



АТЕСТАЦІЙНА РОБОТА

Методи та засоби аналізу мережного трафіку

Виконав:
студент гр. СПзм-18-2
Сидоров Д.С.

Керівник:
проф. каф. ЕОМ,
д.т.н. Міхаль О.П.

МЕТА РОБОТИ ТА ЗАВДАННЯ

2

Метою роботи є дослідження методів та засобів аналізу мережного трафіку.

Завдання:

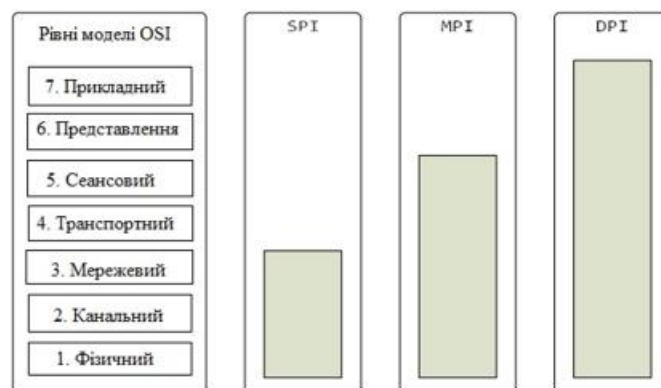
- аналіз існуючих моделей та методів аналізу мережних пакетів, враховучи особливості передачі даних по мережі (втрата окремих пакетів, стиснення і шифрування даних, вкладене тунелювання);
- розробка аналізатора мережного трафіку з використанням досліджених методів.

Об'єктом дослідження є передача пакетів даних у комп'ютерній мережі.

ОБЛАСТІ ПРАКТИЧНОГО ЗАСТОСУВАННЯ 3

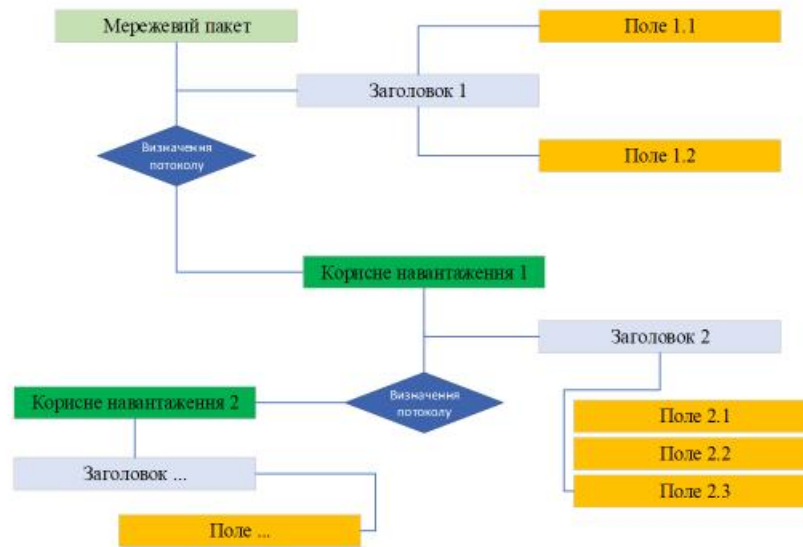
- виявлення проблем в роботі мережі;
- тестування (налагодження) мережевих протоколів;
- запобігання мережевих атак;
- класифікація трафіку.

КЛАСИ ПРОВЕДЕНОГО РОЗБОРУ МЕРЕЖЕВИХ ПАКЕТІВ 4



ВИДІЛЕННЯ І РОЗБІР ЗАГОЛОВКІВ ПРОТОКОЛІВ В ПАКЕТІ

5



ФУНКЦІОНАЛЬНІ ВИМОГИ ДО АНАЛІЗАТОРІВ ТРАФІКУ

6

1. Відновлення та подальший розбір потоків даних з урахуванням можливого переупорядкування та / або втрати окремих пакетів.
2. Аналіз вкладених тунелів.
3. Аналіз зашифрованих (модифікованих) даних.
4. Автоматичне визначення протоколу вищого рівня.
5. Локалізація та відтворення помилок розбору.
6. Додавання підтримки протоколів без внесення змін до вже існуючих модулів розбору.
7. Переносимість модулів розбору між режимами offline- та online-аналізу.

ЗАСОБИ АНАЛІЗУ МЕРЕЖНОГО ТРАФІКУ

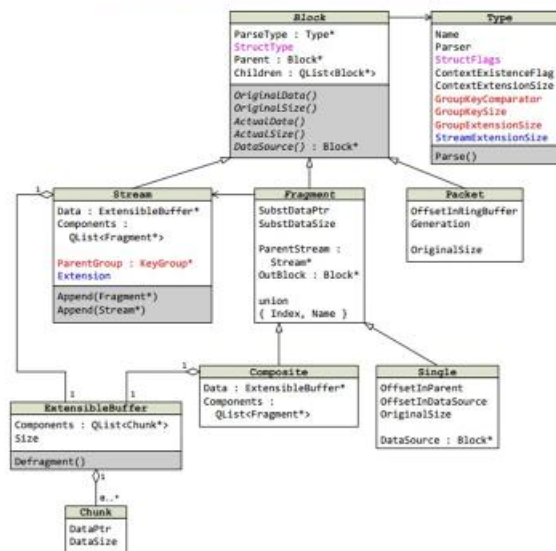
7



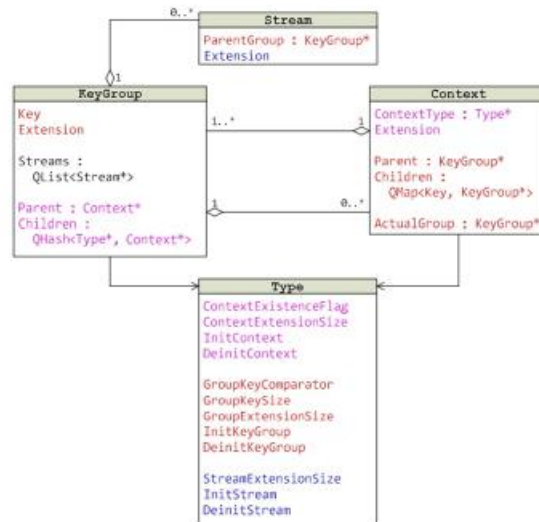
##	SRC (IP:PORT)	DST (IP:PORT)	Коректне відновлення потоку		
			Wireshark v.2.2.3	Snort v.2.9.7.0	Bro v.2.5
Trace11.pcap					
1	68.142.205.139:80	192.168.0.105:25168	-	-	+
2	68.142.205.139:80	192.168.0.105:25175	-	-	+
3	68.142.205.139:80	192.168.0.105:25180	+	-	+
Trace12.pcap					
4	74.125.19.113:443	192.168.0.105:24044	+	-	+
5	68.142.205.139:80	192.168.0.105:24053	-	-	+
6	68.142.205.139:80	192.168.0.105:24060	+	-	+
7	68.142.205.139:80	192.168.0.105:24089	+	-	+
Trace13.pcap					
8	74.125.224.96:443	24.6.173.220:61960	+	-	+
Trace14.pcap					
9	174.46.74.87:80	192.168.2.7:43542	-	-	+

ДІАГРАМА КЛАСІВ ДЛЯ ОПИСУ РОЗІБРАНИХ МЕРЕЖЕВИХ ПАКЕТІВ

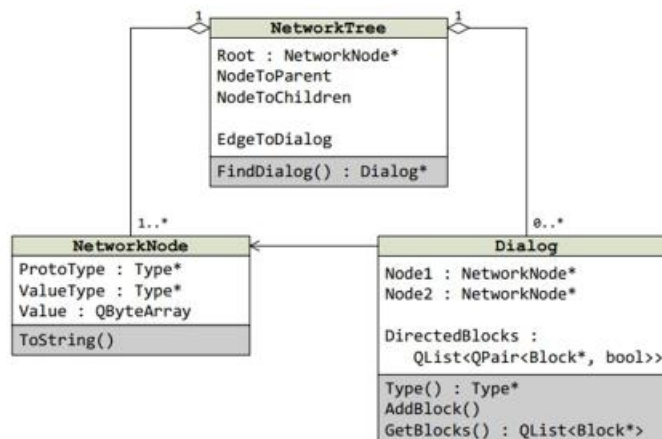
8



ДІАГРАМА КЛАСІВ ДЛЯ ОПИСУ ЛОГІЧНИХ З'ЄДНАНЬ 9

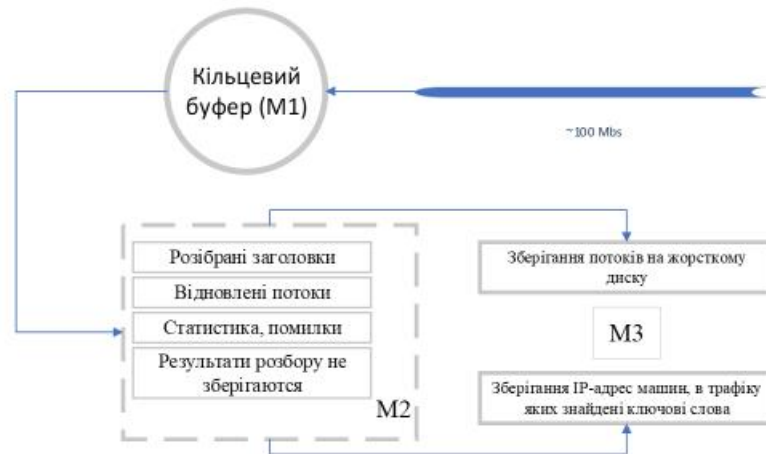


ДІАГРАМА КЛАСІВ ДЛЯ ОПИСУ УЧАСНИКІВ МЕРЕЖЕВОГО ОБМІНУ 10



АНАЛІЗАТОР ТРАФІКУ

11



РЕЗУЛЬТАТИ РОБОТИ

12

Block tree

```

build(Tcp, parse(HTTP_STREAM), [0x0:0x32391]
  1, Packet(Http, [0x0:0x32391]
    StartLine(CRLFString), HTTP/1.1 200 OK\r\n
    Option(CRLFString), Server: nginx\r\n
    Option(CRLFString), Date: Tue, 14 Jan 2020 17:53:40 GMT\r\n
    Option(CRLFString), Content-Type: text/html; charset=windows-1251\r\n
    Option(CRLFString), Transfer-Encoding: chunked\r\n
    Option(CRLFString), Connection: keep-alive\r\n
    Option(CRLFString), Keep-Alive: timeout=25\r\n
    Option(CRLFString), Expires: Tue, 14 Jan 2020 17:54:10 GMT\r\n
    Option(CRLFString), Cache-Control: max-age=30\r\n
    Option(CRLFString), Serv: ng\r\n
    Option(CRLFString), \r\n
    ChunkedData(ChunkedData)
      1, Chunk(Chunk), [0x10:0xFCB]
      2, Chunk(Chunk), [0x100B:0x74F]
      3, Chunk(Chunk), [0x182A:0x488]
      4, Chunk(Chunk), [0x1CB2:0x8]
      5, Chunk(Chunk), [0x1CBA:0x472]
      6, Chunk(Chunk), [0x212C:0x780]
      7, Chunk(Chunk), [0x280C:0x48C]
      8, Chunk(Chunk), [0x2D68:0x43]
      9, Chunk(Chunk), [0x2D48:0x596]
      10, Chunk(Chunk), [0x3301:0x392]
      11, Chunk(Chunk), [0x3693:0x473]
      12, Chunk(Chunk), [0x4106:0x1008]
      13, Chunk(Chunk), [0x510E:0x1008]
      14, Chunk(Chunk), [0x6116:0x1008]
  )
)

```

Components: 154

##	Offset	Size	From	Info
23	0x5FDC	0x580	Tcp(46)	SrcPort: 80, DstPort: 43542, Flags: Ack
24	0x615C	0x580	Tcp(48)	SrcPort: 80, DstPort: 43542, Flags: Ack
25	0x64DC	0x580	Tcp(50)	SrcPort: 80, DstPort: 43542, Flags: Ack
26	0x705C	0x580	Tcp(52)	SrcPort: 80, DstPort: 43542, Flags: Ack
27	0x75DC	0x580	Tcp(54)	SrcPort: 80, DstPort: 43542, Flags: Ack
28	0x785C	0x580	Tcp(56)	SrcPort: 80, DstPort: 43542, Flags: Ack, Push
29	0x80DC	0x8	Tcp(58)	SrcPort: 80, DstPort: 43542, Flags: Ack
30	0x80E7	0x580	Tcp(60)	SrcPort: 80, DstPort: 43542, Flags: Ack
31	0x8667	0x580	Tcp(62)	SrcPort: 80, DstPort: 43542, Flags: Ack
32	0x88E7	0x580	Tcp(64)	SrcPort: 80, DstPort: 43542, Flags: Ack
33	0x9167	0x580	Tcp(66)	SrcPort: 80, DstPort: 43542, Flags: Ack
34	0x96E7	0x580	Tcp(68)	SrcPort: 80, DstPort: 43542, Flags: Ack
35	0x9C67	0x580	Tcp(70)	SrcPort: 80, DstPort: 43542, Flags: Ack
36	0xA1E7	0x580	Tcp(72)	SrcPort: 80, DstPort: 43542, Flags: Ack
37	0xA7E7	0x580	Tcp(74)	SrcPort: 80, DstPort: 43542, Flags: Ack
38	0xACET	0x580	Tcp(76)	SrcPort: 80, DstPort: 43542, Flags: Ack
39	0xB267	0x580	Tcp(78)	SrcPort: 80, DstPort: 43542, Flags: Ack
40	0xB7E7	0x580	Tcp(80)	SrcPort: 80, DstPort: 43542, Flags: Ack
41	0xBD67	0x580	Tcp(82)	SrcPort: 80, DstPort: 43542, Flags: Ack
42	0xC2E7	0x580	Tcp(84)	SrcPort: 80, DstPort: 43542, Flags: Ack
43	0xC8E7	0x580	Tcp(86)	SrcPort: 80, DstPort: 43542, Flags: Ack
44	0xCDE7	0x580	Tcp(88)	SrcPort: 80, DstPort: 43542, Flags: Ack
45	0xD367	0x580	Tcp(90)	SrcPort: 80, DstPort: 43542, Flags: Ack
46	0xD8E7	0x580	Tcp(200)	SrcPort: 80, DstPort: 43542, Flags: Ack
47	0xDE67	0x580	Tcp(92)	SrcPort: 80, DstPort: 43542, Flags: Ack

РЕЗУЛЬТАТИ РОБОТИ

13

No.	Source	Destination	Protocol	Info
1	1.1.1.1	2.2.2.2	ICMP	Echo (ping) request id=0x0004, seq=0/0, ttl=255 (reply in 2)
2	2.2.2.2	1.1.1.1	ICMP	Echo (ping) reply id=0x0004, seq=0/0, ttl=255 (request in 1)
3	1.1.1.1	2.2.2.2	ICMP	Echo (ping) request id=0x0004, seq=1/256, ttl=255 (reply in 4)
4	2.2.2.2	1.1.1.1	ICMP	Echo (ping) reply id=0x0004, seq=1/256, ttl=255 (request in 3)
5	1.1.1.1	2.2.2.2	ICMP	Echo (ping) request id=0x0004, seq=2/512, ttl=255 (reply in 6)
6	2.2.2.2	1.1.1.1	ICMP	Echo (ping) reply id=0x0004, seq=2/512, ttl=255 (request in 5)
7	1.1.1.1	2.2.2.2	ICMP	Echo (ping) request id=0x0004, seq=3/768, ttl=255 (reply in 8)
8	2.2.2.2	1.1.1.1	ICMP	Echo (ping) reply id=0x0004, seq=3/768, ttl=255 (request in 7)
9	1.1.1.1	2.2.2.2	ICMP	Echo (ping) request id=0x0004, seq=4/1024, ttl=255 (reply in 10)
10	2.2.2.2	1.1.1.1	ICMP	Echo (ping) reply id=0x0004, seq=4/1024, ttl=255 (request in 9)

> Frame 1: 134 bytes on wire (1072 bits), 134 bytes captured (1072 bits)
 > Ethernet II, Src: c2:00:57:75:00:00 (c2:00:57:75:00:00), Dst: c2:01:57:75:00:00 (c2:01:57:75:00:00)
 > Internet Protocol Version 4, Src: 10.0.0.1, Dst: 10.0.0.2
 > Internet Protocol Version 4, Src: 1.1.1.1, Dst: 2.2.2.2
 > Internet Control Message Protocol

The image shows a Wireshark interface. On the left, a packet list shows several HTTP and SSL packets. The right pane shows the details of a selected packet (No. 942), including fields like Version, Class, Label, Payload Size, and Src/Dst IP addresses. The bottom pane shows a hex dump of the packet data.

РЕЗУЛЬТАТИ РОБОТИ

14

The image shows a Wireshark interface displaying an HTTP stream. The 'Block tree' on the left shows the structure of the stream, including '1, Packet(Http)', 'StartLine(CRLFString)', and various options like 'Date', 'Content-Type', and 'Cache-Control'. The 'Data' pane on the right shows the raw data of the stream, including the status line 'HTTP/1.1 200 OK' and the body content.

ВИСНОВКИ

15

Проведено дослідження методів та засобів аналізу мережного трафіку. Проаналізовано методи відновлення потоків даних для протоколів довільного рівня, в тому числі прикладного, які стійкі до втрат окремих мережових пакетів, а також їх переупорядкування. Запропонована архітектура системи поглибленого аналізу мережного трафіку, що дозволяє розробляти і налагоджувати модулі підтримки протоколів на попередньо збереженому трафіку (offline) і згодом використовувати ці модулі в режимі online. Розроблені та реалізовано програмні інструменти для проведення поглибленого аналізу мережного трафіку в online і offline режимах.