

작성자	박혜미
분석 일자	2024.05.16
작성 일자	2024.05.16
분석 대상	evidence06.pcap
문서 버전	1.0
작성자 E-mail	parkm0708@naver.com

0. 목차

1. 문제3

2. 분석 도구3

3. 환경3


4. Write-Up.....4

5. Flag..... 12

6. 별도 첨부 13

7. Reference 15

1. 문제

URL	https://forensicscontest.com/2010/05/21/puzzle-6-anns-aurora
문제 내용	<p>Ann Dercover는 SaucyCorp의 비밀 소스 레시피를 따릅니다. 그녀는 SaucyCorp의 서버에 원격으로 액세스할 수 있는 방법을 알아내기 위해 수석 개발자인 Vick Timmes를 추적하고 있습니다. 어느 날 밤 정찰을 수행하던 중 그녀는 그가 자신의 노트북 (10.10.10.70)에 로그인하고 SaucyCorp의 본사에 VPN을 연결하는 것을 목격했습니다. Ann은 국제 해킹 조직과의 연결을 활용하여 Internet Explorer에 대한 제로데이 익스플로잇을 획득 하고 Vick Timmes를 대상으로 클라이언트 측 스피어 피싱 공격을 시작합니다. Ann은 비밀 소스 레시피를 개선하는 방법에 대한 팁이 포함된 이메일을 Vick에게 신중하게 작성하여 보냅니다. Vick은 자신이 탐내던 제품 개발 부사장 직함(및 코너 사무실)을 얻을 수 있는 기회를 보고 링크를 클릭합니다. 앤은 공격할 준비가 되어 있습니다...</p> <p>당신은 법의학 수사관입니다. 귀하의 임무는 Ann의 공격이 포함된 패킷 캡처를 분석하고 , 타임라인을 구축하고, 다음을 포함한 증거를 제출하는 것입니다 .</p> <p>(문제가 너무 긴 관계로 원문은 [6. 별도 첨부]한다.)</p>
문제 파일	 evidence06.pcap
문제 유형	네트워크 포렌식
난이도	1 / 5

2. 분석 도구

도구명	다운로드 링크	Version
Wireshark	https://www.wireshark.org/download.html	4.2.4
WinMD5	https://www.winmd5.com/	1.20
NetworkMiner	https://www.netresec.com/?page=NetworkMiner	2.8

3. 환경

OS
Windows 11 Home

4. Write-Up

파일명	evidence03.pcap
용량	1.46MB
SHA256	fa5fc1ffad525688626c301372b37e101efcbbbd124f9781f5701648e6a02be3
Timestamp	2024-05-13 00:57:59

문제가 긴데, 요약하면 Ann 이 Vick 의 컴퓨터를 VPN 으로 접속하여 스피어 피싱 공격을 하려 하니 Ann 의 행동을 감시하라는 문제이다.

1.Vick Timmes 의 원래 웹 요청의 전체 URI 는 무엇이었나요? (URI 에 포트를 포함해 주세요.)

Source	Destination	Protocol	Length	Info
10.10.10.70	10.10.10.10	HTTP	351	GET /index.php HTTP/1.1

[그림 1] HTTP가 가장 먼저 요청

```
Hypertext Transfer Protocol
> GET /index.php HTTP/1.1\r\n
Accept: image/gif, image/x-bitmap, image/png\r\n
Accept-Language: en-us\r\n
Accept-Encoding: gzip, deflate\r\n
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; .NET CLR 2.0.50727)\r\n
Host: 10.10.10.10:8080\r\n
Connection: Keep-Alive\r\n
```

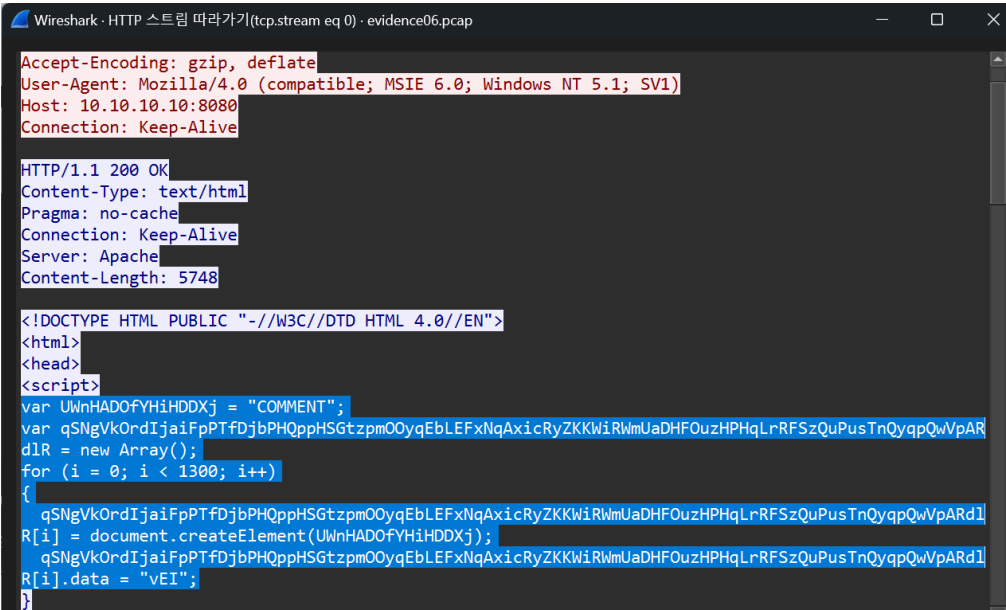
[그림 2] [그림 1]의 상세 정보

Vick Timmes 의 IP 는 10.10.10.70 임으로 해당 IP 에서 가장 먼저 웹을 요청한 것을 보면 문제를 풀 수 있다. 그중, 전체 url 과 포트까지 포함하라 하였으므로 Host 에 적혀 있는 포트까지 포함한다.

1 번 답은 <https://10.10.10.10:8080/index.php> 이다.

[WHS-2] .iso

2.이에 대응하여 악성 웹 서버는 난독화된 JavaScript 를 다시 보냈습니다. 이 코드의 시작 부분에서 공격자는 "COMMENT"라는 라벨이 붙은 1300 개의 요소로 배열을 만든 다음 해당 데이터 요소를 문자열로 채웠습니다. 이 문자열의 값은 무엇이었나요?



```

Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
Host: 10.10.10.10:8080
Connection: Keep-Alive

HTTP/1.1 200 OK
Content-Type: text/html
Pragma: no-cache
Connection: Keep-Alive
Server: Apache
Content-Length: 5748

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<html>
<head>
<script>
var UWnHADOfYHiHDDXj = "COMMENT";
var qSNgVkOrdIjaiFpPTfDjbPHQppHSGtzpmOOyqEbLEFxnqAxicRyZKKWiRwMUA DHFOuzHPHqLrRFSzQuPusTnQyqpQwVpARd1R = new Array();
for (i = 0; i < 1300; i++)
{
qSNgVkOrdIjaiFpPTfDjbPHQppHSGtzpmOOyqEbLEFxnqAxicRyZKKWiRwMUA DHFOuzHPHqLrRFSzQuPusTnQyqpQwVpARd1R[i] = document.createElement(UWnHADOfYHiHDDXj);
qSNgVkOrdIjaiFpPTfDjbPHQppHSGtzpmOOyqEbLEFxnqAxicRyZKKWiRwMUA DHFOuzHPHqLrRFSzQuPusTnQyqpQwVpARd1R[i].data = "vEI";
}

```

[그림 3] HTTP Stream 중 "COMMENT" 부분

HTTP 스트림을 살펴보니 문제에 나온 "COMMENT" 부분이 보인다. 해당 부분을 해석하면 다음과 같다.

코드	해석
var UWnHADOfYHiHDDXj = "COMMENT";	변수 UWnHADOfYHiHDDXj 에 "COMMENT"라는 문자열을 할당한다.
var qSNgVkOrdIjaiFpPTfDjbPHQppHSGtzpmOOyqEbLEFxnqAxicRyZKKWiRwMUA DHFOuzHPHqLrRFSzQuPusTnQyqpQwVpARd1R = new Array();	배열을 선언한다.
for (i = 0; i < 1300; i++)	1300 번 반복한다.
qSNgVkOrdIjaiFpPTfDjbPHQppHSGtzpmOOyqEbLEFxnqAxicRyZKKWiRwMUA DHFOuzHPHqLrRFSzQuPusTnQyqpQwVpARd1R[i] = document.createElement(UWnHADOfYHiHDDXj);	새로운 HTML 요소를 생성하고, 이를 배열에 할당한다.
qSNgVkOrdIjaiFpPTfDjbPHQppHSGtzpmOOyqEbLEFxnqAxicRyZKKWiRwMUA DHFOuzHPHqLrRFSzQuPusTnQyqpQwVpARd1R[i].data = "vEI";	각 요소의 data 속성에 "vEI"라는 문자열을 할당한다.

[표 1] JavaScript 분석

[표 1]를 보면 data 에 vEI 라는 문자열을 할당하고 있으므로 2 번의 답은 vEI 이다.

[WHS-2] .iso

3. Vick 의 컴퓨터는 객체에 대해 두 번째 HTTP 요청을 했습니다.

3-a. 요청된 개체의 파일 이름은 무엇입니까?

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.10.70	10.10.10.10	HTTP	351	GET /index.php HTTP/1.1
2	0.000098	10.10.10.10	10.10.10.70	TCP	60	8080 → 1035 [ACK] Seq=1 Ack=298 Win=8576 Len=0
3	0.345405	10.10.10.10	10.10.10.70	TCP	1514	8080 → 1035 [ACK] Seq=1 Ack=298 Win=8576 Len=1460
4	0.345484	10.10.10.10	10.10.10.70	TCP	1514	8080 → 1035 [ACK] Seq=1461 Ack=298 Win=8576 Len=1460
5	0.345608	10.10.10.10	10.10.10.70	TCP	1514	8080 → 1035 [ACK] Seq=2921 Ack=298 Win=8576 Len=1460
6	0.345729	10.10.10.10	10.10.10.70	TCP	1514	8080 → 1035 [ACK] Seq=4381 Ack=298 Win=8576 Len=1460
7	0.345824	10.10.10.10	10.10.10.70	TCP	60	1035 → 8080 [ACK] Seq=298 Ack=5841 Win=65535 Len=0
8	0.345929	10.10.10.10	10.10.10.70	HTTP	86	HTTP/1.1 200 OK (text/html)
9	0.462112	10.10.10.70	10.10.10.10	HTTP	415	GET /index.phpmfKSxSANkeTeNrah.gif HTTP/1.1

[그림 4] Vick이 웹을 요청한 2번째 패킷

Vick이 HTTP를 요청한 2번째 패킷을 보면 내용에 파일명을 찾을 수 있다. 따라서 3-a의 답은 index.phpmfKSxSANkeTeNrah.gif이다.

3-b. 반환된 객체의 MD5sum은 무엇입니까?

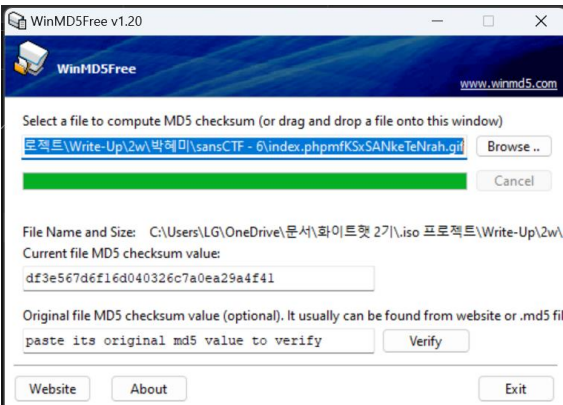
Wireshark · 내보내기 · HTTP 객체 목록

텍스트 필터:

패킷	호스트 이름	내용 유형	크기	파일 이름
8	10.10.10.10:8080	text/html	5748 bytes	index.php
11	10.10.10.10:8080	image/gif	43 bytes	index.phpmfKSxSANkeTeNrah.gif

[그림 5] 객체 내보내기

객체 내보내기를 사용하여 해당 파일을 저장한다.



[그림 6] WinMD5로 MD5 값 추출

추출한 파일의 MD5 값을 WinMD5 도구를 사용하여 추출한다.

따라서 3-b 의 답은 df3e567d6f16d040326c7a0ea29a4f41 이다.

[WHS-2] .iso

4. 4444 의 TCP 세션은 언제 열렸습니까? (패킷 캡처가 시작된 이후의 초 수를 10 분의 1 초로 반올림하여 제공합니다. 즉, 49.5 초)

No.	Time	Source	Destination	Protocol	Length	Info
13	1.265851	10.10.10.70	10.10.10.10	TCP	62	1036 → 4444 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM
14	1.265922	10.10.10.10	10.10.10.70	TCP	62	4444 → 1036 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 SACK_PERM
15	1.266218	10.10.10.70	10.10.10.10	TCP	60	1036 → 4444 [ACK] Seq=1 Ack=1 Win=65535 Len=0

[그림 7] tcp.port == 4444 윗부분

TCP의 4444의 포트를 검색하기 위해 tcp.port == 4444 필터링을 사용한다. 해당 내용을 보니 [그림 7]과 같이 4444의 포트와 연결된 것을 볼 수 있다. 그 중 캡처가 시작된 이후를 보라고 하였으니 [그림 7]의 3개를 볼 수 있다. Time 부분에서 반올림을 하라고 하였으니 4번의 답은 1.3초이다.

5. 포트 4444의 TCP 세션이 언제 닫혔습니까? (패킷 캡처가 시작된 이후의 초 수를 10분의 1초로 반올림하여 제공합니다. 즉, 49.5초)

1562	87.587095	10.10.10.10	10.10.10.70	TCP	60	4444 → 1036 [FIN, ACK] Seq=1239099 Ack=96139 Win=62780 Len=0
1563	87.587154	10.10.10.70	10.10.10.10	TCP	60	1036 → 4444 [FIN, ACK] Seq=96139 Ack=1239099 Win=64773 Len=0

[그림 8] tcp.port == 4444 아랫부분

해당 문제 또한 4 번 문제처럼 tcp.port == 4444 를 필터링하여 알아낼 수 있다. 닫힌 것을 찾아야 하니 가장 밑으로 내려가 FIN 을 확인한다. 이것 또한 반올림을 하라고 하였으니 5 번의 답은 87.6 초이다.

6. 패킷 17 에서는 악성 서버가 클라이언트에 파일을 보냈습니다.

6-a. 어떤 종류의 파일이었나요?

0080	01 4c cd 21 54 68 69 73 20 70 72 6f 67 72 61 6d	·L·!This program
0090	20 63 61 6e 6e 6f 74 20 62 65 20 72 75 6e 20 69	cannot be run i
00a0	6e 20 44 4f 53 20 6d 6f 64 65 2e 0d 0d 0a 24 00	n DOS mo de...\$.·

[그림 9] 17번 패킷 상세 내용

17 번 패킷의 상세 내용을 살펴보니 "This program cannot be run in DOS mode"라는 것이 있다. 해당 내용을 살펴본 결과 주로 Windows 운영 체제에서 실행할 수 없는 프로그램을 실행하려고 할 때 발생하는 오류라고 한다. 이를 미루어 보아 아마 파일은 Windows 의 실행 파일이었을 것으로 보인다.

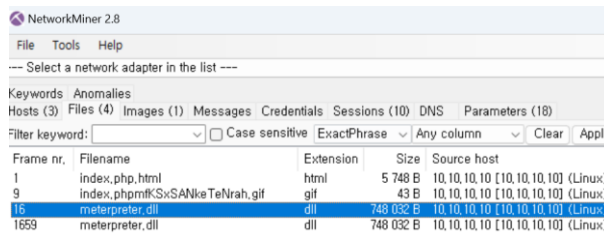
[WHS-2] .iso

6-b. 파일의 MD5sum 은 무엇이었나요?

16	1.526557	10.10.10.10	10.10.10.70	TCP	60 4444 → 1036 [PSH, ACK] Seq=1 Ack=1 Win=5840 Len=4
17	1.529777	10.10.10.10	10.10.10.70	TCP	1514 4444 → 1036 [ACK] Seq=5 Ack=1 Win=5840 Len=1460

[그림 10] 17번 패킷

17번 패킷을 확인해 보니 4444 포트에서 1036 포트를 보내고 있다.



Frame nr.	Filename	Extension	Size	Source host
1	index.php.html	html	5 749 B	10.10.10.10 [10.10.10.10] (Linux)
9	index.phpmkKSwSANkeTeNrah.gif	gif	43 B	10.10.10.10 [10.10.10.10] (Linux)
15	meterpreter.dll	dll	748 032 B	10.10.10.10 [10.10.10.10] (Linux)
1659	meterpreter.dll	dll	748 032 B	10.10.10.10 [10.10.10.10] (Linux)

[그림 11] NetworkMiner를 사용하여 사용 하여 해당 파일 찾기

1036 포트에 보내고 있는 파일은 [그림 11]에 선택되어 있는 [meterpreter.dll] 파일이다.



Name	MD5
meterpreter.dll	b062cb8344cd3e296d8868fbef289c7c

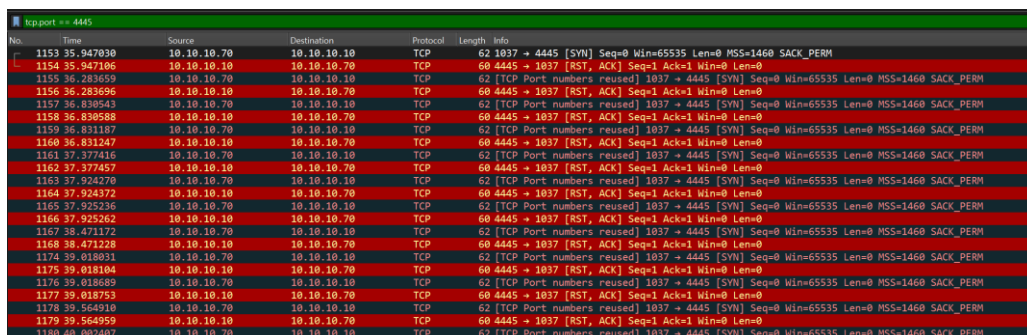
[그림 12] meterpreter.dll 파일의 MD5값

상세 보기를 눌러 MD5 값을 얻어낼 수 있다.

따라서 6-b의 답은 b062cb8344cd3e296d8868fbef289c7c이다.

7. Vick의 컴퓨터는 포트 4444의 원래 연결이 닫힌 후에도 반복적으로 포트 4445를 통해 악성 서버에 다시 연결을 시도했습니다. 이러한 반복적인 연결 시도 실패와 관련하여:

7-a. TCP 초기 시퀀스 번호(ISN)는 얼마나 자주 변경됩니까? (하나를 선택하세요.)



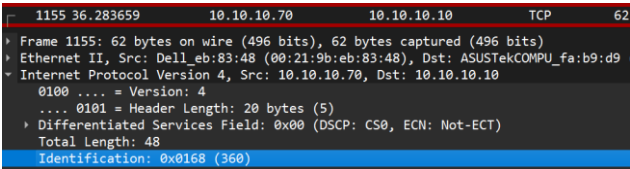
No.	Time	Source	Destination	Protocol	Length	Info
1153	35.947030	10.10.10.70	10.10.10.10	TCP	62	1037 → 4445 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM
1154	35.947106	10.10.10.10	10.10.10.70	TCP	60	4445 → 1037 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1155	36.283659	10.10.10.70	10.10.10.10	TCP	62	[TCP Port numbers reused] 1037 → 4445 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM
1156	36.283696	10.10.10.10	10.10.10.70	TCP	60	4445 → 1037 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1157	36.830543	10.10.10.70	10.10.10.10	TCP	62	[TCP Port numbers reused] 1037 → 4445 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM
1158	36.830588	10.10.10.10	10.10.10.70	TCP	60	4445 → 1037 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1159	36.831187	10.10.10.70	10.10.10.10	TCP	62	[TCP Port numbers reused] 1037 → 4445 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM
1160	36.831247	10.10.10.10	10.10.10.70	TCP	60	4445 → 1037 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1161	37.377416	10.10.10.70	10.10.10.10	TCP	62	[TCP Port numbers reused] 1037 → 4445 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM
1162	37.377457	10.10.10.10	10.10.10.70	TCP	60	4445 → 1037 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1163	37.924270	10.10.10.70	10.10.10.10	TCP	62	[TCP Port numbers reused] 1037 → 4445 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM
1164	37.924372	10.10.10.10	10.10.10.70	TCP	60	4445 → 1037 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1165	37.925236	10.10.10.70	10.10.10.10	TCP	62	[TCP Port numbers reused] 1037 → 4445 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM
1166	37.925262	10.10.10.10	10.10.10.70	TCP	60	4445 → 1037 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1167	38.471172	10.10.10.70	10.10.10.10	TCP	62	[TCP Port numbers reused] 1037 → 4445 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM
1168	38.471228	10.10.10.10	10.10.10.70	TCP	60	4445 → 1037 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1174	39.018031	10.10.10.70	10.10.10.10	TCP	62	[TCP Port numbers reused] 1037 → 4445 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM
1175	39.018104	10.10.10.10	10.10.10.70	TCP	60	4445 → 1037 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1176	39.018689	10.10.10.70	10.10.10.10	TCP	62	[TCP Port numbers reused] 1037 → 4445 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM
1177	39.018753	10.10.10.10	10.10.10.70	TCP	60	4445 → 1037 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1178	39.564910	10.10.10.70	10.10.10.10	TCP	62	[TCP Port numbers reused] 1037 → 4445 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM
1179	39.564959	10.10.10.10	10.10.10.70	TCP	60	4445 → 1037 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1180	40.002407	10.10.10.70	10.10.10.10	TCP	62	[TCP Port numbers reused] 1037 → 4445 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM

[그림 13] tcp.port == 4445

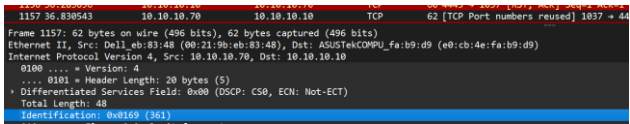
[WHS-2] .iso

포트 4445에 연결을 시도했다고 했으니 tcp.port == 4445를 필터링한다. [그림 13]과 같이 계속해서 연결을 실패하는 것을 볼 수 있는데 자세히 보니 3번째 패킷마다 다시 연결을 시도하는 것을 볼 수 있다. 따라서 7-a의 답은 “세 번째 패킷마다”이다.

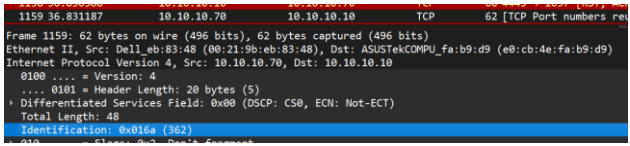
7-b. IP ID는 얼마나 자주 변경되나요? (하나를 선택하세요.)



[그림 14] 1155번 패킷



[그림 15] 1157번 패킷



[그림 16] 1158번 패킷

같은 IP의 ID(Identification)를 살펴보니 매 패킷마다 변경되는 것을 볼 수 있다. 따라서 7-b의 답은 “모든 패킷”이다.

7-c. 소스 포트는 얼마나 자주 변경되니까? (하나를 선택하세요.)

No.	Time	Source	Destination	Protocol	Length	Info
1153	35.947030	10.10.10.70	10.10.10.10	TCP	62	1037 → 4445 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM

[그림 17] 1번째 출발지 포트

No.	Time	Source	Destination	Protocol	Length	Info
1198	47.732517	10.10.10.70	10.10.10.10	TCP	62	1038 → 4445 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM

[그림 18] 2번째 출발지 포트

No.	Time	Source	Destination	Protocol	Length	Info
1313	59.462957	10.10.10.70	10.10.10.10	TCP	62	1039 → 4445 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM

[그림 19] 3번째 출발지 포트

[그림 17, 18, 19]를 비교해보니 약 12초마다 변경되는 것을 확인할 수 있다. 따라서 7-c의 답은 “10~15 초마다”이다.

[WHS-2] .iso

8. 결국 악성 서버가 응답하여 새로운 연결을 열었습니다. 포트 4445의 TCP 연결이 처음으로 성공적으로 완료된 것은 언제입니까? (패킷 캡처가 시작된 이후의 초 수를 10분의 1초로 반올림하여 제공합니다. 즉, 49.5초)

1655	123.236781	10.10.10.10	10.10.10.70	TCP	60 4445 → 1044 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1656	123.674199	10.10.10.70	10.10.10.10	TCP	62 [TCP Port numbers reused] 1044 → 4445 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM
1657	123.674296	10.10.10.10	10.10.10.70	TCP	62 4445 → 1044 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 SACK_PERM
1658	123.674586	10.10.10.70	10.10.10.10	TCP	60 1044 → 4445 [ACK] Seq=1 Ack=1 Win=65535 Len=0
1659	123.905791	10.10.10.10	10.10.10.70	TCP	60 4445 → 1044 [PSH, ACK] Seq=1 Ack=1 Win=5840 Len=4
1660	123.909415	10.10.10.10	10.10.10.70	TCP	1514 4445 → 1044 [ACK] Seq=5 Ack=1 Win=5840 Len=1460
1661	123.909516	10.10.10.10	10.10.10.70	TCP	1514 4445 → 1044 [ACK] Seq=1465 Ack=1 Win=5840 Len=1460

[그림 20] 연결 성공 부분

결국 연결에 성공했다고 하였으므로, 연결된 곳을 찾아 내려간다. 보아하니 123.7초에 연결에 성공한 것을 볼 수 있다. 따라서 8번의 답은 123.7초이다.

9. 이후, 악성 서버는 포트 4445를 통해 클라이언트에 실행 파일을 보냈습니다. 이 실행 파일의 MD5 sum은 얼마였습니까?

1655	123.236781	10.10.10.10	10.10.10.70	TCP	60 4445 → 1044 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
1656	123.674199	10.10.10.70	10.10.10.10	TCP	62 [TCP Port numbers reused] 1044 → 4445 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM
1657	123.674296	10.10.10.10	10.10.10.70	TCP	62 4445 → 1044 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 SACK_PERM
1658	123.674586	10.10.10.70	10.10.10.10	TCP	60 1044 → 4445 [ACK] Seq=1 Ack=1 Win=65535 Len=0
1659	123.905791	10.10.10.10	10.10.10.70	TCP	60 4445 → 1044 [PSH, ACK] Seq=1 Ack=1 Win=5840 Len=4
1660	123.909415	10.10.10.10	10.10.10.70	TCP	1514 4445 → 1044 [ACK] Seq=5 Ack=1 Win=5840 Len=1460
1661	123.909516	10.10.10.10	10.10.10.70	TCP	1514 4445 → 1044 [ACK] Seq=1465 Ack=1 Win=5840 Len=1460

[그림 21] PSH

실행 파일을 보냈다고 했으니 PSH를 살펴보면 된다.

0080	01 4c cd 21 54 68 69 73	20 70 72 6f 67 72 61 6d	·L·!This program
0090	20 63 61 6e 6e 6f 74 20	62 65 20 72 75 6e 20 69	cannot be run i
00a0	6e 20 44 4f 53 20 6d 6f	64 65 2e 0d 0d 0a 24 00	n DOS mo de...\$.·

[그림 22] 1659번 패킷의 상세 정보

해당 패킷 또한 "This program cannot be run in DOS mode"가 있는 것을 보아 Windows 실행 파일로 보여진다.

meterpreter.dll - File Details	
Name	meterpreter.dll
MD5	b062cb8344cd3e296d8868fbef289c7c

[그림 23] meterpreter.dll 파일의 MD5 값

NetworkMiner를 사용하여 MD5의 값을 알아낸다.

따라서 9번의 답은 b062cb8344cd3e296d8868fbef289c7c이다.

[WHS-2] .iso

10. 포트 4445 의 TCP 연결이 언제 닫혔습니까? (패킷 캡처가 시작된 이후의 초 수를 10 분의 1 초로 반올림하여 제공합니다. 즉, 49.5 초)

2551	198.439756	10.10.10.10	10.10.10.70	TCP	91 4445 → 1044 [PSH, ACK] Seq=833055 Ack=11379 Win=62780 Len=37
2552	198.440669	10.10.10.70	10.10.10.10	TCP	60 1044 → 4445 [FIN, ACK] Seq=11379 Ack=833092 Win=65232 Len=0
2553	198.441346	10.10.10.10	10.10.10.70	TCP	60 4445 → 1044 [FIN, ACK] Seq=833092 Ack=11380 Win=62780 Len=0
2554	198.441738	10.10.10.70	10.10.10.10	TCP	60 1044 → 4445 [ACK] Seq=11380 Ack=833093 Win=65232 Len=0

tcp.port == 4445 의 가장 밑을 보니 FIN 을 찾을 수 있다. 이를 보아 포트 4445 의 TCP 연결은 198.4 초에 닫힌 것을 알 수 있다. 따라서 10 번의 답은 198.4 초이다.



5. Flag

1 번: <https://10.10.10.10:8080/indes.php>

2 번: vEI

3-a 번: index.phpmfKSxSANkeTeNrah.gif

3-b 번: df3e567d6f16d040326c7a0ea29a4f41

4 번: 1.3 초

5 번: 87.6 초

6-a 번: Windows 실행 파일

6-b 번: b062cb8344cd3e296d8868fbef289c7c

7-a 번: 세 번째 패킷마다

7-b 번: 모든 패킷

7-c 번: 10~15 초마다

8 번: 123.7 초

9 번: b062cb8344cd3e296d8868fbef289c7c

10 번: 198.4 초

6. 별도 첨부

-문제 원문

Ann Dercover 는 SaucyCorp 의 비밀 소스 레시피를 따릅니다. 그녀는 SaucyCorp 의 서버에 원격으로 액세스할 수 있는 방법을 알아내기 위해 수석 개발자인 Vick Timmes 를 추적하고 있습니다. 어느 날 밤 정찰을 수행하던 중 그녀는 그가 자신의 노트북(10.10.10.70)에 로그인하고 SaucyCorp 의 본사에 VPN 을 연결하는 것을 목격했습니다.

Ann 은 국제 해킹 조직과의 연결을 활용하여 [Internet Explorer 에 대한 제로데이 익스플로잇](#) 을 획득 하고 Vick Timmes 를 대상으로 클라이언트 측 스피어 피싱 공격을 시작합니다. Ann 은 비밀 소스 레시피를 개선하는 방법에 대한 팁이 포함된 이메일을 Vick 에게 신중하게 작성하여 보냅니다. Vick 은 자신이 탐내던 제품 개발 부사장 직함(및 코너 사무실)을 얻을 수 있는 기회를 보고 링크를 클릭합니다. 앤은 공격할 준비가 되어 있습니다...

당신은 법의학 수사관입니다. 귀하의 임무는 Ann 의 공격이 포함된 [패킷 캡처](#)를 분석하고 , 타임라인을 구축하고, 다음을 포함한 [증거를 제출하는 것입니다](#) .

1. Vick Timmes 의 원래 웹 요청의 전체 URI 는 무엇이었나요? (URI 에 포트를 포함해 주세요.)
2. 이에 대응하여 악성 웹 서버는 난독화된 JavaScript 를 다시 보냈습니다. 이 코드의 시작 부분에서 공격자는 "COMMENT"라는 라벨이 붙은 1300 개의 요소로 배열을 만든 다음 해당 데이터 요소를 문자열로 채웠습니다. 이 문자열의 값은 무엇이었나요?
3. Vick 의 컴퓨터는 객체에 대해 두 번째 HTTP 요청을 했습니다.
 - a. 요청된 객체의 파일 이름은 무엇입니까?
 - b. 반환된 객체의 MD5sum 은 무엇입니까?
4. 포트 4444 의 TCP 세션은 언제 열렸습니까? (패킷 캡처가 시작된 이후의 초 수를 10 분의 1 초로 반올림하여 제공합니다. 즉, 49.5 초)
5. 포트 4444 의 TCP 세션이 언제 닫혔습니까? (패킷 캡처가 시작된 이후의 초 수를 10 분의 1 초로 반올림하여 제공합니다. 즉, 49.5 초)
6. 패킷 17 에서는 악성 서버가 클라이언트에 파일을 보냈습니다.
 - a. 어떤 종류의 파일이었나요? 하나를 선택하세요:
 - Windows 실행 파일
 - GIF 이미지
 - PHP 스크립트
 - 압축 파일
 - 암호화된 데이터
 - b. 파일의 MD5sum 은 무엇이었나요?
7. Vick 의 컴퓨터는 포트 4444 의 원래 연결이 닫힌 후에도 반복적으로 포트 4445 를 통해 악성 서버에 다시 연결을 시도했습니다. 이러한 반복적인 연결 시도 실패와 관련하여:
 - a. TCP 초기 시퀀스 번호(ISN)는 얼마나 자주 변경됩니까? (하나를 선택하세요.)
 - 모든 패킷

- 세 번째 패킷마다
 - 10~15 초마다
 - 30~35 초마다
 - 60 초마다
- b. IP ID 는 얼마나 자주 변경되나요? (하나를 선택하세요.)
- 모든 패킷
 - 세 번째 패킷마다
 - 10~15 초마다
 - 30~35 초마다
 - 60 초마다
- c. 소스 포트는 얼마나 자주 변경되니까? (하나를 선택하세요.)
- 모든 패킷
 - 세 번째 패킷마다
 - 10~15 초마다
 - 30~35 초마다
 - 60 초마다
8. 결국 악성 서버가 응답하여 새로운 연결을 열었습니다. 포트 4445 의 TCP 연결이 처음으로 성공적으로 완료된 것은 언제입니까? (패킷 캡처가 시작된 이후의 초 수를 10 분의 1 초로 반올림하여 제공합니다. 즉, 49.5 초)
9. 이후, 악성 서버는 포트 4445 를 통해 클라이언트에 실행 파일을 보냈습니다. 이 실행 파일의 MD5 sum 은 얼마였습니까?
10. 포트 4445 의 TCP 연결이 언제 닫혔습니까? (패킷 캡처가 시작된 이후의 초 수를 10 분의 1 초로 반올림하여 제공합니다. 즉, 49.5 초)
- 증거 파일은 다음과 같습니다: [evidence06.pcap](#)
- MD5(evidence06.pcap) = efac05c50c0ae92bf0818e98763920bd
 - SHA256(evidence06.pcap)=
fa5fc1ffad525688626c301372b37e101efcbabd124f9781f5701648e6a02be3

7. Reference

- [URL]