Exercise 07: Brute Forcing Glassfish

Jan-Matthias Braun

28. of September, 2022

Penetration testing exercise using metasploit on Glassfish. The exercise targets the Glassfish application server running under Windows in the metasploitable~3 image. The test is performed via a brute force attack against the login page.

In this exercise we need the Kali machine (for penetration testing) and the vulnerable machine is the Metasploitable3 Windows image. It is the only exercise to be performed against the Windows image and the last metasploitable exercise of the course.

Please note that all questions (full lines) marked with † should be answered in the report which you submit as assignment 2. Note on the length of answers: Questions which are described as *discussion* or *open* target an appropriate level of elaboration as you see fit, whereas the other questions can typically be answered with a sentence. Nonetheless, there are no length requirements on the answers.

Background

What is Glassfish? Its an open-source application server (Wikipedia on GlassFish). We are going to try to see if we can hack into Glassfish, which is listening on port 4848 (e.g., https://10.0.2.10:4848/). Glassfish uses https, does this secure the service? Perhaps. But...

Question: †what does https actually provide protection for? Consult the Wikipedia article on HTTPS.

While scanning for vulnernable services in Exercise~03, *Glassfish* running on port 4848 should have come up. Besides exploits for specific vulnerabilities, there are always ways to target services in a generic way. How about the possibility of, for example, brute forcing access to gain login information?

Typically, you would dig through known vulnerabilities using, e.g., use *Nessus*, *GVM*, or *metasploit*, *searchsploit* on the terminal, or online resources like exploitdb and the CVE.

Depending on the degree of automation and the range of your penetration test, this can take hours or even days. This is the same with most services, vulnerabilities and exploits you will encounter in the wild.

To make our lives easier, we'll use a metasploit exploit to gain login information for Glassfish. On other applications, websites, etc., an attacker would need different Metasploit modules to perform similar tasks. And of course, there are other tools available, e.g., hydra, which has the nice feature of allowing parallelised brute force attacks. Independently of the tool, the approach is typically the same and can be used for other applications, websites, etc (especially if they have a username). So in essence, the approach will give you a template for how an attacker may be using brute force attacks against other applications, websites, and so on, to gather user information. It will also highlight three important takeaway messages:

1. How social engineering is useful for such kind of attacks (and how its important to try to protect yourself from social engineering)
2. The importance of using secure passwords.
3. The option of hiding user names in official communications and documents.

Enough yapping—let's do this.

Brute Force Attack

As announced, we are going to use a Metasploit exploit from our Kali VM. Power up Kali and the Metasploitable3 Windows image. In Kali lets get into Metasploit. On the terminal, type `msfconsole`. In metasploit, let's search for exploits targeting Glassfish:

```
msf> search Glassfish
```

Today, we are interested in the module: "auxiliary/scanner/http/glassfish_login". It is a GlassFish brute force utility. We will try to use it to get login information for Glassfish. It will try to log in using a list of usernames and passwords that you supply it with. These lists are read from files which are called dictionaries. This way of brute force attack is therefore also often called a dictionary attack. We need to create two files:

user.txt that will have usernames that we hope that at least one is used for login into Glassfish and

pass.txt that will contain passwords that we hope that at least one is used for login into Glassfish

Now work with me here because I will try to make it a bit more realistic. In our two files, we will supply the exploit with passwords and usernames based on our social engineering and digging around for common passwords and usernames. (Hypothetically) We have done some information gathering and social engineering targeting the machine and we know that it is a customized server based on Windows that is called Metasploitable. We also do some research and we know that there are some common usernames and passwords that people often use: e.g., admin, password, Password123 etc. A bit of searching online and you can find sources that show common usernames and passwords. For example, Wikipedia has a list of common passwords and also a list of 10 000 most common password. Relevant lists are generated from leaked password databases.

Thus, the principle is to try common passwords and usernames, adding our own candidates to the list based on our social engineering efforts for this Windows machine that is running Glassfish.

First, let's see which directory you are in. When typing in `pwd` in metasploit, you will likely see that you are in `/home/kali`. If you are in a different directory—that's fine. Just type in the following: `cd /home/kali`. This is to make sure that we are all in the same directory in case you need help. Of course, having all files we are using in the home directory will not result in a tidy directory structure, but it will make our lives easier because we need to type less.

```
msf > pwd
[*] exec: pwd

/home/kali
msf >
```

Open another terminal and while staying in your home directory, type in the following to create an empty *pass.txt* file:

```
gedit pass.txt
```

To edit the file with *gedit*. In case *gedit* is not installed or you do not like it, you can use any other regular text editor by navigating to the folder where the files are, for example *kate*. Of course, you can also fire up your favourite editor and then save the file to the proper name. The instructions above are for starting a the *gedit* editor from the terminal. You can use whatever works best for you, for example text terminal based alternatives like *nano*, *emacs*, or *vim*. Search online for *flame war emacs vs vim* to enable you to make an informed choice (cmp. this Wikipedia article). :-) On a well configured system, `$EDITOR pass.txt` should also work nicely.

Then type in the following list and save. Btw - you can also do a simple copy from here and paste in Kali with your cursor if you have enabled the shared clipboard for your Kali VM (see notes below).

```
1234567
password
Password
Password123
p455w0rd
P455w0rd
admin
Admin
Server
Starwars
starwars
metasploitable
Metasploitable
Metasploit
metasploit
meta
Meta
sploit
Sploit
```

To save in *gedit*, use the menu or *CTRL—S*. When using *nano*, press *CTRL-X*, then type *Y*, and then hit enter.

Let's do the same for the username file. Create a file from the terminal:

```
gedit user.txt
```

and add usernames:

```
111111
123456
12345678
abc123
abramov
account
accounting
admin
Admin
Administrator
user
User
vagrant
Vagrant
```

```
admin
Admin
metasploitable
Metasploitable
metasploitable3
Metasploitable3
Metasploit
metasploit
meta
Meta
sploit
Sploit
```

Again, save the file and exit the editor.

Okay now you should have user.txt and pass.txt files. Let's go back to Mesaploit and use the `auxiliary/scanner/http/glassfish_login` module. Use `show options` to see all relevant parameters for this exploit and set them appropriately.

Then run `show options` again to double-check that the information that is shown is correct, e.g., that you have set the correct IP and PORT values that correspond to the victim (which is your Metasploitable3 Windows VM).

Well, let's go ahead and try to brute force and see if we can find login information for this Glassfish instance. Type in "exploit" and hit enter... and let Metasploit do the work for you...

```
msf auxiliary(scanner/http/glassfish_login) > exploit
```

Now sit back and wait. Metaploit will test all combinations of the username and password dictionaries provided in `user.txt` and `pass.txt`. Please note that in real-word situations you could be waiting for a long time depending on how long your username and password dictionaries are. The quality of your dictionary files determines the success chance of your brute force attack.

You will notice a bunch of messages indicating that a specific combination of username and password has failed. When Metasploit is done, scroll back up slowly and see if there is one that was a success.

And there you go. You have found usable login information. Go ahead and try to login from the browser in Kali: https://10.0.2.10:4848/ and go ahead and snoop around.

Congrats! You may now inform the administrator of Metasploitable3-Windows image of the security issue.

Questions:

† Which username/password combination did you find?

† Discuss which security relevant problems are we testing with a brute force attack?

† Discuss what would be your suggestions to the admin in order to address and mitigate this issue?

† How is this attack type related to the internet of things, internet routers, and, e.g., virtual machines?

† Do you know a way in which https could make the connection more secure against this kind of attack?

Notes

Saving metasploit's output In case you do not want to scroll back through the metasploit output, given large enough dictionaries, this will be difficult, use `spool filename.txt` to have all output being copied to the file *filename.txt* in the current directory. Use `spool off` to disable logging.

Reducing the output Use set VERBOSE false to only log successful brute force results.

Hidden menu bar In case you have hidden the menu bar in virtual box, you can use *CTRL—c* to show it again. This way you can easily access the shared clipboard setting.

Shared clipboard VirtualBox allows to share the copy & paste buffer between virtual machines and host OS. In the menu bar, select "Devices" -> "Shared Clipboard" -> "Bidirectional", or any other direction of choice. Note, that this way also passwords in the clipboard may be readable from the VM. Alternatively, in the VirtualBox program, select your VM, choose "Settings", "General", "Advanced" and there the directionality of clipboard transfer.

Copy & paste under Linux Note that selected text typically already is copied under Unix systems. Use the middle mouse button (or both outer buttons simultaneously) to paste.