실시간 대응 전략

2021.02.05.

1. 대응 모듈 전개

1-1) 대응 모듈 배포

① 업로드

- 파이선 스크립트를 난독화 시킨다.
- 난독화된 파이선 스크립트를 PyInstaller를 이용하여 하나의 실행 파일로 추출함으로 써, 대응모듈이 탈취되어도 즉각적인 코드 분석이 어렵도록 만든다.

```
pip3 install pyinstaller
```

```
pyinstaller --name Response --onefile --noconsole main.py
```

- 추출된 실행 파일을 'Response'이름으로 Github 페이지에 업로드 한다.
  (https://github.com/GoldBigDragon/GoldBigDragon.github.io/tree/master/resources/files/BoB)

② CnC 서버 통신상태 확인

- CnC 서버의 IP를 확인하고, API 서버가 운용되는 Port 및 방화벽 접근 권한을 확인 한다.
- 확인된 주소 및 포트를 Github 제어 페이지에 수정/기록한다.
  (https://github.com/GoldBigDragon/GoldBigDragon.github.io/blob/master/version/BoB.json)

③ 다운로드

- 대응 모듈 전개지점은 공격 수행 여지가 적은 /mnt 경로로 한다.
- 전개지점에서 wget 을 통해 대응 모듈을 다운로드한다.

대응모듈 다운로드

```
sudo wget -O Response https://github.com/GoldBigDragon/GoldBigDragon.github.io/raw/master/resources/files/BoB/Response
```

- 실행 권한을 부여한다.

실행 권한 부여

```
sudo chmod +x ./Response
```

1-2) 대응 모듈 전개

① API 서버 구동

- API 서버를 구동시킨다.

```
node ./main.js
```

② 데이터베이스 초기화
- 데이터베이스를 초기화 시켜 정확한 데이터를 수집할 수 있도록 한다.

```
TRUNCATE `realtimeresponse`.`accountactivity`;
TRUNCATE `realtimeresponse`.`accountpasswd`;
TRUNCATE `realtimeresponse`.`arp`;
TRUNCATE `realtimeresponse`.`filetimelogs`;
TRUNCATE `realtimeresponse`.`history`;
TRUNCATE `realtimeresponse`.`hosts`;
TRUNCATE `realtimeresponse`.`internetconnection`;
TRUNCATE `realtimeresponse`.`lastlog`;
TRUNCATE `realtimeresponse`.`majorlogs`;
TRUNCATE `realtimeresponse`.`packettraffic`;
TRUNCATE `realtimeresponse`.`processlsmod`;
TRUNCATE `realtimeresponse`.`processlsof`;
TRUNCATE `realtimeresponse`.`processstatus`;
TRUNCATE `realtimeresponse`.`rootkit`;
TRUNCATE `realtimeresponse`.`socketconnection`;
TRUNCATE `realtimeresponse`.`systemtime`;
TRUNCATE `realtimeresponse`.`systemversion`;
TRUNCATE `realtimeresponse`.`weblogs`;
```

③ 대응모듈 전개
- 패킷 스니퍼가 네트워크 장비에 접근해야하므로, 실행 시 관리자권한을 요한다.
- 공격측의 탈취를 대비하여 인자 값으로 BoB9DigitalForensics를 입력하지 않으면 작동하지 않게 되어있다. (ps 입력 시 입력 값이 보이긴 하지만, 그저 대응 프로그램이니 건들지 말라는 뜻으로 전해질 것이다)
- & 연결자를 사용하여 앞의 명령어를 백그라운드 실행으로 돌리고, 동시에 기록을 지운다.

```
sudo ./Response BoB9DigitalForensics & history -c
```

```
root       2502  0.0  0.3  52700  3784 tty1     S    04:50   0:00 sudo ./Response BoB9DigitalForensi
root       2503  0.0  0.3  10400  3284 tty1     S    04:50   0:00 ./Response BoB9DigitalForensics
root       2504 17.9  3.7 763480 37476 tty1     Sl   04:50   0:33 ./Response BoB9DigitalForensics
root       3784  0.0  0.3  52700  3784 tty1     S    04:52   0:00 sudo ./Response BoB9DigitalForensi
root       3785  0.0  0.3  10400  3348 tty1     S    04:52   0:00 ./Response BoB9DigitalForensics
root       3787 20.0  3.6 763516 36208 tty1     Sl   04:52   0:15 ./Response BoB9DigitalForensics
user       5241  0.0  0.3  37364  3180 tty1     R+   04:53   0:00 ps -aux
```

④ 데이터 수집 확인
- 전개 이후 데이터베이스에 정상적으로 데이터가 수집되는지 확인한다.

⑤ 흔적 제거
- 전개 이후 history -c 명령어를 입력하여 추적 단서를 제거한다.

2. 대응 모듈 기능

- status 컬럼은 ORI, ADD, DEL 3가지 값을 가지며, 각각 대응모듈 실행 초기부터 존재하던 값, 공격 진행 이후 추가된 값, 공격 진행 이후 제거된 값을 뜻한다.
- 시간 관련 컬럼은 모두 yyyy-MM-dd HH:mm:ss 포맷으로 저장된다.
- IP주소, 시간 값은 '실시간' 대응 시 가독성을 위하여 정수형 대신 문자열로 저장한다.
- PATH, USERNAME과 같이 가변적 길이를 갖는 항목은 무조건 LONGTEXT로 저장하여 크래커가 악의적으로 파일 명을 길게 작성하여 API 서버에 오류를 발생시키는 것을 조기에 차단시킨다.

2-1) 네트워크

① 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

② 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

③ 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)

2-2) 프로세스

① 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		

② 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		

2-3) 로그파일

① 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		

② 웹로그 수집

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		

2-4) 기타

① 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)

② 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)

③ 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		

④ 명령 수행 내역 수집

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

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

⑤ 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

⑥ 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

⑦ 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

⑧ 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

⑨ 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
시스템 파일 변조여부 수집	/system/post-rootkit	5초	rootkit

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

⑩ 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		

2-5) 데이터 송수신

① 암호화

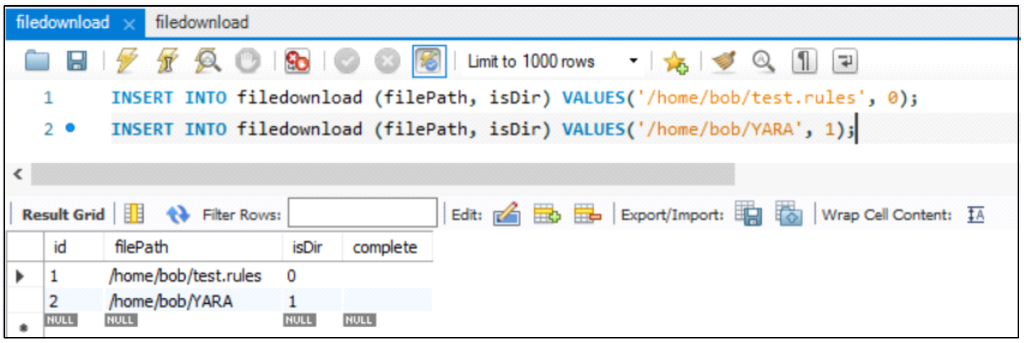
- 공격팀에서 송신 데이터를 도청할 경우를 대비하여 모든 통신을 암호화 시킨다.
- RSA4096, AES256 등 강력한 암호화 방식은 컴퓨팅 자원이 많이 필요하기 때문에 제외시킨다.
- 초당 수천 건의 통신이 이루어지므로, 단순한 암호화 규칙을 생성한다.
  - ⓐ json 형식 데이터를 배열로 만든다.
  - ⓑ 배열을 역순으로 뒤집는다.
  - ⓒ 배열 속 값들을 16진수로 변환시킨다.
  - ⓓ 배열 맨 앞에 문자 b를 추가하여 16진수 자체를 HxD로 옮겼을 때 모두 깨져 보이도록 만든다.

ⓔ 배열을 문자열로 변환시킨다.

- 서버에서 암호화된 데이터를 수신했을 때, 복호화 과정은 아래와 같다.
  ⓐ 문자열의 가장 앞자리를 제거한다.
  ⓑ 문자열을 16진수로 변환시킨다.
  ⓒ 16진수를 배열로 만든다.
  ⓓ 배열을 역순으로 뒤집는다.
  ⓔ 배열을 문자열로 변환시킨 다음, json 형태로 파싱 한다.

② 파일 수집

- 파일 수집이 필요할 경우, filedownload 테이블에 파일 이름 혹은 디렉터리 명을 입력하면 된다.
- Ubuntu의 파일 경로 특성 때문에 반드시 루트 디렉터리부터 시작하여야 하며, ./ ../ 등을 사용할 수 없다.
- 사용 예시



③ 데이터베이스 접근

- 112.148.181.37:5000 주소에 ID와 PW 모두 root 입력 시 접근 가능하다.