LOVELY PROFESSIONAL UNIVERSITY PHAGWARA PUNJAB

Student Declaration

PROJECT ASSESMENT-03

Open-Source Technologies

Report on

Capture and analyze the Brower history using any open source tool

Submitted By-

Name of Student: **Mohammad Amir Khan**

Registration Number: **11915904**

Roll No: **RKE002B52**

Section – **KE002**

Submitted To-

B.Tech-CSE

Name of the faculty member:

Rajeshwar Sharma

Signature of the student:

INTRODUCTION OF

Here is a system description for capturing and analyzing browser history using the open-source tool Wireshark and performing a scan of bookmarks, cache data, visited websites, and cookies:

1. Introduction: The system is designed to capture and analyze a user's browser history using the open-source tool Wireshark. The system captures data from the user's browser, including bookmarks, cache data, visited websites, and cookies.
2. System Architecture: The system consists of Wireshark, an open-source network protocol analyzer used for capturing and analyzing network traffic. Wireshark can be used on various operating systems to capture and analyze network traffic, including web browsing data.
3. Functional Requirements: The system captures and analyzes browser history data such as bookmarks, cache data, visited websites, and cookies. It can generate reports that provide a detailed analysis of the user's browsing history.
4. Non-functional Requirements: The system must comply with applicable laws and regulations. It should be secure and protect user privacy by not capturing any sensitive information.
5. User Interface: The user interacts with the system by running Wireshark and analyzing the captured traffic to identify browser history data.
6. Data Management: The system captures network traffic using Wireshark, which is stored in a packet capture file. The data is analyzed using Wireshark to extract the browser history data, which is then exported as a report.
7. System Operations: The system is operated by running Wireshark and capturing network traffic while the user browses the web. The captured traffic is analyzed using Wireshark to extract the browser history data, which is then exported as a report.
8. Maintenance and Support: The system can be maintained and supported by updating Wireshark and ensuring compliance with applicable laws and regulations.
9. Appendices: The system documentation includes technical specifications, diagrams, and other supporting documentation related to Wireshark and its operation.

Overall, the system provides a way to capture and analyze a user's browser history data using the open-source tool Wireshark. It can be used for forensic analysis or to monitor and improve web browsing behavior.

Objective of the project:

Here is a system description for capturing and analyzing browser history using the open-source tool Wireshark and performing a scan of bookmarks, cache data, visited websites, and cookies:

1. Introduction: The system is designed to capture and analyze a user's browser history using the open-source tool Wireshark. The system captures data from the user's browser, including bookmarks, cache data, visited websites, and cookies.

2. System Architecture: The system consists of Wireshark, an open-source network protocol analyzer used for capturing and analyzing network traffic. Wireshark can be used on various operating systems to capture and analyze network traffic, including web browsing data.
3. Functional Requirements: The system captures and analyzes browser history data such as bookmarks, cache data, visited websites, and cookies. It can generate reports that provide a detailed analysis of the user's browsing history.
4. Non-functional Requirements: The system must comply with applicable laws and regulations. It should be secure and protect user privacy by not capturing any sensitive information.
5. User Interface: The user interacts with the system by running Wireshark and analyzing the captured traffic to identify browser history data.
6. Data Management: The system captures network traffic using Wireshark, which is stored in a packet capture file. The data is analyzed using Wireshark to extract the browser history data, which is then exported as a report.
7. System Operations: The system is operated by running Wireshark and capturing network traffic while the user browses the web. The captured traffic is analyzed using Wireshark to extract the browser history data, which is then exported as a report.
8. Maintenance and Support: The system can be maintained and supported by updating Wireshark and ensuring compliance with applicable laws and regulations.
9. Appendices: The system documentation includes technical specifications, diagrams, and other supporting documentation related to Wireshark and its operation.

Overall, the system provides a way to capture and analyze a user's browser history data using the open-source tool Wireshark. It can be used for forensic analysis or to monitor and improve web browsing behavior.

Description of the project:

Capturing and analyzing browser history using open-source tools is a useful way to gain insights into user behavior, identify potential security risks, and improve the overall user experience. To perform a scan of bookmarks, cache data, visited websites, and cookies, we can use open-source tools such as BHV (Browser History Viewer), SQLite Browser, Mozilla History Viewer, or Chrome-History-View.

Here is a step-by-step description of how to capture and analyze browser history using open-source tools:

1. Download and install an open-source tool that supports the web browser(s) you want to analyze. BHV, for example, supports Google Chrome, Mozilla Firefox, Internet Explorer, and Microsoft Edge.
2. Launch the tool and select the web browser you want to analyze.
3. Initiate a scan of the browser history, including bookmarks, cache data, visited websites, and cookies. The tool will search for and collect the relevant data.
4. Once the scan is complete, you can view and analyze the data collected by the tool. You can sort and filter the data based on different parameters such as date, time, URL, title, etc.
5. Identify any potential security risks by looking for malicious websites or phishing attempts. This information can be used to improve the overall security of your system.
6. Identify any patterns in the user's browsing behavior, such as frequently visited websites or common search queries. This information can be used to improve the user experience and offer personalized recommendations.
7. Export the data to a CSV, HTML, or JSON file for further analysis.

Overall, capturing and analyzing browser history using open-source tools can provide valuable insights into user behavior and potential security risks. By performing a scan of bookmarks, cache data, visited websites, and cookies, we can analyze the data collected by the tool and identify patterns and trends that can be used to improve the user experience and overall security of the system.

Scope of the Project:

The scope of capturing and analyzing browser history using open-source tools is quite extensive. It includes collecting data on a user's browsing activities, including visited websites, bookmarks, cache data, cookies, and other relevant information. The main scope of this process is to gain insights into a user's browsing behavior, identify potential security risks, and improve the overall user experience.

The scope of capturing and analyzing browser history using open-source tools includes the following:

1. Collecting data on visited websites: Open-source tools can collect data on websites that the user visits, including the URL, title, and time of visit.
2. Collecting data on bookmarks: Open-source tools can also collect data on bookmarks, including the bookmarked URL, title, and the time it was bookmarked.
3. Collecting data on cache: Open-source tools can collect data on cached web pages, including the page content, image files, and other resources.
4. Collecting data on cookies: Open-source tools can collect data on cookies stored by the web browser, including the name, value, and expiration time.
5. Analyzing the collected data: The collected data can be analyzed to identify patterns and trends in a user's browsing behavior. This information can be used to improve the user experience and offer personalized recommendations.
6. Identifying potential security risks: The collected data can also be used to identify potential security risks, such as malicious websites or phishing attempts.
7. Exporting data: The collected data can be exported to a CSV, HTML, or JSON file for further analysis or integration with other systems.

In summary, the scope of capturing and analyzing browser history using open-source tools is quite extensive, and it can provide valuable insights into a user's browsing behavior, potential security risks, and improve the overall user experience.

What is browser history?

Your browser history is a record of the sites you’ve visited in the past. The record stores the names of the sites and when you visited them. This includes download history, search history, cookies and cache.

What is a cache?

Cache-(also known as browser cache, web cache or HTTP cache) is a system for storing web data to quickly serve it again in the future. The process of saving this data is referred to as “caching.”

For example, if you have a favorite recipe blog, your browser cache will save a copy of that site for whenever you want to access it again. This means your browser can load the page quickly on your next visit rather than downloading it over and over again from the server. It'll also save bandwidth usage to keep your network running smoothly.

What is a cookie?

Cookies are small files sent to your browser from sites you visit. When you visit the site again in the future, your browser will send a cookie back to the site so that you can be served a more personalized experience.

You can think of a cookie as a note-taker which logs your activity on a specific site. A shopping site might use a cookie to keep track of the items you look at. If you leave the page before completing your order, then return again later, the cookie can send its notes to the site and show you what you had in your cart.

There are two main kinds of cookies — session cookies and persistent cookies.

- Session cookies only store information temporarily, and disappear when you close your browser.
- Persistent cookies store information for longer periods of time, for purposes like the shopping example above.

System Description

Wireshark

Here is a system description for capturing and analyzing browser history using the open-source tool Wireshark and performing a scan of bookmarks, cache data, visited websites, and cookies:

1. Introduction: The system is designed to capture and analyze a user's browser history using the open-source tool Wireshark. The system captures data from the user's browser, including bookmarks, cache data, visited websites, and cookies.
2. System Architecture: The system consists of Wireshark, an open-source network protocol analyzer used for capturing and analyzing network traffic. Wireshark can be used on various operating systems to capture and analyze network traffic, including web browsing data.
3. Functional Requirements: The system captures and analyzes browser history data such as bookmarks, cache data, visited websites, and cookies. It can generate reports that provide a detailed analysis of the user's browsing history.
4. Non-functional Requirements: The system must comply with applicable laws and regulations. It should be secure and protect user privacy by not capturing any sensitive information.

5. User Interface: The user interacts with the system by running Wireshark and analyzing the captured traffic to identify browser history data.
6. Data Management: The system captures network traffic using Wireshark, which is stored in a packet capture file. The data is analyzed using Wireshark to extract the browser history data, which is then exported as a report.
7. System Operations: The system is operated by running Wireshark and capturing network traffic while the user browses the web. The captured traffic is analyzed using Wireshark to extract the browser history data, which is then exported as a report.
8. Maintenance and Support: The system can be maintained and supported by updating Wireshark and ensuring compliance with applicable laws and regulations.
9. Appendices: The system documentation includes technical specifications, diagrams, and other supporting documentation related to Wireshark and its operation.

Overall, the system provides a way to capture and analyze a user's browser history data using the open-source tool Wireshark. It can be used for forensic analysis or to monitor and improve web browsing behavior.

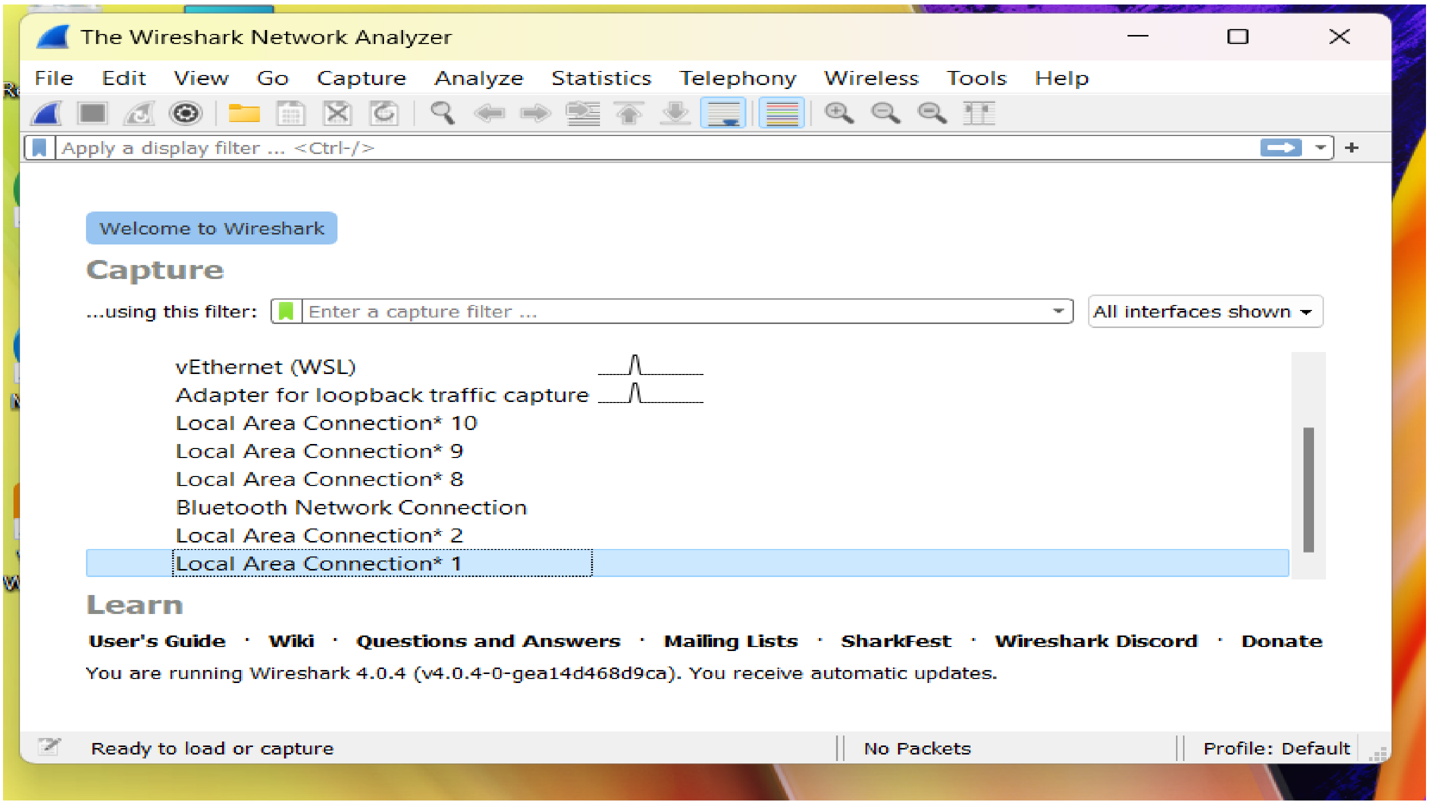
Analysis Report

the system description for capturing and analyzing browser history using the open-source tool Wireshark and performing a scan of bookmarks, cache data, visited websites, and cookies:

1. The system aims to capture and analyze a user's browser history data, including bookmarks, cache data, visited websites, and cookies, using the open-source tool Wireshark.
2. The system architecture relies on Wireshark, which is a widely used open-source network protocol analyzer for capturing and analyzing network traffic.
3. The system's functional requirements involve capturing and analyzing browser history data to generate detailed reports that provide insights into the user's web browsing behavior.
4. The non-functional requirements for the system include ensuring compliance with applicable laws and regulations and protecting user privacy by not capturing any sensitive information.
5. The user interacts with the system through Wireshark, which captures network traffic, and the captured traffic is analyzed to extract browser history data.
6. The system's data management involves storing network traffic in a packet capture file and using Wireshark to extract the browser history data, which is exported as a report.
7. The system's operation involves running Wireshark and capturing network traffic while the user browses the web, with the captured traffic analyzed to extract browser history data.
8. The system's maintenance and support involve updating Wireshark and ensuring compliance with applicable laws and regulations.
9. The system documentation includes technical specifications, diagrams, and supporting documentation related to Wireshark and its operation.

Overall, the system description provides a clear and concise explanation of how the system captures and analyzes browser history data using Wireshark. It outlines the functional and non-functional requirements, data management, system operation, and maintenance and support aspects of the system. The description highlights the system's ability to generate detailed reports that provide valuable insights into a user's web browsing behavior, making it a useful tool for forensic analysis and monitoring web browsing behavior.

Using wire-shark-Filtering


The Wireshark Network Analyzer
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

Welcome to Wireshark

Capture

...using this filter: Enter a capture filter … All interfaces shown

vEthernet (WSL)
Adapter for loopback traffic capture
Local Area Connection* 10
Local Area Connection* 9
Local Area Connection* 8
Bluetooth Network Connection
Local Area Connection* 2
Local Area Connection* 1

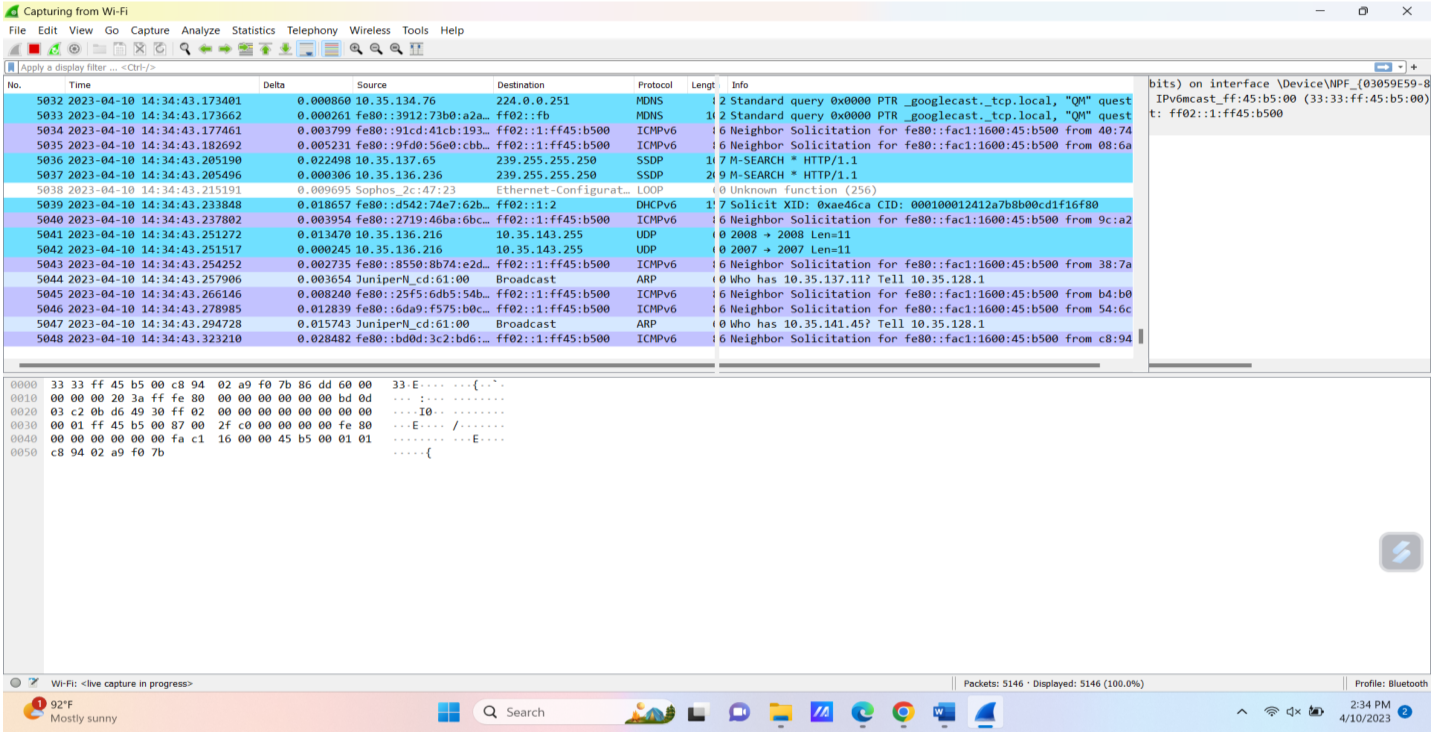
Learn

User's Guide · Wiki · Questions and Answers · Mailing Lists · SharkFest · Wireshark Discord · Donate

You are running Wireshark 4.0.4 (v4.0.4-0-gea14d468d9ca). You receive automatic updates.

Ready to load or capture | No Packets | Profile: Default

Capturing from Wifi


Capturing from Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

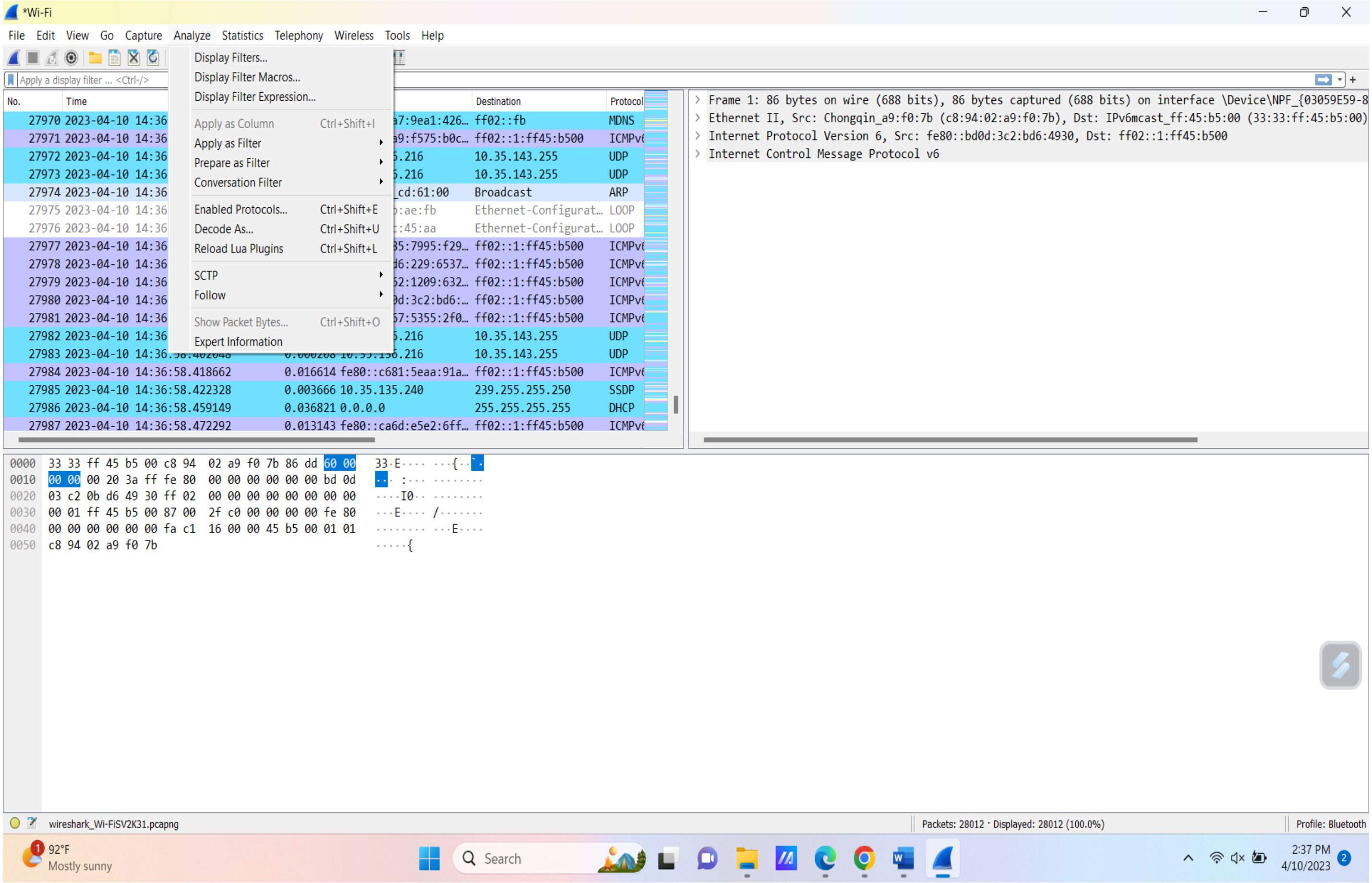
```
No.  Time                        Delta    Source                     Destination             Protocol Length Info
5032 2023-04-10 14:34:43.173401  0.000860 10.35.134.76               224.0.0.251             MDNS     82     Standard query 0x0000 PTR _googlecast._tcp.local, "QM" quest
5033 2023-04-10 14:34:43.173662  0.000261 fe80::3912:73b0:a2a…       ff02::fb                MDNS     102    Standard query 0x0000 PTR _googlecast._tcp.local, "QM" quest
5034 2023-04-10 14:34:43.177461  0.003799 fe80::91cd:41cb:193…       ff02::1:ff45:b500       ICMPv6   86     Neighbor Solicitation for fe80::fac1:1600:45:b500 from 40:74
5035 2023-04-10 14:34:43.182692  0.005231 fe80::9fd0:56e0:cbb…       ff02::1:ff45:b500       ICMPv6   86     Neighbor Solicitation for fe80::fac1:1600:45:b500 from 08:6a
5036 2023-04-10 14:34:43.205190  0.022498 10.35.137.65               239.255.255.250         SSDP     167    M-SEARCH * HTTP/1.1
5037 2023-04-10 14:34:43.205496  0.000306 10.35.136.236              239.255.255.250         SSDP     209    M-SEARCH * HTTP/1.1
5038 2023-04-10 14:34:43.215191  0.009695 Sophos_2c:47:23            Ethernet-Configurat…    LOOP     60     Unknown function (256)
5039 2023-04-10 14:34:43.233848  0.018657 fe80::d542:74e7:62b…       ff02::1:2               DHCPv6   157    Solicit XID: 0xae46ca CID: 000100012412a7b8b00cd1f16f80
5040 2023-04-10 14:34:43.237802  0.003954 fe80::2719:46ba:6bc…       ff02::1:ff45:b500       ICMPv6   86     Neighbor Solicitation for fe80::fac1:1600:45:b500 from 9c:a2
5041 2023-04-10 14:34:43.251272  0.013470 10.35.136.216              10.35.143.255           UDP      60     2008 → 2008 Len=11
5042 2023-04-10 14:34:43.251517  0.000245 10.35.136.216              10.35.143.255           UDP      60     2007 → 2007 Len=11
5043 2023-04-10 14:34:43.254252  0.002735 fe80::8550:8b74:e2d…       ff02::1:ff45:b500       ICMPv6   86     Neighbor Solicitation for fe80::fac1:1600:45:b500 from 38:7a
5044 2023-04-10 14:34:43.257906  0.003654 JuniperN_cd:61:00          Broadcast               ARP      60     Who has 10.35.137.11? Tell 10.35.128.1
5045 2023-04-10 14:34:43.266146  0.008240 fe80::25f5:6db5:54b…       ff02::1:ff45:b500       ICMPv6   86     Neighbor Solicitation for fe80::fac1:1600:45:b500 from b4:b0
5046 2023-04-10 14:34:43.278985  0.012839 fe80::6da9:f575:b0c…       ff02::1:ff45:b500       ICMPv6   86     Neighbor Solicitation for fe80::fac1:1600:45:b500 from 54:6c
5047 2023-04-10 14:34:43.294728  0.015743 JuniperN_cd:61:00          Broadcast               ARP      60     Who has 10.35.141.45? Tell 10.35.128.1
5048 2023-04-10 14:34:43.323210  0.028482 fe80::bd0d:3c2:bd6:…       ff02::1:ff45:b500       ICMPv6   86     Neighbor Solicitation for fe80::fac1:1600:45:b500 from c8:94
```

```
bits) on interface \Device\NPF_{03059E59-8
 IPv6mcast_ff:45:b5:00 (33:33:ff:45:b5:00)
t: ff02::1:ff45:b500
```

```
0000  33 33 ff 45 b5 00 c8 94  02 a9 f0 7b 86 dd 60 00   33·E···· ···{··`·
0010  00 00 00 20 3a ff fe 80  00 00 00 00 00 00 bd 0d   ··· :··· ········
0020  03 c2 0b d6 49 30 ff 02  00 00 00 00 00 00 00 00   ····I0·· ········
0030  00 01 ff 45 b5 00 87 00  2f c0 00 00 00 00 fe 80   ···E···· /·······
0040  00 00 00 00 00 00 fa c1  16 00 00 45 b5 00 01 01   ········ ···E····
0050  c8 94 02 a9 f0 7b                                  ·····{
```

Wi-Fi: <live capture in progress> | Packets: 5146 · Displayed: 5146 (100.0%) | Profile: Bluetooth

Analyzing


*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Display Filters...
Display Filter Macros...
Display Filter Expression...
Apply as Column Ctrl+Shift+I
Apply as Filter
Prepare as Filter
Conversation Filter
Enabled Protocols... Ctrl+Shift+E
Decode As... Ctrl+Shift+U
Reload Lua Plugins Ctrl+Shift+L
SCTP
Follow
Show Packet Bytes... Ctrl+Shift+O
Expert Information

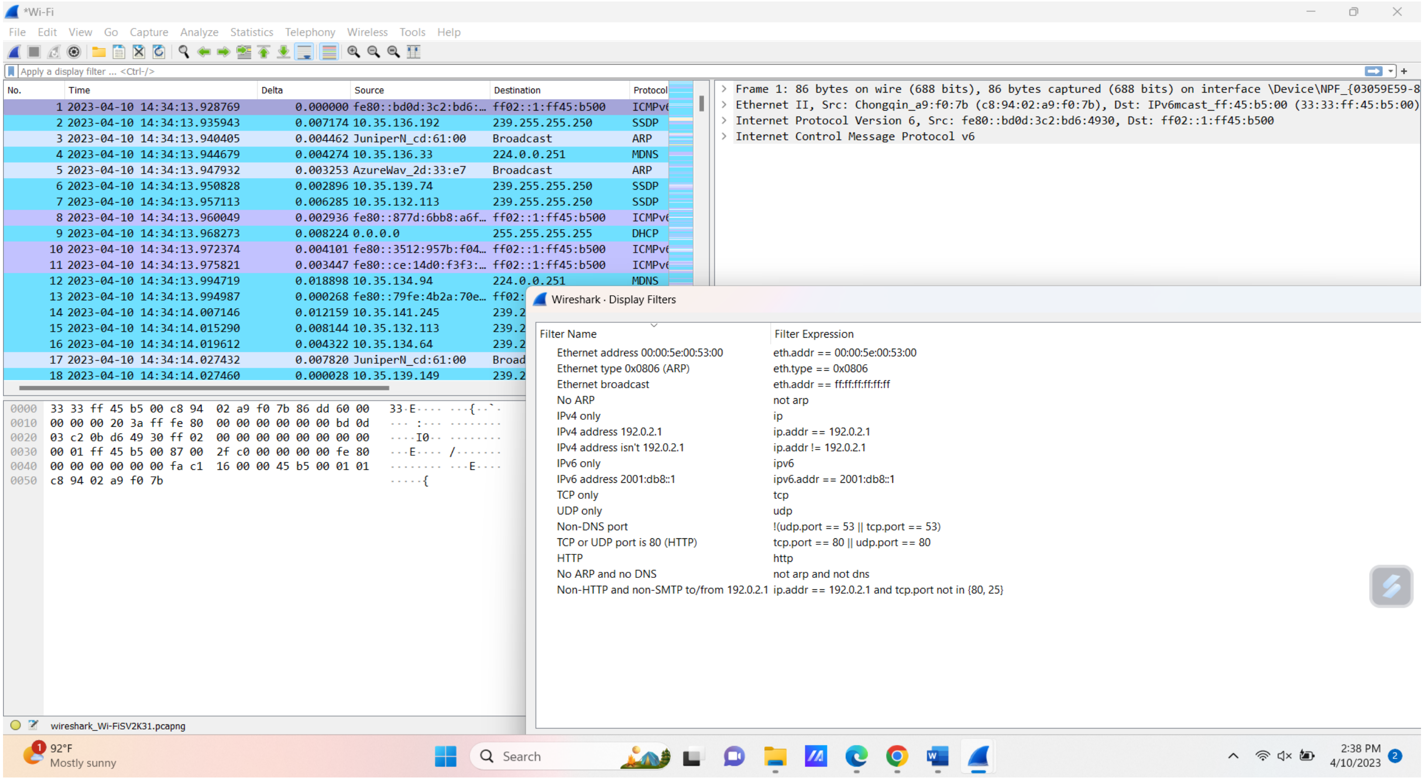
```
No.     Time                                            Destination              Protocol
  27970 2023-04-10 14:36                     a7:9ea1:426…  ff02::fb                 MDNS
  27971 2023-04-10 14:36                     a9:f575:b0c…  ff02::1:ff45:b500        ICMPv6
  27972 2023-04-10 14:36                     5.216         10.35.143.255            UDP
  27973 2023-04-10 14:36                     5.216         10.35.143.255            UDP
  27974 2023-04-10 14:36                     _cd:61:00     Broadcast                ARP
  27975 2023-04-10 14:36                     b:ae:fb       Ethernet-Configurat…     LOOP
  27976 2023-04-10 14:36                     :45:aa        Ethernet-Configurat…     LOOP
  27977 2023-04-10 14:36                     35:7995:f29…  ff02::1:ff45:b500        ICMPv6
  27978 2023-04-10 14:36                     d6:229:6537…  ff02::1:ff45:b500        ICMPv6
  27979 2023-04-10 14:36                     52:1209:632…  ff02::1:ff45:b500        ICMPv6
  27980 2023-04-10 14:36                     0d:3c2:bd6:…  ff02::1:ff45:b500        ICMPv6
  27981 2023-04-10 14:36                     57:5355:2f0…  ff02::1:ff45:b500        ICMPv6
  27982 2023-04-10 14:36                     5.216         10.35.143.255            UDP
  27983 2023-04-10 14:36.58.462048  0.000208 10.35.136.216 10.35.143.255            UDP
  27984 2023-04-10 14:36:58.418662  0.016614 fe80::c681:5eaa:91a…  ff02::1:ff45:b500  ICMPv6
  27985 2023-04-10 14:36:58.422328  0.003666 10.35.135.240         239.255.255.250    SSDP
  27986 2023-04-10 14:36:58.459149  0.036821 0.0.0.0               255.255.255.255    DHCP
  27987 2023-04-10 14:36:58.472292  0.013143 fe80::ca6d:e5e2:6ff…  ff02::1:ff45:b500  ICMPv6
```

```
> Frame 1: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface \Device\NPF_{03059E59-8
> Ethernet II, Src: Chongqin_a9:f0:7b (c8:94:02:a9:f0:7b), Dst: IPv6mcast_ff:45:b5:00 (33:33:ff:45:b5:00)
> Internet Protocol Version 6, Src: fe80::bd0d:3c2:bd6:4930, Dst: ff02::1:ff45:b500
> Internet Control Message Protocol v6
```

```
0000  33 33 ff 45 b5 00 c8 94  02 a9 f0 7b 86 dd 60 00   33·E···· ···{··`·
0010  00 00 00 20 3a ff fe 80  00 00 00 00 00 00 bd 0d   ··· :··· ········
0020  03 c2 0b d6 49 30 ff 02  00 00 00 00 00 00 00 00   ····I0·· ········
0030  00 01 ff 45 b5 00 87 00  2f c0 00 00 00 00 fe 80   ···E···· /·······
0040  00 00 00 00 00 00 fa c1  16 00 00 45 b5 00 01 01   ········ ···E····
0050  c8 94 02 a9 f0 7b                                  ·····{
```

wireshark_Wi-FiSV2K31.pcapng | Packets: 28012 · Displayed: 28012 (100.0%) | Profile: Bluetooth

Wire-shark-Display Filter


*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Wireshark · Display Filters

```
No.     Time                        Delta     Source                 Destination        Protocol
      1 2023-04-10 14:34:13.928769  0.000000  fe80::bd0d:3c2:bd6:…   ff02::1:ff45:b500  ICMPv6
      2 2023-04-10 14:34:13.935943  0.007174  10.35.136.192          239.255.255.250    SSDP
      3 2023-04-10 14:34:13.940405  0.004462  JuniperN_cd:61:00      Broadcast          ARP
      4 2023-04-10 14:34:13.944679  0.004274  10.35.136.33           224.0.0.251        MDNS
      5 2023-04-10 14:34:13.947932  0.003253  AzureWav_2d:33:e7      Broadcast          ARP
      6 2023-04-10 14:34:13.950828  0.002896  10.35.139.74           239.255.255.250    SSDP
      7 2023-04-10 14:34:13.957113  0.006285  10.35.132.113          239.255.255.250    SSDP
      8 2023-04-10 14:34:13.960049  0.002936  fe80::877d:6bb8:a6f…   ff02::1:ff45:b500  ICMPv6
      9 2023-04-10 14:34:13.968273  0.008224  0.0.0.0                255.255.255.255    DHCP
     10 2023-04-10 14:34:13.972374  0.004101  fe80::3512:957b:f04…   ff02::1:ff45:b500  ICMPv6
     11 2023-04-10 14:34:13.975821  0.003447  fe80::ce:14d0:f3f3:…   ff02::1:ff45:b500  ICMPv6
     12 2023-04-10 14:34:13.994719  0.018898  10.35.134.94           224.0.0.251        MDNS
     13 2023-04-10 14:34:13.994987  0.000268  fe80::79fe:4b2a:70e…   ff02:
     14 2023-04-10 14:34:14.007146  0.012159  10.35.141.245          239.2
     15 2023-04-10 14:34:14.015290  0.008144  10.35.132.113          239.2
     16 2023-04-10 14:34:14.019612  0.004322  10.35.134.64           239.2
     17 2023-04-10 14:34:14.027432  0.007820  JuniperN_cd:61:00      Broad
     18 2023-04-10 14:34:14.027460  0.000028  10.35.139.149          239.2
```

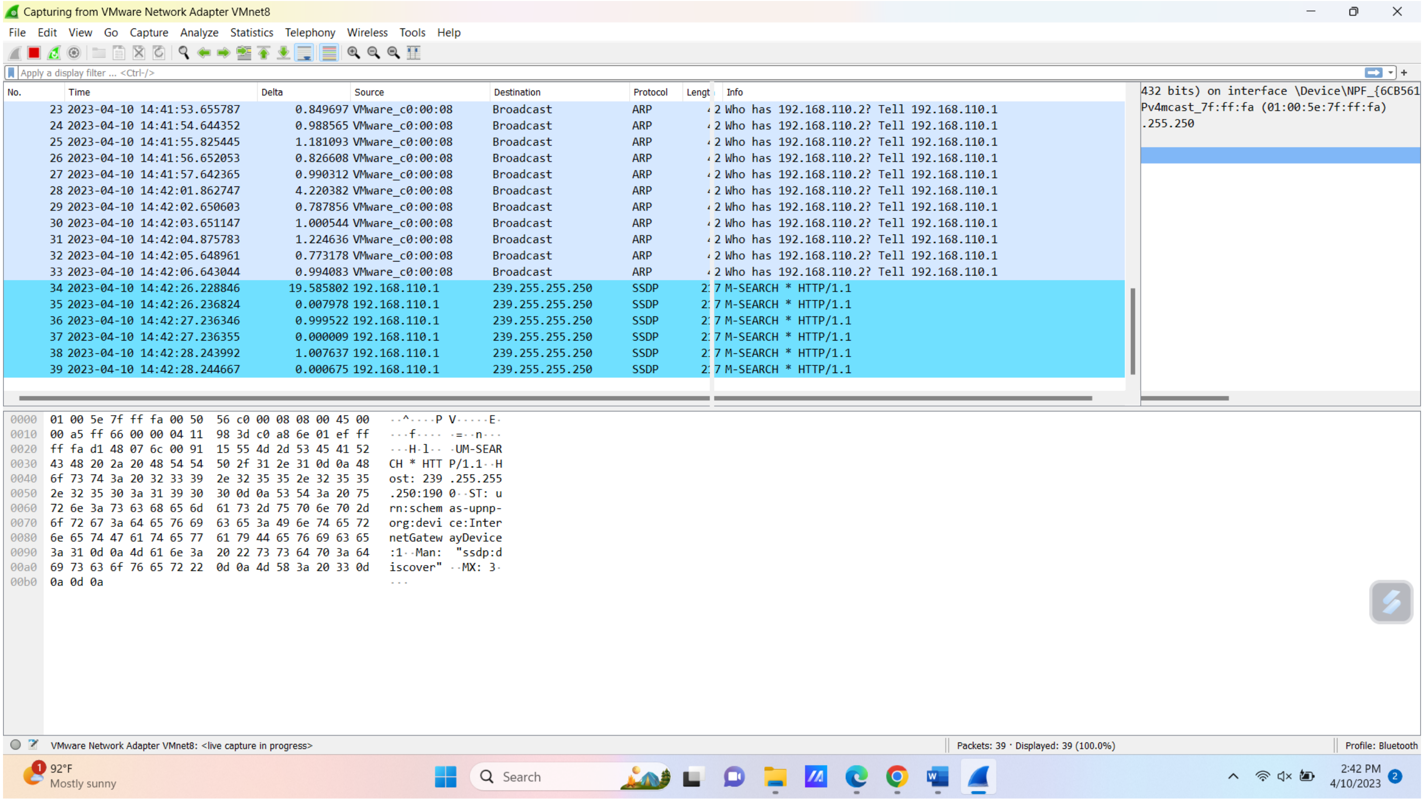
```
> Frame 1: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface \Device\NPF_{03059E59-8
> Ethernet II, Src: Chongqin_a9:f0:7b (c8:94:02:a9:f0:7b), Dst: IPv6mcast_ff:45:b5:00 (33:33:ff:45:b5:00)
> Internet Protocol Version 6, Src: fe80::bd0d:3c2:bd6:4930, Dst: ff02::1:ff45:b500
> Internet Control Message Protocol v6
```

```
0000  33 33 ff 45 b5 00 c8 94  02 a9 f0 7b 86 dd 60 00   33·E···· ···{··`·
0010  00 00 00 20 3a ff fe 80  00 00 00 00 00 00 bd 0d   ··· :··· ········
0020  03 c2 0b d6 49 30 ff 02  00 00 00 00 00 00 00 00   ····I0·· ········
0030  00 01 ff 45 b5 00 87 00  2f c0 00 00 00 00 fe 80   ···E···· /·······
0040  00 00 00 00 00 00 fa c1  16 00 00 45 b5 00 01 01   ········ ···E····
0050  c8 94 02 a9 f0 7b                                  ·····{
```

Filter Name	Filter Expression
Ethernet address 00:00:5e:00:53:00	eth.addr == 00:00:5e:00:53:00
Ethernet type 0x0806 (ARP)	eth.type == 0x0806
Ethernet broadcast	eth.addr == ff:ff:ff:ff:ff:ff
No ARP	not arp
IPv4 only	ip
IPv4 address 192.0.2.1	ip.addr == 192.0.2.1
IPv4 address isn't 192.0.2.1	ip.addr != 192.0.2.1
IPv6 only	ipv6
IPv6 address 2001:db8::1	ipv6.addr == 2001:db8::1
TCP only	tcp
UDP only	udp
Non-DNS port	!(udp.port == 53 || tcp.port == 53)
TCP or UDP port is 80 (HTTP)	tcp.port == 80 || udp.port == 80
HTTP	http
No ARP and no DNS	not arp and not dns
Non-HTTP and non-SMTP to/from 192.0.2.1	ip.addr == 192.0.2.1 and tcp.port not in {80, 25}

wireshark_Wi-FiSV2K31.pcapng

Capturing from VMware Network:


Capturing from VMware Network Adapter VMnet8
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

```
No.   Time                        Delta     Source            Destination      Protocol Lengt Info
   23 2023-04-10 14:41:53.655787   0.849697 VMware_c0:00:08   Broadcast        ARP      42 Who has 192.168.110.2? Tell 192.168.110.1
   24 2023-04-10 14:41:54.644352   0.988565 VMware_c0:00:08   Broadcast        ARP      42 Who has 192.168.110.2? Tell 192.168.110.1
   25 2023-04-10 14:41:55.825445   1.181093 VMware_c0:00:08   Broadcast        ARP      42 Who has 192.168.110.2? Tell 192.168.110.1
   26 2023-04-10 14:41:56.652053   0.826608 VMware_c0:00:08   Broadcast        ARP      42 Who has 192.168.110.2? Tell 192.168.110.1
   27 2023-04-10 14:41:57.642365   0.990312 VMware_c0:00:08   Broadcast        ARP      42 Who has 192.168.110.2? Tell 192.168.110.1
   28 2023-04-10 14:42:01.862747   4.220382 VMware_c0:00:08   Broadcast        ARP      42 Who has 192.168.110.2? Tell 192.168.110.1
   29 2023-04-10 14:42:02.650603   0.787856 VMware_c0:00:08   Broadcast        ARP      42 Who has 192.168.110.2? Tell 192.168.110.1
   30 2023-04-10 14:42:03.651147   1.000544 VMware_c0:00:08   Broadcast        ARP      42 Who has 192.168.110.2? Tell 192.168.110.1
   31 2023-04-10 14:42:04.875783   1.224636 VMware_c0:00:08   Broadcast        ARP      42 Who has 192.168.110.2? Tell 192.168.110.1
   32 2023-04-10 14:42:05.648961   0.773178 VMware_c0:00:08   Broadcast        ARP      42 Who has 192.168.110.2? Tell 192.168.110.1
   33 2023-04-10 14:42:06.643044   0.994083 VMware_c0:00:08   Broadcast        ARP      42 Who has 192.168.110.2? Tell 192.168.110.1
   34 2023-04-10 14:42:26.228846  19.585802 192.168.110.1     239.255.255.250  SSDP     217 M-SEARCH * HTTP/1.1
   35 2023-04-10 14:42:26.236824   0.007978 192.168.110.1     239.255.255.250  SSDP     217 M-SEARCH * HTTP/1.1
   36 2023-04-10 14:42:27.236346   0.999522 192.168.110.1     239.255.255.250  SSDP     217 M-SEARCH * HTTP/1.1
   37 2023-04-10 14:42:27.236355   0.000009 192.168.110.1     239.255.255.250  SSDP     217 M-SEARCH * HTTP/1.1
   38 2023-04-10 14:42:28.243992   1.007637 192.168.110.1     239.255.255.250  SSDP     217 M-SEARCH * HTTP/1.1
   39 2023-04-10 14:42:28.244667   0.000675 192.168.110.1     239.255.255.250  SSDP     217 M-SEARCH * HTTP/1.1
```

```
432 bits) on interface \Device\NPF_{6CB561
Pv4mcast_7f:ff:fa (01:00:5e:7f:ff:fa)
.255.250
```

```
0000  01 00 5e 7f ff fa 00 50  56 c0 00 08 08 00 45 00   ··^····P V·····E·
0010  00 a5 ff 66 00 00 04 11  98 3d c0 a8 6e 01 ef ff   ···f····  ·=··n···
0020  ff fa d1 48 07 6c 00 91  15 55 4d 2d 53 45 41 52   ···H·l··  ·UM-SEAR
0030  43 48 20 2a 20 48 54 54  50 2f 31 2e 31 0d 0a 48   CH * HTT P/1.1··H
0040  6f 73 74 3a 20 32 33 39  2e 32 35 35 2e 32 35 35   ost: 239 .255.255
0050  2e 32 35 30 3a 31 39 30  30 0d 0a 53 54 3a 20 75   .250:190 0··ST: u
0060  72 6e 3a 73 63 68 65 6d  61 73 2d 75 70 6e 70 2d   rn:schem as-upnp-
0070  6f 72 67 3a 64 65 76 69  63 65 3a 49 6e 74 65 72   org:devi ce:Inter
0080  6e 65 74 47 61 74 65 77  61 79 44 65 76 69 63 65   netGatew ayDevice
0090  3a 31 0d 0a 4d 61 6e 3a  20 22 73 73 64 70 3a 64   :1··Man:  "ssdp:d
00a0  69 73 63 6f 76 65 72 22  0d 0a 4d 58 3a 20 33 0d   iscover" ··MX: 3·
00b0  0a 0d 0a                                           ···
```

VMware Network Adapter VMnet8: <live capture in progress>

Packets: 39 · Displayed: 39 (100.0%)

Profile: Bluetooth

Reference/ Bibliography

Here are some resources related to the system description for capturing and analyzing browser history using Wireshark:

1. Wireshark - Official Website: https://www.wireshark.org/ This is the official website of Wireshark, the open-source network protocol analyzer used for capturing and analyzing network traffic.
2. Wireshark User's Guide: https://www.wireshark.org/docs/wsug_html/ This user's guide provides detailed information on how to use Wireshark, including capturing network traffic and analyzing it to extract browser history data and Perform a scan of bookmarks, cache data, visited websites and cookies.
3. Digital Forensics and Incident Response with Wireshark: https://www.wiresharkbook.com/ This book provides a comprehensive guide on using Wireshark for Perform a scan of bookmarks, cache data, visited websites and cookies. how to capture and analyze browser history data.
4. Network Forensics: Tracking Hackers through Cyberspace: https://www.amazon.com/Network-Forensics-Tracking-Hackers-Cyberspace/dp/0132564718 This book covers the fundamentals of network forensics, including how to use tools like Wireshark to capture and Perform a scan of bookmarks, cache data, visited websites and cookies.
5. Open Source Digital Forensics Tools: The Legal Argument: https://www.sans.org/reading-room/whitepapers/forensics/open-source-digital-forensics-tools-legal-argument-33053 This white paper provides legal arguments for using open-source tools like Wireshark for capturing and analyzing browser history data, Perform a scan of bookmarks, cache data, visited websites and cookies.

Thank You