



작성자	윤지원
분석 일자	2024.05.08
작성 일자	2024.05.08
분석 대상	Pcap 파일
문서 버전	1
작성자 E-mail	yoonjw0827@gmail.com

0. 목차

1. 문제3

2. 분석 도구3

3. 환경3


4. Write-Up.....4

5. Flag.....8

6. 별도 첨부9

7. Reference 10

1. 문제

URL	https://dreamhack.io/wargame/challenges/605?writeup_page=1
문제 내용	Do shark sleep?
문제 파일	 dump.pcap
문제 유형	forensics
난이도	2.5 / 5

2. 분석 도구

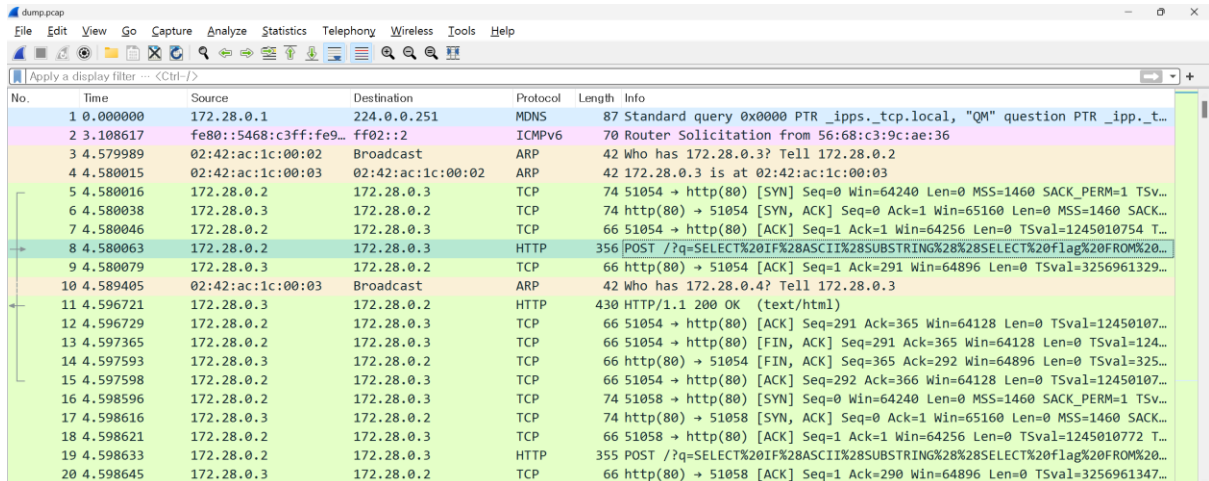
도구명	다운로드 링크	Version
wireshark	Wireshark · Download	3.4.7

3. 환경

OS
Windows 11 64-bit

4. Write-Up

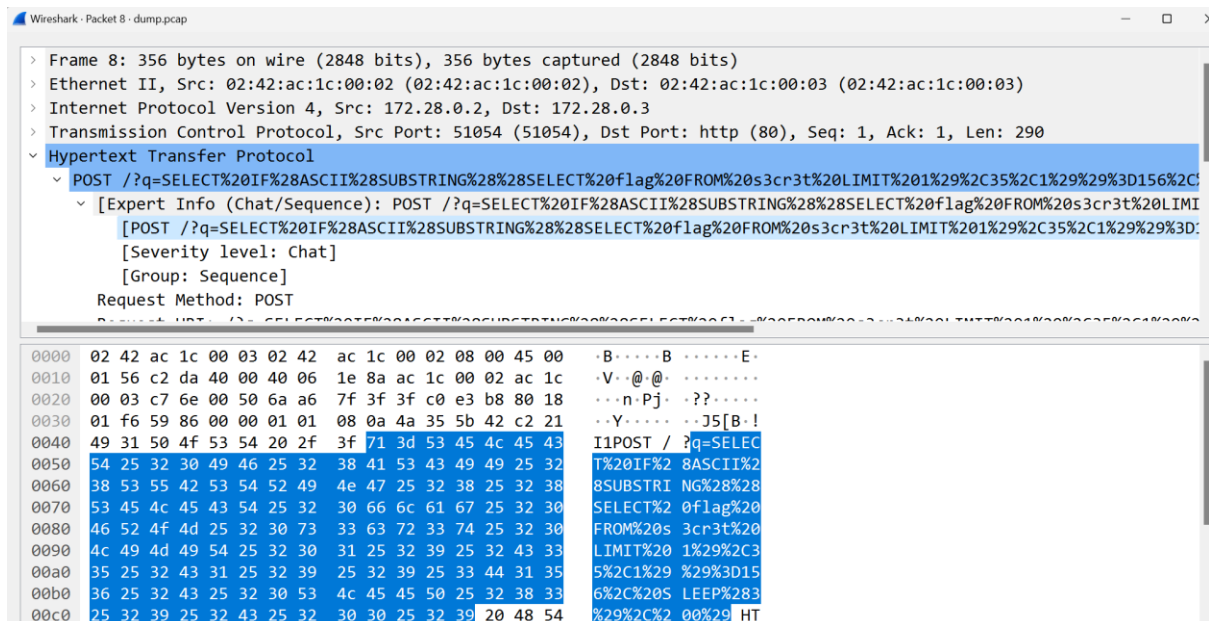
파일명	dump.pcap
용량	14,524KB
SHA256	208a6dda15caae1d83a70990e79099cfd26b820164ebe76777dbf9afe1967fe0
Timestamp	2022-08-26 18:03:00



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	172.28.0.1	224.0.0.251	MDNS	87	Standard query 0x0000 PTR _ipps._tcp.local, "QM" question PTR _ipp.t...
2	3.108617	fe80::5468:c3ff:fe9...	ff02::2	ICMPv6	70	Router Solicitation from 56:68:c3:9c:ae:36
3	4.579989	02:42:ac:1c:00:02	Broadcast	ARP	42	Who has 172.28.0.3? Tell 172.28.0.2
4	4.580015	02:42:ac:1c:00:03	02:42:ac:1c:00:02	ARP	42	172.28.0.3 is at 02:42:ac:1c:00:03
5	4.580016	172.28.0.2	172.28.0.3	TCP	74	51054 → http(80) [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 Tsv...
6	4.580038	172.28.0.3	172.28.0.2	TCP	74	http(80) → 51054 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK...
7	4.580046	172.28.0.2	172.28.0.3	TCP	66	51054 → http(80) [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1245010754 T...
8	4.580063	172.28.0.2	172.28.0.3	HTTP	356	POST /?q=SELECT%20IF%28ASCII%28SUBSTRING%28%28SELECT%20flag%20FROM%20...]
9	4.580079	172.28.0.3	172.28.0.2	TCP	66	http(80) → 51054 [ACK] Seq=1 Ack=291 Win=64896 Len=0 TSval=3256961329...
10	4.589405	02:42:ac:1c:00:03	Broadcast	ARP	42	Who has 172.28.0.4? Tell 172.28.0.3
11	4.596721	172.28.0.3	172.28.0.2	HTTP	430	HTTP/1.1 200 OK (text/html)
12	4.596729	172.28.0.2	172.28.0.3	TCP	66	51054 → http(80) [ACK] Seq=291 Ack=365 Win=64128 Len=0 TSval=12450107...
13	4.597365	172.28.0.2	172.28.0.3	TCP	66	51054 → http(80) [FIN, ACK] Seq=291 Ack=365 Win=64128 Len=0 TSval=124...
14	4.597593	172.28.0.3	172.28.0.2	TCP	66	http(80) → 51054 [FIN, ACK] Seq=365 Ack=292 Win=64896 Len=0 TSval=325...
15	4.597598	172.28.0.2	172.28.0.3	TCP	66	51054 → http(80) [ACK] Seq=292 Ack=366 Win=64128 Len=0 TSval=12450107...
16	4.598596	172.28.0.2	172.28.0.3	TCP	74	51058 → http(80) [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 Tsv...
17	4.598616	172.28.0.3	172.28.0.2	TCP	74	http(80) → 51058 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK...
18	4.598621	172.28.0.2	172.28.0.3	TCP	66	51058 → http(80) [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1245010772 T...
19	4.598633	172.28.0.2	172.28.0.3	HTTP	355	POST /?q=SELECT%20IF%28ASCII%28SUBSTRING%28%28SELECT%20flag%20FROM%20...
20	4.598645	172.28.0.3	172.28.0.2	TCP	66	http(80) → 51058 [ACK] Seq=1 Ack=290 Win=64896 Len=0 TSval=3256961347...

[사진 1] wireshark를 이용하여 연 dump.pcap

문제 파일을 압축 해제 하였을 때, dump.pcap 파일이 존재하고 있음을 확인하였고, wireshark 이
용시 사용되는 파일형태이기 때문에 바로 wireshark를 이용하여 열어보았습니다. 8번 패킷에서
flag라는 단어를 발견하고 이를 더블클릭하여 다음과 같은 화면을 볼 수 있었습니다.



Wireshark - Packet 8 - dump.pcap

- Frame 8: 356 bytes on wire (2848 bits), 356 bytes captured (2848 bits)
- Ethernet II, Src: 02:42:ac:1c:00:02 (02:42:ac:1c:00:02), Dst: 02:42:ac:1c:00:03 (02:42:ac:1c:00:03)
- Internet Protocol Version 4, Src: 172.28.0.2, Dst: 172.28.0.3
- Transmission Control Protocol, Src Port: 51054 (51054), Dst Port: http (80), Seq: 1, Ack: 1, Len: 290
- Hypertext Transfer Protocol
 - POST /?q=SELECT%20IF%28ASCII%28SUBSTRING%28%28SELECT%20flag%20FROM%20s3cr3t%20LIMIT%201%29%2C35%2C1%29%29%3D156%2C...
 - [Expert Info (Chat/Sequence): POST /?q=SELECT%20IF%28ASCII%28SUBSTRING%28%28SELECT%20flag%20FROM%20s3cr3t%20LIMIT%201%29%2C35%2C1%29%29%3D156%2C...
 - [Severity level: Chat]
 - [Group: Sequence]
 - Request Method: POST

Raw data (hex):

```

0000  02 42 ac 1c 00 03 02 42 ac 1c 00 02 08 00 45 00  ·B·····B·····E·
0010  01 56 c2 da 40 00 40 06 1e 8a ac 1c 00 02 ac 1c  ·V·@·@· ······
0020  00 03 c7 6e 00 50 6a a6 7f 3f 3f c0 e3 b8 80 18  ····n·Pj· ·??·
0030  01 f6 59 86 00 00 01 01 08 0a 4a 35 5b 42 c2 21  ··Y······J5[B·!
0040  49 31 50 4f 53 54 20 2f 3f 71 3d 53 45 4c 45 43  I1POST / ?q=SELEC
0050  54 25 32 30 49 46 25 32 38 41 53 43 49 49 25 32  T%20IF%2 8ASCII%2
0060  38 53 55 42 53 54 52 49 4e 47 25 32 38 25 32 38  8SUBSTRIN G%28%28
0070  53 45 4c 45 43 54 25 32 30 66 6c 61 67 25 32 30  SELECT%2 0flag%20
0080  46 52 4f 4d 25 32 30 73 33 63 72 33 74 25 32 30  FROM%20s 3cr3t%20
0090  4c 49 4d 49 54 25 32 30 31 25 32 39 25 32 43 33  LIMIT%20 1%29%2C3
00a0  35 25 32 43 31 25 32 39 25 32 39 25 33 44 31 35  5%2C1%29 %29%3D15
00b0  36 25 32 43 25 32 30 53 4c 45 45 50 25 32 38 33  6%2C%20S LEEP%283
00c0  25 32 39 25 32 43 25 32 30 30 25 32 39 20 48 54  %29%2C%2 00%29 HT
  
```

[사진 2] 8번 패킷을 자세히 살펴본 모습

[WHS-2] .iso

해당 패킷 내용은 POST 메소드로 패킷을 전송하였다는 내용이었고, flag부분이 url인코딩 되어있다는 사실을 알 수 있었습니다. url 인코딩은 url로 사용할 수 있지만 의미가 왜곡될 수 있는 문자들을 16진수 값으로 변환한 것이기 때문에 이를 디코딩하면 url을 다시 원래 형태로 볼 수 있습니다. 따라서 url decoding을 검색하였더니 <https://www.urldecoder.org/> 이라는 사이트를 발견할 수 있었습니다. 아래의 값을 사이트에 집어넣은 다음 decode를 하였더니 다음과 같은 화면을 볼 수 있었습니다.

Decode from URL-encoded format
Simply enter your data then push the decode button.

POST /?q=SELECT%20IF%28ASCII%28SUBSTRING%28%28SELECT%20flag%20FROM%20s3cr3t%20LIMIT%201%29%2C35%2C1%29%29%3D156%2C%20SLEEP%283%29%2C%200%29

For encoded binaries (like images, documents, etc.) use the file upload form a little further down on this page.

UTF-8 Source character set.

☐ Decode each line separately (useful for when you have multiple entries).

☒ Live mode OFF Decodes in real-time as you type or paste (supports only the UTF-8 character set).

< DECODE > Decodes your data into the area below.

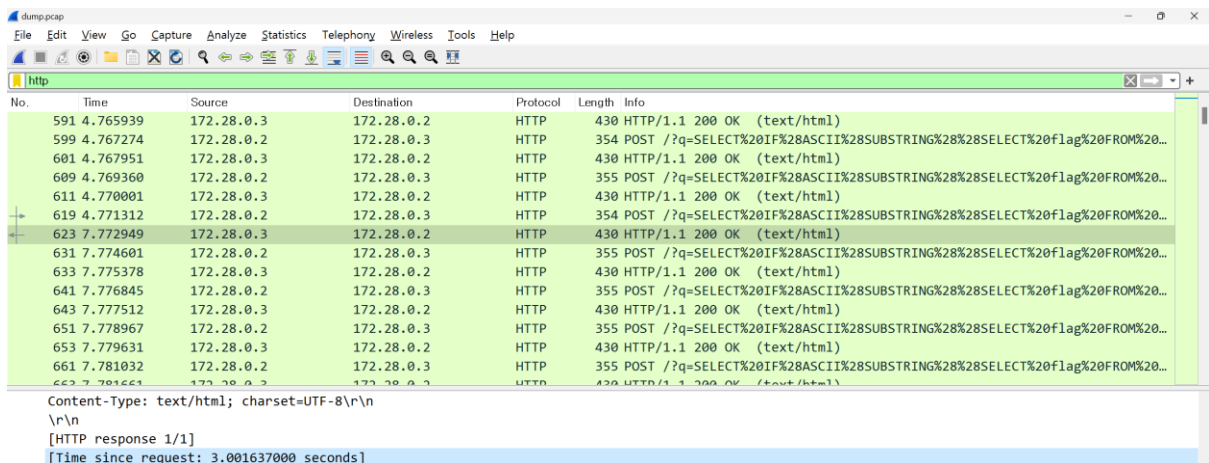
POST/?q=SELECT IF(ASCII(SUBSTRING((SELECT flag FROM s3cr3t LIMIT 1),35,1))=156, SLEEP(3), 0)

[사진 3] urldecoder를 통해 살펴본 전체 url

디코딩된 값은 다음과 같습니다.

- SELECT flag FROM s3cr3t LIMIT 1 : s3cr3t에서 flag값을 1개씩 가져옵니다.
- ASCII ~ = 156 : 나오는 값은 ASCII 값으로 나오며, 156과 일치합니다.
- IF(ASCII ~ = 156, SLEEP(3), 0) : ASCII 값이 156으로 나올 때, 3초 SLEEP한다. 그렇지 않으면 0이 반환됩니다.
- 이때, SUBSTRING 함수에 따라 35 위치에 들어가는 숫자는 글자의 배열 순서를 나타냅니다.

따라서 flag 값을 찾기 위해서는 3초 단위로 HTTP 패킷을 분석하면 될 것이라고 생각하였고, HTTP 패킷들만 분석하여 timestamp가 3초 이상인 것들의 정보를 순서대로 합치면 온전한 flag 값이 나올 것임을 예상하였습니다.



No.	Time	Source	Destination	Protocol	Length	Info
591	4.765939	172.28.0.3	172.28.0.2	HTTP	430	HTTP/1.1 200 OK (text/html)
599	4.767274	172.28.0.2	172.28.0.3	HTTP	354	POST /?q=SELECT%20IF%28ASCII%28SUBSTRING%28%28SELECT%20flag%20FROM%20...
601	4.767951	172.28.0.3	172.28.0.2	HTTP	430	HTTP/1.1 200 OK (text/html)
609	4.769360	172.28.0.2	172.28.0.3	HTTP	355	POST /?q=SELECT%20IF%28ASCII%28SUBSTRING%28%28SELECT%20flag%20FROM%20...
611	4.770001	172.28.0.3	172.28.0.2	HTTP	430	HTTP/1.1 200 OK (text/html)
619	4.771312	172.28.0.2	172.28.0.3	HTTP	354	POST /?q=SELECT%20IF%28ASCII%28SUBSTRING%28%28SELECT%20flag%20FROM%20...
623	7.772949	172.28.0.3	172.28.0.2	HTTP	430	HTTP/1.1 200 OK (text/html)
631	7.774601	172.28.0.2	172.28.0.3	HTTP	355	POST /?q=SELECT%20IF%28ASCII%28SUBSTRING%28%28SELECT%20flag%20FROM%20...
633	7.775378	172.28.0.3	172.28.0.2	HTTP	430	HTTP/1.1 200 OK (text/html)
641	7.776845	172.28.0.2	172.28.0.3	HTTP	355	POST /?q=SELECT%20IF%28ASCII%28SUBSTRING%28%28SELECT%20flag%20FROM%20...
643	7.777512	172.28.0.3	172.28.0.2	HTTP	430	HTTP/1.1 200 OK (text/html)
651	7.778967	172.28.0.2	172.28.0.3	HTTP	355	POST /?q=SELECT%20IF%28ASCII%28SUBSTRING%28%28SELECT%20flag%20FROM%20...
653	7.779631	172.28.0.3	172.28.0.2	HTTP	430	HTTP/1.1 200 OK (text/html)
661	7.781032	172.28.0.2	172.28.0.3	HTTP	355	POST /?q=SELECT%20IF%28ASCII%28SUBSTRING%28%28SELECT%20flag%20FROM%20...

Content-Type: text/html; charset=UTF-8\r\n
\r\n
[HTTP response 1/1]
[Time since request: 3.001637000 seconds]

[사진 4] 619번 패킷에서 623번 패킷으로 넘어갈 때 timestamp가 3초 이상인 모습

위의 사진을 예시로 보면, 619번 패킷에서 623번 패킷으로 넘어갈 때 Time이 4초대에서 7초대로 갑자기 증가하는 것을 볼 수 있습니다. 이렇게 저는 timestamp가 3초씩 건너뛰어지는 패킷 총 39개를 찾아냈습니다. 몇 번째 패킷에서 어떤 flag 값이 나왔는지는 별도 첨부해 기재하겠습니다.



이진 번역기
이진, 10 진수, 16 진수, ASCII 코드 및 일반 텍스트 간 변환

검색

텍스트 ⇄ 이진
텍스트 ⇄ 아스키
 텍스트 ⇄ 16 진수
 텍스트 ⇄ 십진법
 이진 ⇄ 소수
 이진 ⇄ 육각
 십진 ⇄ 육각

명확한 부

G_oBmn1__5wh7{I33iL4jsl}c7
dT_EnQPN_p0c4

명확한 부

71 95 111 66 109 110 49 95 95
 53 119 104 55 123 73 51 51 105
 76 52 106 115 73 125 99 55 100
 84 95 69 110 81 80 78 95 112 48
 99 52

텍스트를 아스키에 >

< 텍스트에 아스키

[사진 5] 이진 번역기를 이용하여 찾은 아스키 형태의 플래그 값을 텍스트로 변환한 모습

이렇게 나온 flag 값들을 쭉 나열하여 이진 번역기에 입력한 다음, 텍스트로 변환하면 [사진 5]와 같은 값이 나옵니다. 이것은 전혀 일반적인 flag 값의 형태를 띄고 있지 않기 때문에 찾은 flag 값들의 나열 순서가 문제일 것이라고 생각하였습니다. 따라서 urldecoder를 이용하여 찾은 SUBSTRING 함수의 문자열 순서대로 flag 값을 나열한 형태는 다음과 같습니다.

71 111 78 123 84 49 109 69 95 66 52 115 51 100 95 53 81 76 95 73 110 106 51 99 55 105 48 110
 95 119 73 55 104 95 80 99 52 112 125



이진 번역기

이진, 10 진수, 16 진수, ASCII 코드 및 일반 텍스트 간 변환

검색



★

텍스트 ⇄ 이진

텍스트 ⇄ 아스키

텍스트 ⇄ 16 진수

텍스트 ⇄ 십진법

이진 ⇄ 소수

이진 ⇄ 육각

십진 ⇄ 육각

명확한

부

GoN{T1mE_B4s3d_5QL_Inj3c7

i0n_wl7h_Pc4p}

텍스트를 아스키에 >

명확한

부

71 111 78 123 84 49 109 69 95

66 52 115 51 100 95 53 81 76 95

73 110 106 51 99 55 105 48 110

95 119 73 55 104 95 80 99 52

112 125

< 텍스트에 아스키

[사진 6] 순서대로 나열한 flag 값을 이진 번역기에 넣어 텍스트로 변환한 모습

순서대로 나열한 flag 값을 이진 번역기에 넣어 텍스트로 변환하면 일반적인 flag의 형태를 얻을 수 있었습니다. 이를 정답란에 입력하니 정답이라는 문구가 출력되었습니다.

5. Flag

GoN{T1mE_B4s3d_5QL_Inj3c7i0n_wl7h_Pc4p}

6. 별도 첨부

패킷 번호	플래그 값	순서	패킷 번호	플래그 값	순서
619 – 623	71	1	40195 – 40197	106	22
5391 – 5393	95	19	41875 – 41877	115	12
8251 – 8253	111	2	46255 – 46257	73	20
9891 – 9894	66	10	47225 – 47227	125	39
10162 – 10164	109	7	50205 – 50207	99	36
11743 – 11745	110	28	51455 – 51457	55	25
15623 – 15625	49	6	56275 – 56277	100	14
18013 – 18015	95	9	56395 – 56397	84	5
18863 – 18865	95	15	62915 – 62917	95	34
19253 – 19255	53	16	65075 – 65078	69	8
22943 – 22945	119	30	67617 – 67622	110	21
26163 – 26165	104	33	73860 – 73863	81	17
26333 – 26336	55	32	78781 – 78783	80	35
26504 – 26506	123	4	84381 – 84384	78	3
29235 – 29237	73	31	85362 – 85365	95	29
29615 – 29617	51	23	88993 – 88995	112	38
34365 – 34367	51	13	89683 – 89685	48	27
36775 – 36777	105	26	96193 – 96195	99	24
37905 – 37907	76	18	99453 – 99455	52	37
38595 – 38597	52	11			

[표 1] wireshark를 이용하여 찾은 플래그 값들 목록

7. Reference

- [URL]