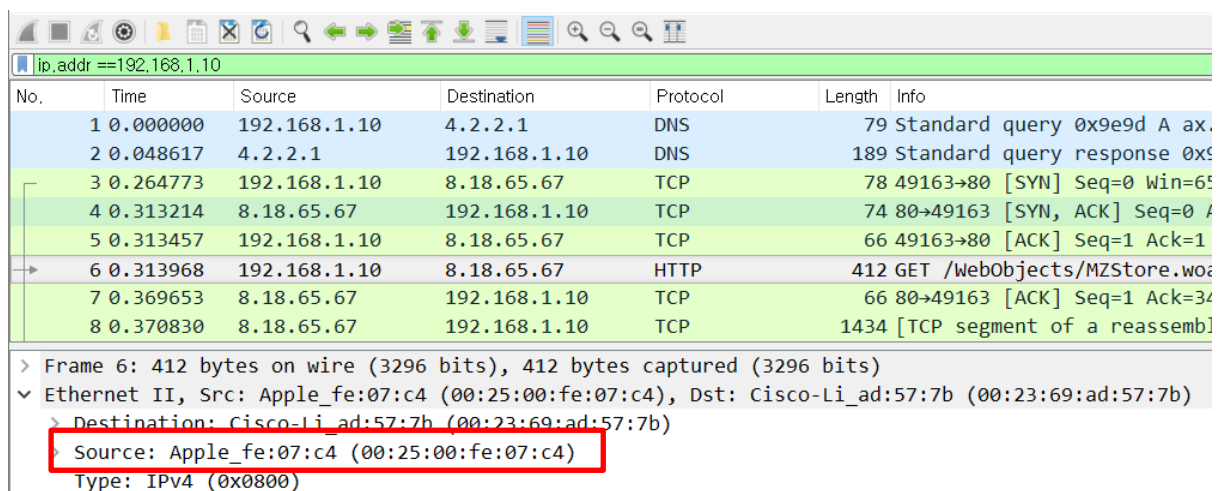


Puzzle #3: Ann's AppleTV

<http://forensicscontest.com/2009/12/28/anns-appletv>

1. What is the MAC address of Ann's AppleTV?

우선 Ann은 AppleTV의 IP를 192.168.1.10으로 정적으로 받아 사용한다고 하였다. 따라서 `ip.addr == 192.168.1.10`으로 필터링해주었다.



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.10	4.2.2.1	DNS	79	Standard query 0x9e9d A ax.
2	0.048617	4.2.2.1	192.168.1.10	DNS	189	Standard query response 0xs
3	0.264773	192.168.1.10	8.18.65.67	TCP	78	49163→80 [SYN] Seq=0 Win=65
4	0.313214	8.18.65.67	192.168.1.10	TCP	74	80→49163 [SYN, ACK] Seq=0
5	0.313457	192.168.1.10	8.18.65.67	TCP	66	49163→80 [ACK] Seq=1 Ack=1
6	0.313968	192.168.1.10	8.18.65.67	HTTP	412	GET /WebObjects/MZStore.woa
7	0.369653	8.18.65.67	192.168.1.10	TCP	66	80→49163 [ACK] Seq=1 Ack=34
8	0.370830	8.18.65.67	192.168.1.10	TCP	1434	[TCP segment of a reassembl

> Frame 6: 412 bytes on wire (3296 bits), 412 bytes captured (3296 bits)
v Ethernet II, Src: Apple_fe:07:c4 (00:25:00:fe:07:c4), Dst: Cisco-Li_ad:57:7b (00:23:69:ad:57:7b)
 > Destination: Cisco-Li_ad:57:7b (00:23:69:ad:57:7b)
 > Source: Apple_fe:07:c4 (00:25:00:fe:07:c4)
 Type: IPv4 (0x0800)

패킷에서 2계층인 Ethernet부분을 보면 보내는 쪽(source)에서 AppleTV의 MAC주소를 알 수 있다.

Ann의 AppleTV의 맥주소는 00:25:00:fe:07:c4이다.

2. What User-Agent string did Ann's AppleTV use in HTTP requests?

No.6 번의 info에 GET이라는 request method를 보면 http 요청하는 패킷이라는 것을 알 수 있다. HTTP계층을 보면, User-Agent를 바로 확인 할 수 있다.

No.	Time	Source	Destination	Protocol	Length	Info
4	0.313214	8.18.65.67	192.168.1.10	TCP	74	80→49163 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 M...
5	0.313457	192.168.1.10	8.18.65.67	TCP	66	49163→80 [ACK] Seq=1 Ack=1 Win=65535 Len=0 TSval...
6	0.313968	192.168.1.10	8.18.65.67	HTTP	412	GET /WebObjects/MZStore.woa/wa/viewGrouping?id=3...
7	0.369653	8.18.65.67	192.168.1.10	TCP	66	80→49163 [ACK] Seq=1 Ack=347 Win=6432 Len=0 TSva...
8	0.370820	8.18.65.67	192.168.1.10	TCP	1434	[TCP segment of a reassembled PDU]

> Frame 6: 412 bytes on wire (3296 bits), 412 bytes captured (3296 bits)
 > Ethernet II, Src: Apple_fe:07:c4 (00:25:00:fe:07:c4), Dst: Cisco-Li_ad:57:7b (00:23:69:ad:57:7b)
 > Internet Protocol Version 4, Src: 192.168.1.10, Dst: 8.18.65.67
 > Transmission Control Protocol, Src Port: 49163, Dst Port: 80, Seq: 1, Ack: 1, Len: 346
 > Hypertext Transfer Protocol
 > GET /WebObjects/MZStore.woa/wa/viewGrouping?id=39 HTTP/1.1\r\n
 Accept: */*\r\n
 Accept-Language: en\r\n
 Accept-Encoding: gzip, deflate\r\n
 Cookie: s_vi=[CS]v1|259c176a85010c29-6000010d80115d7f[CE]\r\n
 User-Agent: AppleTV/2.4\r\n
 If-Modified-Since: Fri, 25 Dec 2009 04:42:31 GMT\r\n
 X-Apple-Store-Front: 143441-1 3\r\n

User-Agent는 AppleTV/2.4이다.

3. What were Ann's first four search terms on the AppleTV (all incremental searches count)?

385	GET	/WebObjects/MZSearch.woa/wa/incrementalSearch?media=movie&q=h	HTTP/1.1
-----	-----	---	----------

Method Path Version of Protocol

위의 사진은 No. 43의 패킷 info를 확인한 것이다.

HTTP의 Request 헤더의 첫줄에는 Method, Path, Version of Protocol이 들어간다. 여기서 Path를 보면 사용자가 무엇을 검색한지 알 수 있다.

> [Expert Info (Chat/Sequence): GET /WebObjects/MZSearch.woa/wa/incrementalSearch?media=movie&q=h HTTP/1.1\r\n] Request Method: GET Request URI: /WebObjects/MZSearch.woa/wa/incrementalSearch?media=movie&q=h Request URI Path: /WebObjects/MZSearch.woa/wa/incrementalSearch Request URI Query: media=movie&q=h Request URI Query Parameter: media=movie Request URI Query Parameter: q=h

패킷의 HTTP부분을 확인해보면 Path부분에서 ?전까지는 경로를 나타내고, 그 뒤에는 요청한 자원의 이름이 들어간다. Search?는 보통 검색엔진에서 검색할 때 많이 보인다. 또한, 대부분 q= 또는 query=뒤에 있는 내용이 사용자가 입력한 값이 된다.

추가로 모든 증분검색을 포함하여 4개를 찾으라고 하였다. 증분검색이란 사용자가 글자를 입력하는 도중에 계속적으로 해당하는 내용을 찾아주는 기능이

다.

```
385 GET /WebObjects/MZSearch.woa/wa/incrementalSearch?media=movie&q=h HTTP/1.1
386 GET /WebObjects/MZSearch.woa/wa/incrementalSearch?media=movie&q=ha HTTP/1.1
387 GET /WebObjects/MZSearch.woa/wa/incrementalSearch?media=movie&q=hac HTTP/1.1
388 GET /WebObjects/MZSearch.woa/wa/incrementalSearch?media=movie&q=hack HTTP/1.1
```

따라서 ip.addr==192.168.1.10 && http로 필터링 하고 이 형태를 순서대로 찾으면 Ann이 hack을 검색한 사실을 알 수 있고 증분검색이 되었기 때문에 한글자씩 요청해 4개의 요청 패킷이 보내진 것을 볼 수 있다.

4. What was the title of the first movie Ann clicked on?

```
375 GET /WebObjects/MZStore.woa/wa/viewMovie?id=333441649&s=143441 HTTP/1.1
697 HTTP/1.1 200 OK
407 GET /WebObjects/MZStore.woa/wa/relatedItemsShelf?ct-id=3&id=333441649&storeFrontId=143441&mt=6 HTTP/1.1
573 GET /b/ss/applesuperglobal/1/G.6--NS?pageName=Movie%20Page-US-Hackers-Iain%20Softley-333441649&pccr=true&h5=app...
642 HTTP/1.1 200 OK (GIF89a) (GIF89a) (image/gif)
729 HTTP/1.1 200 OK
423 GET /us/r1000/032/Video/f0/48/dd/mzi.pizbdeal.enc.jpg?downloadKey2=1265245618_f3a714a27ea9388f7c07104353e1d763 ...
348 HTTP/1.1 200 OK (image/jpeg)
373 GET /us/r1000/038/Music/2e/10/15/mzi.qdnwlnpu.170x170-75.jpg HTTP/1.1
```

ip.addr == 192.168.1.10 && http로 필터링하고 info를 보다가 ViewMovie부분(No.307)이 수상한 것을 볼 수 있었다. HTTP Stream을 확인해보았다. 또 그 밑에 downloadkey2(No.331)를 보고 영화를 다운받는다는 것을 추측하였다.

Viewmovie부분(No.307)의 HTTP Stream을 확인하였다.

```
<key>title</key><string>Hackers</string>
<key>store-version</key><string>1.0</string>
<key>pings</key>
<array>
  <string>http://metrics.apple.com/b/ss/applesuperglobal/1/G.6--NS?pageName=Movie%20Page-US-Hackers-Iain%20Softley-333441649&pccr=true&h5=appleitmsnatv%2Cappleitmsustv&ch=Movie%20Page&g=http%3A%2F%2Fax.itunes.apple.com%2FWebObjects%2FMZStore.woa%2Fwa%2FviewMovie%3Fid%3D333441649%26s%3D143441</string>
</array>
<key>items</key>
<array>
  <dict>
    <key>type</key><string>movie</string>
    <key>title</key><string>Hackers</string>
    <key>unmodified-title</key><string>Hackers</string>
    <key>item-id</key><integer>333441649</integer>
    <key>url</key><string>http://ax.itunes.apple.com/WebObjects/MZStore.woa/wa/viewMovie?id=333441649&s=143441</string>
    <key>release-date</key><date>1998-08-25T07:00:00Z</date>
    <key>genre-name</key><string>Drama</string>
```

Hackers라는 타이틀의 영화를 본 것을 알 수 있다.

5. What was the full URL to the movie trailer (defined by “preview-url”)?

4번에서 본 HTTP Stream에서 preview-url을 검색했다.

```
<key>preview-url</key><string>http://a227.v.phobos.apple.com/us/r1000/008/Video/62/bd/1b/mzm.plqacyqb..640x278.h264lc.d2.p.m4v</string>
```

movie trailer 전체의 url을 찾을 수 있었다.

6. What was the title of the second movie Ann clicked on?

4번과 같은 방식으로 찾았다.

```
<dict>
  <key>type</key><string>movie</string>
  <key>title</key><string>Sneakers</string>
  <key>unmodified-title</key><string>Sneakers</string>
```

두번째 영화는 Sneakers이다.

7. What was the price to buy it (defined by “price-display”)?

6번에서 본 HTTP Stream에서 price-display를 검색하였다.

```
<key>price</key><real>9.99000</real>
<key>price-display</key><string>$9.99</string>
```

가격은 \$9.99인 것을 확인할 수 있다.

8. What was the last full term Ann searched for?

3번과 같은 방법으로 찾았다. 이번엔 밑에서부터 찾았더니 밑에서 4번째인 No.1766번에서 답을 찾을 수 있었다.

1766	158.251617	192.168.1.10	8.18.65.89	HTTP	404 GET /WebObjects/MZSearch.woa/wa/incrementalS...
1769	158.362253	8.18.65.89	192.168.1.10	HTTP/XML	170 HTTP/1.1 200 OK
1771	158.371395	192.168.1.10	66.235.132.121	HTTP	627 GET /b/ss/applesuperglobal/1/G.6--NS?pcr=tr...
1772	158.420464	66.235.132.121	192.168.1.10	HTTP	643 HTTP/1.1 200 OK (GIF89a) (GIF89a) (image/gi...

[Expert Info (Chat/Sequence): GET /WebObjects/MZSearch.woa/wa/incrementalSearch?media=movie&q=iknowyourewatchingme
 [GET /WebObjects/MZSearch.woa/wa/incrementalSearch?media=movie&q=iknowyourewatchingme HTTP/1.1\r\n
 [Severity level: Chat]
 [Group: Sequence]

Request Method: GET

Request URI: /WebObjects/MZSearch.woa/wa/incrementalSearch?media=movie&q=iknowyourewatchingme
 Request URI Path: /WebObjects/MZSearch.woa/wa/incrementalSearch
 Request URI Query: media=movie&q=iknowyourewatchingme
 Request Version: HTTP/1.1

마지막으로 검색한 문장은 iknowyourewatchingme이다.