954e4e1
192.168.200.0/24 > 192.168.200.128  » [18:04:45] [net.sniff.http.response] http 8.241.136.126:80 304 Not Modified -> DESKTOP-IH3FGVV.local (0 B ?)
192.168.200.0/24 > 192.168.200.128  » [18:04:46] [net.sniff.https] sni DESKTOP-IH3FGVV.local > https://self.events.data.microsoft.com
192.168.200.0/24 > 192.168.200.128  » [18:04:46] [net.sniff.https] sni DESKTOP-IH3FGVV.local > https://self.events.data.microsoft.com
192.168.200.0/24 > 192.168.200.128  » [18:04:47] [net.sniff.dns] dns gateway > DESKTOP-IH3FGVV.local : onedscolprdcus01.centralus.cloudapp.azure.com is 52.182.141.63
192.168.200.0/24 > 192.168.200.128  » [18:04:47] [net.sniff.dns] dns gateway > DESKTOP-IH3FGVV.local : onedscolprdcus01.centralus.cloudapp.azure.com is 52.182.141.63
192.168.200.0/24 > 192.168.200.128  » [18:04:47] [net.sniff.https] sni DESKTOP-IH3FGVV.local > https://functional.events.data.microsoft.com
192.168.200.0/24 > 192.168.200.128  » [18:04:47] [net.sniff.https] sni DESKTOP-IH3FGVV.local > https://functional.events.data.microsoft.com
192.168.200.0/24 > 192.168.200.128  » [18:04:58] [net.sniff.https] sni DESKTOP-IH3FGVV.local > https://edge.microsoft.com
192.168.200.0/24 > 192.168.200.128  » [18:04:58] [net.sniff.https] sni DESKTOP-IH3FGVV.local > https://edge.microsoft.com
192.168.200.0/24 > 192.168.200.128  » [18:04:59] [net.sniff.dns] dns gateway > DESKTOP-IH3FGVV.local : dual-a-0001.a-msedge.net is 204.79.197.200, 13.107.21.200
192.168.200.0/24 > 192.168.200.128  » [18:04:59] [net.sniff.dns] dns gateway > DESKTOP-IH3FGVV.local : dual-a-0001.a-msedge.net is 204.79.197.200, 13.107.21.200
192.168.200.0/24 > 192.168.200.128  » [18:04:59] [net.sniff.https] sni DESKTOP-IH3FGVV.local > https://www.bing.com
192.168.200.0/24 > 192.168.200.128  » [18:04:59] [net.sniff.https] sni DESKTOP-IH3FGVV.local > https://www.bing.com
```

Capturing and Analyzing packets on wireshark

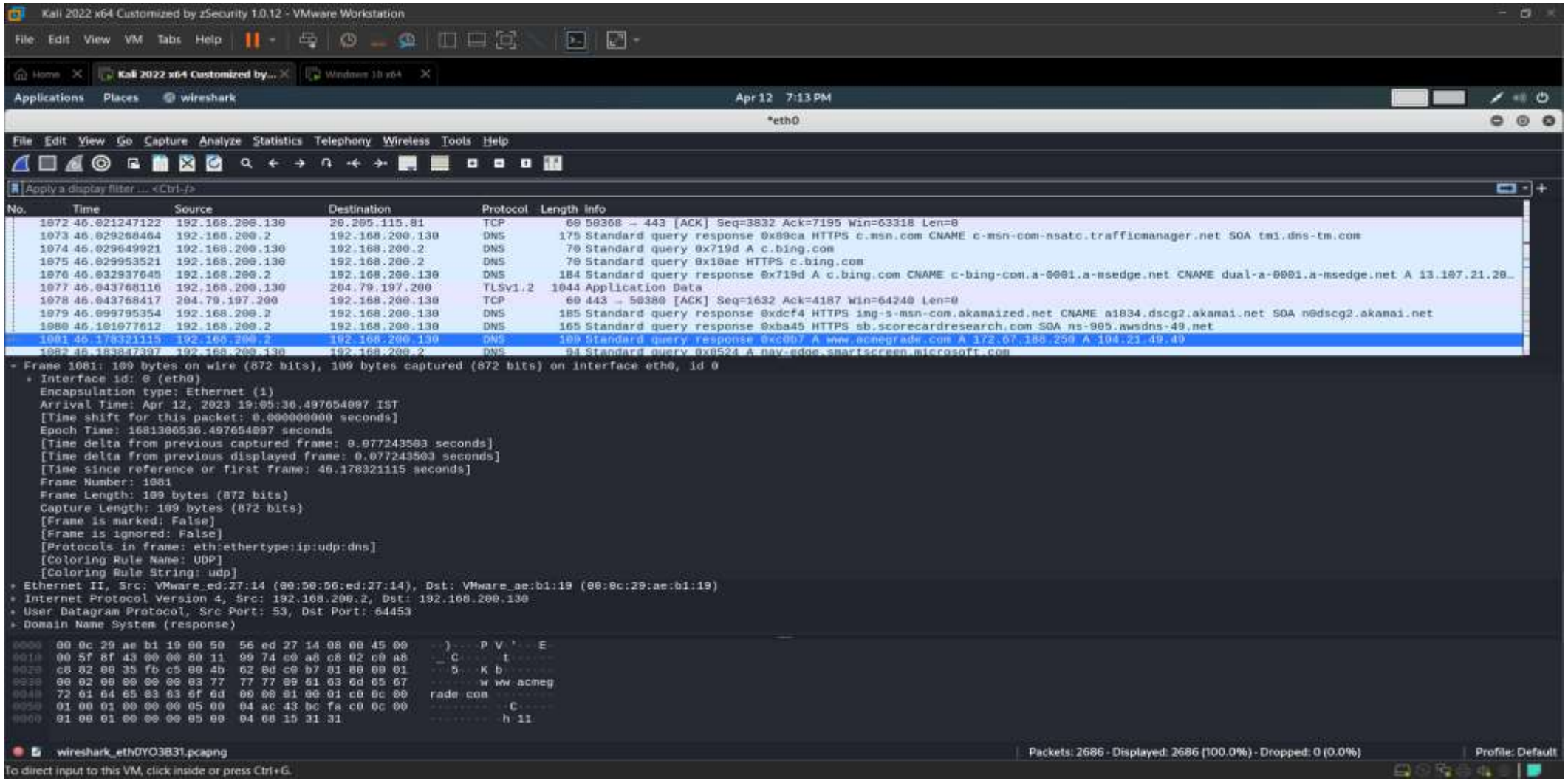
Searched for acmegrade in bing on my target machine



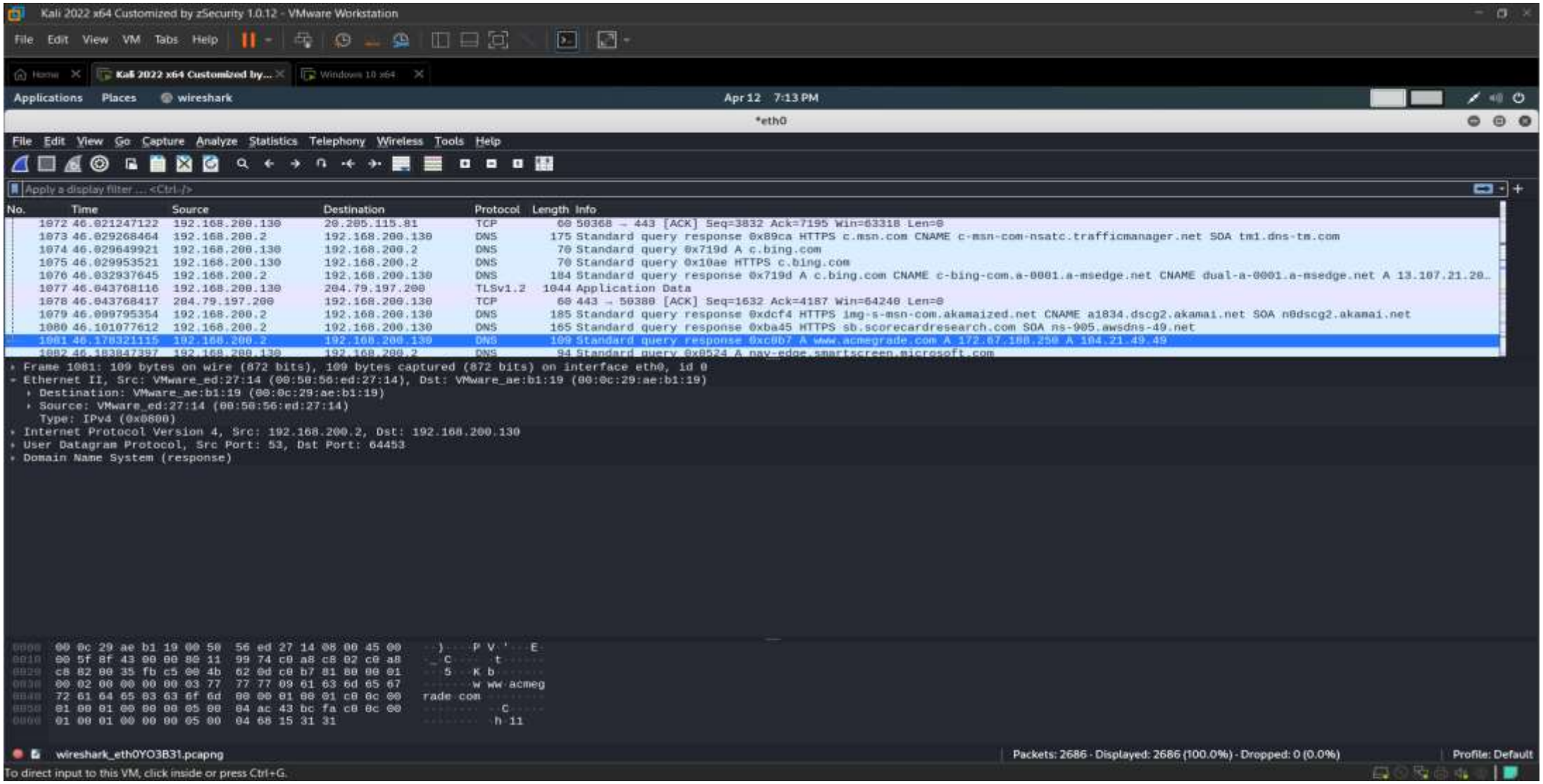
Displaying the searched item in wireshark



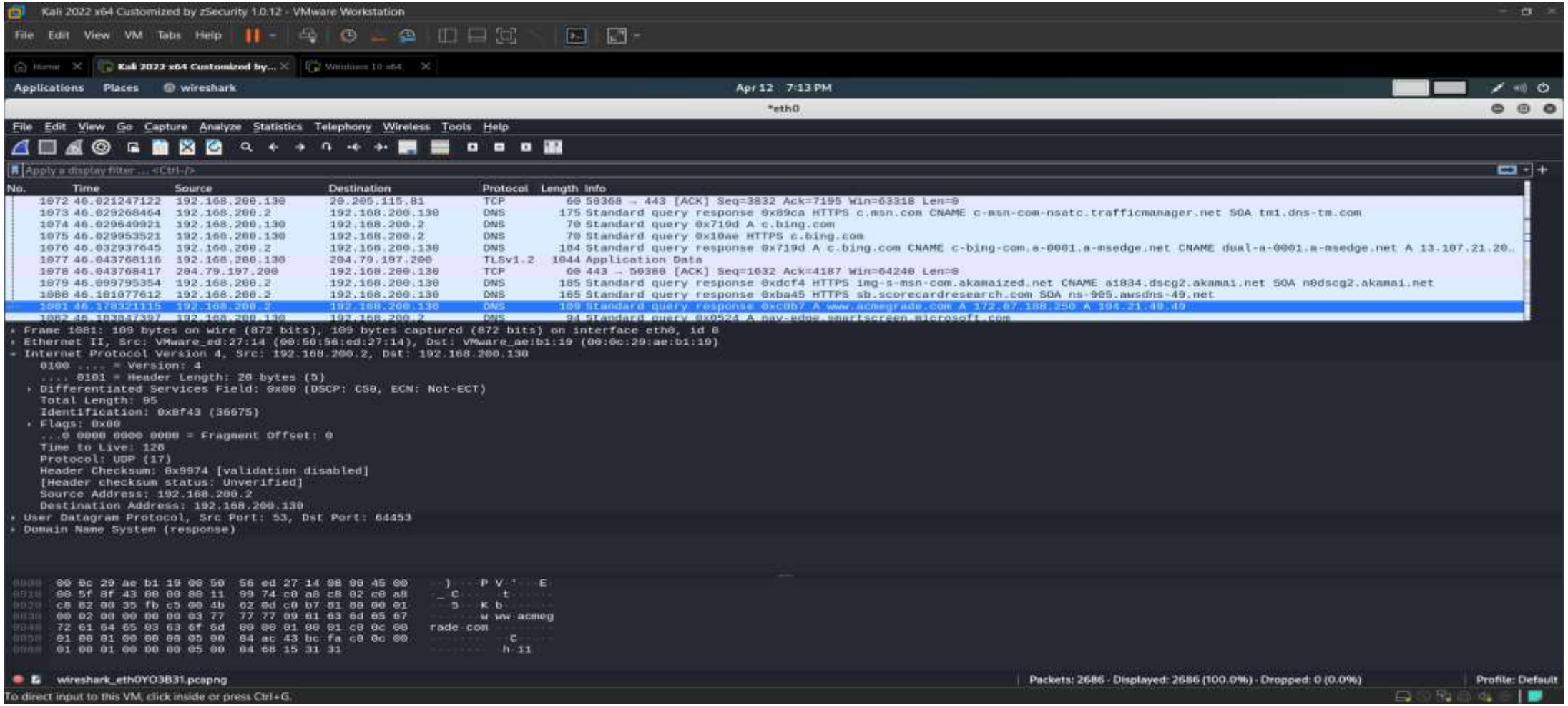
About Frame


Kali 2022 x64 Customized by zSecurity 1.0.12 - VMware Workstation
File Edit View VM Tabs Help
Applications Places wireshark
Apr 12 7:13 PM
*eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
No. Time Source Destination Protocol Length Info
1072 46.021247122 192.168.200.130 20.205.115.81 TCP 60 50368 → 443 [ACK] Seq=3832 Ack=7195 Win=63318 Len=0
1073 46.029268464 192.168.200.2 192.168.200.130 DNS 175 Standard query response 0x89ca HTTPS c.msn.com CNAME c-msn-com-nsatc.trafficmanager.net SOA tm1.dns-tm.com
1074 46.029649921 192.168.200.130 192.168.200.2 DNS 70 Standard query 0x719d A c.bing.com
1075 46.029953521 192.168.200.130 192.168.200.2 DNS 70 Standard query 0x10ae HTTPS c.bing.com
1076 46.032937645 192.168.200.2 192.168.200.130 DNS 184 Standard query response 0x719d A c.bing.com CNAME c-bing-com.a-0001.a-msedge.net CNAME dual-a-0001.a-msedge.net A 13.107.21.20
1077 46.043768116 192.168.200.130 204.79.197.200 TLSv1.2 1044 Application Data
1078 46.043768417 204.79.197.200 192.168.200.130 TCP 60 443 → 50380 [ACK] Seq=1632 Ack=4187 Win=64240 Len=0
1079 46.099795354 192.168.200.2 192.168.200.130 DNS 185 Standard query response 0xdcf4 HTTPS img-s-msn-com.akamaized.net CNAME a1834.dscg2.akamai.net SOA n0dscg2.akamai.net
1080 46.101077612 192.168.200.2 192.168.200.130 DNS 165 Standard query response 0xba45 HTTPS sb.scorecardresearch.com SOA ns-905.awsdns-49.net
1081 46.178321115 192.168.200.2 192.168.200.130 DNS 109 Standard query response 0xc0b7 A www.acmegrade.com A 172.67.188.250 A 104.21.49.49
1082 46.183847397 192.168.200.130 192.168.200.2 DNS 94 Standard query 0x0524 A nav-edge.smartscreen.microsoft.com
Frame 1081: 109 bytes on wire (872 bits), 109 bytes captured (872 bits) on interface eth0, id 0
Interface id: 0 (eth0)
Encapsulation type: Ethernet (1)
Arrival Time: Apr 12, 2023 19:05:36.497654097 IST
[Time shift for this packet: 0.000000000 seconds]
Epoch Time: 1681306536.497654097 seconds
[Time delta from previous captured frame: 0.077243503 seconds]
[Time delta from previous displayed frame: 0.077243503 seconds]
[Time since reference or first frame: 46.178321115 seconds]
Frame Number: 1081
Frame Length: 109 bytes (872 bits)
Capture Length: 109 bytes (872 bits)
[Frame is marked: False]
[Frame is ignored: False]
[Protocols in frame: eth:ethertype:ip:udp:dns]
[Coloring Rule Name: UDP]
[Coloring Rule String: udp]
Ethernet II, Src: VMware_ed:27:14 (00:50:56:ed:27:14), Dst: VMware_ae:b1:19 (00:0c:29:ae:b1:19)
Internet Protocol Version 4, Src: 192.168.200.2, Dst: 192.168.200.130
User Datagram Protocol, Src Port: 53, Dst Port: 64453
Domain Name System (response)
0000 00 0c 29 ae b1 19 00 50 56 ed 27 14 08 00 45 00
0010 00 5f 8f 43 00 00 80 11 99 74 c0 a8 c8 02 c0 a8
0020 c8 82 00 35 fb c5 00 4b 62 8d c0 b7 81 80 00 01
0030 00 02 00 00 00 00 03 77 77 77 09 61 63 6d 65 67
0040 72 61 64 65 03 63 6f 6d 00 00 01 00 01 c0 0c 00
0050 01 00 01 00 00 00 05 00 04 ac 43 bc fa c0 0c 00
0060 01 00 01 00 00 00 05 00 04 68 15 31 31
wireshark_eth0YO3B31.pcapng
Packets: 2686 · Displayed: 2686 (100.0%) · Dropped: 0 (0.0%)
Profile: Default
To direct input to this VM, click inside or press Ctrl+G.

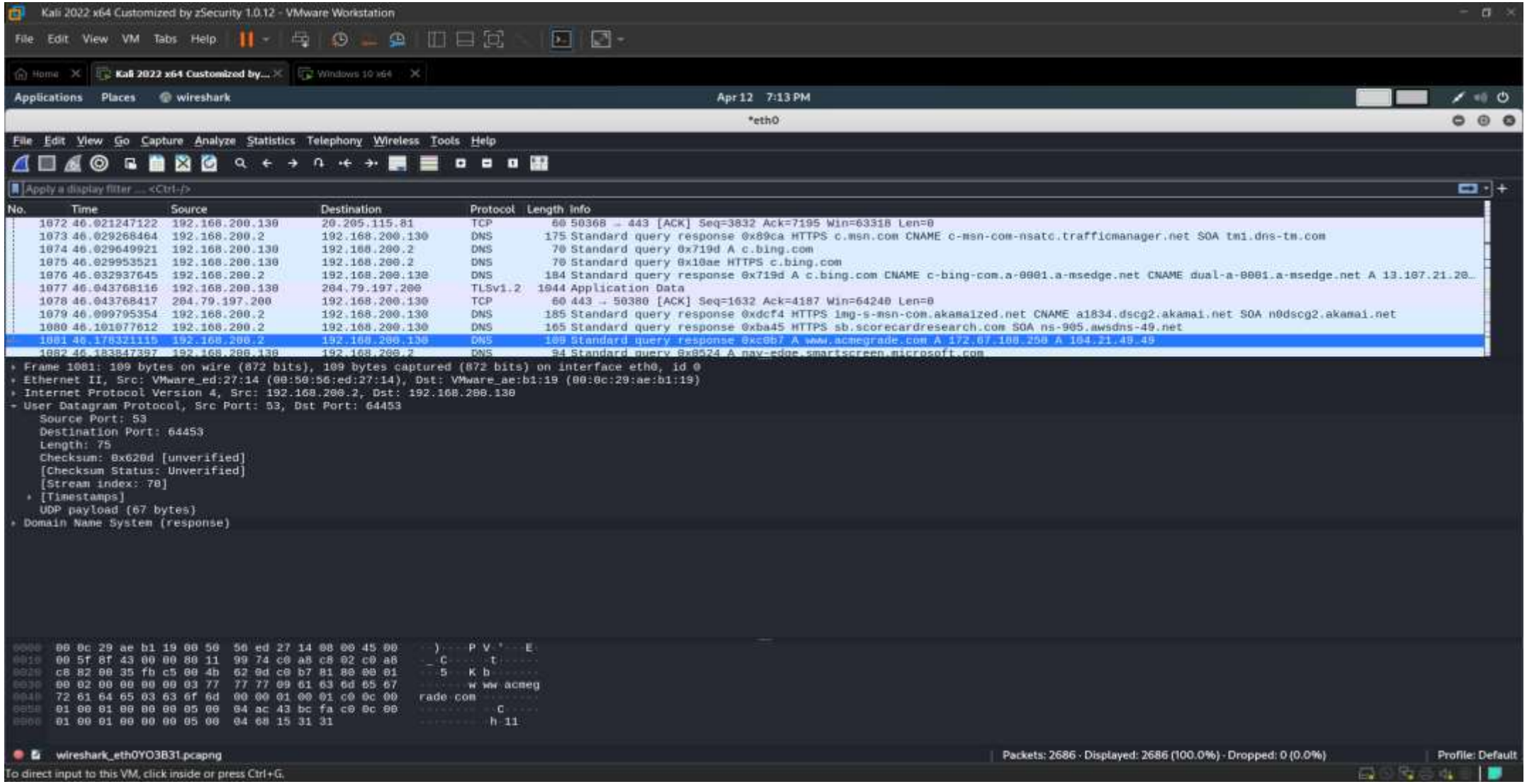
About Destination source and type



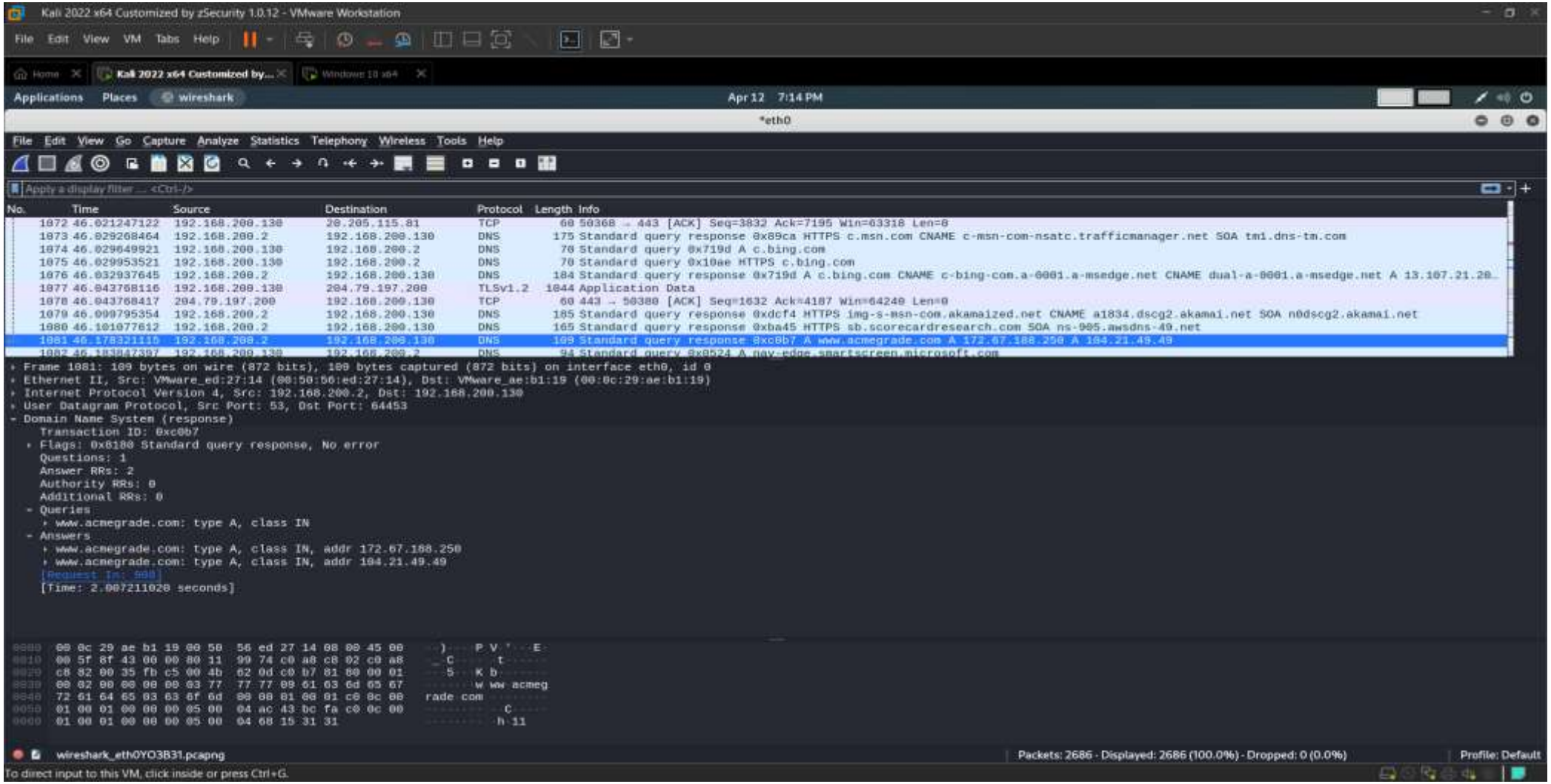
About Internet Protocol



About User Datagram Protocol



About Domain Name System



Caplets function and the commands