



# [Ann's Dark Tangent] Write-Up

작성자	박혜미
분석 일자	2024.05.17
작성 일자	2024.05.18
분석 대상	evidence-defcon2010.pcap
문서 버전	2.0
작성자 E-mail	<a href="mailto:parkm0708@naver.com">parkm0708@naver.com</a>

0. 목차

1. 문제 .....3

2. 분석 도구 .....3

3. 환경 .....3


4. Write-Up.....4

5. Flag..... 10

6. 별도 첨부 ..... 11

7. Reference ..... 12

### 1. 문제

URL	<a href="https://forensicscontest.com/2011/07/31/puzzle-7-anns-dark-tangent-defcon-2010">https://forensicscontest.com/2011/07/31/puzzle-7-anns-dark-tangent-defcon-2010</a>
문제 내용	Ann은 Dark Tangent와의 만남을 주선했습니다. 당신은 법의학 수사관입니다. 그들의 목적지를 알아낼 수 있나요? 네트워크 트래픽 사본은 다음과 같습니다. <a href="#">증거-defcon2010.pcap</a> MD5sum: 7c416421a626600f86e3702df0cac8ef
문제 파일	 evidence-defcon2010.pcap
문제 유형	네트워크 포렌식
난이도	2 / 3

### 2. 분석 도구

도구명	다운로드 링크	Version
Wireshark	<a href="https://www.wireshark.org/download.html">https://www.wireshark.org/download.html</a>	4.2.4
Aircrack-ng	<a href="https://www.aircrack-ng.org/">https://www.aircrack-ng.org/</a>	1.7
HxD	<a href="https://mh-nexus.de/en/hxd/">https://mh-nexus.de/en/hxd/</a>	2.5

### 3. 환경

OS
Windows 11 Home

4. Write-Up

파일명	evidence-defcon2010.pcap
용량	62.2MB
SHA256	680cd7437f6d985e7456349a0e7bd299cd21cacfd7271431b97549265241a541
Timestamp	2024-05-17 15:07:32

Time	Source	Destination	Protocol	Length	Info
1 0.000000		Cisco_3d:fb:70 (00:...	802.11	10	Clear-to-send, Flags=.....
2 0.000000		Cisco_3d:fb:70 (00:...	802.11	10	Clear-to-send, Flags=.....
3 0.000000		Cisco_3d:fb:70 (00:...	802.11	10	Clear-to-send, Flags=.....
4 0.000000		Cisco_3d:fb:70 (00:...	802.11	10	Clear-to-send, Flags=.....
5 0.000000		Cisco_3d:fb:70 (00:...	802.11	10	Clear-to-send, Flags=.....
6 0.000000		Cisco_3d:fb:70 (00:...	802.11	10	Clear-to-send, Flags=.....
7 0.000068	CiscoLinksys_b3:cc:...	Broadcast	802.11	110	Beacon frame, SN=2064, FN=0, Flags=....., BI=100, SSID="w00t"
8 -0.000001		Cisco_3d:fb:70 (00:...	802.11	10	Clear-to-send, Flags=.....
9 0.003073		Cisco_3d:fb:70 (00:...	802.11	10	Clear-to-send, Flags=.....
10 0.006144		Cisco_3d:fb:70 (00:...	802.11	10	Clear-to-send, Flags=.....
11 0.009728		Cisco_3d:fb:70 (00:...	802.11	10	Clear-to-send, Flags=.....
12 0.012288		Cisco_3d:fb:70 (00:...	802.11	10	Clear-to-send, Flags=.....
13 0.016384	Cisco_3d:fb:70 (00:...	Cisco_b9:a4:10 (00:...	802.11	16	Request-to-send, Flags=.....

[그림 1] Sans 3, 6과 다른 화면

문제 파일을 열어보니 Sans 3, 6 번과 다른 화면의 Wireshark 가 보인다. 802.11 프로토콜에 대하여 검색하니 이는 무선랜 환경에서의 패킷이 암호화된 것이라고 한다. 따라서 해당 문제에서는 이 암호화된 패킷을 복호화 해야 한다.

```
fopen failed: No such file or directory          Got 98923 out of 95000 IVsStarting PTW attack with 98923 ivs.
KEY FOUND! [ 4A:7D:B5:08:CD ]
Decrypted correctly: 100%
C:\Users\LG\OneDrive\문서\화이트햇 2기\프로젝트\Write-Up\2w\백해미\sansCTF - 7>aircrack-ng.exe evidence-defcon2010.pcap
Reading packets, please wait...
C:\Users\LG\OneDrive\문서\화이트햇 2기\프로젝트\Write-Up\2w\백해미\sansCTF - 7>
Read 426642 packets.

# BSSID      ESSID      Encryption
1 00:1C:10:B3:CC:F0  w00t      WEP (98923 IVs)

Choosing first network as target.
Reading packets, please wait...
Opening evidence-defcon2010.pcap
Read 426642 packets.

1 potential targets
Attack will be restarted every 5000 captured ivs.
```

[그림 2] aircrack-ng.exe evidence-defcon2010.pcap

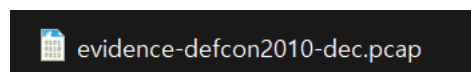
aircrack-ng 를 사용하여 무선랜 패킷을 분석해 보는데 키를 찾았다며 [ 4A:7D:B5:08:CD ] 라는 키 값이 보인다.

## [WHS-2] .iso

```
C:\Users\LG\Downloads\aircrack-ng-1.7-win\aircrack-ng-1.7-win\bin>airdecap-ng.exe -w 4A:7D:B5:08:CD evidence-defcon2010.pcap
Total number of stations seen      3
Total number of packets read      426642
Total number of WEP data packets  187650
Total number of WPA data packets  0
Number of plaintext data packets  0
Number of decrypted WEP packets   187650
Number of corrupted WEP packets   0
Number of decrypted WPA packets   0
Number of bad TKIP (WPA) packets  0
Number of bad CCMP (WPA) packets  0
```

[그림 3] airdecap-ng.exe -w 4A:7D:B5:08:CD evidence-defcon2010.pcap

airdecap-ng 를 사용하여 패킷을 복호화한다.



[그림 4] evidence-defcon2010-dec.pcap 생성

-dec 가 붙은 복호화 된 파일이 생성된다.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.5.5.113	205.188.250.75	IMAP	75	Request: 37 NOOP
2	0.205334	CiscoLinksys_b3:cc:...	Broadcast	ARP	42	Who has 10.5.5.113? Tell 10.5.5.1
3	0.207357	Apple_3b:4e:52	CiscoLinksys_b3:cc:...	ARP	42	10.5.5.113 is at d8:a2:5e:3b:4e:52
4	0.207398	Intel_05:a3:08	Broadcast	ARP	42	Who has 10.5.5.113? Tell 10.5.5.1
5	0.211476	205.188.250.75	10.5.5.113	IMAP	88	Response: 37 OK NOOP completed
6	0.229398	Intel_05:a3:08	Broadcast	ARP	42	Who has 10.5.5.113? Tell 10.5.5.1
7	0.229372	10.5.5.113	205.188.250.75	TCP	66	50045 → 143 [ACK] Seq=10 Ack=23 Win=32826 Len=0 TSval=787815491 TSecr=3518285019
8	0.231974	Intel_05:a3:08	Broadcast	ARP	42	Who has 10.5.5.113? Tell 10.5.5.1
9	0.231974	Intel_05:a3:08	Broadcast	ARP	42	Who has 10.5.5.113? Tell 10.5.5.1

[그림 5] 복호화 된 evidence-defcon2010-dec.pcap

정상적으로 출력되는 것을 확인할 수 있다. 여기서 IMAP 프로토콜은 이메일 관련 프로토콜이라고 한다. 따라서 아마 두 사람 간의 전자이메일의 내용이 있을 텐데 그 내용에 약속 장소가 있을 것 같다.

```
U1WZ4m1B3JmCmSy30f84f1f0a1qf1000327XLB1f2V0aV0XpGcZ1RNCe10f3ERX00W1Qj7f0
oS4Pb/x84EJmekXwEQ5kdKIRUejBDCJAqrKbvewZQMXVWU5wrUfEH+poRR16QIGyVjahmpjAAHzB
ZQggABRNdBT8JE0c5ijG6A4wxEq0KU0oAEPKDCHzvQim1M6fKYn9I20kEACAEii74Ltwv0WI6X
tKPP5ZhFJB7hCDr44AIf7RRVHhCENbzGqWGiD+WAlA5QKQbEZjKAFBwCqWl/ARsYc6QMj85pvj
GLfYAxrKIAMUWP/62EfBCSpGFRcs4RbkyLH/QfSxjnH8YhBOMEED7D4VArBIARYQQh1QQUoLM8QO
ANGBRNLhAQDQPYJ0AFGgAVYIAaQsDDjkIAKOA7PwAmPsAUEiAU6gAEWYAEXcAEVeIF6d3dUQQHB
p08ZAAV80E1/kAURGIgFsAER0HhTIQE4UARFoAR2QAv08AzQkA72BxGVAAAZgDcNgQ83AACFQA1n
wAEHkE/6FAC30wBM2IT4hIRbAVzARQAPsAEwEAVT4AeokApc2IVem[1368 bytes missing in capture file].1BKHFJXB
uEP+dvq/rohNxoLmBAFH8ABG8AA7AQcXgCulcKv/1pX7AoR7joh
gCADhTEVBZAAQABPbAEHvuxUuAH95EiIsAHUfCkKJuyKKsELsCCVDEALlgBAOAAewA08v+Yi/tw
D9swDaqQCGSgsigLBZ3HohBhAgDwNfRADq1wBRdwAAVQAC7LgYEVgCRABaFwC7hwDOjgHC4AFRMA
```

[그림 6] IMAP의 TCP 스트림

IMAP 에서 TCP 스트림을 들어가 봤는데 이곳에 “missing in capture file”이 있는 것을 보아 송수신 간에 문제가 있는 것 같다.

## [WHS-2] .iso

```
--Apple-Mail-2-875303692
Content-Disposition: inline;
    filename=IMG_0002.GIF
Content-Type: image/gif;
    name=IMG_0002.GIF
Content-Transfer-Encoding: base64

R0lGODlhkACdAPcAAAAAAEBAQICAgMDAwQEBAUFbQYGBGcHBwgICAKJCQoKCgsLCwwMDA0NDQ4O
Dg8PDxAQEBERERISEhMTExQUFBUVFRYWfHcXfXgYGBkZGRoaGhsbGxwchB0dHR4eHh8fHyAgICEh
ISiIiMjIyQkJCULJSYmJicnJygoKCKpKSoqKisrKywsLC0tLS4uLi8vLzAwMDEMTIyMjMzMQ0
NDU1NTY2Njc3Nzg4ODk5OTk5OTk5OTk5OTk5OTk5OTk5OTk5OTk5OTk5OTk5OTk5OTk5OTk5OTk5
R0hISElJSUpKSkLs0xMTE1NTU5OTk5OTk5OTk5OTk5OTk5OTk5OTk5OTk5OTk5OTk5OTk5OTk5OT
WltbW1xcXF1dXV5eXl9fX2BgYGFhYWJiYmNjY2RkZGVlZWZmZmdnZ2hoaGlpaWpqaamtra2xsbG1t
bW5ubm9vb3BwcHFxcXJycnNzc3R0dHV1dXZ2dnd3d3h4eHl5eXp6ent7e3x8fH19fX5+fn9/f4CA
gIGBgYKCGoODg4SEhIWFhYaGhoeHh4iIiImJiYqKiouLi4yMjI2NjY6Ojo+Pj5CQkJGRkZKSkpOT
k5SULjWVlZaWlpeXl5iYmJmZmZqampubm5ycnJ2dnZ6enp+fn6CgoKGhoaKioqOjo6SkpKWlpaam
pqpnp6ioqKmpqaqqqurq6ysrK2tra6urq+vr7CwsLGxsbKysrOzs7S0tLW1tba2tre3t7i4uLm5
ubq6uru7u7y8vL29vb6+vr+/v8DAwMHBwCLCwsPDw8TEwMXFxcGxsfHx8jIyMnJycrKysvLy8zM
zMNzc7Ozs/Pz9DQ0NHR0dLS0tPT09TU1NXV1dbW1tfX19jY2NnZ2dra2tvb29zc3N3d3d7e3t/f
3+Dg4OHh4eLi4uPj4+Tk5OX15ebm5ufn5+jo6Onp6erq6uvr6+zs703t7e7u7u/v7/Dw8PHx8fLy
```

[그림 7] SMTP의 TCP 스트림

따라서 SMTP의 TCP 스트림을 살펴봤는데 이곳엔 수신 실패가 일어나지 않았다. 해당 문자는 Base64로 인코딩되어 있기에 디코딩을 진행했다.

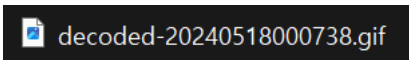
Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	52	30	6C	47	4F	44	6C	68	6B	41	43	64	41	50	63	41	R0lGODlhkACdAPcA
00000010	41	41	41	41	41	41	45	42	41	51	49	43	41	67	4D	44	AAAAAAEBAQICAgMD
00000020	41	77	51	45	42	41	55	46	42	51	59	47	42	67	63	48	AwQEBAUFbQYGBGcH
00000030	42	77	67	49	43	41	6B	4A	43	51	6F	4B	43	67	73	4C	BwgICAKJCQoKCgsL
00000040	43	77	77	4D	44	41	30	4E	44	51	34	4F	0D	0A	44	67	CwwMDA0NDQ4O..Dg
00000050	38	50	44	78	41	51	45	42	45	52	45	52	49	53	45	68	8PDxAQEBERERISEh
00000060	4D	54	45	78	51	55	46	42	55	56	46	52	59	57	46	68	MTExQUFBUVFRYWfH
00000070	63	58	46	78	67	59	47	42	6B	5A	47	52	6F	61	47	68	cXfXgYGBkZGRoaGh
00000080	73	62	47	78	77	63	48	42	30	64	48	52	34	65	48	68	sbGxwchB0dHR4eHh
00000090	38	66	48	79	41	67	49	43	45	68	0D	0A	49	53	49	69	8fHyAgICEh..ISiI
000000A0	49	69	4D	6A	49	79	51	6B	4A	43	55	6C	4A	53	59	6D	IiMjIyQkJCULJSYm
000000B0	4A	69	63	6E	4A	79	67	6F	4B	43	6B	70	4B	53	6F	71	JicnJygoKCKpKSoq
000000C0	4B	69	73	72	4B	79	77	73	4C	43	30	74	4C	53	34	75	KisrKywsLC0tLS4u
000000D0	4C	69	38	76	4C	7A	41	77	4D	44	45	78	4D	54	49	79	Li8vLzAwMDEMTIy
000000E0	4D	6A	4D	7A	4D	7A	51	30	0D	0A	4E	44	55	31	4E	54	MjMzMQ0..NDU1NT
000000F0	59	32	4E	6A	63	33	4E	7A	67	34	4F	44	6B	35	4F	54	Y2Njc3Nzg4ODk5OT
00000100	6F	36	4F	6A	73	37	4F	7A	77	38	50	44	30	39	50	54	o6Ojs7Ozw8PD09PT

[그림 8] HxD

해당 내용을 HxD에 넣어 저장한 뒤 디코딩해준다.

(사용 사이트: <https://www.base64decode.org/>)

[WHS-2] .iso



[그림 9] 디코딩

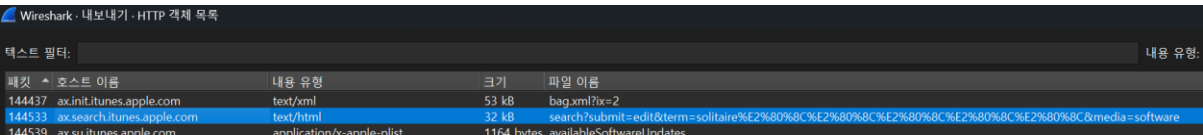
복호화 하니 .gif 파일이 하나 생성된다.



[그림 10] decoded -20240518000738.gif

파일을 열어보니 App Store – App Name, Podcast Title, YouTube Video Title, Google Earth City Name, AIM Buddy Name 이 적혀 있다.

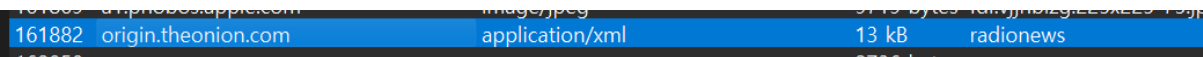
1. App Store – App Name



[그림 11] Http 객체 목록 1

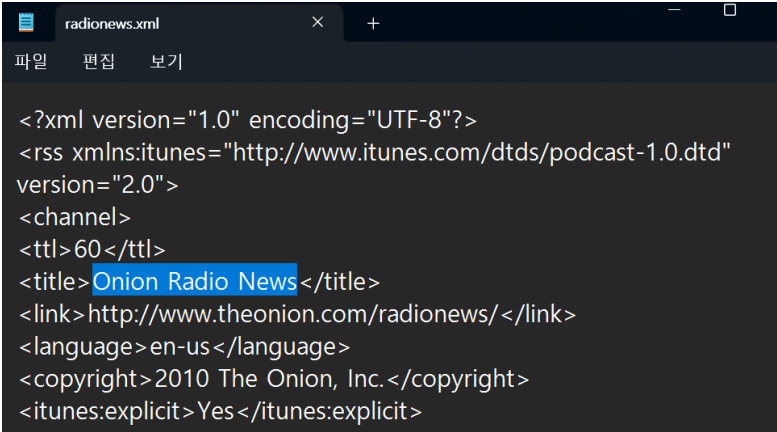
Http 객체를 확인하면 App Store 는 itunes 이고 **solitaire** 라는 앱을 검색한 것을 찾을 수 있다.

2. Podcast Title



[그림 12] Http 객체 목록 2

Http 객체를 확인해 보니 podcast 와 관련 있어 보이는 .xml 파일을 찾을 수 있다. 해당 파일을 저장한다.

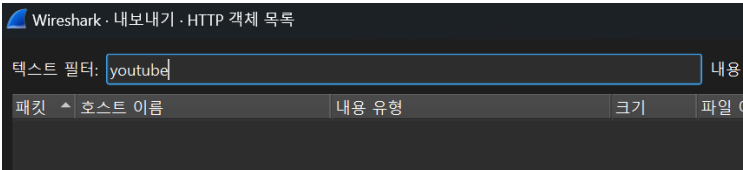


```
<?xml version="1.0" encoding="UTF-8"?>
<rss xmlns:itunes="http://www.itunes.com/dtds/podcast-1.0.dtd"
version="2.0">
<channel>
<ttl>60</ttl>
<title>Onion Radio News</title>
<link>http://www.theonion.com/radionews/</link>
<language>en-us</language>
<copyright>2010 The Onion, Inc.</copyright>
<itunes:explicit>Yes</itunes:explicit>
</channel>
</rss>
```

[그림 13] radionews

파일 안에 **Onion Radio News** 라는 podcast 제목을 찾을 수 있다.

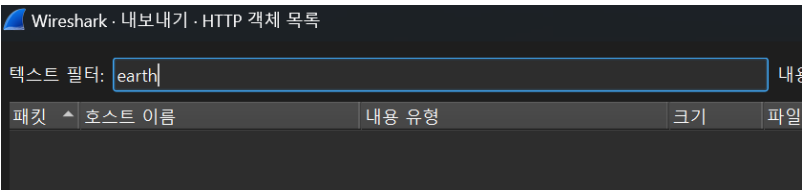
3. YouTube Video Title



[그림 14] 아무것도 뜨지 않는 Youtube

Http 객체에서 youtube 를 찾아보았지만 해당 내용이 존재하지 않았다. 인터넷에 찾아보니 **Cry For Help** 라는 제목이었다.

4. Google Earth City Name



[그림 15] 아무것도 뜨지 않는 Earth

이것 또한 Http 객체에서 찾아보려 했으나 해당 내용이 존재하지 않았다. (왜 나만.) 이것 또한 인터넷에 찾아보니 **Hacker Valley** 라고 한다. (아무래도 파일이 조금 깨진 것 같다.)



5. AIM Buddy Name

Wireshark · 내보내기 · HTTP 객체 목록

텍스트 필터: aim

패킷	호스트 이름	내용 유형	크기	파일
187548	205.188.193.215	application/json	447 bytes	fetch
187582	api.aim.net	application/json	586 bytes	get
187584	api.aim.net	application/json	106 bytes	get
187606	api.aim.net	application/json	181 bytes	send
187609	api.aim.net	application/json	100 bytes	send
187618	205.188.193.215	application/json	487 bytes	fetch
187629	205.188.193.215	application/json	610 bytes	fetch
187646	api.aim.net	application/json	100 bytes	send

[그림 16] aim

Http 객체에서 aim 을 검색해 보았더니 aim 메신저를 주고받을 때 json 파일을 사용하고 있다.

1876	859.361488	10.5.5.113	64.12.236.203	HTTP	450 GET /im/sendIM?aimsid=001.1035016094.1397756565%3Asneakyg33k&autoResponse=false&comscoreChannel=&displa
Transmission Control Protocol, Src Port: 50321, Dst Port: 80, Seq: 398, Ack: 444, Len: 396					
Hypertext Transfer Protocol					
GET /im/sendIM?aimsid=001.1035016094.1397756565%3Asneakyg33k&autoResponse=false&comscoreChannel=&displa					
[ [truncated]Expert Info (Chat/Sequence): GET /im/sendIM?aimsid=001.1035016094.1397756565%3Asneakyg33k&autoResponse=false&comscoreChannel=&displa					
[GET /im/sendIM?aimsid=001.1035016094.1397756565%3Asneakyg33k&autoResponse=false&comscoreChannel=&displa					
[Severity level: Chat]					
[Group: Sequence]					
Request Method: GET					
Request URI: /im/sendIM?aimsid=001.1035016094.1397756565%3Asneakyg33k&autoResponse=false&comscoreChannel=&displa					
Request URI Path: /im/sendIM					
Request URI Query: aimsid=001.1035016094.1397756565%3Asneakyg33k&autoResponse=false&comscoreChannel=&displa					
Request URI Query Parameter: aimsid=001.1035016094.1397756565%3Asneakyg33k					
Request URI Query Parameter: autoResponse=false					
Request URI Query Parameter: comscoreChannel=					
Request URI Query Parameter: displaySMSegmentData=false					
Request URI Query Parameter: f=json					
Request URI Query Parameter: k=ip1NaUE17G7o_OHY					
Request URI Query Parameter: message=Do%20you%20have%20the%20cash%3F					
Request URI Query Parameter: offlineIM=true					
Request URI Query Parameter: r=33					
Request URI Query Parameter: t=interOptic					

[그림 17] aim buddy id

해당 패킷의 상세 내용을 보니 aim buddy id 로 추정되는 **interOptic** 가 존재한다.



## 5. Flag

1. solitaire
2. Onion Radio News
3. Cry For Help
4. Hacker Valley
5. inter0pt1c

sochi



## 6. 별도 첨부

## 7. Reference

- [URL]