Giga.ctf

Context

This web app was a challenge for NorthSec 2019.

+ The challenge had a lot of solves, but some people did not understand **how** they solved it.
+ Difficulty is: Easy to Medium

The vulnerability is heavily inspired by a real vulnerability that affected MegaUpload.

Tools

In order to solve this challenge, you will need:

+ A tool to read pcap (Wireshark/Ethereal is the recommended tool)
+ A text editor, to read source code
+ A browser

Info:

To solve this, you need the pcap at:

https://github.com/montrehack/challenges/blob/master/2020-07-22_giga-ctf/capture.pcap

The website is:

http://giga.montrehack.ca

Ready?

https://github.com/montrehack/challenges/blob/master/2020-07-22_giga-ctf/capture.pcap

Or

montrehack.ca

Can anyone name this attack?

Session Puzzling

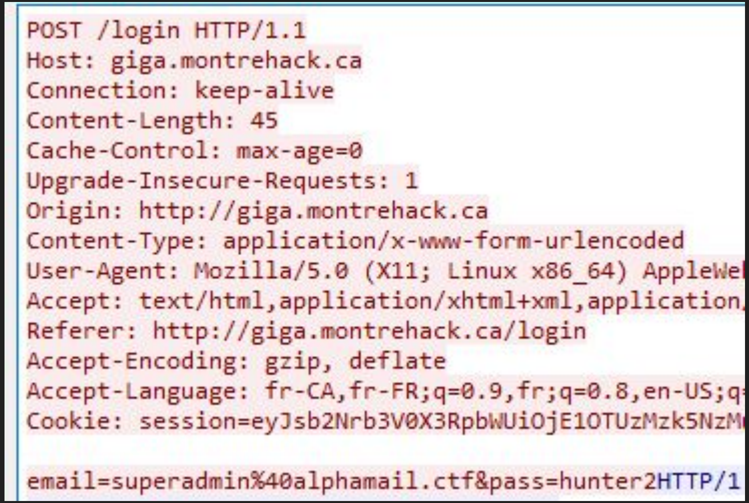
Session Puzzling is a logic based attack when two different process use the SAME session variable name. This oftentimes allow an attacker to bypass some logic flaws.

Step 1:

Using Wireshark, you can see the username and password:

Username: superadmin@alphamail.ctf

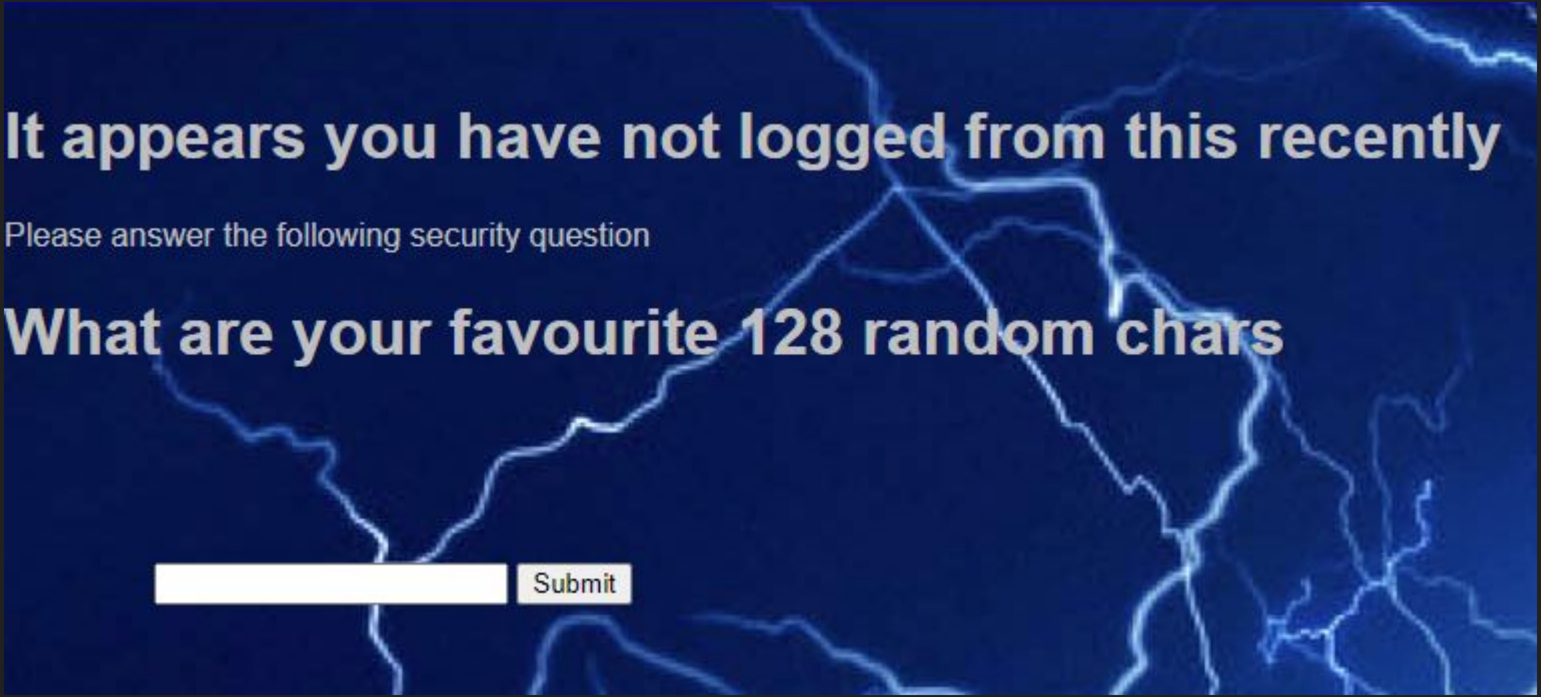
Password: hunter2



```
POST /login HTTP/1.1
Host: giga.montrehack.ca
Connection: keep-alive
Content-Length: 45
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://giga.montrehack.ca
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWe
Accept: text/html,application/xhtml+xml,application
Referer: http://giga.montrehack.ca/login
Accept-Encoding: gzip, deflate
Accept-Language: fr-CA,fr-FR;q=0.9,fr;q=0.8,en-US;q
Cookie: session=eyJsb2Nrb3V0X3RpbWUiOjE1OTUzMzk5NzM

email=superadmin%40alphamail.ctf&pass=hunter2HTTP/1
```

Step 2: Visit the website



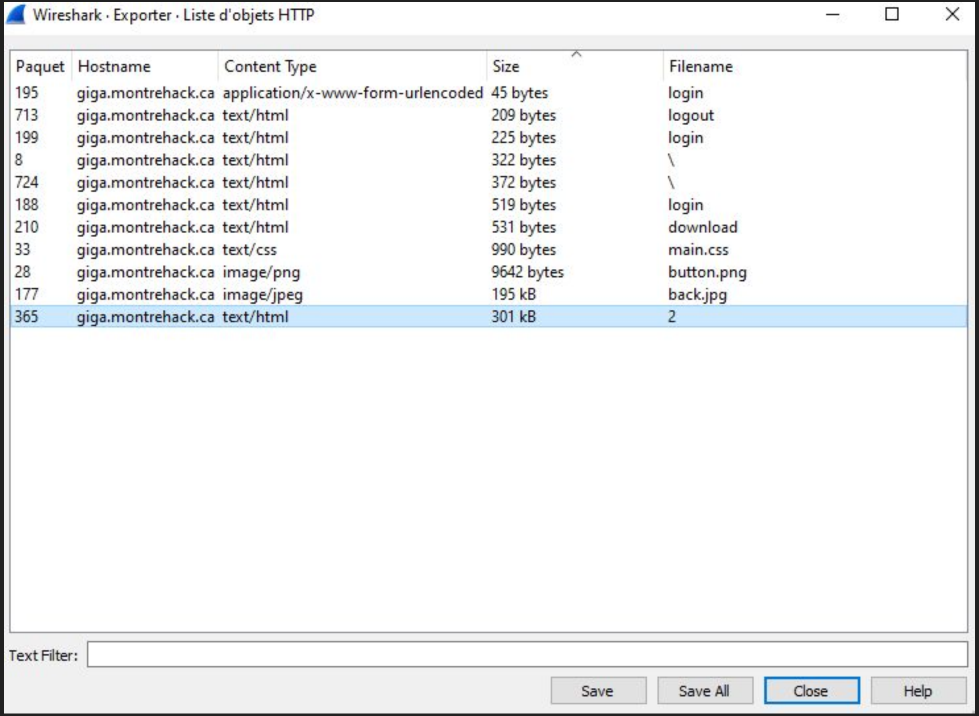
It appears you have not logged from this recently

Please answer the following security question

What are your favourite 128 random chars

Step 3: Look further

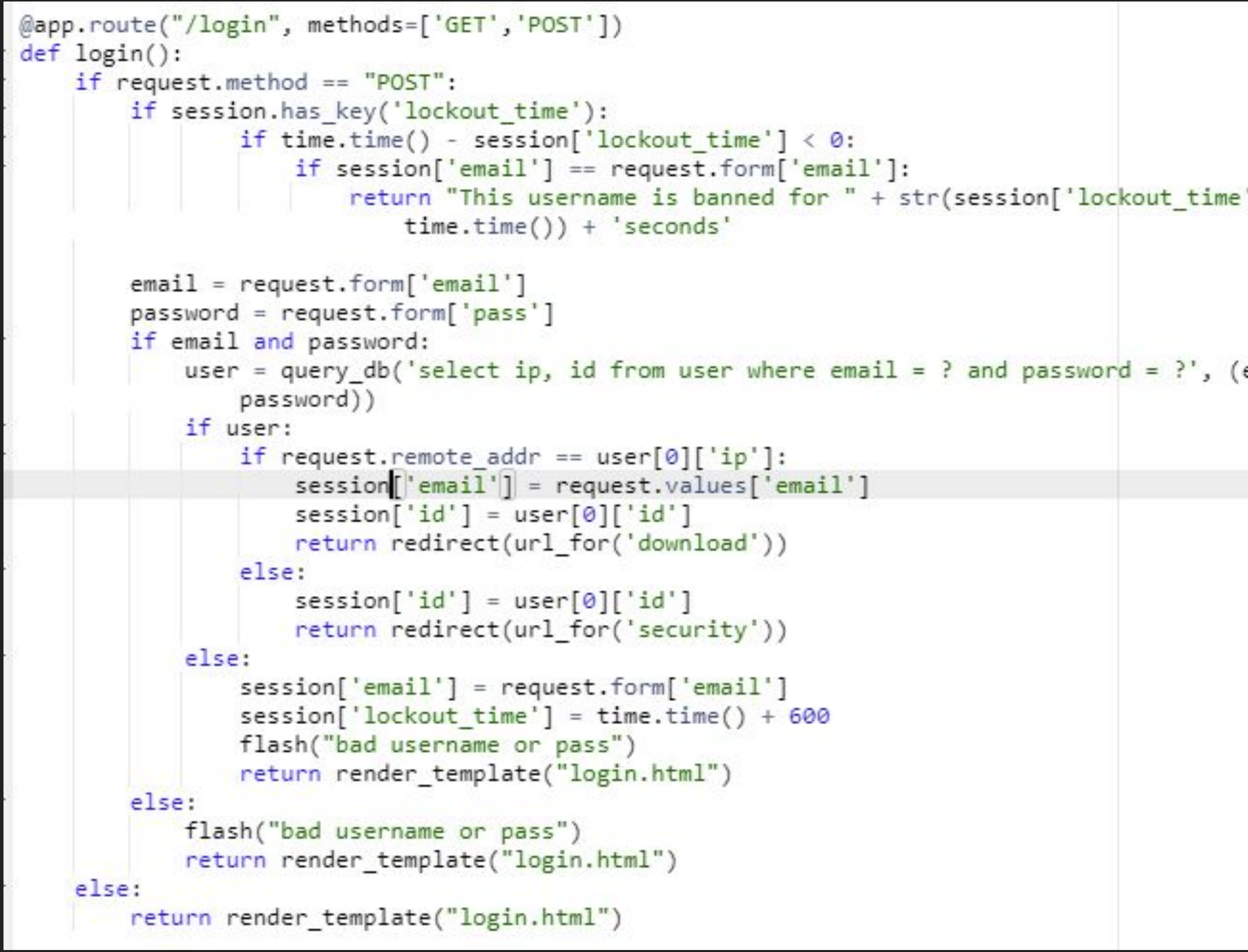
You can extract the source code in the pcap, using wireshark.



Wireshark · Exporter · Liste d'objets HTTP

Paquet	Hostname	Content Type	Size	Filename
195	giga.montrehack.ca	application/x-www-form-urlencoded	45 bytes	login
713	giga.montrehack.ca	text/html	209 bytes	logout
199	giga.montrehack.ca	text/html	225 bytes	login
8	giga.montrehack.ca	text/html	322 bytes	\
724	giga.montrehack.ca	text/html	372 bytes	\
188	giga.montrehack.ca	text/html	519 bytes	login
210	giga.montrehack.ca	text/html	531 bytes	download
33	giga.montrehack.ca	text/css	990 bytes	main.css
28	giga.montrehack.ca	image/png	9642 bytes	button.png
177	giga.montrehack.ca	image/jpeg	195 kB	back.jpg
365	giga.montrehack.ca	text/html	301 kB	2

4 : Read the code



```
@app.route("/login", methods=['GET','POST'])
def login():
    if request.method == "POST":
        if session.has_key('lockout_time'):
                if time.time() - session['lockout_time'] < 0:
                    if session['email'] == request.form['email']:
                        return "This username is banned for " + str(session['lockout_time'
                            time.time()) + 'seconds'

        email = request.form['email']
        password = request.form['pass']
        if email and password:
            user = query_db('select ip, id from user where email = ? and password = ?', (e
                password))
            if user:
                if request.remote_addr == user[0]['ip']:
                    session['email'] = request.values['email']
                    session['id'] = user[0]['id']
                    return redirect(url_for('download'))
                else:
                    session['id'] = user[0]['id']
                    return redirect(url_for('security'))
            else:
                session['email'] = request.form['email']
                session['lockout_time'] = time.time() + 600
                flash("bad username or pass")
                return render_template("login.html")
        else:
            flash("bad username or pass")
            return render_template("login.html")
    else:
        return render_template("login.html")
```

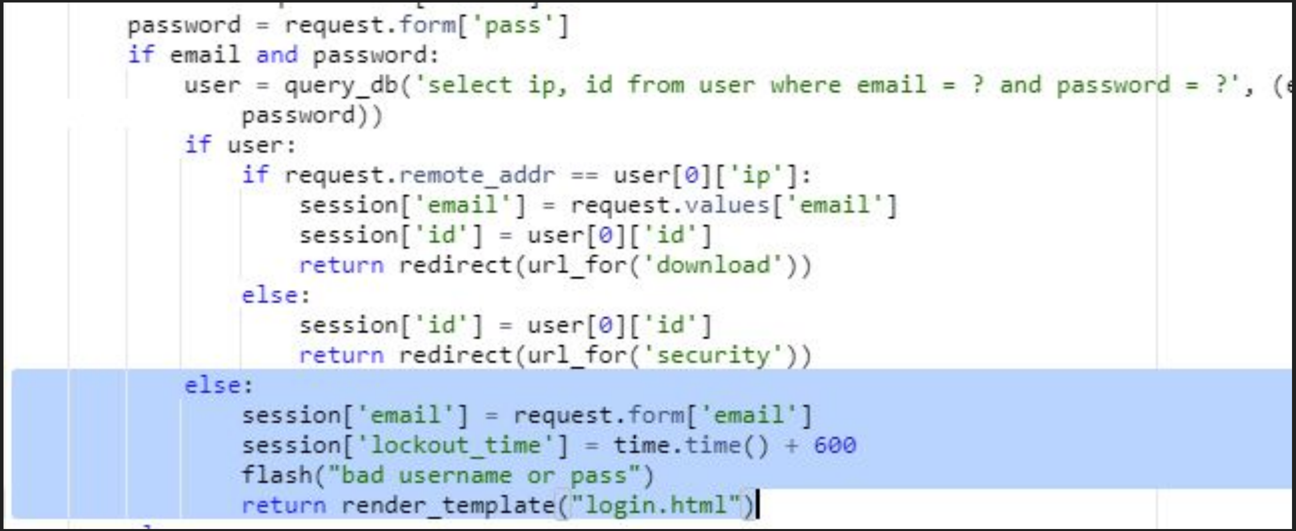
4b: Email in session

So, if i have a email in my session, I am able to bypass the secret question.

Idea 1: Un-base64 the session, add a ‘email’ function ✗

What else?

4c: How to add email in session?



```
password = request.form['pass']
if email and password:
    user = query_db('select ip, id from user where email = ? and password = ?', (e
        password))
    if user:
        if request.remote_addr == user[0]['ip']:
            session['email'] = request.values['email']
            session['id'] = user[0]['id']
            return redirect(url_for('download'))
        else:
            session['id'] = user[0]['id']
            return redirect(url_for('security'))
    else:
        session['email'] = request.form['email']
        session['lockout_time'] = time.time() + 600
        flash("bad username or pass")
        return render_template("login.html")
```

We lock ourselves! This way, the variable “Email” will be used.

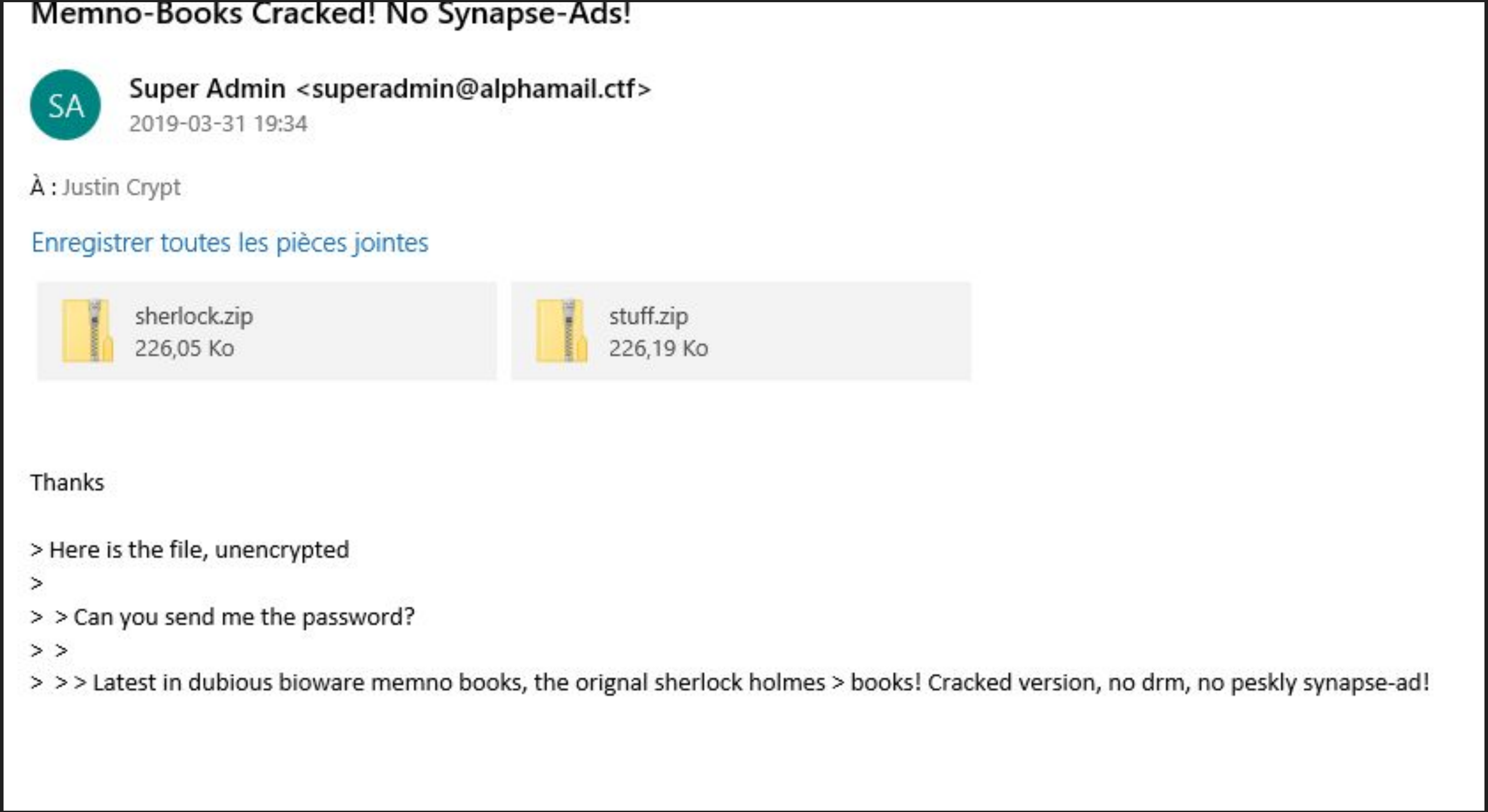
Demo

Bonus

Bonus challenge: Stuff.zip

In the pcap, there is another file.

You can obtain this file from wireshark, with a similar technique.


Memno-Books Cracked! No Synapse-Ads!

SA

Super Admin <superadmin@alphamail.ctf>
2019-03-31 19:34

À : Justin Crypt

Enregistrer toutes les pièces jointes

sherlock.zip
226,05 Ko

stuff.zip
226,19 Ko

Thanks

> Here is the file, unencrypted
>
> > Can you send me the password?
> >
> > > Latest in dubious bioware memno books, the orignal sherlock holmes > books! Cracked version, no drm, no peskly synapse-ad!

Decrypting stuff.zip

In the zip file, you will see 2 files:
Pg1691.txt and flag.txt

Both files are protected by a password, which we dont know :(

How can we know the password?

What do we know?

We also know the content of the file pg1691.txt. The sender was kind enough to send it in an unencrypted form.

Solution: Known plaintext attack

In a (legacy) zip file, you can decrypt the whole content of a zip file if you have a partial knowledge of the content on the file.

There is a tool to do this: pkcrack (https://github.com/keyunluo/pkcrack)

Syntax:

```
pkcrack -C stuff.zip -c pg1661.txt -P sherlock.zip -p
pg1661.txt -d FLAG_HERE -a
```