Information Gathering (Using GUI Base Tool with Url Target)

Wireshark is a powerful network analysis tool that allows you to capture, filter, and inspect network packets. It can be used for information gathering and troubleshooting network communication issues. Here's a step-by-step guide on how to use Wireshark for information gathering:

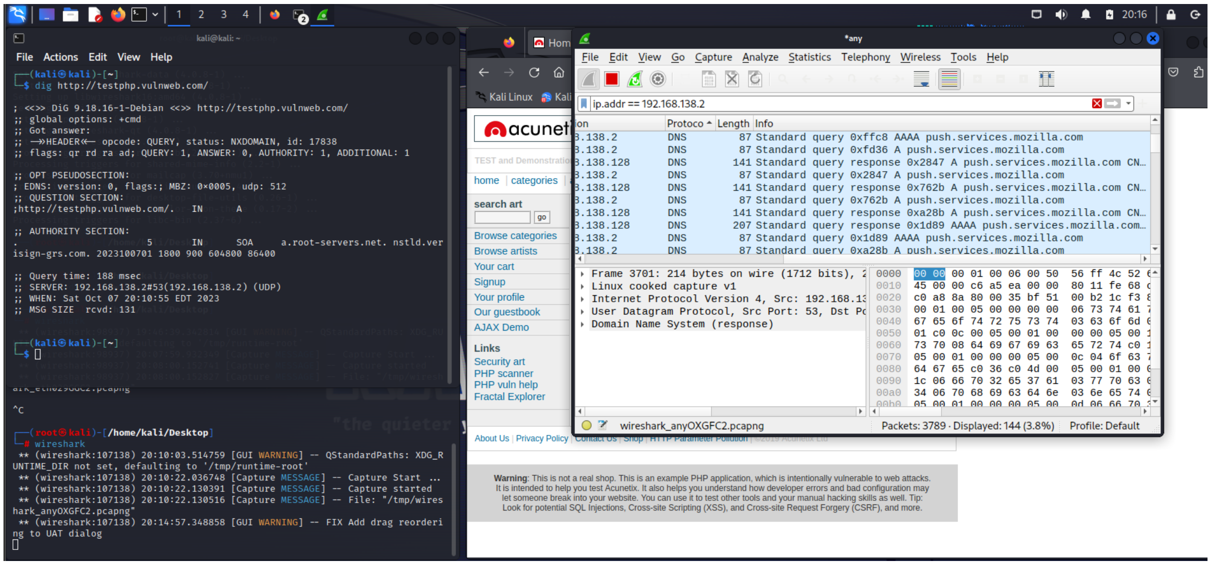
Capturing Packets:

Download and install Wireshark from the official website.

Launch Wireshark and select the network interface you want to capture packets from. For example, if you want to capture traffic on your wireless network, select your wireless interface.

By default, Wireshark captures all packets on the network. If you want to capture only specific packets, you can apply filters. Click on "Capture > Options" to configure advanced features and apply filters if necessary.

Once you've configured the capture options, click the interface's name, and Wireshark will start capturing packets in real-time.



kali@kali: ~
File Actions Edit View Help
*any
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
20:16

```
┌──(kali㉿kali)-[~]
└─$ dig http://testphp.vulnweb.com/

; <<>> DiG 9.18.16-1-Debian <<>> http://testphp.vulnweb.com/
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 17838
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; MBZ: 0×0005, udp: 512
;; QUESTION SECTION:
;http://testphp.vulnweb.com/.   IN      A

;; AUTHORITY SECTION:
.                       5       IN      SOA     a.root-servers.net. nstld.ver
isign-grs.com. 2023100701 1800 900 604800 86400

;; Query time: 188 msec
;; SERVER: 192.168.138.2#53(192.168.138.2) (UDP)
;; WHEN: Sat Oct 07 20:10:55 EDT 2023
;; MSG SIZE  rcvd: 131


┌──(kali㉿kali)-[~]
└─$ 
```

```
^C

┌──(root㉿kali)-[/home/kali/Desktop]
└─# wireshark
 ** (wireshark:107138) 20:10:03.514759 [GUI WARNING] -- QStandardPaths: XDG_R
UNTIME_DIR not set, defaulting to '/tmp/runtime-root'
 ** (wireshark:107138) 20:10:22.036748 [Capture MESSAGE] -- Capture Start ...
 ** (wireshark:107138) 20:10:22.130391 [Capture MESSAGE] -- Capture started
 ** (wireshark:107138) 20:10:22.130516 [Capture MESSAGE] -- File: "/tmp/wires
hark_anyOXGFC2.pcapng"
 ** (wireshark:107138) 20:14:57.348858 [GUI WARNING] -- FIX Add drag reorderi
ng to UAT dialog
```

ip.addr == 192.168.138.2

```
ion           Protoco  Length Info
3.138.2       DNS          87 Standard query 0xffc8 AAAA push.services.mozilla.com
3.138.2       DNS          87 Standard query 0xfd36 A push.services.mozilla.com
3.138.128     DNS         141 Standard query response 0x2847 A push.services.mozilla.com CN…
3.138.2       DNS          87 Standard query 0x2847 A push.services.mozilla.com
3.138.128     DNS         141 Standard query response 0x762b A push.services.mozilla.com CN…
3.138.2       DNS          87 Standard query 0x762b A push.services.mozilla.com
3.138.128     DNS         141 Standard query response 0xa28b A push.services.mozilla.com CN…
3.138.128     DNS         207 Standard query response 0x1d89 AAAA push.services.mozilla.com…
3.138.2       DNS          87 Standard query 0x1d89 AAAA push.services.mozilla.com
3.138.2       DNS          87 Standard query 0xa28b A push.services.mozilla.com
```

```
Frame 3701: 214 bytes on wire (1712 bits), 2
Linux cooked capture v1
Internet Protocol Version 4, Src: 192.168.13
User Datagram Protocol, Src Port: 53, Dst Po
Domain Name System (response)
```

```
0000  00 00 00 01 00 06 00 50  56 ff 4c 52 6
0010  45 00 00 c6 a5 ea 00 00  80 11 fe 68 c
0020  c0 a8 8a 80 00 35 bf 51  00 b2 1c f3 8
0030  00 01 00 05 00 00 00 00  06 73 74 61 7
0040  67 65 6f 74 72 75 73 74  03 63 6f 6d 0
0050  01 c0 0c 00 05 00 01 00  00 00 05 00 1
0060  73 70 08 64 69 67 69 63  65 72 74 c0 1
0070  05 00 01 00 00 00 05 00  0c 04 6f 63 7
0080  64 67 65 c0 36 c0 4d 00  05 00 01 00 0
0090  1c 06 66 70 32 65 37 61  03 77 70 63 0
00a0  34 06 70 68 69 63 64 6e  03 6e 65 74 0
00b0  05 00 01 00 00 00 05 00  0d 06 66 70 3
```

wireshark_anyOXGFC2.pcapng   Packets: 3789 · Displayed: 144 (3.8%)   Profile: Default

Warning: This is not a real shop. This is an example PHP application, which is intentionally vulnerable to web attacks. It is intended to help you test Acunetix. It also helps you understand how developer errors and bad configuration may let someone break into your website. You can use it to test other tools and your manual hacking skills as well. Tip: Look for potential SQL Injections, Cross-site Scripting (XSS), and Cross-site Request Forgery (CSRF), and more.

To stop capturing packets, click the red "Stop" button near the top left corner of the window.

Analyzing Packets:

After capturing packets, Wireshark displays them in the Packet List pane.

The Packet List pane shows a list of captured packets with columns like No., Time, Source, Protocol, Length, and Info.

Clicking on a packet in the Packet List pane displays detailed information about that packet in the Packet Details pane.

The Packet Bytes pane displays the packet in hexadecimal format.

You can apply filters to analyze specific types of packets or conversations. Right-click on a packet and select "Follow" to see only the packets that are part of that conversation.

Use the Wireshark menu options and toolbar buttons to navigate, filter, and analyze the captured packets.

Additional Wireshark Commands:

Wireshark can also be used from the command line with various options:

wireshark -h: Displays available command-line parameters for Wireshark.

wireshark -a duration:300 -i eth1 -w wireshark.pcap: Captures traffic on the Ethernet interface "eth1" for five minutes and saves it to a file named "wireshark.pcap". The -a option automatically stops the capture after the specified duration, and the -i option specifies the interface to capture from.

Remember, when using Wireshark for information gathering, ensure that you have the necessary permissions and legal rights to capture and analyze network traffic. Wireshark is a powerful tool that requires a good understanding of networking protocols and packet analysis techniques to effectively gather information from network traffic.