디지털 포렌식 과제

방화벽 로그 분석 시나리오

디지털 포렌식 트랙 서종찬

최종 제출일 `23.07.16 (일)

Index

분석 대상 조사

분석 대상 개요		
File Name	SIZE	Hash (sha-256)
security.log	19.9 GB	81e115ada42852d4ce6744ad6a5afcdd4084f1a1950fff3639289fda79

<특이 사항>

- security.log 파일에는 개행(\n)이 없는 상태로 저장되어 있어 Database 에 가공된 형태로 삽입이 불가.

- 로그는 2020-01-01 ~ 2023-07-20 까지 방화벽 로그를 대상으로 수집.

- 회사 내부의 소스코드가 유출, 도용된 것으로 추정, 회사 전체 로그를 분석하여 행위를 파악하는 것이 목적.

<회사 보안 정책>

- 5 대의 서버, 사원 500 명(내국인 250 명, 외국인 250 명)
- 모든 사원은 지정된 기기에서만 접근
- 모든 사원은 현지 시간 (미국 UTC-8, 한국 UTC+9)으로 업무시간(06:00 ~ 22:00)에만 접속 가능

<조사 준비 사항>

1. 빠른 인덱싱을 위한 Database 구축
2. Python 을 통해 데이터 가공 및 Database 에 값 추가
3. 회사 보안 정책을 기반으로 쿼리 작성
   - 업무 시간 외에 접속한 로그
   - 사용자의 IP 와 MAC 이 일치하지 않는 로그
   - 서버의 IP 와 MAC, PORT 가 일치하지 않는 로그

데이터 베이스 구축

사용 프로그램	MySQL (8.1)
데이터베이스 명	Server_db
테이블 명	fw_log, employee_list, server_list

fw_log	
Column_NAME	Type
id	INTEGER (PRIMARY KEY)
date_added	Datetime
source_ip	INTEGER (UNSIGNED)
source_mac	BIGINT (UNSIGNED)
source_port	INTEGER
server_name	VARCHAR(255)
destination_ip	INTEGER (UNSIGNED)
destination_mac	BIGINT UNSIGNED

Employee_list	
Column_NAME	Type
No	INTEGER (PRIMARY KEY)
Name	VARCHAR(50)
IP	INTEGER (UNSIGNED)
MAC	BIGINT (UNSIGNED)

Server_list	
Column_NAME	Type
No	INTEGER (PRIMARY KEY)
Name	VARCHAR(50)
IP	INTEGER (UNSIGNED)
MAC	BIGINT (UNSIGNED)
PORT	INTEGER (UNSIGNED)

인덱싱 현황
CREATE INDEX idx_date_added ON fw_log (date_added);
CREATE INDEX idx_source_ip ON fw_log (source_ip);
CREATE INDEX idx_source_mac ON fw_log (source_mac);
CREATE INDEX idx_destination_ip ON fw_log (destination_ip);
CREATE INDEX idx_destination_mac ON fw_log (destination_mac);

분석 시작

1.5 억개를 insert 하는 Python 프로그램 작성하는 것이 우선이다. 그런데 문제점은 개행(\n)이 없기 때문에 데이터 하나를 뽑아내는데 신중을 가해야 한다는 점이다. 테이블의 칼럼은 고정적인 길이를 가지는 값도 있지만 가변적인 길이를 가지는 값도 존재한다. 아이피와 MAC 주소, 바이트 크기 등은 정해진 길이가 있는 것이 아니기 때문에 고적길이만큼 바이트를 읽는 것은 불가능하다. 패턴을 발견하고 해당 패턴을 기준으로 데이터를 잘라서 insert 하는 기술이 중요하다. 먼저 눈에 띄는 것은 날짜이다.

```
un.py > ip_to_int
    import pymysql.cursors
    import socket
    import struct
    from datetime import datetime
    import sys
    phase = ["Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov","Dec"]
```

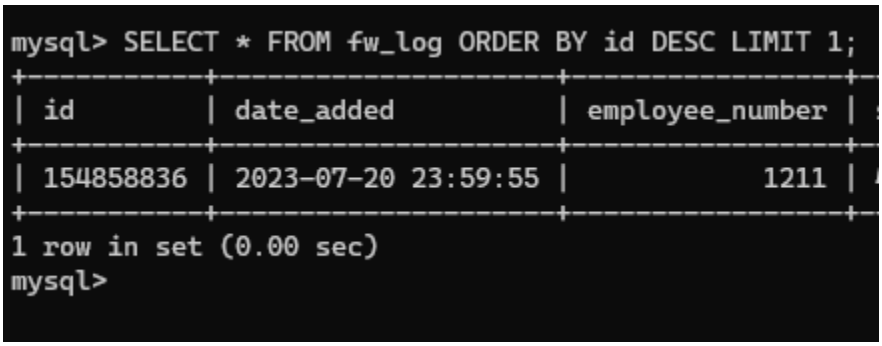
모든 데이터의 시작 3 글자는 해당 phase 의 12 개의 원소 중의 하나임은 자명하다. 해당 특성을 통해 데이터를 분류할 것이다.

```
 char_list.append(char)
# print(''.join( char_list[-3:] ),len(char_list),''.join(char_list))
 if len(char_list) > 100 and ''.join( char_list[-3:] ) in phase:
     #print(''.join(char_list[:-3]))
     raw = str(''.join(char_list[:-3])).split(" ")
```

또한 데이터를 분석한 결과, 한 데이터의 전체 바이트 길이가 최소 100 은 넘는다는 것을 알았다. 즉 데이터가 Jan xxx … xxx … 이런식으로 가다가 끝의 3 글자가 'Jan' 같은 phase 의 원소라면 끝의 3 글자를 다시 데이터의 첫글자로 넣고 나머지 부분을 데이터로 보는 알고리즘을 구상했다. 해당 알고리즘을 단계화 시켜보면 다음과 같다.

1. 바이트코드를 한 글자씩 읽는다.
2. 해당 바이트코드를 str 로 변환 후 문자열 변수에 계속 더한다.
3. 계속 더하면서 길이가 100 이상이 넘어갈 시 끝의 3 글자를 검사한다.
4. 해당 끝의 3 글자가 1~12 월의 정보를 나타내는 문자열이라면
5. 해당 3 글자를 제외한 나머지 부분을 데이터로 파싱해 DB 에 insert 한다.
6. 문자열 변수의 초기값을 해당 3 글자로 변환 후 위의 과정을 반복한다.

이 때 5 번의 DB insert 과정에서 ip 와 mac 을 정수로 바꾸는 기능이 있으며, 속도를 최대화하기 위해 pypy 를 사용하여 insert 작업을 진행하였다. 최종적인 파일은 run.py 에서 확인할 수 있다.



총 15,485,836 개의 데이터가 추가된 것을 볼 수 있다. 그리고 인덱싱을 설정하였기 때문에 해당 정보를 알아내는데 상당히 빠른 속도를 내는 것을 볼 수 있다.

사용 쿼리
SELECT * FROM fw_log ORDER BY id DESC LIMIT 1;

이제 사원 정보와 서버 정보 역시 DB 에 값을 추가할 것이다. 초기 설치 과정에서 mysql workbench 를 설치하지 않아 부득이하게 csv 를 gui 환경에서 넣을 수 없었고, 직접 파이썬

코드를 구현하여 insert 를 진행하였다. 해당 파일은 insertServer_list.py 와 insertEmployee.py 에서 확인할 수 있다. 회사의 보안 정책을 위반한 사람을 찾는 쿼리는 다음과 같다.

사용한 쿼리
mysql> SELECT fw.id -> FROM fw_log AS fw -> LEFT JOIN employee_list AS emp ON fw.source_ip = emp.IP AND fw.source_mac = -> emp.MAC -> WHERE emp.No IS NULL;

쿼리를 설명하자면 fw_log 에 찍힌 사용자의 ip 와 mac 이 employee_list 의 명시된 ip-mac 쌍이 아닌 것을 출력하는 쿼리이다.

```
mysql> SELECT fw.id
    -> FROM fw_log AS fw
    -> LEFT JOIN employee_list AS emp ON fw.source_ip = emp.IP AND fw.source_mac = emp.MAC
    -> WHERE emp.No IS NULL;
+----------+
| id       |
+----------+
| 65127278 |
| 65127313 |
| 86201252 |
| 86201253 |
| 86201254 |
| 86201255 |
| 86201256 |
| 86201257 |
| 86201258 |
| 86201259 |
| 86201260 |
| 86201261 |
| 86201262 |
| 86201263 |
| 86201264 |
| 86201265 |
| 86201266 |
| 86201267 |
| 86201268 |
| 86201269 |
| 86201270 |
| 86201271 |
| 86201272 |
| 86201273 |
| 86201274 |
| 86201275 |
+----------+
26 rows in set (3 min 1.77 sec)
```

해당 결과는 다음과 같다. 총 26 건의 로그가 발견되었고 이는 정보 유출과 관련된 로그다.

```
mysql> select * from fw_log where id = 65127278;
+----------+---------------------+-----------------+------------+----------------+-------------+
| id       | date_added          | employee_number | source_ip  | source_mac     | source_port |
+----------+---------------------+-----------------+------------+----------------+-------------+
| 65127278 | 2021-06-29 10:30:14 |            1369 | 1230651821 | 18039931858064 |       34333 |
+----------+---------------------+-----------------+------------+----------------+-------------+
1 row in set (0.01 sec)
```

사원번호 1369 는 외국인의 id 이고 해당 mac 을 employee_list 에 조회해 보면 한국인의 mac 주소임을 알 수 있다.

```
mysql> select * from employee_list where MAC = 18039931858064;
+------+--------+-----------+----------------+
| No   | Name   | IP        | MAC            |
+------+--------+-----------+----------------+
| 1021 | 김기연 | 463986692 | 18039931858064 |
+------+--------+-----------+----------------+
1 row in set (0.01 sec)
```

```
mysql> select * from fw_log where id = 65127313;
+----------+---------------------+-----------------+------------+----------------+
| id       | date_added          | employee_number | source_ip  | source_mac     |
+----------+---------------------+-----------------+------------+----------------+
| 65127313 | 2021-06-29 10:30:40 |            1268 | 1231241802 | 18039931858064 |
+----------+---------------------+-----------------+------------+----------------+
1 row in set (0.00 sec)
```

또한 다음 로그도 비슷한 시간대에 회원번호 1268 인 사원이 동일한 MAC 으로 접근했다. 다른 사람의 MAC 주소를 통해 접속을 시도한 것을 볼 때 정보 유출을 위한 ARP 스푸핑이 의심된다. 해당 두 사람의 정보는 다음과 같다.

```
mysql> select * from employee_list where No = 1369 or No = 1268;
+------+-------------+------------+----------------+
| No   | Name        | IP         | MAC            |
+------+-------------+------------+----------------+
| 1268 | Arleen Choi | 1231241802 | 48742778176928 |
| 1369 | Ian Choi    | 1230651821 |   136956671582 |
+------+-------------+------------+----------------+
2 rows in set (0.01 sec)
```

또한 그 이후의 로그인 86201252 ~ 86201275 번 대의 로그는 연속적인 로그로, 정보유출 행위의 실제 로그로 추정된다.

```
mysql> SELECT fl.date_added, en.NAME, fl.source_ip, fl.source_mac, fl.server_name, fl.size_kb
    -> FROM fw_log AS fl
    -> INNER JOIN employee_list AS en ON fl.employee_number = en.No
    -> WHERE fl.id >= 86201253 AND fl.id <= 86201275;
+---------------------+----------+------------+-----------------+-------------+---------+
| date_added          | NAME     | source_ip  | source_mac      | server_name | size_kb |
+---------------------+----------+------------+-----------------+-------------+---------+
| 2021-12-23 00:02:21 | Ian Choi | 1230651821 | 273777359852657 | NAS_2       |   80012 |
| 2021-12-23 00:02:21 | Ian Choi | 1230651821 | 273777359852657 | NAS_2       |   80632 |
| 2021-12-23 00:02:22 | Ian Choi | 1230651821 | 273777359852657 | NAS_2       |   80012 |
| 2021-12-23 00:02:22 | Ian Choi | 1230651821 | 273777359852657 | NAS_2       |   80011 |
| 2021-12-23 00:02:22 | Ian Choi | 1230651821 | 273777359852657 | NAS_2       |   80832 |
| 2021-12-23 00:02:23 | Ian Choi | 1230651821 | 273777359852657 | NAS_2       |   81153 |
| 2021-12-23 00:02:23 | Ian Choi | 1230651821 | 273777359852657 | NAS_2       |   83522 |
| 2021-12-23 00:02:23 | Ian Choi | 1230651821 | 273777359852657 | NAS_2       |   85553 |
| 2021-12-23 00:02:23 | Ian Choi | 1230651821 | 273777359852657 | NAS_2       |   86239 |
| 2021-12-23 00:02:23 | Ian Choi | 1230651821 | 273777359852657 | NAS_2       |   82385 |
| 2021-12-23 00:02:24 | Ian Choi | 1230651821 | 273777359852657 | NAS_2       |   83235 |
| 2021-12-23 00:02:24 | Ian Choi | 1230651821 | 273777359852657 | NAS_2       |   87713 |
| 2021-12-23 00:02:24 | Ian Choi | 1230651821 | 273777359852657 | NAS_2       |   83731 |
| 2021-12-23 00:02:24 | Ian Choi | 1230651821 | 273777359852657 | NAS_2       |   83474 |
| 2021-12-23 00:02:25 | Ian Choi | 1230651821 | 273777359852657 | NAS_2       |   85474 |
| 2021-12-23 00:02:25 | Ian Choi | 1230651821 | 273777359852657 | NAS_2       |   81737 |
| 2021-12-23 00:02:25 | Ian Choi | 1230651821 | 273777359852657 | NAS_2       |   84357 |
| 2021-12-23 00:02:26 | Ian Choi | 1230651821 | 273777359852657 | NAS_2       |   82437 |
| 2021-12-23 00:02:26 | Ian Choi | 1230651821 | 273777359852657 | NAS_2       |   88213 |
| 2021-12-23 00:02:26 | Ian Choi | 1230651821 | 273777359852657 | NAS_2       |   81324 |
| 2021-12-23 00:02:27 | Ian Choi | 1230651821 | 273777359852657 | NAS_2       |   82386 |
| 2021-12-23 00:02:28 | Ian Choi | 1230651821 | 273777359852657 | NAS_2       |   81236 |
| 2021-12-23 00:02:28 | Ian Choi | 1230651821 | 273777359852657 | NAS_2       |    1251 |
+---------------------+----------+------------+-----------------+-------------+---------+
23 rows in set (0.01 sec)
```

사용한 쿼리
SELECT fl.date_added, en.NAME, fl.source_ip, fl.source_mac, fl.server_name, fl.size_kb ->FROM fw_log AS fl ->INNER JOIN employee_list AS en ON fl.employee_number = en.No ->WHERE fl.id >= 86201253 AND fl.id <= 86201275;

employee_list 를 fw_log 와 INNER JOIN 하여 분석한 결과, Ian Choi 라는 사람이 현지 시각 UTC-8 을 적용했을 때 2021-12-22 16:02:21 부터 2021-12-22 16:02:28 초 동안 인가되지 않은 MAC 을 통해 NAS_2 에서 데이터를 유출해 간 것을 볼 수 있다.

```
mysql> select * from employee_list where MAC = 273777359852657;
Empty set (0.00 sec)

mysql>
```

employee_list 에 존재하지 않는 mac 이므로 새로운 외부 장치일 것으로 추측된다. 또한 접근한 NAS_2 의 포트 역시 비인가된 포트이다.

```
mysql> select * from server_list where MAC = 33322;
Empty set (0.01 sec)

mysql>
```

유출된 데이터의 양은 다음과 같다.

```
mysql> select sum(size_kb) from fw_log where id >= 86201253 and id <= 86201275;
+--------------+
| sum(size_kb) |
+--------------+
|      1826919 |
+--------------+
1 row in set (0.01 sec)
```

이는 약 1.74 GB 에 해당하는 양이다.

또한 1219 번 이하의 사원 번호를 가지는 사람은 한국인으로, 다음과 같이 일반적인 사용자는 UTC-0 기준으로 해당 시간에는 업무시간이 아님을 알아냈다.

결론

최종적으로 알아낸 것은 Ian Choi 와 Arleen Choi 가 가장 유력한 용의자이며, 실제 정보유출을 행한 사람은 Ian Choi 로 추정됩니다. 대략적인 시나리오를 요약하자면 2021-06-29 새벽 2 시경 Ian Choi 와 Arleen Choi 는 김기연의 MAC 을 토용해 ARP 스푸핑 후 범죄를 계획하였으며 약 6 개월이 지난 2021-12-22 일 오후 4 시경 약 1.7GB 의 데이터를 비인가 장치를 통해 NAS_2 서버에 비인가 PORT 로 접속하여 유출한 것으로 추측됩니다.

해당 시나리오를 분석하면서 데이터 정재 과정의 중요성을 알게 되었으며 직접 쿼리를 짜면서 분석하는 과정이 상당히 흥미로웠습니다. 그러나 아쉬운 점은 해당 보고서의 시나리오는 상당히 빠른 시간안에 구해냈지만 그 이후로 약 일주일간 추가적인 행위를 찾기위해 많은 시도를 했음에도 불구하고 본 보고서의 내용만큼 명확한 증거는 발견해내지 못하였습니다. 그러다 결국 다른 과제의 마감일에 치여 분석을 그만두고 제출을 결정하게 되었는데, 좀 더 심층적인 분석을 해보고 싶었으나 그러지 못한 점이 아쉽습니다.