Computer Security

1.

Fuzzing: Fuzz testing (fuzzing) is a quality assurance technique used to discover coding errors and security loopholes in software, operating systems or networks. It involves inputting massive amounts of random data, called fuzz, to the test subject in an attempt to make it crash. If a vulnerability is found, a software tool called a fuzzer can be used to identify potential causes. [1]

Fuzzing tool used: BurpSuite Community Edition
Web Application used: bWapp
Attack/ Bug Hacked: SQL Injection

Setup VM 1

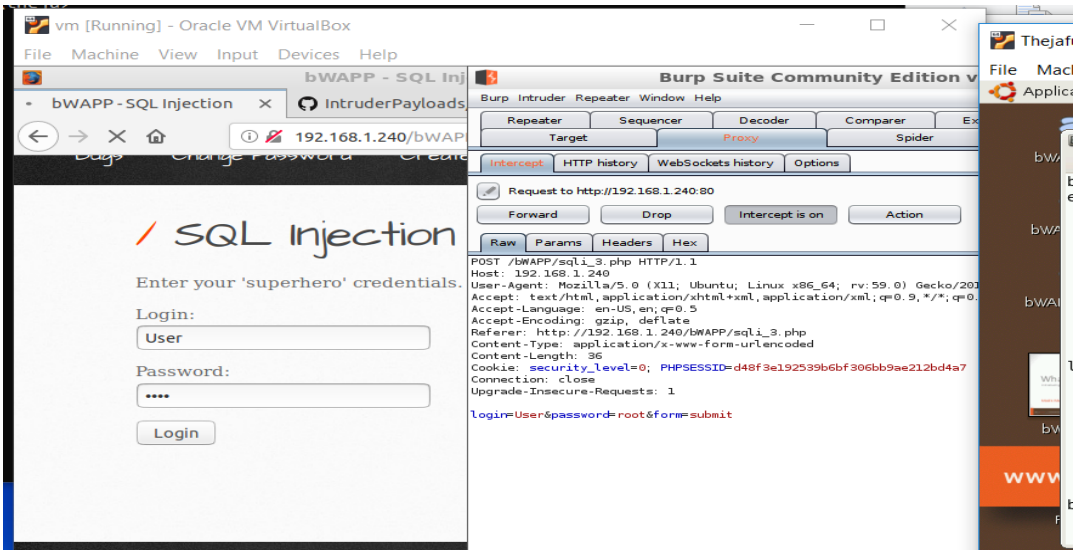
- bWapp was downloaded and hosted on a VM (Ubuntu 64-bit).
- Network type was set to Bridged Adapter.
- IP was noted

Setup VM 2

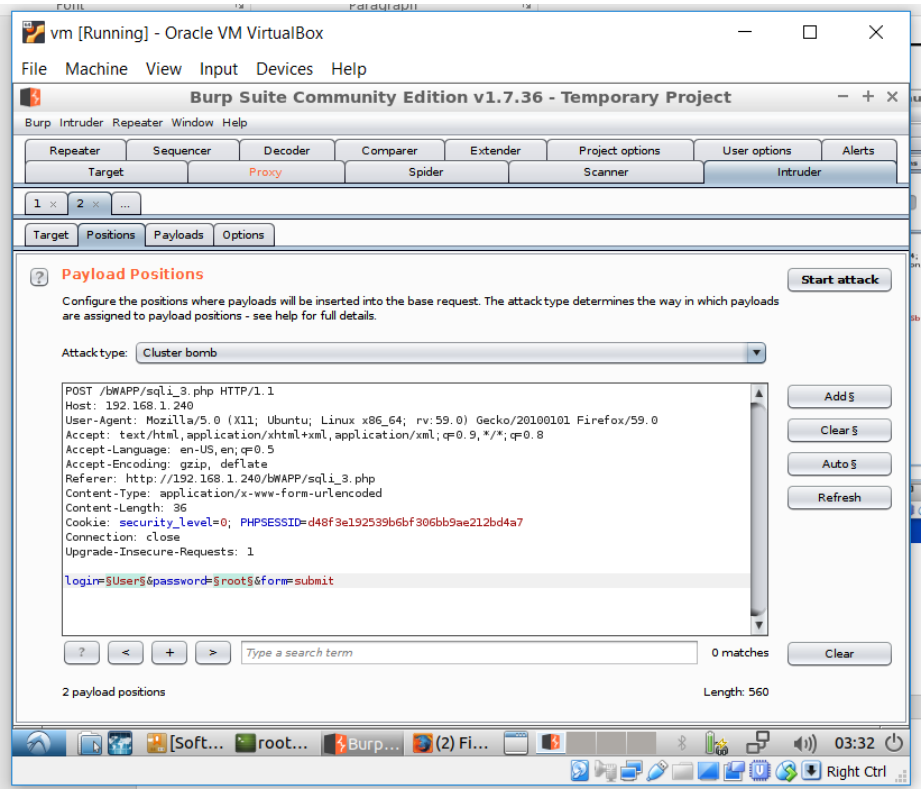
- Network type was set to Bridged Adapter.
- bWapp web application was accessed in another VM (Ubuntu 64-bit) using the URL IP/bWapp/login.php
- Login Credential: bee/bug
- Bug “SQL Injection” was selected from the dropdown to hack.
- BurpSuite Community Edition was downloaded and launched (Intercept was on to capture GET and POST requests.)
- Mozillla Firefox was set to Manual Proxy

Steps for SQL Attack:

Tried to login by inputting a Username and Password (User and root was tried) essentially to capture the request.



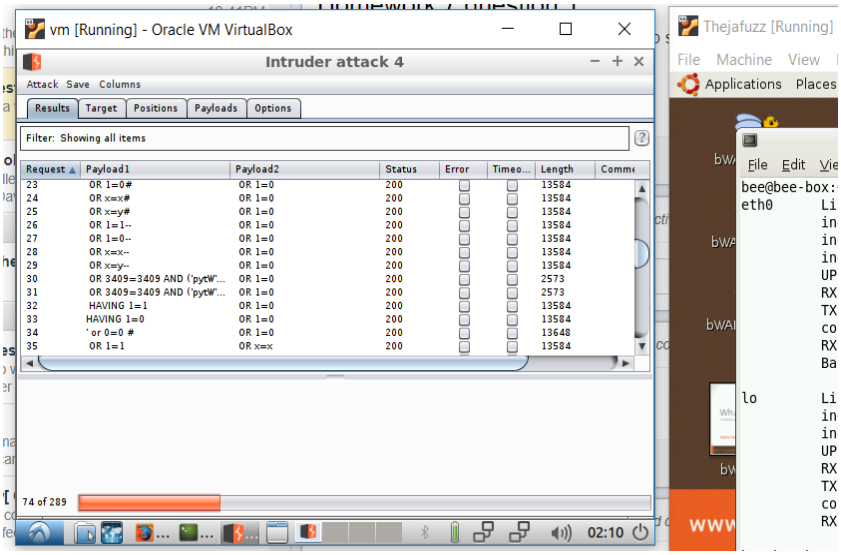
The captured request was sent to Intruder and the attack was changed to cluster bomb, with 2 payloads the username and password.



Intruder Payload was taken from the following specified github repository:

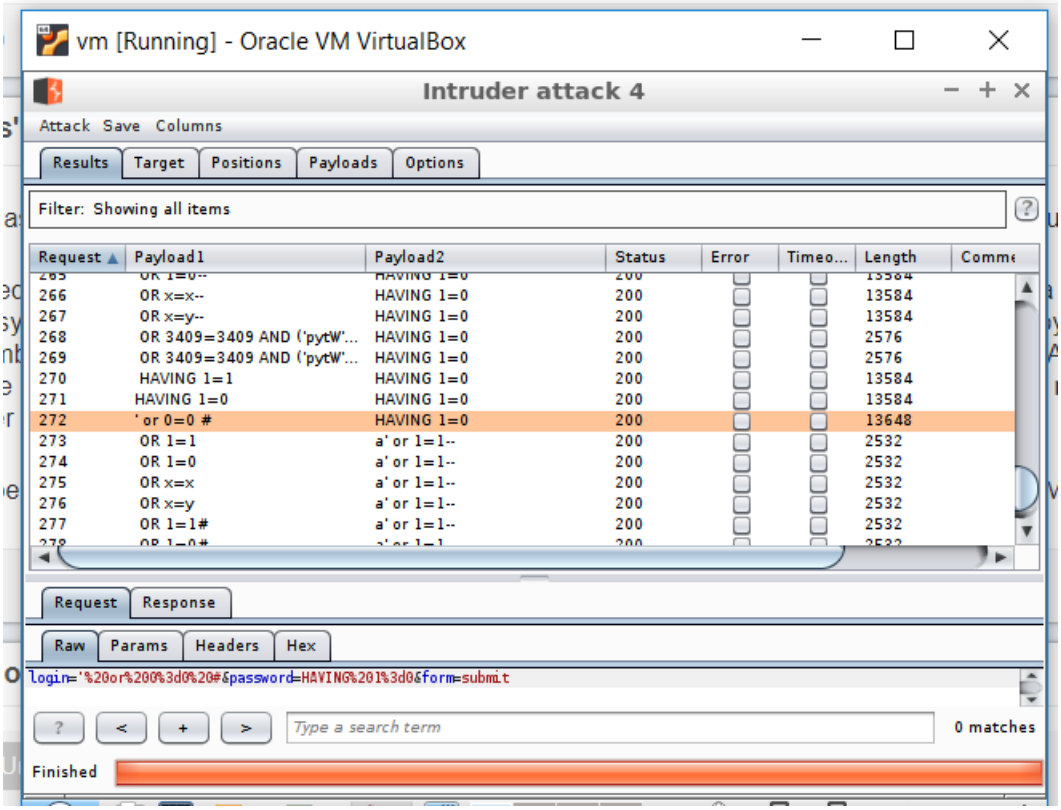
https://github.com/1N3/IntruderPayloads/tree/master/FuzzLists [2]

After payload simple list was added to both the payloads, attack was started.

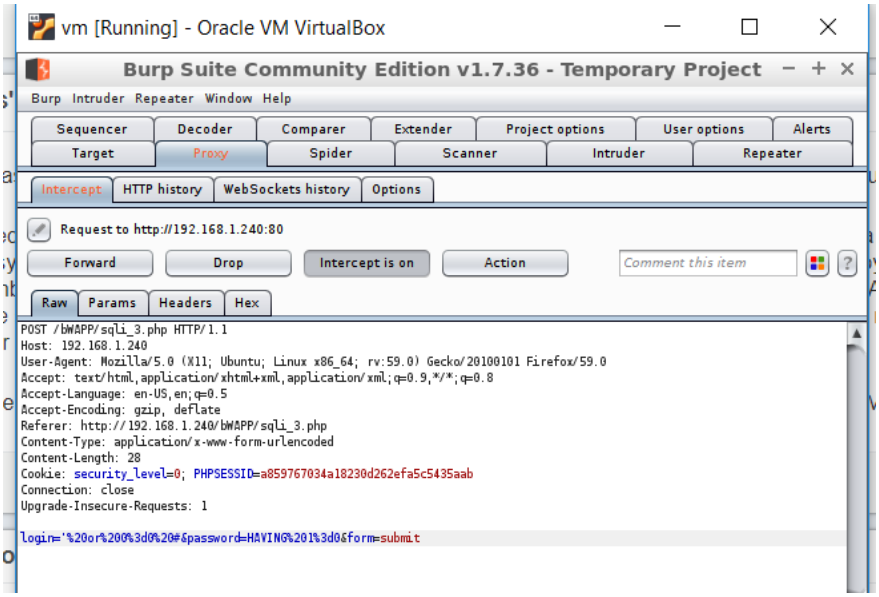


Observations:

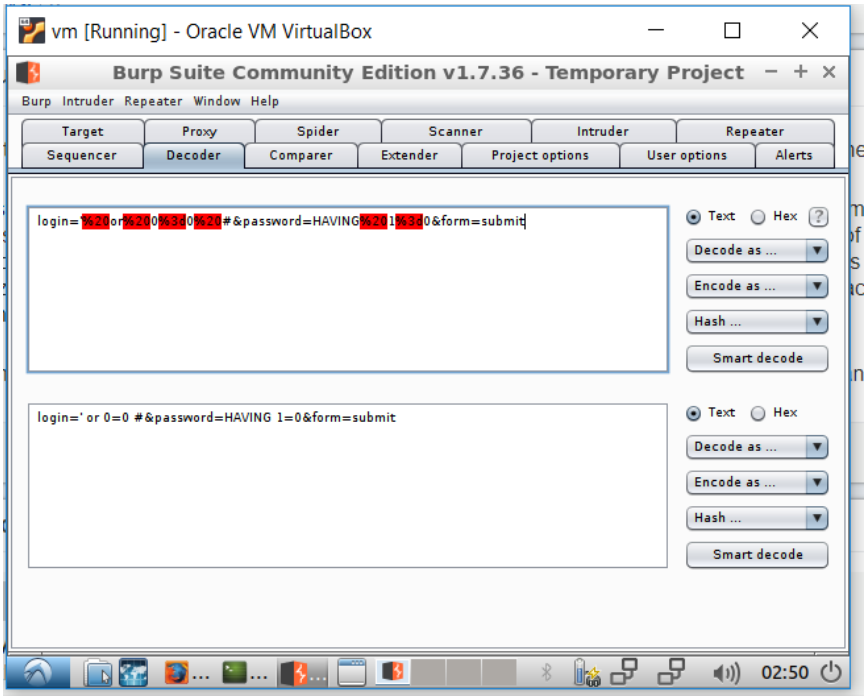
The length change should be observed and the one with major difference suggests that there is something abnormal happening at that point.



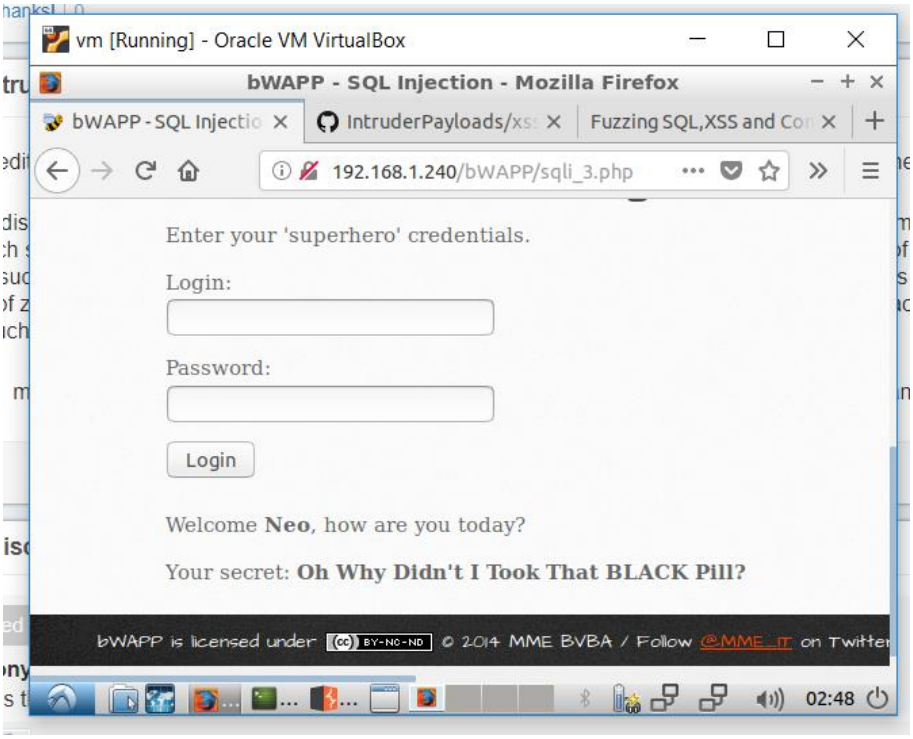
The login, password from the raw data for the highlighted shows major difference, so I use that as the input and try to perform the attack.



Decoding the above to verify the condition

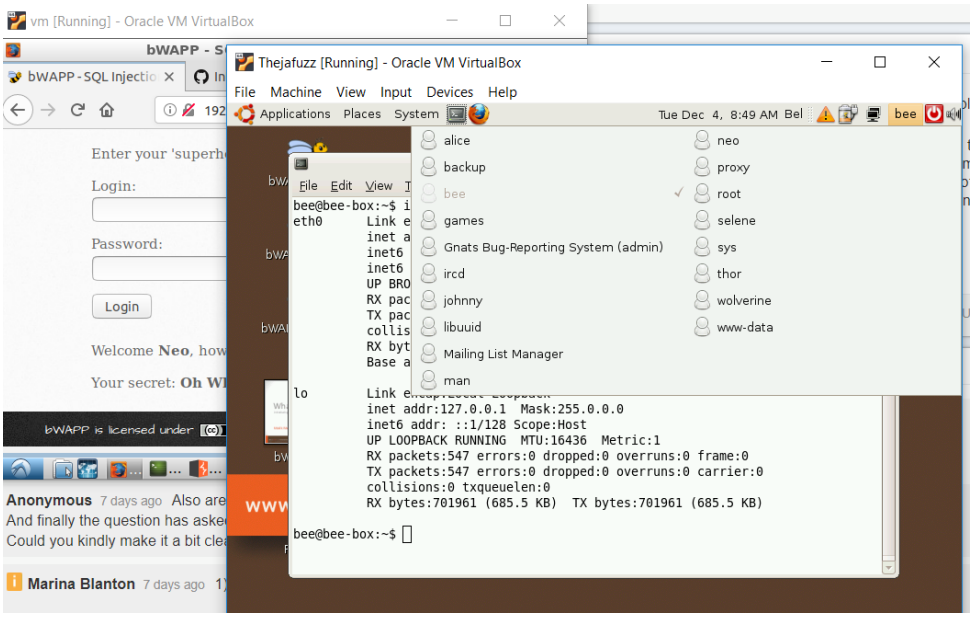


This is now forwarded and if we get in, the attack was successful.



The above picture shows that we have logged into the site, without knowing the user name or password using fuzzing of so many combinations and by observing the change in its behavior.

To verify if there is a user called neo, I checked the bWapp hosted VM for the users.



2 * 1024 * 1024/ (32*4000) = **16.384**
Therefore 16 systems are required

For 10 Mbps:
10 * 1024 * 1024/ (32*4000) = **81.92**

[1] https://searchsecurity.techtarget.com/definition/fuzz-testing

[2] https://github.com/1N3/IntruderPayloads/tree/master/FuzzLists