



РУСЛАН СКУРАТОВСЬКИЙ

МЕТОДИЧНИЙ ПОСІБНИК
для виконання практики
у лабораторії “Кіберполігон”
АНАЛІЗ МЕРЕЖЕВОГО ТРАФІКУ
З ВИКОРИСТАННЯМ WIRESHARK



Wireshark

МІЖРЕГІОНАЛЬНА
АКАДЕМІЯ УПРАВЛІННЯ ПЕРСОНАЛОМ



МЕТОДИЧНИЙ ПОСІБНИК
для виконання практики
у лабораторії “Кіберполігон”
АНАЛІЗ МЕРЕЖЕВОГО ТРАФІКУ
З ВИКОРИСТАННЯМ WIRESHARK

Київ
2024

Підготовлено кандидатом фізико-математичних наук, завідуючим лабораторією Інституту безпеки, доцентом кафедри обчислювальної математики і комп'ютерного моделювання *Р. Скуратовським*

Схвалено Вченою радою Міжрегіональної Академії управління персоналом (протокол № 6 від 15.05.24)

Методичний посібник для виконання практики у лабораторії “Кіберполігон” “Аналіз мережевого трафіку з використанням Wireshark” / уклад. Р. Скуратовський. Київ: МАУП, 2024. 48 с.

Методичний посібник містить доступно викладений матеріал як по застосуванню аналізатора мережевого трафіка Wireshark Network Analyzer, так і методи аналізу і знаходження патологічного трафіку. Схарактеризовано переваги Wireshark Network Analyzer. Згідно з вивченою в курсі дискретної математики теорією відносності створено і описано взаємозв'язок фільтрування мережевого трафіку, що реалізоване в Wireshark на рівні протоколів. Представлені всі команди на рівні фільтрів логічного типу і знакові системи, які реалізують такі відношення, як виключення, перетин, об'єднання. В роботі чудово представлено мережевий інтерфейс даної програми. Також описані фільтри для транспортних протоколів (наприклад, TCP, UDP).

Важливість зазначеної програми Wireshark в сучасному кіберсвіті є очевидною, оскільки вона дозволяє визначити можливі загрози для безпеки мережі, виявити паразитивний, вірусний і закільцьований трафіки, дамп трафіку і різного типу мережеві атаки.

Для студентів магістратури спеціальностей кібербезпека і комп'ютерні науки.

© Міжрегіональна Академія
управління персоналом (МАУП), 2024

ВСТУП

Аналіз мережного трафіку є вкрай важливим для забезпечення стабільності, ефективності та безпеки мережі. Основні причини, чому аналіз мережного трафіку є важливим, включають:

1. Виявлення кіберзагроз та злочинної діяльності: аналіз мережевого трафіку дає змогу виявляти кіберзагрози та злочинну діяльність, такі як вторгнення, спам, фішинг та ін. Це допомагає запобігти можливим кібератакам і захистити мережу від стороннього втручання.
2. Підвищення ефективності мережі: аналіз мережевого трафіку уможливорює знаходити проблеми з пропускнуою спроможністю, визначати трафік, який споживає більше ресурсів мережі, та забезпечувати оптимальне використання ресурсів мережі, що підвищує ефективність її роботи.
3. Відновлення роботи мережі: аналіз мережевого трафіку може допомогти відновити роботу мережі після виникнення проблем, таких як відмова обладнання або програмного забезпечення.
4. Поліпшення безпеки мережі: аналіз мережевого трафіку може допомогти виявити вразливості мережі та допомогти виправити їх, щоб уникнути можливих кібератак.
5. Підтримка рішень: аналіз мережевого трафіку може допомогти приймати обґрунтовані рішення щодо конфігурації мережі, вибору обладнання та відповідності вимогам безпеки.

У цілому аналіз мережевого трафіку є важливим інструментом для забезпечення безпеки та ефективності мережі, а також для виявлення та усунення проблем.

Ми пропонуємо найбільш доступну і водночас просунуту програму мережевого аналізатора — Wireshark (сніфера). Безперечним залишається те, що є конкурентоздатним доповнення до існуючих Real-Time систем виявлення атак.

Її конкурент — Network Traffic Analyzer, NTA недоступна у вільному доступі.

Для успішного виконання лабораторної роботи необхідно знати стек протоколів TCP/IP, протокол UDP.

Завдання

Уявіть, що Ви є аналітиком комп'ютерної безпеки, який отримав відомість про підозрілий мережевий трафік на одному з серверів вашої організації. Ваше завдання — дослідити цей трафік з використанням програми Wireshark та визначити можливі загрози для безпеки мережі. Виявити паразитивний, вірусний і закільцьований трафіки, дамп трафіку.

Рекомендації для покрокового виконання:

1. Запустіть **Wireshark** та виберіть мережевий інтерфейс, через який ви хочете аналізувати трафік.
2. Надайте кілька скрінів, де є мережевий інтерфейс Wireshark. Опишіть їх вербально, зокрема їх переваги і недоліки.
3. Прокоментуйте, як Ви вибрали джерело трафіку і підкріпіть це скрінами.

Означення. **API** — applicated program interface.

На рис. 1. зображено API-інтерфейс версії 1.12.3 Wireshark Network Analyzer.

- 3.1. Почекайте деякий час, щоб зібрати достатню кількість пакетів.
- 3.2. Проаналізуйте заголовки пакетів, щоб виявити незвичайний або підозрілий трафік. Зверніть увагу на такі параметри, як джерело та призначення пакетів, тип протоколу тощо.
4. У командний рядок ввести команду `ssh demo@test.rebex.net`. та віддалено підключитися до тестового сервера.

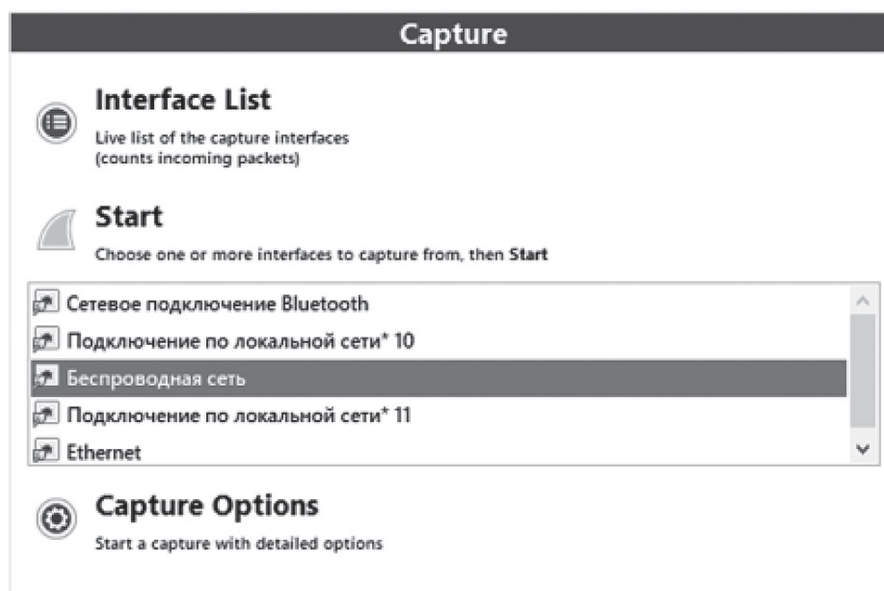
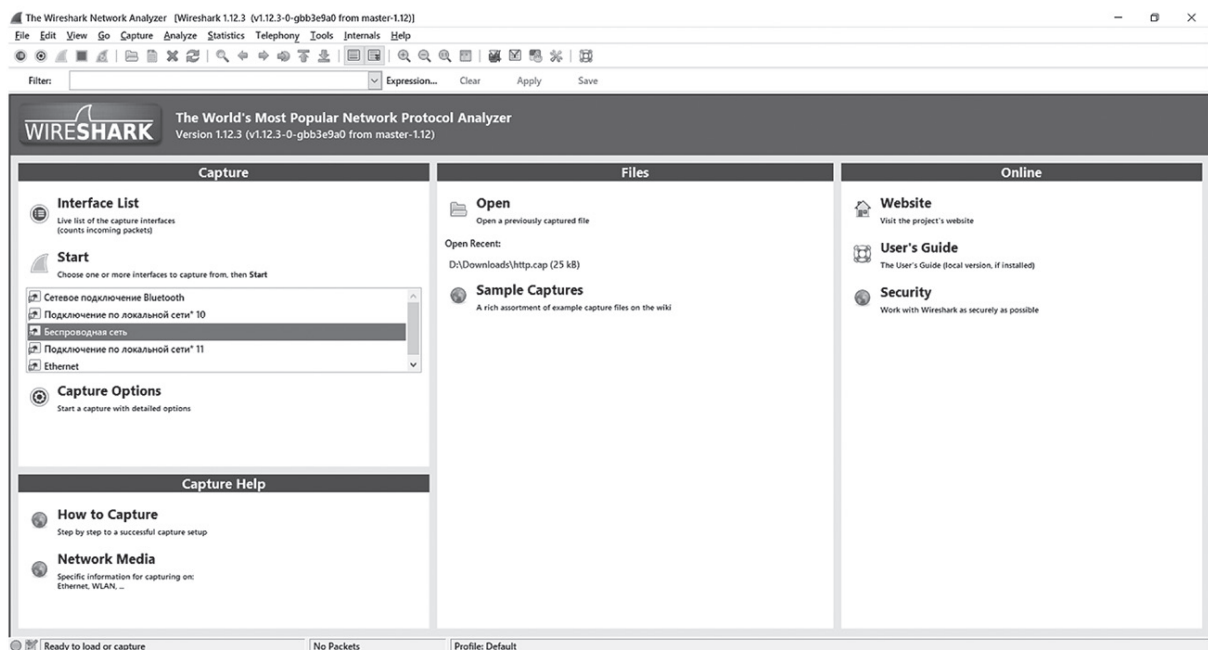


Рис. 1. API-інтерфейс версії 1.12.3 Wireshark Network Analyzer

Після відфільтрувати дані за протоколом SSH. #6 (див. рис. 2).

А тепер опишемо і застосуємо різні фільтри по IP та по типу протокола.

Відношення фільтрування

Створити фільтр можна вручну як у рядку запитів, так і за допомогою контекстно-залежного меню.

Wireshark 1.12.3 (v1.12.3-0-gba3e9ad from master.1.12.3)

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filters: Expression... Clear Apply Save

1

No.	Time	Source	Destination	Protocol	Length	Info
15	1.0678170	192.168.3.251	188.43.75.34	HTTP	104	GET /ncsi.txt HTTP/1.1
16	1.0678910	192.168.3.251	192.168.3.74	DNS	88	Standard query 0xfa7d A au-per.phantom.avira-vpn.com
17	1.0678900	192.168.3.251	192.168.3.74	DNS	84	Standard query 0x3a8c A nl.phantom.avira-vpn.com
18	1.0687770	192.168.3.251	192.168.3.74	DNS	84	Standard query 0xd642 A at.phantom.avira-vpn.com
19	1.0689770	192.168.3.251	192.168.3.74	DNS	88	Standard query 0x0fe8 A au-syd.phantom.avira-vpn.com
20	1.0691270	192.168.3.251	192.168.3.74	DNS	84	Standard query 0x79ef A be.phantom.avira-vpn.com
21	1.1243550	188.43.75.34	192.168.3.251	HTTP	238	HTTP/1.1 200 OK (text/plain)
22	1.1244890	192.168.3.251	188.43.75.34	TCP	54	54350->80 [ACK] Seq=151 Ack=553 Win=63320 Len=0
23	1.1247800	192.168.3.74	192.168.3.251	DNS	104	Standard query response 0xfa7d A 103.77.234.146
24	1.1256140	192.168.3.251	192.168.3.74	DNS	84	Standard query 0xf9ea A bg.phantom.avira-vpn.com
25	1.1256460	192.168.3.251	188.43.75.34	HTTP	104	GET /ncsi.txt HTTP/1.1
26	1.1259480	192.168.3.251	103.77.234.146	ICMP	74	Echo (ping) request id=0x0001, seq=307/13057, ttl=128 (reply in 132)
27	1.1266340	192.168.3.251	192.168.3.74	DNS	84	Standard query 0xd83b A br.phantom.avira-vpn.com
28	1.1267630	192.168.3.74	192.168.3.251	DNS	116	Standard query response 0xd642 A 185.216.34.90 A 185.210.219.234
29	1.1271730	192.168.3.74	192.168.3.251	DNS	148	Standard query response 0x3a8c A 95.168.164.194 A 94.75.218.175 A 185.59.222.90 A 83.149.73.14
30	1.1274800	192.168.3.251	185.216.34.90	ICMP	74	Echo (ping) request id=0x0001, seq=308/13313, ttl=128 (reply in 71)
31	1.1278100	192.168.3.251	95.168.164.194	ICMP	74	Echo (ping) request id=0x0001, seq=309/13569, ttl=128 (reply in 66)
32	1.1281400	192.168.3.251	192.168.3.74	DNS	88	Standard query 0xad10 A gb-lon.phantom.avira-vpn.com
33	1.1284100	192.168.3.251	192.168.3.74	DNS	88	Standard query 0xd660 A gb-mnc.phantom.avira-vpn.com
34	1.1285860	192.168.3.74	192.168.3.251	DNS	104	Standard query response 0x0fe8 A 103.77.233.114
35	1.1292110	192.168.3.251	103.77.233.114	ICMP	74	Echo (ping) request id=0x0001, seq=310/13825, ttl=128 (reply in 136)
36	1.1298960	192.168.3.251	192.168.3.74	DNS	84	Standard query 0x2284 A hu.phantom.avira-vpn.com
37	1.1305180	192.168.3.74	192.168.3.251	DNS	100	Standard query response 0x79ef A 82.102.19.162
38	1.1311690	192.168.3.251	82.102.19.162	ICMP	74	Echo (ping) request id=0x0001, seq=311/14081, ttl=128 (reply in 70)
39	1.1318550	192.168.3.74	192.168.3.251	DNS	331	Standard query response 0x7e71 CNAME api-eu.phantom.avira-vpn.com CNAME prod-vpn-api-20220523-190
40	1.1318840	192.168.3.251	192.168.3.74	DNS	84	Standard query 0x2810 A de.phantom.avira-vpn.com

2

3

4

5

Frame 1: 666 bytes on wire (5328 bits), 666 bytes captured (5328 bits) on interface 0
 Ethernet II, Src: 94:08:53:90:71:0b (94:08:53:90:71:0b), Dst: IP4cast::ff:fa (01:00:5e:7f:fa)
 Internet Protocol Version 4, Src: 192.168.3.251 (192.168.3.251), Dst: 239.255.255.250 (239.255.255.250)
 User Datagram Protocol, Src Port: 59840 (59840), Dst Port: 3702 (3702)
 Data (624 bytes)

0000 01 00 5e 7f fa 94 08 53 90 71 0b 08 00 45 00 ..A.....S.q...E.
 0010 02 8c 10 88 00 01 11 f2 3b c0 a8 03 fb ef ff:.....
 0020 ff fa e9 c0 0e 76 02 78 af 29 3c 3f 78 6d 6c 20v.x.<?xml
 0030 76 65 72 73 69 6f 6e 3d 22 31 2e 30 22 20 65 6eversion="1.0" en
 0040 63 6f 64 69 6e 67 3d 22 75 74 66 2d 38 22 3f 3eutf-8"?>

Безпроводная сеть: «Live capture in progress» File C:\Users\... | Packets: 1369 - Displayed: 136... | Profile: Default

Рис. 2. Занурения в бездротову мережу 1.12.3 Wireshark



The screenshot displays a network traffic capture in Wireshark. The main focus is on a TCP connection. The packet list on the left shows a SYN packet (Seq=188,251,236,209) and a corresponding SYN-ACK packet (Seq=188,251,236,209, Ack=61209). The packet details on the right show the TCP header for the SYN-ACK packet, including the 'Window size scaling factor' field set to 1 (unknown). The packet bytes pane at the bottom shows the raw data of the packet.

Packet List:

No.	Time	Source	Destination	Protocol	Length	Info
837	0.946219	148.251.236.209	10.0.0.8	TCP	40	8443 → 61209 [ACK] Seq=188251236209 Win=0 Len=0
1180	14.917273	148.251.236.209	10.0.0.8	TCP	40	8443 → 61209 [ACK] Seq=188251236209 Win=0 Len=0
1580	19.584761	148.251.236.209	10.0.0.8	TCP	40	8443 → 61209 [ACK] Seq=188251236209 Win=0 Len=0
1586	19.630556	148.251.236.209	10.0.0.8	TCP	40	8443 → 61209 [ACK] Seq=188251236209 Win=0 Len=0
1600	20.914046	148.251.236.209	10.0.0.8	TCP	40	8443 → 61209 [ACK] Seq=188251236209 Win=0 Len=0
1853	22.727592	148.251.236.209	10.0.0.8	TCP	40	8443 → 61209 [ACK] Seq=188251236209 Win=0 Len=0
574	7.124613	148.251.236.209	10.0.0.8	TCP	40	8443 → 61209 [ACK] Seq=188251236209 Win=0 Len=0
20025	190.858239	148.251.236.209	10.0.0.8	TCP	40	8443 → 61209 [ACK] Seq=188251236209 Win=0 Len=0
3972	43.913632	148.251.236.209	10.0.0.8	TCP	40	8443 → 61209 [ACK] Seq=188251236209 Win=0 Len=0
4892	52.264098	148.251.236.209	10.0.0.8	TCP	40	8443 → 61209 [ACK] Seq=188251236209 Win=0 Len=0
5148	56.325390	148.251.236.209	10.0.0.8	TCP	40	8443 → 61209 [ACK] Seq=188251236209 Win=0 Len=0
5353	58.646215	148.251.236.209	10.0.0.8	TCP	40	8443 → 61209 [ACK] Seq=188251236209 Win=0 Len=0
5357	58.691832	148.251.236.209	10.0.0.8	TCP	40	8443 → 61209 [ACK] Seq=188251236209 Win=0 Len=0
5420	59.346974	148.251.236.209	10.0.0.8	TCP	40	8443 → 61209 [ACK] Seq=188251236209 Win=0 Len=0
5424	59.389454	148.251.236.209	10.0.0.8	TCP	40	8443 → 61209 [ACK] Seq=188251236209 Win=0 Len=0
20180	191.024613	148.251.236.209	10.0.0.8	TCP	40	8443 → 61209 [ACK] Seq=188251236209 Win=0 Len=0
20811	195.571983	148.251.236.209	10.0.0.8	TCP	40	8443 → 61209 [ACK] Seq=188251236209 Win=0 Len=0
20936	200.320187	148.251.236.209	10.0.0.8	TCP	40	8443 → 61209 [ACK] Seq=188251236209 Win=0 Len=0
21311	203.202476	148.251.236.209	10.0.0.8	TCP	40	8443 → 61209 [ACK] Seq=188251236209 Win=0 Len=0
21367	203.425574	148.251.236.209	10.0.0.8	TCP	40	8443 → 61209 [ACK] Seq=188251236209 Win=0 Len=0
23239	210.719077	148.251.236.209	10.0.0.8	TCP	40	8443 → 61209 [ACK] Seq=188251236209 Win=0 Len=0
25710	217.175255	148.251.236.209	10.0.0.8	TCP	40	8443 → 61209 [ACK] Seq=188251236209 Win=0 Len=0
28882	226.688995	148.251.236.209	10.0.0.8	TCP	40	8443 → 61209 [ACK] Seq=188251236209 Win=0 Len=0
32173	238.792878	148.251.236.209	10.0.0.8	TCP	40	8443 → 61209 [ACK] Seq=188251236209 Win=0 Len=0
33567	243.541764	148.251.236.209	10.0.0.8	TCP	40	8443 → 61209 [ACK] Seq=188251236209 Win=0 Len=0

Packet Details:

- Ethernet II, Src: Intel (08:00:00:00:00:00), Dst: Intel (08:00:00:00:00:00)**
- Internet Protocol Version 4, Src: 148.251.236.209, Dst: 10.0.0.8**
- Transmission Control Protocol, Src Port: 8443, Dst Port: 61209, Seq: 1, Ack: 4267, Len: 0**
- TCP Segment Len: 0**
- Sequence Number: 1 (relative sequence number)**
- Next Sequence Number: 4267 (relative sequence number)**
- Acknowledgment Number: 257673644 (relative ack number)**
- Flags: 0x010 (ACK)**
- Window: 2441**
- Calculated window size: 2441**
- Window size scaling factor: 1 (unknown)**
- Checksum: 0x3db0 [unverified]**

Packet Bytes:

```

0000  45 00 00 28 b0 52 40 00 38 06 19 a8 94 f0 e5 d1  ... .. 8.9m..3
0010  0a 00 00 00 20 f6 e7 19 1a c0 08 be 0f 50 c9 ac  ... ..{Q.$$.
0020  50 10 00 09 3d db 00 00  ... ..8.1m...
  
```

Рис. 4. Не выбране.

Створимо фільтр типу ВІКЛЮЧЕННЯ для конкретного ІР. Це можна зробити прописавши у пошуковому рядку “!”



Рис. 5. Виключення — це знак інверсії, який пишеться як “!” перед ip.src у командному рядку

5. Використовуйте фільтри Wireshark для відсіювання не-потрібної інформації та підкреслення потенційно небезпечного трафіку.

Опишіть ці фільтри ось тут по назвах і по призначенню. Також давайте СКРІН, де використано фільтр.

6. Зверніть увагу на будь-які аномалії у звичайному шаблоні трафіку, такі як надмірна кількість запитів, незвичайний обсяг даних або неправильний тип запитів.
7. Якщо ви виявите підозрілий трафік, спробуйте встановити його джерело та призначення, а також тип даних, які передаються.
8. Після аналізу зробіть звіт з виявленими загрозами та рекомендаціями щодо подальших заходів з усунення можливих проблем безпеки.

Отримані навички

Це завдання допоможе учасникам практично застосувати навички аналізу мережевого трафіку з використанням програми Wireshark та виявлення можливих загроз для безпеки мережі.

Контрольні запитання

- Як за допомогою Wireshark виявити ознаки Ddos атаки у вашій мережі?
- По якому протоколу у вашій мережі отримано найбільше пакетів?
- Якою командою задається фільтр типу АБО?
- Якою командою задається фільтр типу І?
- Створіть фільтр типу ВИКЛЮЧЕННЯ для конкретного ІР.

Примітка. Будьте обережні та дотримуйтесь внутрішніх політик безпеки вашої організації під час аналізу мережевого трафіку.