Lab9 – Mod_security

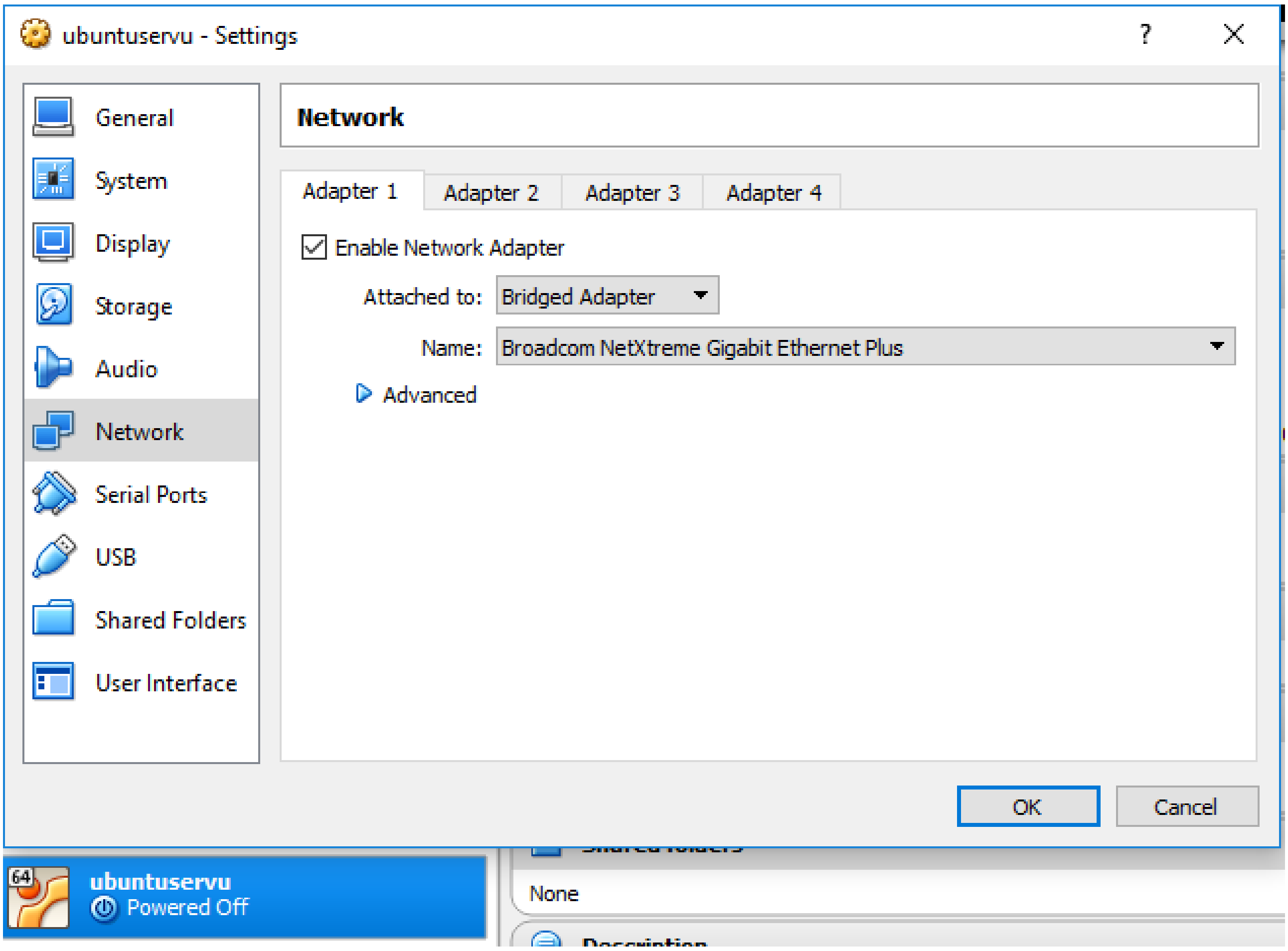
Document your commands or take screenshots. Answer questions in english or finnish.

Use Ubuntu server on Virtual box. Use Bridged network adapter.

Main idea is to Install LAMP on server, then set it up so we can try injections on it.

Next we install mod_security and change settings so it doesn't allow injections.

Then we create custom form to submit data. After that we make custom rules to prevent usage of specific words.



ubuntuservu - Settings

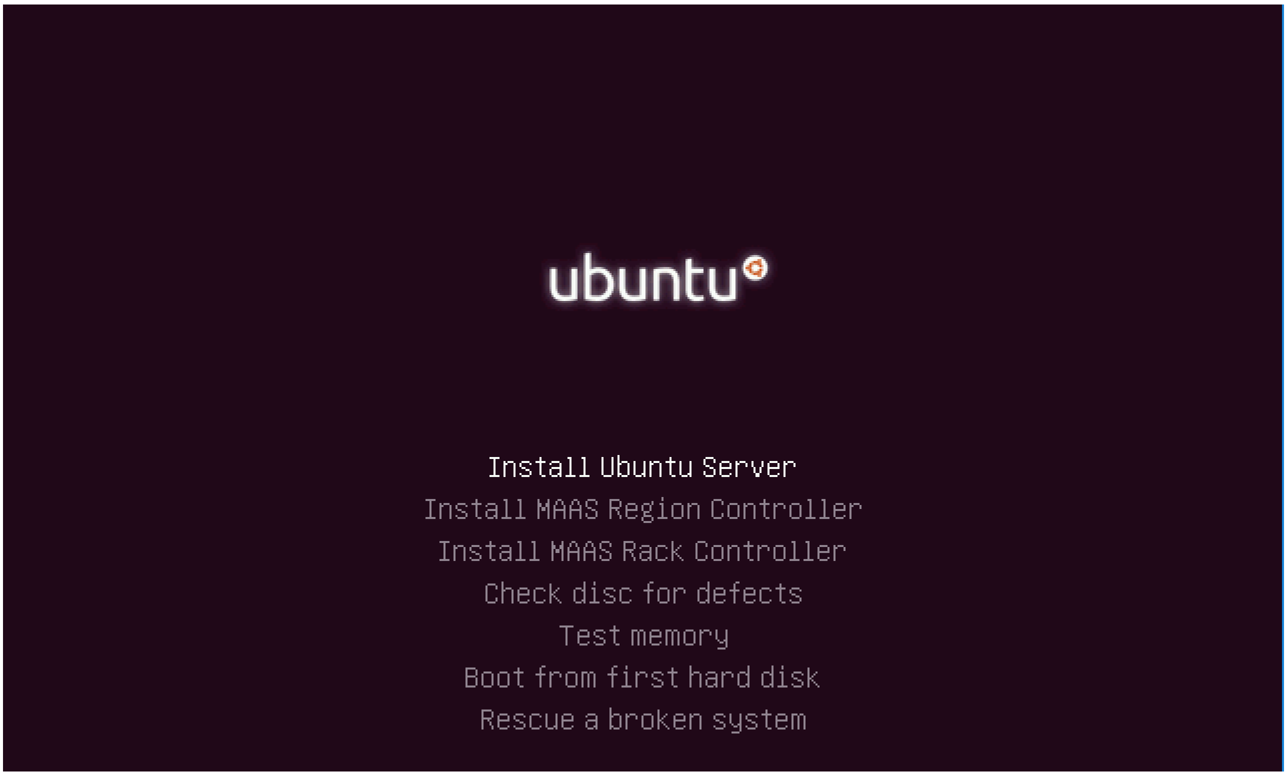
Network

Adapter 1 | Adapter 2 | Adapter 3 | Adapter 4

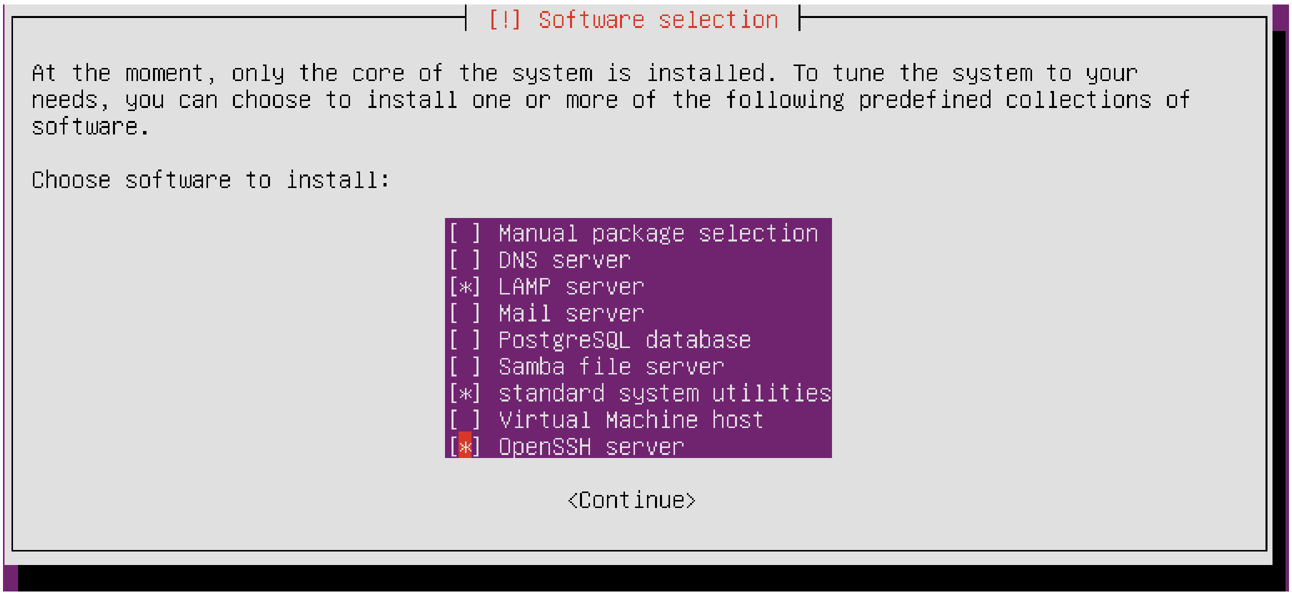
☑ Enable Network Adapter

Attached to: Bridged Adapter

Name: Broadcom NetXtreme Gigabit Ethernet Plus



ubuntu
Install Ubuntu Server
Install MAAS Region Controller
Install MAAS Rack Controller
Check disc for defects
Test memory
Boot from first hard disk
Rescue a broken system



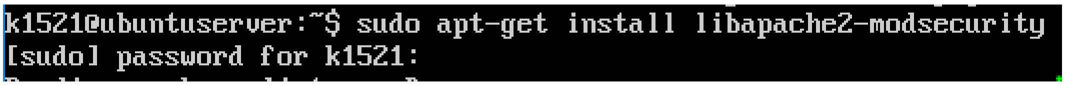
[!] Software selection

At the moment, only the core of the system is installed. To tune the system to your needs, you can choose to install one or more of the following predefined collections of software.

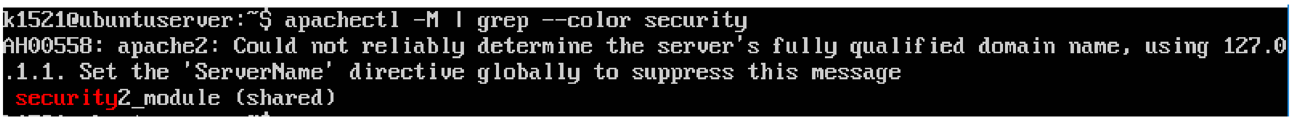
Choose software to install:

```
[ ] Manual package selection
[ ] DNS server
[*] LAMP server
[ ] Mail server
[ ] PostgreSQL database
[ ] Samba file server
[*] standard system utilities
[ ] Virtual Machine host
[*] OpenSSH server
```

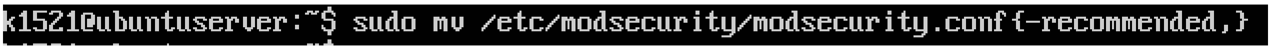
```
k1521@ubuntuserver:~$
```



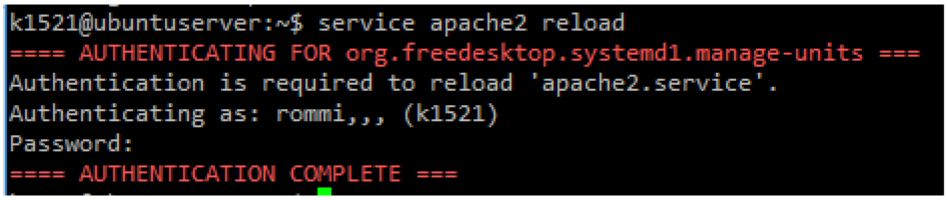
```
k1521@ubuntuserver:~$ sudo apt-get install libapache2-modsecurity
[sudo] password for k1521:
```



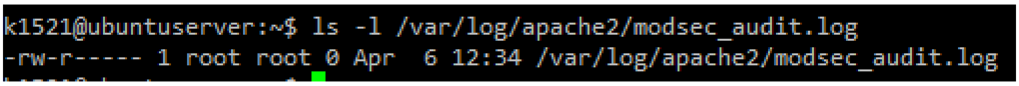
```
k1521@ubuntuserver:~$ apachectl -M | grep --color security
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0
.1.1. Set the 'ServerName' directive globally to suppress this message
 security2_module (shared)
```



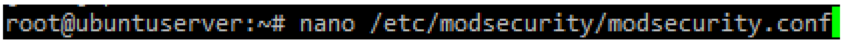
```
k1521@ubuntuserver:~$ sudo mv /etc/modsecurity/modsecurity.conf{-recommended,}
```



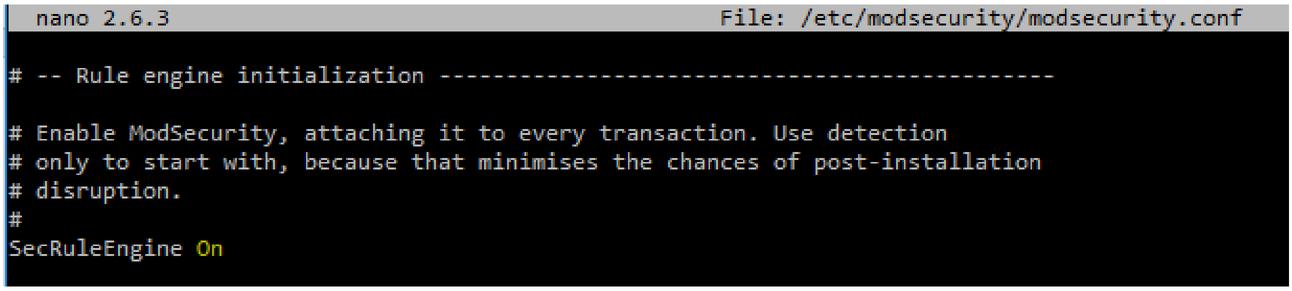
```
k1521@ubuntuserver:~$ service apache2 reload
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ===
Authentication is required to reload 'apache2.service'.
Authenticating as: rommi,,, (k1521)
Password:
==== AUTHENTICATION COMPLETE ===
```



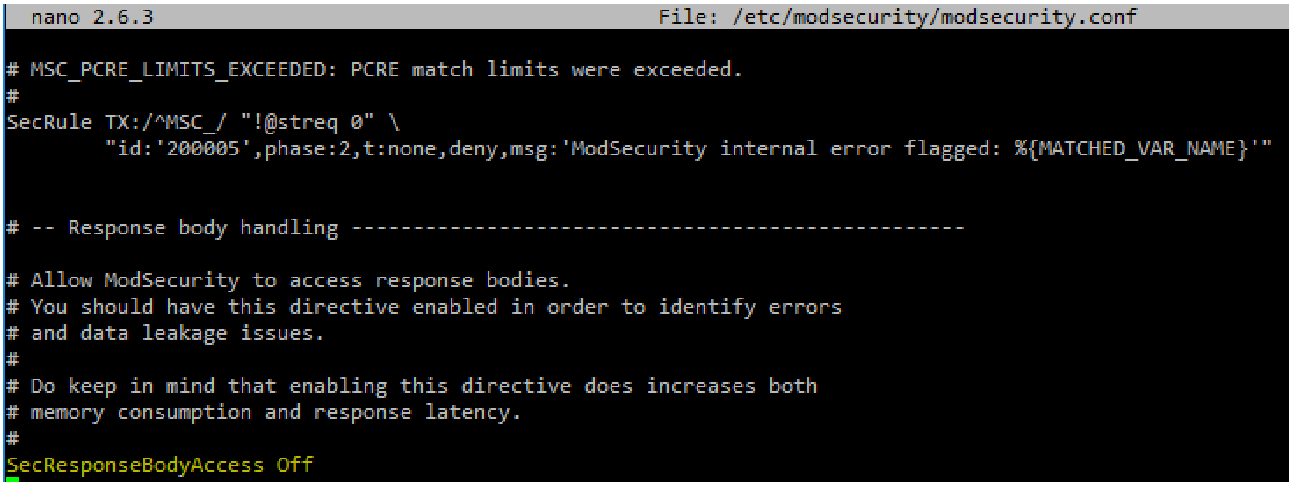
```
k1521@ubuntuserver:~$ ls -l /var/log/apache2/modsec_audit.log
-rw-r----- 1 root root 0 Apr  6 12:34 /var/log/apache2/modsec_audit.log
```



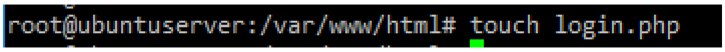
```
root@ubuntuserver:~# nano /etc/modsecurity/modsecurity.conf
```



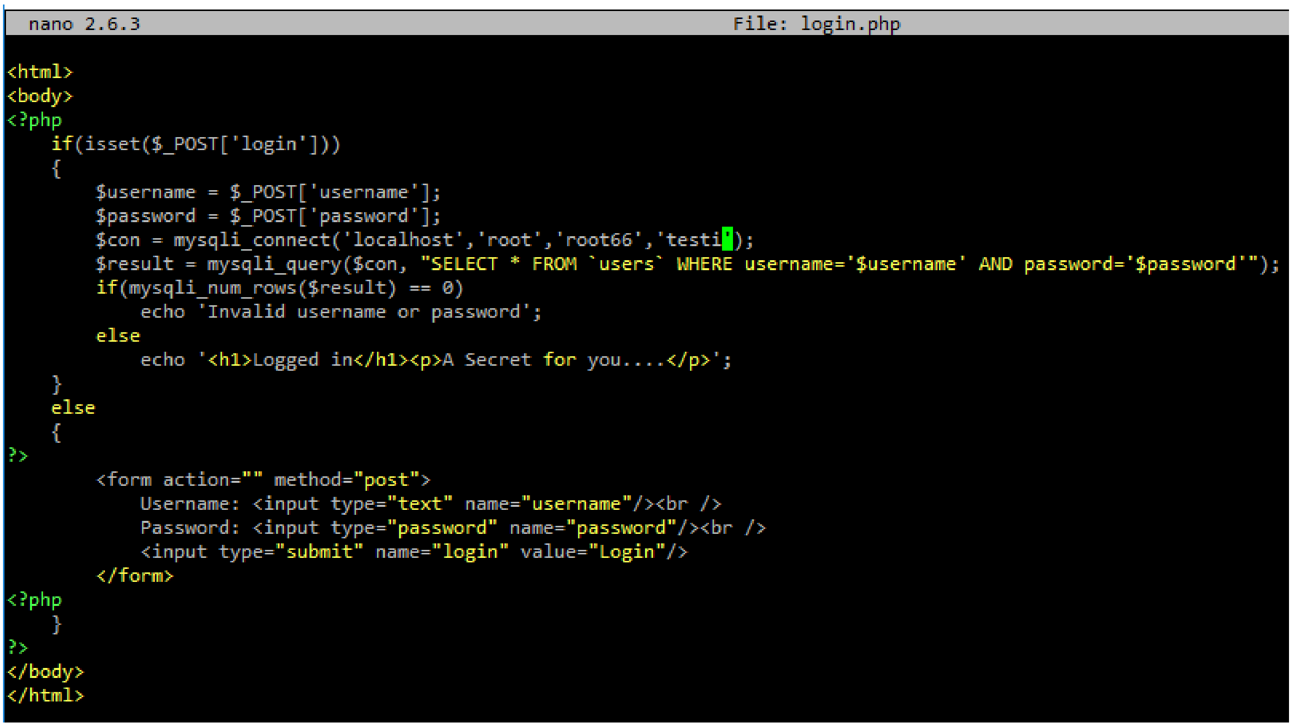
```
  nano 2.6.3                                File: /etc/modsecurity/modsecurity.conf

# -- Rule engine initialization ----------------------------------------------

# Enable ModSecurity, attaching it to every transaction. Use detection
# only to start with, because that minimises the chances of post-installation
# disruption.
#
SecRuleEngine On
```



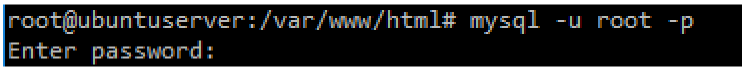
```
  nano 2.6.3                                File: /etc/modsecurity/modsecurity.conf

# MSC_PCRE_LIMITS_EXCEEDED: PCRE match limits were exceeded.
#
SecRule TX:/^MSC_/ "!@streq 0" \
        "id:'200005',phase:2,t:none,deny,msg:'ModSecurity internal error flagged: %{MATCHED_VAR_NAME}'"


# -- Response body handling --------------------------------------------------

# Allow ModSecurity to access response bodies.
# You should have this directive enabled in order to identify errors
# and data leakage issues.
#
# Do keep in mind that enabling this directive does increases both
# memory consumption and response latency.
#
SecResponseBodyAccess Off
```



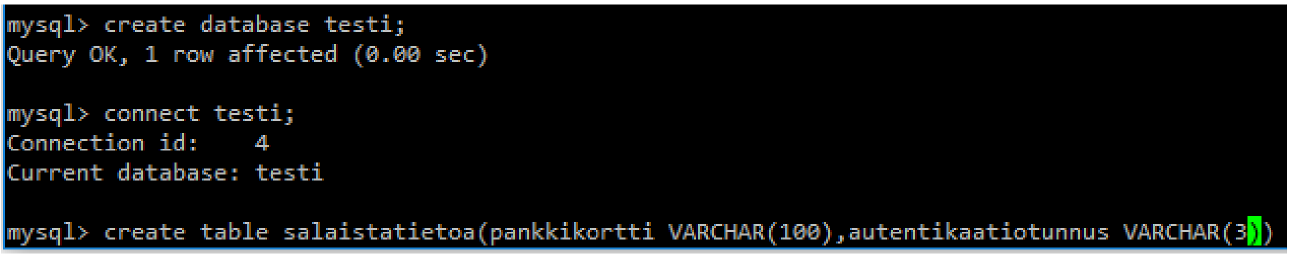
```
root@ubuntuserver:/var/www/html# touch login.php
```



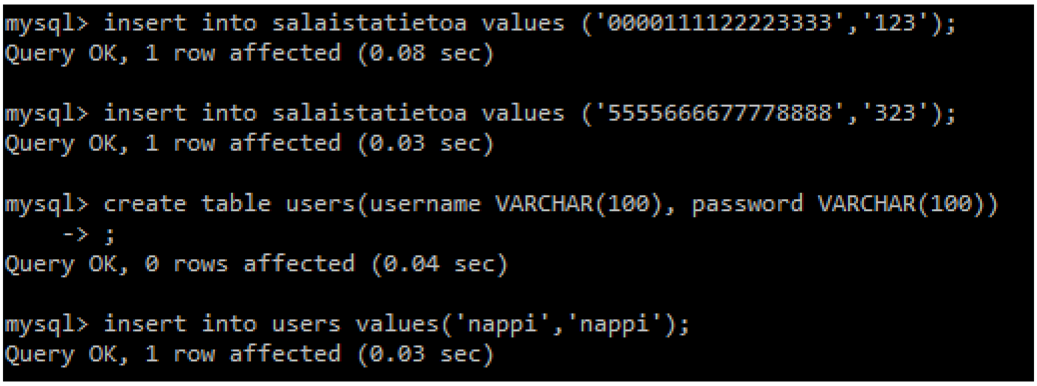
```
  nano 2.6.3                                                      File: login.php

<html>
<body>
<?php
    if(isset($_POST['login']))
    {
        $username = $_POST['username'];
        $password = $_POST['password'];
        $con = mysqli_connect('localhost','root','root66','testi');
        $result = mysqli_query($con, "SELECT * FROM `users` WHERE username='$username' AND password='$password'");
        if(mysqli_num_rows($result) == 0)
            echo 'Invalid username or password';
        else
            echo '<h1>Logged in</h1><p>A Secret for you....</p>';
    }
    else
    {
?>
        <form action="" method="post">
            Username: <input type="text" name="username"/><br />
            Password: <input type="password" name="password"/><br />
            <input type="submit" name="login" value="Login"/>
        </form>
<?php
    }
?>
</body>
</html>
```



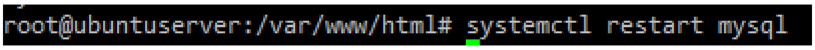
```
root@ubuntuserver:/var/www/html# mysql -u root -p
Enter password:
```



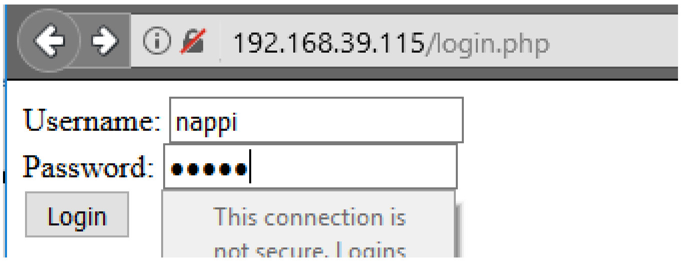
```
mysql> create database testi;
Query OK, 1 row affected (0.00 sec)

mysql> connect testi;
Connection id:    4
Current database: testi

mysql> create table salaistatietoa(pankkikortti VARCHAR(100),autentikaatiotunnus VARCHAR(3))
```



```
mysql> insert into salaistatietoa values ('0000111122223333','123');
Query OK, 1 row affected (0.08 sec)

mysql> insert into salaistatietoa values ('5555666677778888','323');
Query OK, 1 row affected (0.03 sec)

mysql> create table users(username VARCHAR(100), password VARCHAR(100))
    -> ;
Query OK, 0 rows affected (0.04 sec)

mysql> insert into users values('nappi','nappi');
Query OK, 1 row affected (0.03 sec)
```



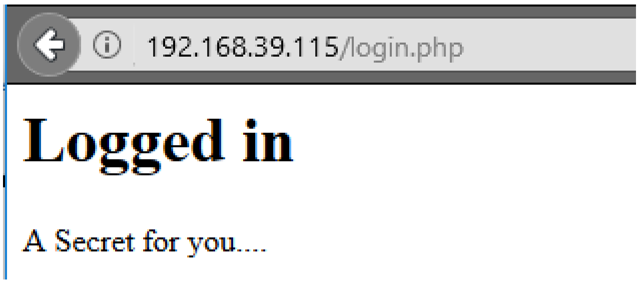
```
root@ubuntuserver:/var/www/html# systemctl restart mysql
```



Username: nappi

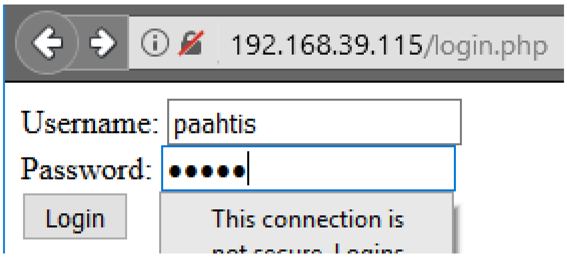
Password: •••••

This connection is not secure. Logins



Logged in

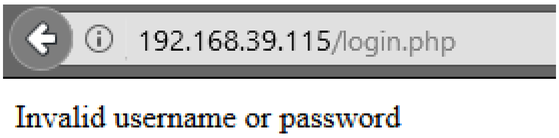
A Secret for you....



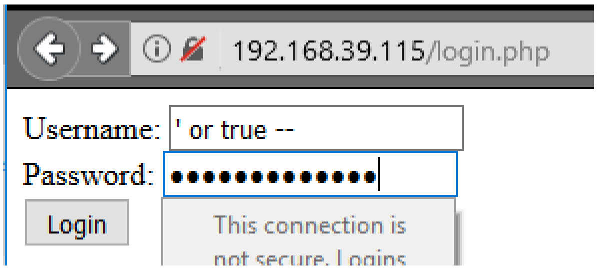
Username: paahtis

Password: •••••

This connection is not secure. Logins

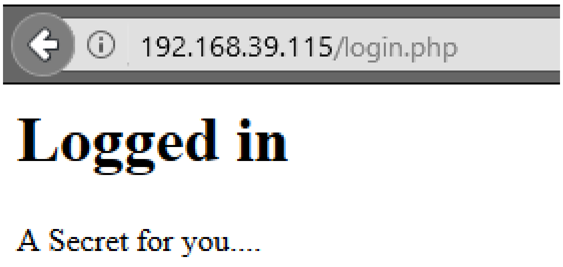


Invalid username or password



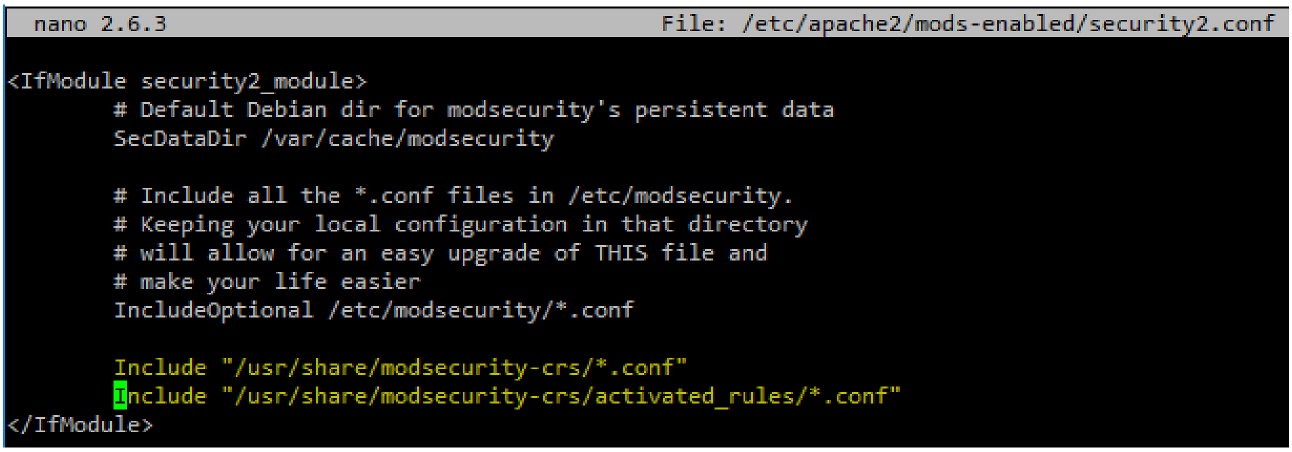
Username: ' or true --

Password: ●●●●●●●●●●●●●

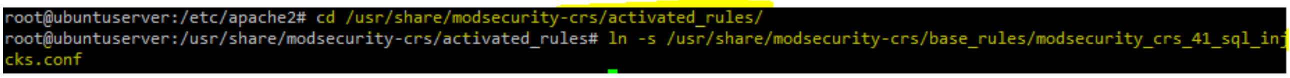


Logged in

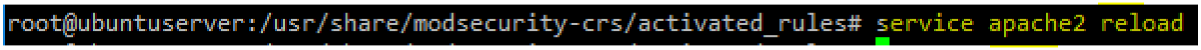
A Secret for you....



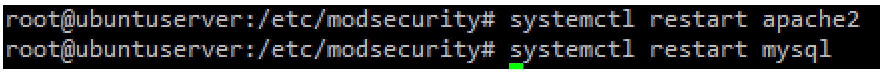
```
  nano 2.6.3                                        File: /etc/apache2/mods-enabled/security2.conf

<IfModule security2_module>
        # Default Debian dir for modsecurity's persistent data
        SecDataDir /var/cache/modsecurity

        # Include all the *.conf files in /etc/modsecurity.
        # Keeping your local configuration in that directory
        # will allow for an easy upgrade of THIS file and
        # make your life easier
        IncludeOptional /etc/modsecurity/*.conf

        Include "/usr/share/modsecurity-crs/*.conf"
        Include "/usr/share/modsecurity-crs/activated_rules/*.conf"
</IfModule>
```



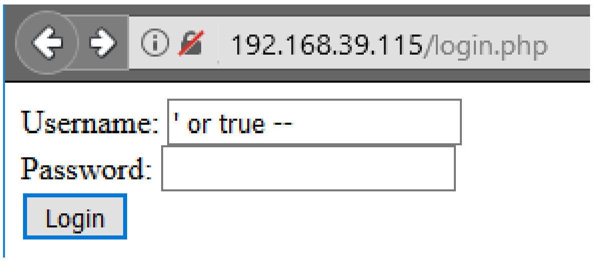
```
root@ubuntuserver:/etc/apache2# cd /usr/share/modsecurity-crs/activated_rules/
root@ubuntuserver:/usr/share/modsecurity-crs/activated_rules# ln -s /usr/share/modsecurity-crs/base_rules/modsecurity_crs_41_sql_inj
cks.conf
```



```
root@ubuntuserver:/usr/share/modsecurity-crs/activated_rules# service apache2 reload
```

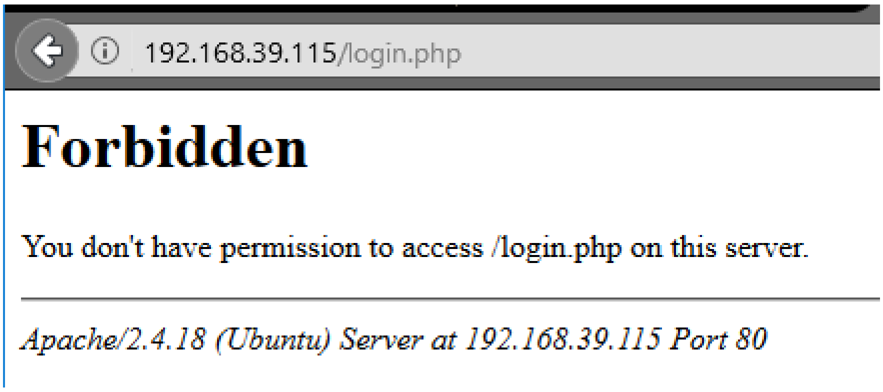


```
root@ubuntuserver:/etc/modsecurity# systemctl restart apache2
root@ubuntuserver:/etc/modsecurity# systemctl restart mysql
```



Username: ' or true --

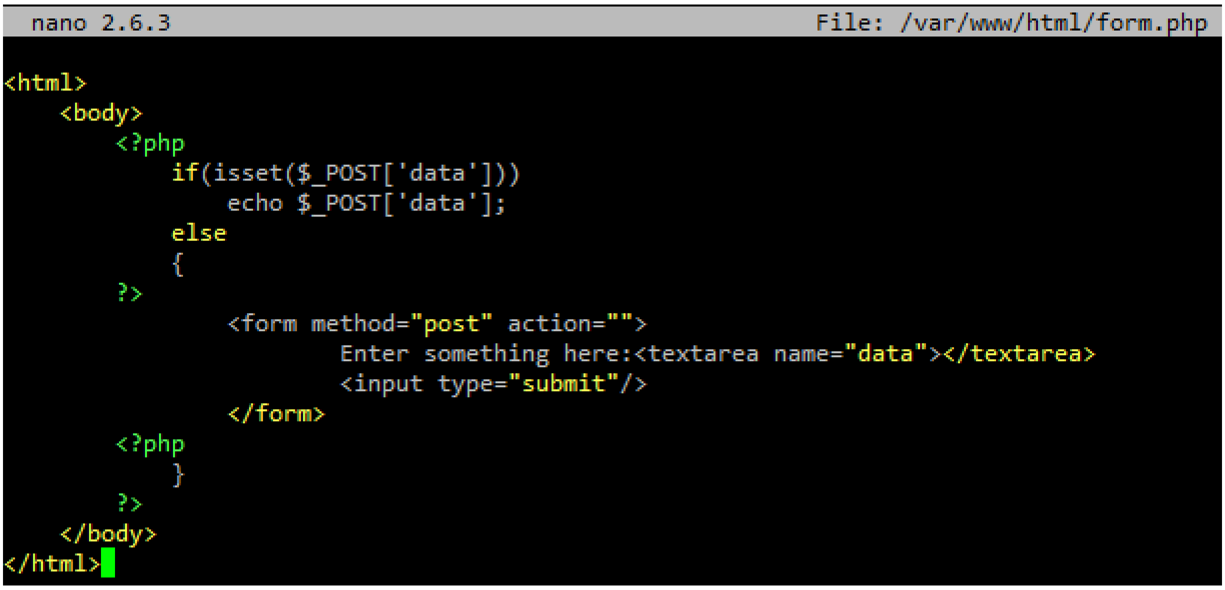
Password:



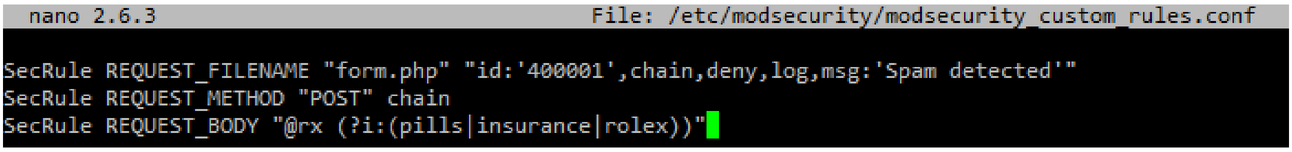
Forbidden

You don't have permission to access /login.php on this server.

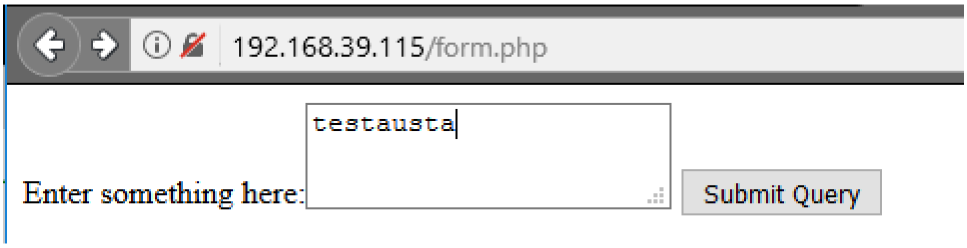
Apache/2.4.18 (Ubuntu) Server at 192.168.39.115 Port 80



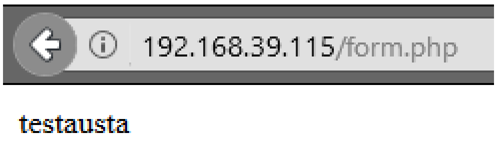
```
 nano 2.6.3                                          File: /var/www/html/form.php

<html>
    <body>
        <?php
            if(isset($_POST['data']))
                echo $_POST['data'];
            else
            {
        ?>
                <form method="post" action="">
                        Enter something here:<textarea name="data"></textarea>
                        <input type="submit"/>
                </form>
        <?php
            }
        ?>
    </body>
</html>
```



```
 nano 2.6.3                               File: /etc/modsecurity/modsecurity_custom_rules.conf

SecRule REQUEST_FILENAME "form.php" "id:'400001',chain,deny,log,msg:'Spam detected'"
SecRule REQUEST_METHOD "POST" chain
SecRule REQUEST_BODY "@rx (?i:(pills|insurance|rolex))"
```



Enter something here: testausta



testausta

This all can be done by following this tutorial: https://www.digitalocean.com/community/tutorials/how-to-set-up-mod_security-with-apache-on-debian-ubuntu

1p for Installing LAMP

1p for setting up database & login site, and proving login & injection works.

1p for installing mod_secure + preventing injection (NOTE where guide says: “nano /etc/apache2/mods-enabled/modsecurity.conf” depending how you installed it might be /etc/apache2/mods-enabled/security2.conf or something like that.)

1p for creating custom form + creating custom rule to block specific words from it, prove that it works.

1p for proving that you can Exclude some directories from rules. For example, create custom form to other directory (/var/www/html/notblocked/form.php) and prove that it doesn’t get checked by rules even original form.php is checked