

CTF Crypto 입문“만” 해보기

Chung-Ang Univ.

CAUtion 0기

김도엽

이번 세미나의 목표

1. CTF의 소개 및 현재 환경에 대한 공유
2. 동아리 내의 CTF, Wargame 관심도 증가 (특히 Crypto)
3. 진입장벽을 낮춤으로써 CTF의 참여를 유도, 많은 참여로 인한 실력 향상 기대
4. 동아리 내 Crypto 풀이 집단 형성 빌드업

1. CTF란?
2. Crypto 문제의 일반적인 형태
3. 문제를 풀기 위한 환경 세팅
 1. Ubuntu
 2. pwntools
 3. Paper and Pen
4. 쉬운 RSA 문제에 대한 Write-up
 1. 2023 ISANG X CAUtion CTF / Crypto / Blinding

1. CTF란?

▶ Catch The Flag

- ▶ 취약점을 분석하고 Flag라는 secret을 찾아내는 보안, 해킹 대회
- ▶ 문제풀이(Jeopardy)와 Attack-Defense 방식으로 나뉨
- ▶ (대부분의 온라인 대회는 Jeopardy) 발견한 Flag를 제출하면 해당 문제를 해결했다고 보는 방식
- ▶ 분야 예시
 - ▶ Crypto, Cryptography 암호학
 - ▶ Reversing 리버스 엔지니어링
 - ▶ Pwn(포너블) 시스템 해킹
 - ▶ Web 웹
 - ▶ Forensics 디지털 포렌식
 - ▶ OSINT 공개정보
 - ▶ MISC(Miscellaneous) 기타

Challenges

Cryptography

Loud ✓
280

Really Loud ✓
316

LCG
489

Loud Revenge
492

Really Loud Revenge
498

LLoud
498

LLoud 2
500

Reverse Engineering

Meow meow meow
460

Fancy
475

Nasty
494

X-rays
498

OP
498

IoT binary
500

10000
500

PWN

one-punch ✓
241

popcorn
486

chatroom
500

1. CTF란?

▶ Flag 형식

- ▶ 대부분 Example{flag} 의 format을 가지고 있음
- ▶ 앞에 CTF 이름 등으로 태그를 놓고 중괄호로 비밀값을 감싸는 format
- ▶ Flag를 유추하는 것을 막기 위해서 알파벳을 숫자로 치환한 장난이 많음 (A = 4, E = 3, g = 9, O = 0)

▶ 배점 방식

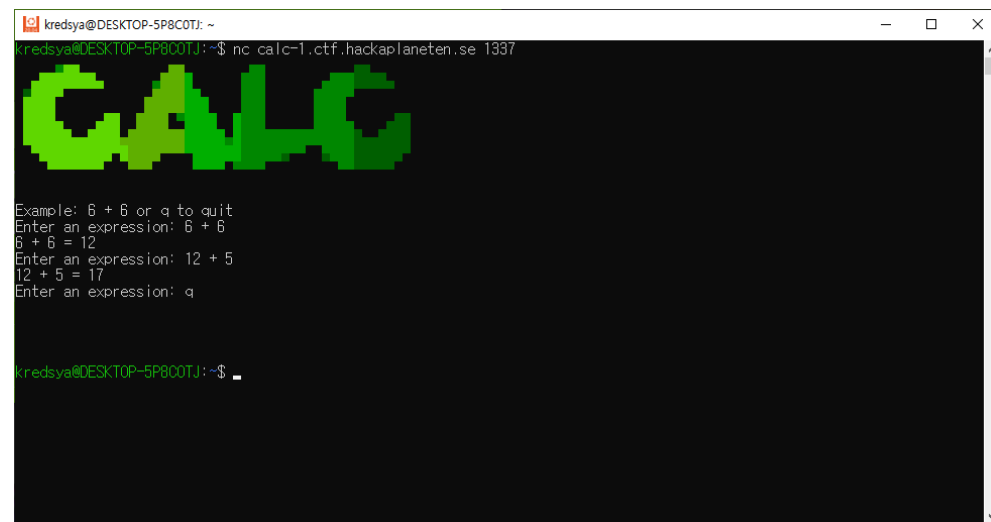
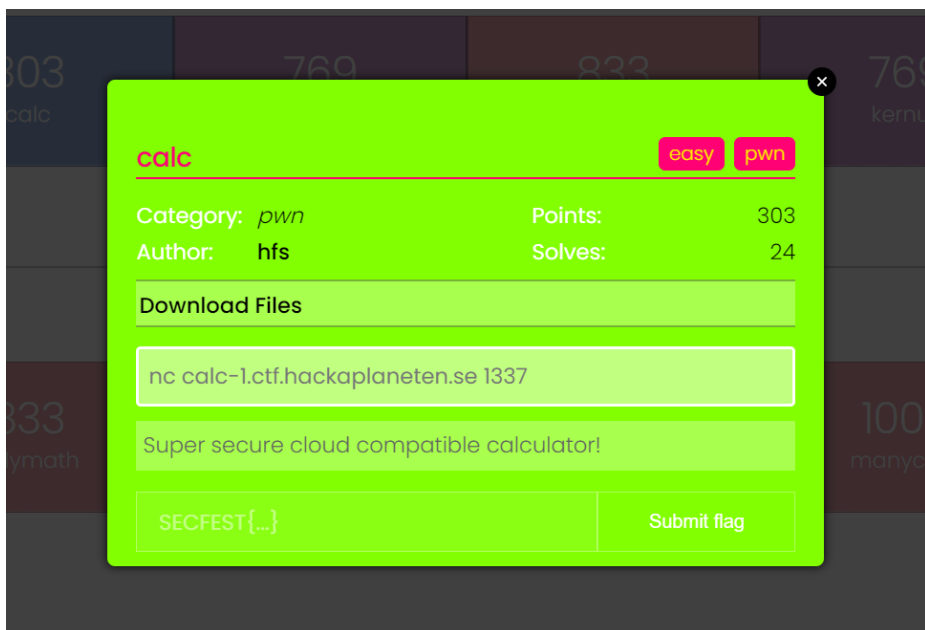
- ▶ Dynamic 방식이 주류
- ▶ 초기 배점은 모두 같음 (거의 500점)
- ▶ 푼 사람이 많아질 수록 해당 문제의 배점이 기하급수적으로 감소함 (쉬운 문제는 200점, 어려우면 480점)
- ▶ 점수를 많이 땀다고 생각하고 자고 왔더니 쪽 떨어져 있는 경우가 종종 발생함
- ▶ 점수 집계 후 동점자 발생 시 먼저 푼 사람이 우선됨

2. Crypto 문제의 일반적인 형태

- ▶ 보통 python 코드가 주어짐
 - ▶ 이는 server 와 client 코드일수도, 같이 주어진 output을 만든 단순 program 일 수도 있음
 - ▶ Flag는 bytes 로 초기화되고, bytes_to_long() 함수에 의해 long(int)으로 바뀐 뒤 연산이 진행됨
 - ▶ 아스키코드값(16진수 2자리)의 나열을 하나의 큰 수로 봄
 - ▶ Python은 Big Integer에 대한 처리가 따로 필요없어서 자주 쓰임
- ▶ Flag는 취약점을 가진 scheme에 의해 암호화됨
 - ▶ Ex1) 개인키 d가 매우 작은 RSA (Wiener's Attack)
 - ▶ Ex2) 충분한 개수의 결과가 노출된 LCG(Linear Congruential Generator)
- ▶ Flag가 보관되는 방법은 다양함
 - ▶ 외부 txt 파일 read, 외부 python 코드에서 import, 내부 코드에서 선언했으나 배포 코드는 모자이크 처리

2. Crypto 문제의 일반적인 형태

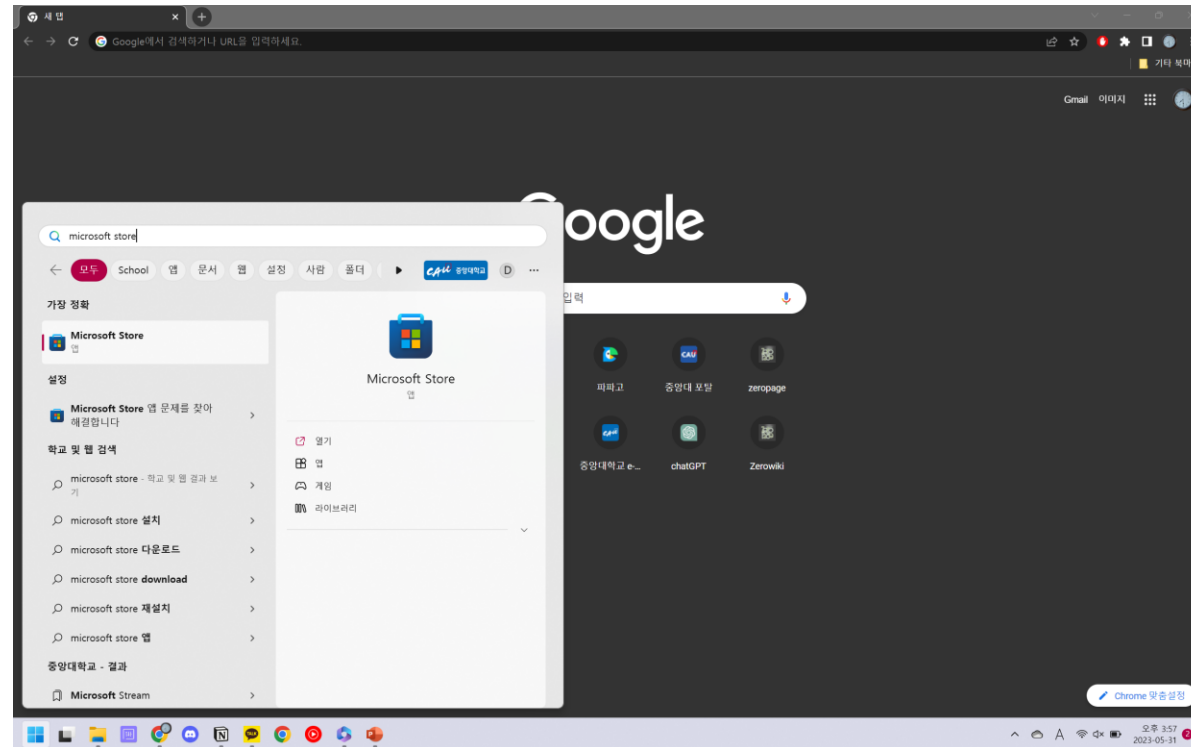
- ▶ 문제에 대한 접근
 - ▶ 코드와 함께 netcat으로 접속할 수 있는 환경을 제공함
 - ▶ nc (host) (port) 형식으로 우분투에서 접속
 - ▶ Python 라이브러리 중 pwntools로도 접속 및 상호작용이 가능함



3. 문제를 풀기 위한 초기 세팅

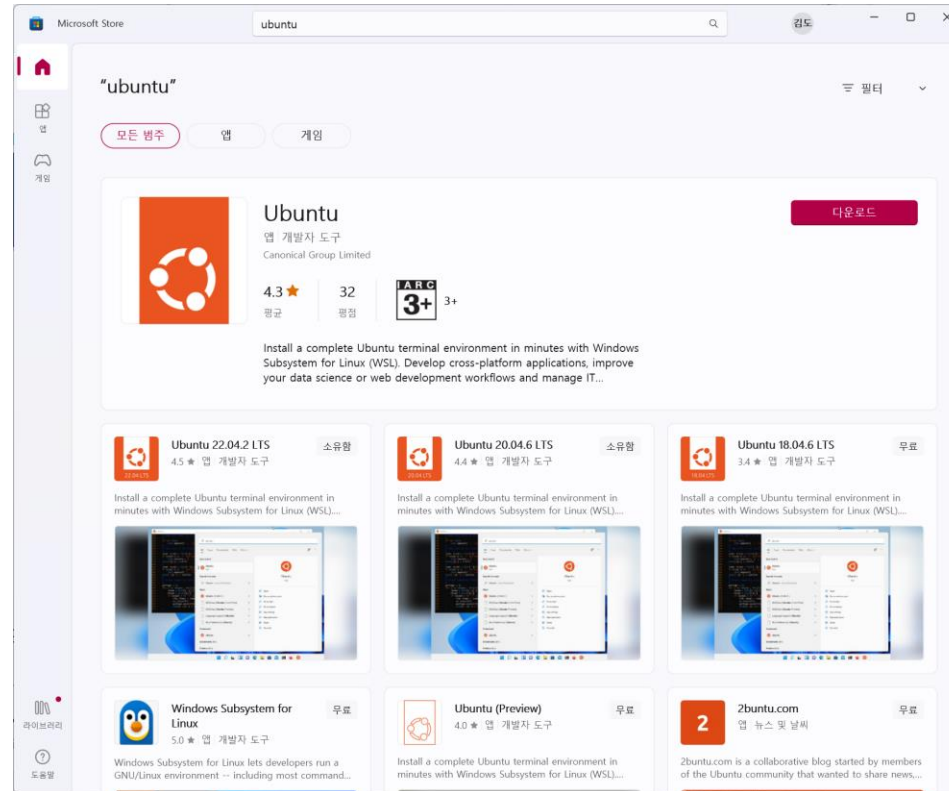
- ▶ nc을 위한 Ubuntu
- ▶ 코드 실행을 위한 Python
- ▶ 코드 분석을 위한 VS Code
- ▶ exploit code 작성을 위한 pwntools
- ▶ pwntools 설치를 위한 pip

3-1. Ubuntu



- ▶ (윈도우 기준) Microsoft Store 실행
- ▶ 그 외, Ubuntu 공식 사이트에 접속하여 설치 진행

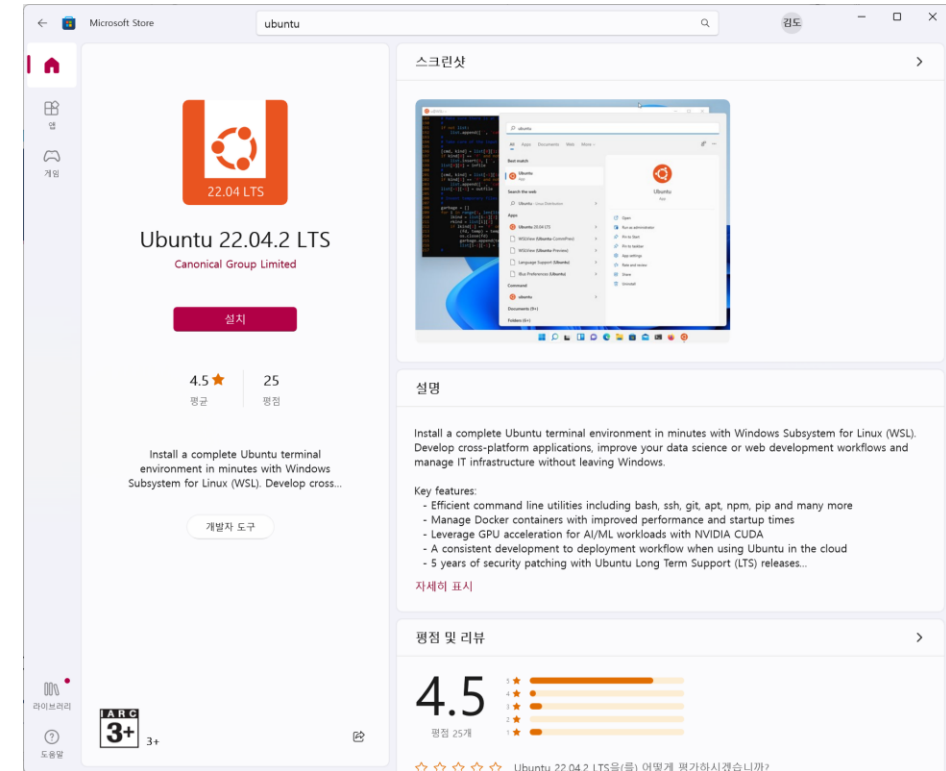
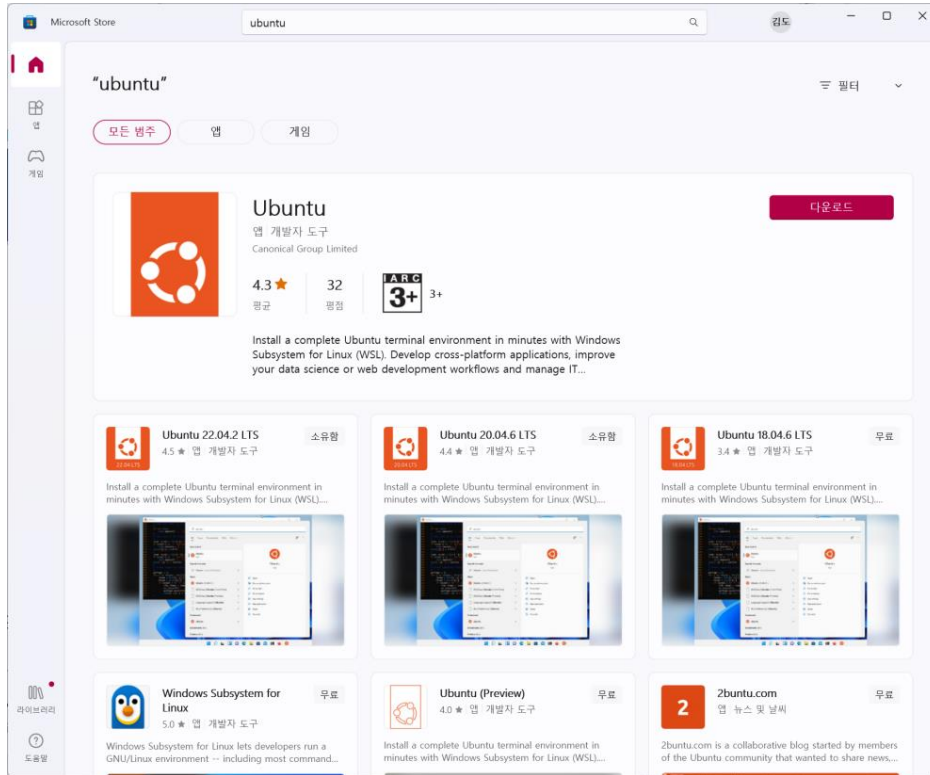
3-1. Ubuntu



▶ Ubuntu 검색

▶ 22.04 또는 20.04 버전 다운로드 (문제에 따라 지원 가능한 버전이 달라질 수도 있음)

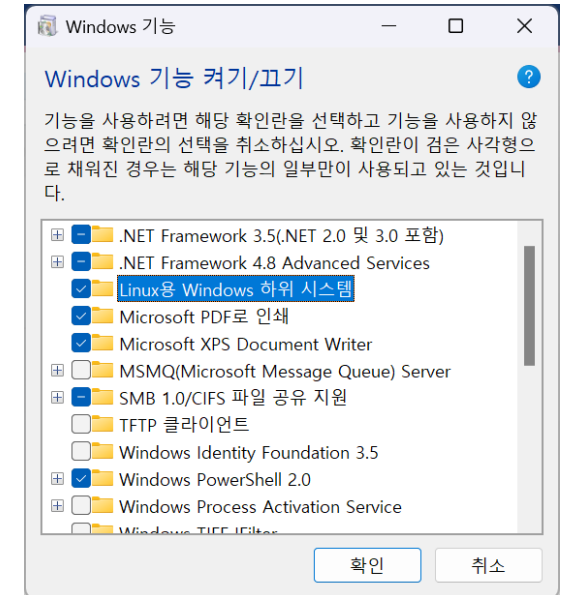
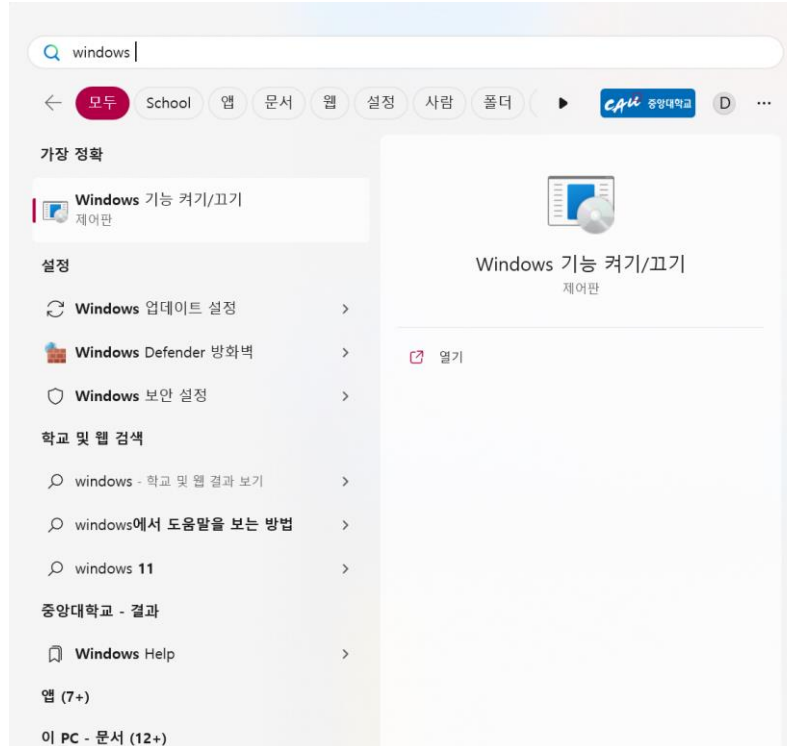
3-1. Ubuntu



- ▶ Ubuntu 검색
- ▶ 22.04 또는 20.04 버전 다운로드 (문제에 따라 지원 가능한 버전이 달라질 수도 있음)

3-1. Ubuntu

```
Ubuntu 20.04.6 LTS
Installing, this may take a few minutes...
WslRegisterDistribution failed with error: 0x8007019e
Error: 0x8007019e Linux? Windows ?? ???? ???? ???? ?? ????
Press any key to continue...
```



- ▶ 최초 실행 시 0x8007019e 에러 발생 시 리눅스 하위 시스템이 없다는 것
- ▶ Windows 기능 켜기/끄기에서 설정. 그 외 오류도 구글링으로 검색 가능.

3-1. Ubuntu

```
Ubuntu 20.04.6 LTS
Installing, this may take a few minutes...
Please create a default UNIX user account. The username does not need to match your Windows username.
For more information visit: https://aka.ms/wslusers
Enter new UNIX username: |
```

```
kredsya@Kredsya: ~
See "man sudo_root" for details.

Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.10.16.3-microsoft-standard-WSL2 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Wed May 31 16:25:20 HST 2023

System load:  0.03          Processes:            8
Usage of /:   0.5% of 250.98GB Users logged in:        0
Memory usage: 2%           IPv4 address for eth0: 192.168.133.124
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

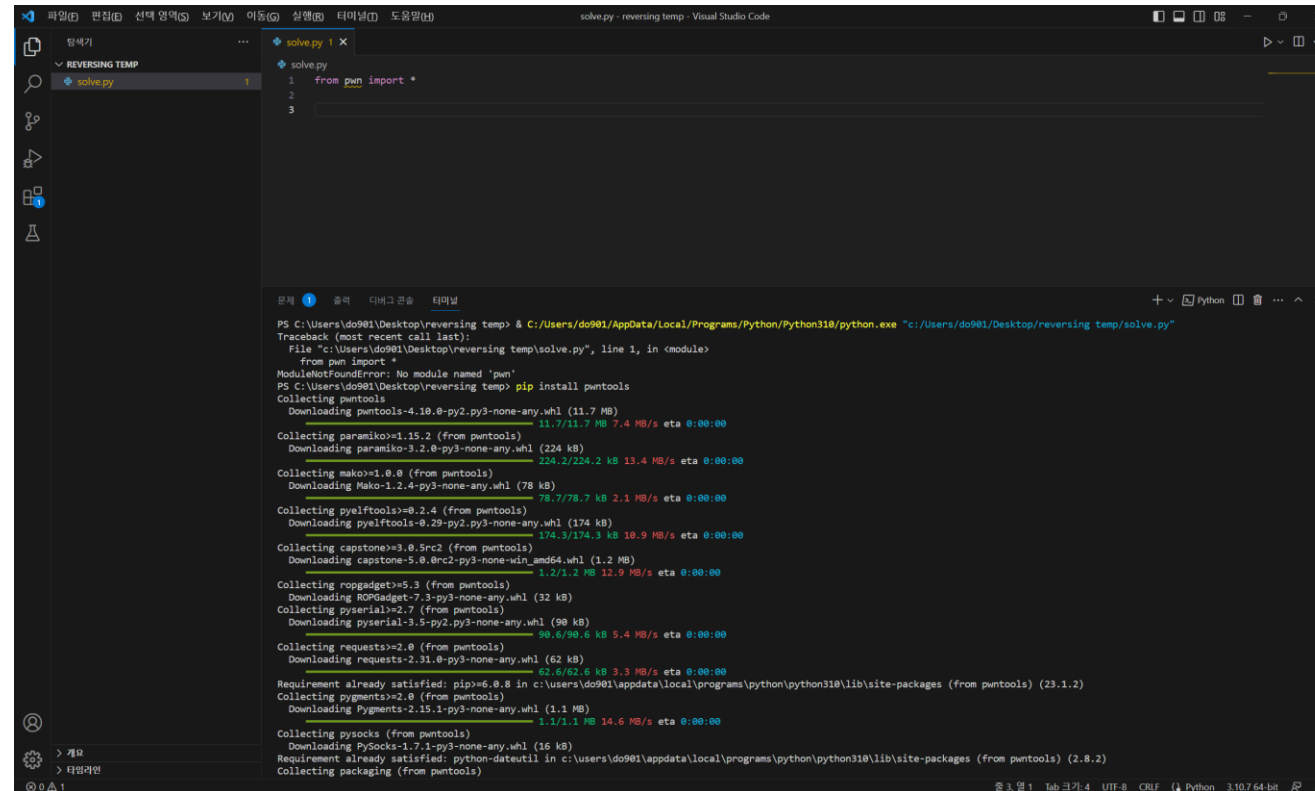
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

This message is shown once a day. To disable it please create the
/home/kredsya/.hushlogin file.
kredsya@Kredsya:~$ |
```

- ▶ 최초 실행 시 자동으로 환경설정
- ▶ 닉네임과 비밀번호 입력하면 설치 끝

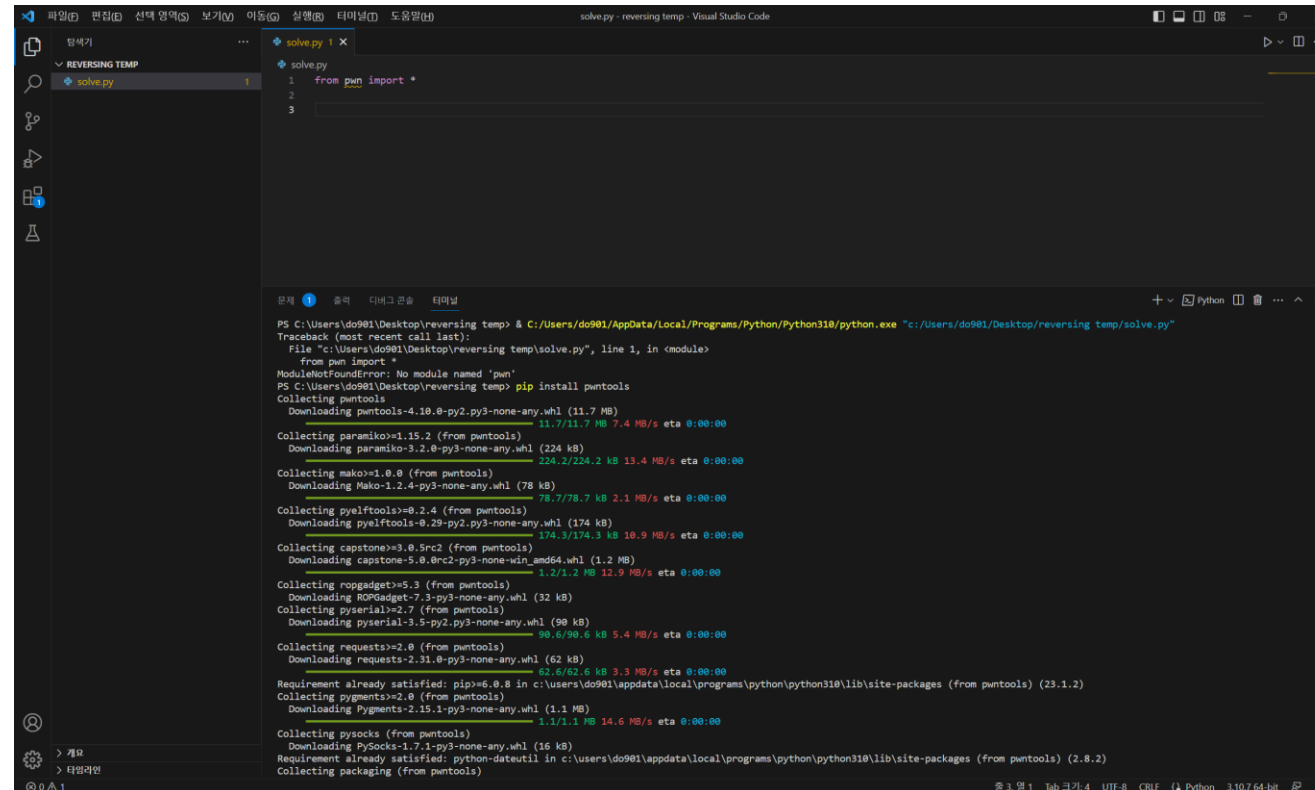
3-2. pwntools



```
PS C:\Users\do901\Desktop\reversing temp> & C:\Users\do901\AppData\Local\Programs\Python\Python310\python.exe "c:\Users\do901\Desktop\reversing temp\solve.py"
Traceback (most recent call last):
  File "c:\Users\do901\Desktop\reversing temp\solve.py", line 1, in <module>
    from pwn import *
ModuleNotFoundError: No module named 'pwn'
PS C:\Users\do901\Desktop\reversing temp> pip install pwntools
Collecting pwntools
  Downloading pwntools-4.10.0-py2.py3-none-any.whl (11.7 MB)
    11.7/11.7 MB 7.4 MB/s eta 0:00:00
Collecting paramiko>=1.15.2 (from pwntools)
  Downloading paramiko-3.2.0-py3-none-any.whl (224 kB)
    224.2/224.2 kB 13.4 MB/s eta 0:00:00
Collecting mako>=1.0.0 (from pwntools)
  Downloading Mako-1.2.4-py3-none-any.whl (78 kB)
    78.7/78.7 kB 2.1 MB/s eta 0:00:00
Collecting pyelftools>=0.29 (from pwntools)
  Downloading pyelftools-0.29-py2.py3-none-any.whl (174 kB)
    174.3/174.3 kB 10.9 MB/s eta 0:00:00
Collecting capstone>=3.0.5rc2 (from pwntools)
  Downloading capstone-5.0.8rc2-py3-none-win_amd64.whl (1.2 MB)
    1.2/1.2 MB 12.9 MB/s eta 0:00:00
Collecting ropgadget>=5.3 (from pwntools)
  Downloading ROPGadget-7.3-py3-none-any.whl (32 kB)
Collecting pyserial>=2.7 (from pwntools)
  Downloading pyserial-3.5-py2.py3-none-any.whl (90 kB)
    90.6/90.6 kB 5.4 MB/s eta 0:00:00
Collecting requests>=2.0 (from pwntools)
  Downloading requests-2.31.0-py3-none-any.whl (62 kB)
    62.6/62.6 kB 3.3 MB/s eta 0:00:00
Requirement already satisfied: pip>=6.0.8 in c:\users\do901\AppData\Local\Programs\Python\Python310\lib\site-packages (from pwntools) (23.1.2)
Collecting pygments>=2.0 (from pwntools)
  Downloading Pygments-2.15.1-py3-none-any.whl (1.1 MB)
    1.1/1.1 MB 14.6 MB/s eta 0:00:00
Collecting pycrypto (from pwntools)
  Downloading PySocks-1.7.1-py3-none-any.whl (16 kB)
Requirement already satisfied: python-dateutil in c:\users\do901\AppData\Local\Programs\Python\Python310\lib\site-packages (from pwntools) (2.8.2)
Collecting packaging (from pwntools)
```

- ▶ pip install pwntools 명령어로 설치 가능
- ▶ pip는 python으로 설치 가능

3-2. pwntools



```
PS C:\Users\do901\Desktop\reversing temp> & C:\Users\do901\AppData\Local\Programs\Python\Python310\python.exe "c:\Users\do901\Desktop\reversing temp\solve.py"
Traceback (most recent call last):
  File "c:\Users\do901\Desktop\reversing temp\solve.py", line 1, in <module>
    from pwn import *
ModuleNotFoundError: No module named 'pwn'
PS C:\Users\do901\Desktop\reversing temp> pip install pwntools
Collecting pwntools
  Downloading pwntools-4.10.0-py2.py3-none-any.whl (11.7 MB)
    11.7/11.7 MB 7.4 MB/s eta 0:00:00
Collecting paramiko>=1.15.2 (from pwntools)
  Downloading paramiko-3.2.0-py3-none-any.whl (224 kB)
    224.2/224.2 kB 13.4 MB/s eta 0:00:00
Collecting mako>=1.0.0 (from pwntools)
  Downloading Mako-1.2.4-py3-none-any.whl (78 kB)
    78.7/78.7 kB 2.1 MB/s eta 0:00:00
Collecting pyelftools>=0.29 (from pwntools)
  Downloading pyelftools-0.29-py2.py3-none-any.whl (174 kB)
    174.3/174.3 kB 10.9 MB/s eta 0:00:00
Collecting capstone>=3.0.5rc2 (from pwntools)
  Downloading capstone-5.0.8rc2-py3-none-win_amd64.whl (1.2 MB)
    1.2/1.2 MB 12.9 MB/s eta 0:00:00
Collecting ropgadget>=5.3 (from pwntools)
  Downloading ROPGadget-7.3-py3-none-any.whl (32 kB)
Collecting pyserial>=2.7 (from pwntools)
  Downloading pyserial-3.5-py2.py3-none-any.whl (90 kB)
    90.6/90.6 kB 5.4 MB/s eta 0:00:00
Collecting requests>=2.0 (from pwntools)
  Downloading requests-2.31.0-py3-none-any.whl (62 kB)
    62.6/62.6 kB 3.3 MB/s eta 0:00:00
Requirement already satisfied: pip>=6.0.8 in c:\users\do901\AppData\Local\Programs\Python\Python310\lib\site-packages (from pwntools) (23.1.2)
Collecting pygments>=2.0 (from pwntools)
  Downloading Pygments-2.15.1-py3-none-any.whl (1.1 MB)
    1.1/1.1 MB 14.6 MB/s eta 0:00:00
Collecting pycrypto (from pwntools)
  Downloading PySocks-1.7.1-py3-none-any.whl (16 kB)
Requirement already satisfied: python-dateutil in c:\users\do901\AppData\Local\Programs\Python\Python310\lib\site-packages (from pwntools) (2.8.2)
Collecting packaging (from pwntools)
```

- ▶ pip install pwntools 명령어로 설치 가능
- ▶ pip는 python으로 설치 가능

3-2. pwntools

문제 정보

📄 해당 문제는 Dreamhack CTF Season 1 Round #5에 출제된 문제입니다.

Description

드림이가 비밀 플래그를 가지고 있는 RSA 서버를 운영하고 있습니다. 서버를 공격해 플래그를 탈취해주세요!

플래그 형식은 DH{...} 입니다.

References

<https://dreamhack.io/lecture/courses/76>

접속 정보

Host: host3.dreamhack.games
Port: 8742/tcp → 9090/tcp

nc host3.dreamhack.games 8742
<http://host3.dreamhack.games:8742/>

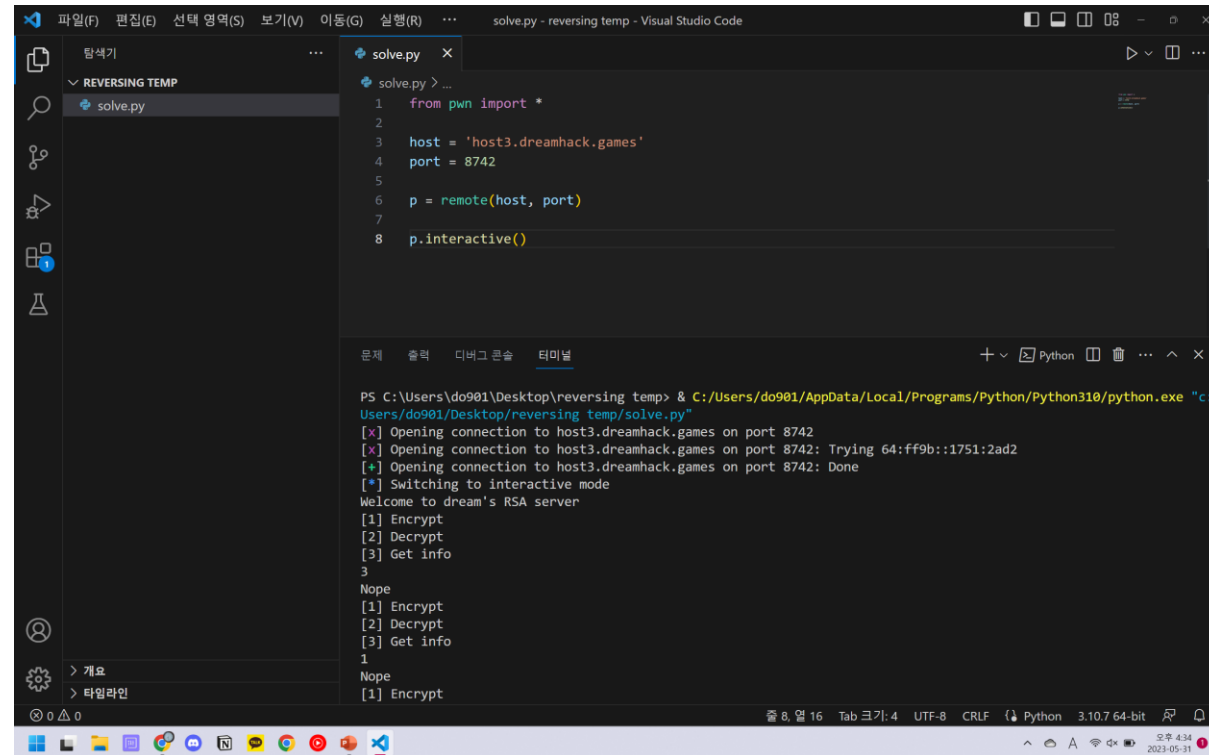
⚠ VM 부팅에 다소 시간이 걸릴 수 있습니다.

문제 파일

📄 문제 파일 다운로드

- ▶ 밑의 nc 주석
- ▶ nc (host) (port)로 이루어져 있음

3-2. pwntools

A screenshot of the Visual Studio Code editor. The main window shows a Python file named 'solve.py' with the following code:

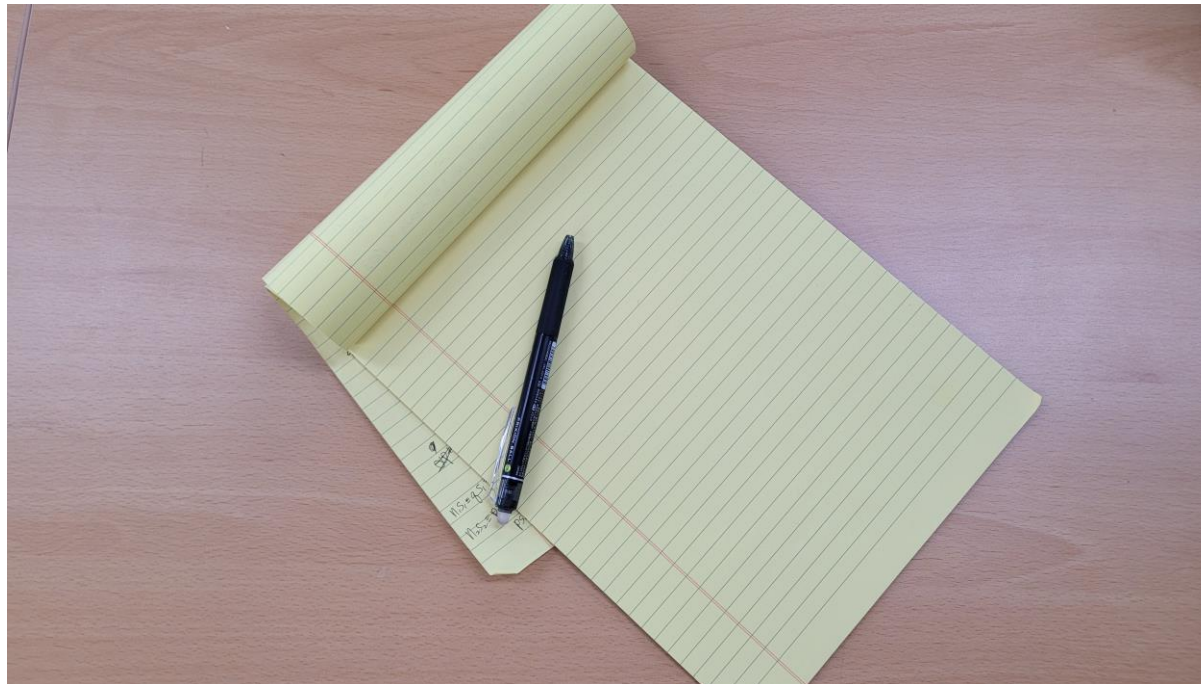
```
1 from pwn import *
2
3 host = 'host3.dreamhack.games'
4 port = 8742
5
6 p = remote(host, port)
7
8 p.interactive()
```

The bottom panel shows the terminal output of the script execution. The command executed is `python.exe "c:/Users/do901/Desktop/reversing temp/solve.py"`. The output shows the script successfully connecting to the host on port 8742 and entering an interactive mode with a menu:

```
PS C:\Users\do901\Desktop\reversing temp> C:/Users/do901/AppData/Local/Programs/Python/Python310/python.exe "c:/
Users/do901/Desktop/reversing temp/solve.py"
[x] Opening connection to host3.dreamhack.games on port 8742
[x] Opening connection to host3.dreamhack.games on port 8742: Trying 64:ff9b::1751:2ad2
[+] Opening connection to host3.dreamhack.games on port 8742: Done
[*] Switching to interactive mode
Welcome to dream's RSA server
[1] Encrypt
[2] Decrypt
[3] Get info
3
Nope
[1] Encrypt
[2] Decrypt
[3] Get info
1
Nope
[1] Encrypt
```

- ▶ `p = remote(host, port)` 명령어 사용 시 nc로 접속
- ▶ `p.sendline(b'string')`, `in = p.recvline()` 등으로 문제 서버 접속을 자동화 할 수 있음

3-3. Paper and Pen



- ▶ 종이와 펜
- ▶ 암호학 = 수학 = 증명도 일부 필요

4-0. Write-up이란?

- ▶ 풀이 방법에 대한 전 과정을 적은 것
 - ▶ Flag는 고정된 값 => 값 공유의 부정행위 발생 위험
 - ▶ 상위 입상자에 한해 Write-up 제출 요구함
 - ▶ Exploit code(해결하기 위해 작성한 코드)가 있으면 같이 첨부함
 - ▶ 제출용 write-up에는 풀이법의 증명까지는 잘 하지 않는 경향이 있음
- ▶ 본 발표에서 Crypto 문제에 대한 Write-up 한 개 살펴볼 예정

4-1. Blinding

▶ 2023 ISANG X CAUtion CTF / Crypto / Blinding

▶ Author : 김도엽

▶ Alice는 서버 구축을 연습하려고 RSA 기반 서명 시스템을 적용한 서버를 만들어보았다. Alice는 자신이 만든 시스템의 보안에 자신이 있어서 동아리 토크방에 자신의 서버에 서명을 보내서 검증에 성공할 때 나오는 flag를 가져오면 밥을 사주겠다고 했다. 당신은 밥을 얻어먹을 수 있겠는가?

▶ 실제 주어지는 python 파일의 flag는 ?로 모자이크 되어서 배포되었음

```
1  from Crypto.Util.number import *
2
3  flag = b"IxC{Blind1ng_i5_on3_of_the_3lment4r7_att4cks}"
4
5  p = getStrongPrime(512)
6  q = getStrongPrime(512)
7  N = p * q
8  phi_N = (p-1) * (q-1)
9  e = 13
10 d = inverse(e, phi_N)
```

4-1. Blinding

- ▶ 메뉴 및 입력
 - ▶ 공개키 확인
 - ▶ 서명
 - ▶ 검증
 - ▶ Challenge(flag)

```
12  ∨ if __name__ == "__main__":
13  ∨     while True:
14         print()
15         print("=== Welcome to Alice's server ===")
16         print("1. get information")
17         print("2. make signature")
18         print("3. verify signature")
19         print("4. challenge")
20         print("0. exit")
21         print("mode : ", end='')
22
23         select = int(input())
```

4-1. Blinding

- ▶ 평범한 동작
 - ▶ (1) 공개키 N, e 확인
 - ▶ (2) hex로 들어오는 message를 개인키로 서명해줌
 - ▶ 단, 'Alice'는 서명해주지 않음
 - ▶ (3) message와 서명을 hex로 받아서 검증해줌

```
25     if select == 1:
26         print(f"N = {N}")
27         print(f"e = {e}")
28
29     elif select == 2:
30         print("msg(hex) : ", end='')
31         msg = input()
32         if msg == b"Alice".hex():
33             print("Don't cheat :<")
34             continue
35
36         sign = format(pow(int(msg, 16), d, N), 'x')
37         print(sign)
38
39     elif select == 3:
40         print("msg(hex) : ", end='')
41         msg = input()
42         print("sign(hex) : ", end='')
43         sign = input()
44
45         sign = format(pow(int(sign, 16), e, N), 'x')
46
47         if msg == sign:
48             print("verifying success")
49         else:
50             print("invalid signature")
```

4-1. Blinding

▶ (4) Challenge

- ▶ Flag를 알아낼 수 있는 곳
- ▶ 'Alice'를 서명한 값을 입력하면 flag를 출력

```
52 elif select == 4:
53     msg = b"Alice".hex()
54     print("sign : ", end='')
55     sign = input()
56
57     sign = str(format(pow(int(sign, 16), e, N), 'x'))
58
59     if msg == sign:
60         print("Here is flag")
61         print(flag)
62     else:
63         print("Try again")
```


4-1. Blinding

▶ Write-up (intended)

- ▶ 「Twenty Years of Attacks on the RSA Cryptosystem」 논문에서 나온 Blinding을 그대로 적용한 문제
- ▶ $e = 13$ 으로 고정되어있음
- ▶ Alice의 hex string인 416c696365 를 그대로 넣으면 “Don’t Cheat :<” 메시지가 출력됨
- ▶ 이 메시지에 2^e 를 곱해서 서명을 요청하면 됨

$$2^{-1}(2^e M)^d \equiv 2^{ed-1} M^d \equiv 2^{k\phi(N)+1-1} M^d \equiv M^d \equiv S \pmod{N}$$

- ▶ 이후에 받은 서명에 2의 역수를 곱해주면 ‘Alice’의 서명을 얻을 수 있게 됨

4-1. Blinding

```
kredsya@DESKTOP-5P8C0TJ: ~  
3. verify signature  
4. challenge  
0. exit  
mode : 3  
msg(hex) : 426f62  
sign(hex) : 43882486217779c0f7f30f71ddcc00dcaa75ba408b3764eb7bf01e5cfd566362ba6345e8e8a7dbfee570ed009e061b25dc38730fbc33  
2a1a982484a4b4d3b15d86a5bd80c5e58c4102ad34ed557fe2c6a067d30b0f10437bd030cb3d3ea44a34d79c468087a3b0a573866d5b0144d1e15370  
6efeebadb78ab224bd0af8dc216c  
verifying success  
  
=== Welcome to Alice's server ===  
1. get information  
2. make signature  
3. verify signature  
4. challenge  
0. exit  
mode : 4  
sign : 9b3b5a43529a076f89a62ad11935723f625065333c65f2c0cf6ff1a267e5714d11d57924fe9d009637abf5537aafc4ab3e51919c5e787ee74  
09ce2355828a6a7b3ed9f4f7c8b5ca6f2c7d37adcbf259dbb813fd6afa24f1939391d0b1828c1781708dda6995878582b2e98194d61e12039e4e9873  
2320f26f12f0e920f1fa819  
Here is flag  
b'!xC{Blind1ng_i5_on3_of_the_3!ment4r7_att4cks}'  
  
=== Welcome to Alice's server ===  
1. get information  
2. make signature  
3. verify signature  
4. challenge  
0. exit  
mode :
```

4-1. Blinding

- ▶ Write-up (unintended)
 - ▶ Write-up by 김*준 학우
 - ▶ $0x416c696365 = 0x1159 * 0x3c574ed$ 이니까 2번에 1159랑 03c574ed를 입력하여 받은 결과를 곱해서 4번에 입력하면 된다.

5. 최신 동향

- ▶ RSA의 경우 지속적으로 비슷한 문제가 나옴
 - ▶ 잘못 선택된 소수 p, q
 - ▶ 잘못된 세팅 $N = p^2$
 - ▶ CRT (Chinese Remainder Theorem)
 - ▶ 일부 비트 노출 공격
- ▶ SVP(Shortest Vector Problem) 문제 등 LLL-algorithm을 의도하는 문제출제 빈번해짐
- ▶ 메르센-트위스터 의사난수, LCG(Linear Congruential Generator) 등 이미 뚫린 의사난수 문제도 적지만 가끔 등장함

Thank you

Q & A