# [Dream Zoo]Write-Up

작성자	심주완
분석 일자	2024.05.16
작성 일자	2024.05.16
분석 대상	zoo.pcap
문서 버전	3
작성자 E-mail	rd002@naver.com





## 0. 목차

1.	문제	3
2.	분석 도구	3
3.	환경	3
4.	Write-Up	4
5.	Flag	8
6.	별도 첨부	9
7.	Reference	10



### 1. 문제

URL	https://dreamhack.io/wargame/challenges/1205
문제 내용	드림핵 공간에는 엄청난 동물원이 있어요. 주어진 pcap 파일을 분석하여 동물원으로 가보세요! 동물원에서 많은 동물들을 살펴보고 플래그를 찾으세요!
문제 파일	zoo.pcap
문제 유형	Network forensics
난이도	2 / 3

## 2. 분석 도구

도구명	다운로드 링크	Version
wireshark	https://www.wireshark.org/	4.2.5
OpenStego	https://github.com/syvaidya/openstego/releases	0.8.6

## 3. 환경

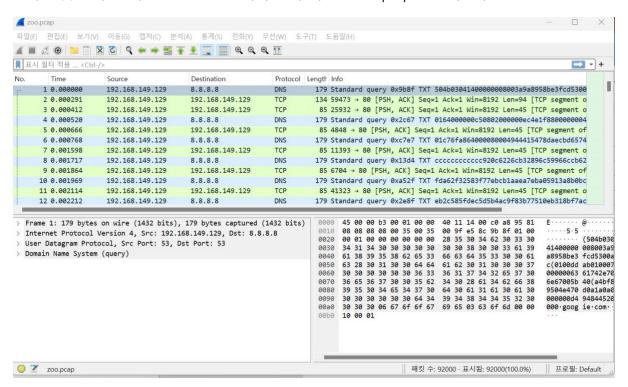
OS	
Window 11 Home	



### 4. Write-Up

파일명	zoo.pcap
용량	13.1 MB
SHA256	011d77a04cf6c313c7eb80ac8639f07ce2f41632c18c10cff5489bacc7792d93
Timestamp	2024-04-10 06:20:46

네트워크 포렌식 문제는 처음이라 쉽지가 않았다. 역시 실전에서 많이 배우는구나 하고 느꼈던 문제인 것 같다. 먼저 wireshark 를 통하여 문제 파일인 zoo.pcap 를 열어보자.



[사진 1] WireShark를 통하여 연 zoo.pcap

다음과 같이 확인할 수 있는데, DNS를 확인해보면 계속 TXT 쿼리문을 날리고 있다. 자세히 살펴보자.



#### [WHS-2] .iso

```
■ Wireshark - 単刻 1 - zoo pcan
   Internet Protocol Version 4, Src: 192.168.149.129, Dst: 8.8.8.8
 > User Datagram Protocol, Src Port: 53, Dst Port: 53
 V Domain Name System (query)
      Transaction ID: 0x9b8f
    > Flags: 0x0100 Standard query
      Questions: 1
      Answer RRs: 0
      Authority RRs: 0
      Additional RRs: 0
   ∨ Oueries
      v 504b0304140000008003a9a8958be3fcd5300ac.0100ddab0100070000006361742e706e67005b40.a4bf89504e470d0a1a0a0000000d49484452000...
            [Name Length: 133]
                                                             ....5.5
      45 00 00 b3 00 01 00 00 08 08 08 08 08 00 35 00 35
                                 00 9f e5 8c 9b 8f 01 00
       00 01 00 00 00 00 00 00
34 31 34 30 30 30 30 30
                                28 35 30 34 62 30 33 30
30 30 38 30 30 33 61 39
                                                           ..... (504b030
41400000 008003a9
       61 38 39 35 38 62 65 33
                                 66 63 64 35 33 30 30 61
                                                            a8958be3 fcd5300a
 c(0100dd ab010007
                                                            00000063 61742e70
6e67005b 40(a4bf8
                                                           9504e470 d0a1a0a0
000000d4 94844520
000 goog ie com··
No.: 1 · Time: 0.000000 · Source: 192.168.149.129 · Destination: 8.8.8.8 · Protocol: DNS · Length....Oac.0100ddab0100070000006361742e706e67005b40.a4bf89504e470d0a1a0a000000d494844520000.googie.com
☑ 패킷 바이트 표시
                                                                                                                        닫기
                                                                                                                                  도움막
```

[사진 2] DNS 패킷

확실히 비정상적인 패킷을 보내고 있다. 의심되는 루트 도메인이 googie.com이라는 정보도 확인할 수 있다. 그렇다면 보내고 있는 정보를 종합하여 어떤 정보를 보내려고(유출하려고) 하는지 확인해볼 수 있다. PowerShell을 통하여 다음과 같은 명령문을 실행시켰다.

이 코드는 총 가지로 나눠진다.

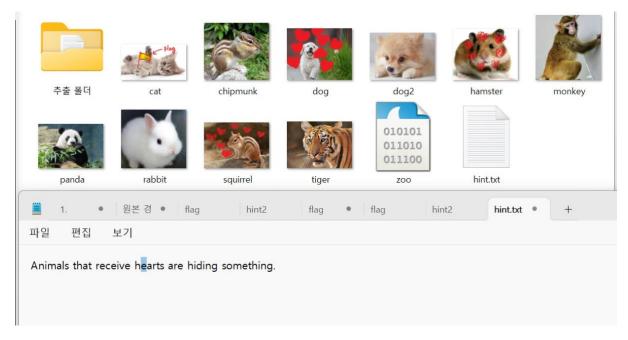
- 1. tshark를 통하여 zoo.pcap를 연다.
- 2. 플래그가 0(사용자가 보낸 형식) 과 쿼리 타입이 16(txt)인 부분만 검색한다.
- 3. 필드 타입으로 나누고, 쿼리의 name 부분(txt형식의 내용이 있는 부분)을 출력한다.
- 4. 출력문에서 Googie.com을 빼고 통합한다.

```
$ xxd -p -r dump > extract
$ file extract
```

다음 코드를 통하여 최종적으로 extract 파일을 만들 수 있었다. 압축을 풀고 내부를 확인하면,



#### [WHS-2] .iso



[사진 3] 추출 파일

다음과 같은 사진들을 확인할 수 있었다.(추출 폴더는 같이 추출된 파일이 아니다. 어떻게 생성하였는지는 뒤에 설명하겠다) 이 이유 때문에 Dream Zoo 라는 제목을 사용했구나 싶었다. hint.txt 의 파일 안에는 다음과 같은 내용이 있었다. 하트가 있는 사진을 조사하라는 내용은 있었지만 어떤 방법을 통하여 조사를 해야할지 막막했다. 이에 대한 답은 TCP 패킷에서 확인할 수 있었다.

```
45 00 00 86 00 01 00 00 40 06 ce 1d c0 a8 95 81
                                                        E . . . . . . . . . . . . . . . . . .
c0 a8 95 81 e8 51 00 50 00 00 00 00 00 00 00
                                                        ·····Q·P ······
                                                        P· ···· GET /syv
  50 18 20 00 ab 0a 00 00
                            47 45 54 20 2f 73 <mark>7</mark>9
                                                        aidya/op enstego,
  61 69 64 79 61 2f 6f
                        70
                            65 6e 73 74 65 67 6f
  72 65 6c 65 61 73 65 73
                                                         releases /downloa
                             2f 64 6f 77 6e 6c 6f 61
  64 2f 6f 70 65 6e 73 74
                            65 67 6f 2d 30 2e 38 2e
                                                        d/openst ego-0.8.
                                                        6/Setup- OpenSteg
  36 2f 53 65 74 75 70 2d
                            4f 70 65 6e 53 74 65 67
  6f 2d 30 2e 38 2e 36 2e 65 78 65 20 48 54 54 50
                                                        o-0.8.6. exe HTTP
  2f 31 2e 31 0d 0a
                                                        /1.1…
```

[사진 4] TCP패킷 내용

OpenStego로 패킷을 날리는 과정을 확인할 수 있었고, 이는 사진 속 스테가노그래피를 확인할 수 있는 프로그램이었다. 이를 통해서 하트가 보이는 동물들의 사진을 OpenStego로 분석해보았다.





[사진 4] 추출 파일 내부

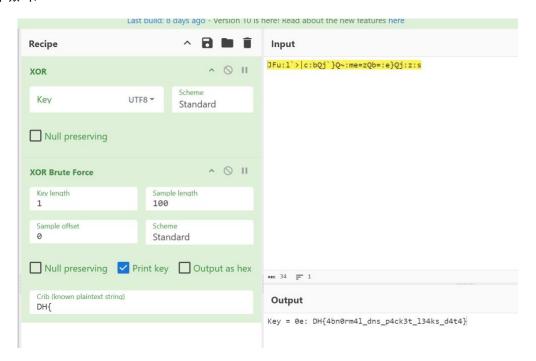
이를 전부 추출한 파일이 아까 확인하였던 추출 폴더이다. hint2.txt 의 파일의 내용은 다음과 같았다.

#### ( cat's hidden ) ( squirrel's hidden ) ( dog's hidden ) = FLAG

flag.txt 파일이 각 cat's hidden 이고, parrot 파일이 squirrel's hidden 이고 giraffe 파일이 dog's hidden 이다. 이에 해당하는 값을 집합시키면 플래그 값은

#### JFu:l`>|c:bQj`}Q~:me=zQb=:e}Qj:z:s XOR 14

를 연산한 값이 될 것이다. 나의 경우 giraffe에 해당하는 값을 14가 아닌 12로 놓고 풀어 계속답이 나오지 않았다. 결국 XOR Brutefoce tool을 사용해서 풀긴 했다. 사이트는 레퍼런스에넣어두었다.



[사진 5] flag



WhiteHat School

### 5. Flag

 $DH\{4bn0rm4l\_dns\_p4ck3t\_l34ks\_d4t4\}$ 



### 6. 별도 첨부

# [WHS-2] .iso



### 7. Reference

https://gchq.github.io/CyberChef/#recipe=XOR(%7B'option':'UTF8','string':''%7D,'Standard',false)XOR\_Brute\_Force(1,100,0,'Standard',false,true,false,'DH%7B')&input=SkZ1OmxgPnxjOmJRamB9UX46bWU9elFiPTplfVFqOno6cw