

Lab 6- Securing Apache Web Server - 2  
Bijon Saha  
2019831007

```
bijon@cyborg: /etc/apache2/sites-enabled
GNU nano 6.2 example.com.conf
<VirtualHost *:80>
    ServerAdmin webmaster@example.com
    ServerName example.com
    ServerAlias www.example.com

    DocumentRoot /var/www/example.com/html

    # Redirect all traffic to HTTPS
    Redirect permanent / https://example.com/

    ErrorLog ${APACHE_LOG_DIR}/example.com_error.log
    CustomLog ${APACHE_LOG_DIR}/example.com_access.log combined
</VirtualHost>

<VirtualHost _default_:443>
    ServerAdmin admin@example.com
    ServerName example.com

    DocumentRoot /var/www/example.com/html

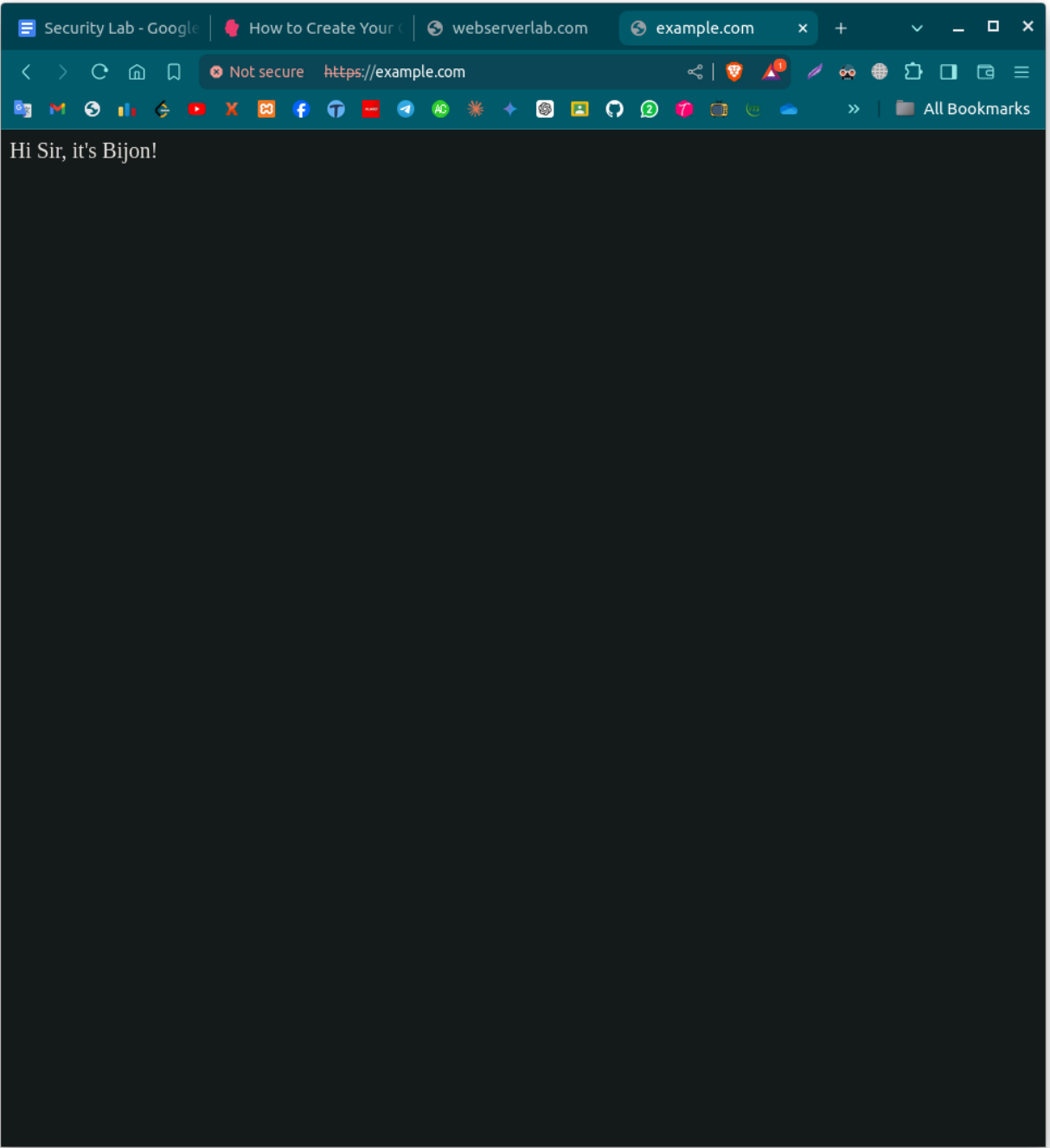
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    SSLEngine on

    # A self-signed (snakeoil) certificate can be created by installing
    # the ssl-cert package. See
    # /usr/share/doc/apache2/README.Debian.gz for more info.
    # If both key and certificate are stored in the same file, only the
    # SSLCertificateFile directive is needed.
    SSLCertificateFile /etc/ssl/certs/apache-selfsigned.crt
    SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key

    # Server Certificate Chain:
    # Point SSLCertificateChainFile at a file containing the
    # concatenation of PEM encoded CA certificates which form the
    # certificate chain for the server certificate. Alternatively
    # the referenced file can be the same as SSLCertificateFile
    # when the CA certificates are directly appended to the server
    # certificate for convinience.
    #SSLCertificateChainFile /etc/apache2/ssl.crt/server-ca.crt

    # Certificate Authority (CA):
    # Set the CA certificate verification path where to find CA
    # certificates for client authentication or alternatively one
    # huge file containing all of them (file must be PEM encoded)
    [ Read 120 lines ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line  M-E Redo
```



```
bijon@cyborg: /etc/apache2/sites-enabled
bijon@cyborg:/etc/apache2/sites-available$ cd
bijon@cyborg:~$ ls
certs  Documents  lab5  Pictures  pt      snap      Videos
Desktop  Downloads  Music  Postman  Public  Templates
bijon@cyborg:~$ sudo a2enmod rewrite
Enabling module rewrite.
To activate the new configuration, you need to run:
  systemctl restart apache2
bijon@cyborg:~$ systemctl restart apache2
bijon@cyborg:~$ cd /etc/apache2/sites-enabled/
bijon@cyborg:/etc/apache2/sites-enabled$ ls
example.com.conf
bijon@cyborg:/etc/apache2/sites-enabled$ sudo nano example.com.conf
bijon@cyborg:/etc/apache2/sites-enabled$ sudo nano example.com.conf
bijon@cyborg:/etc/apache2/sites-enabled$ systemctl restart apache2
bijon@cyborg:/etc/apache2/sites-enabled$ sudo htpasswd -c /etc/apache2/.htpasswd bijon
New password:
Re-type new password:
Adding password for user bijon
bijon@cyborg:/etc/apache2/sites-enabled$ sudo htpasswd /etc/apache2/.htpasswd saha
New password:
Re-type new password:
Adding password for user saha
bijon@cyborg:/etc/apache2/sites-enabled$ cat /etc/apache2/.htpasswd
bijon:$apr1$UkP5Cj.R$ms/nLNOt24okeLRevQedz0
saha:$apr1$g78eouU7$8GUuKvFYLDWKfmdAZUgkM.
bijon@cyborg:/etc/apache2/sites-enabled$ _
```

```

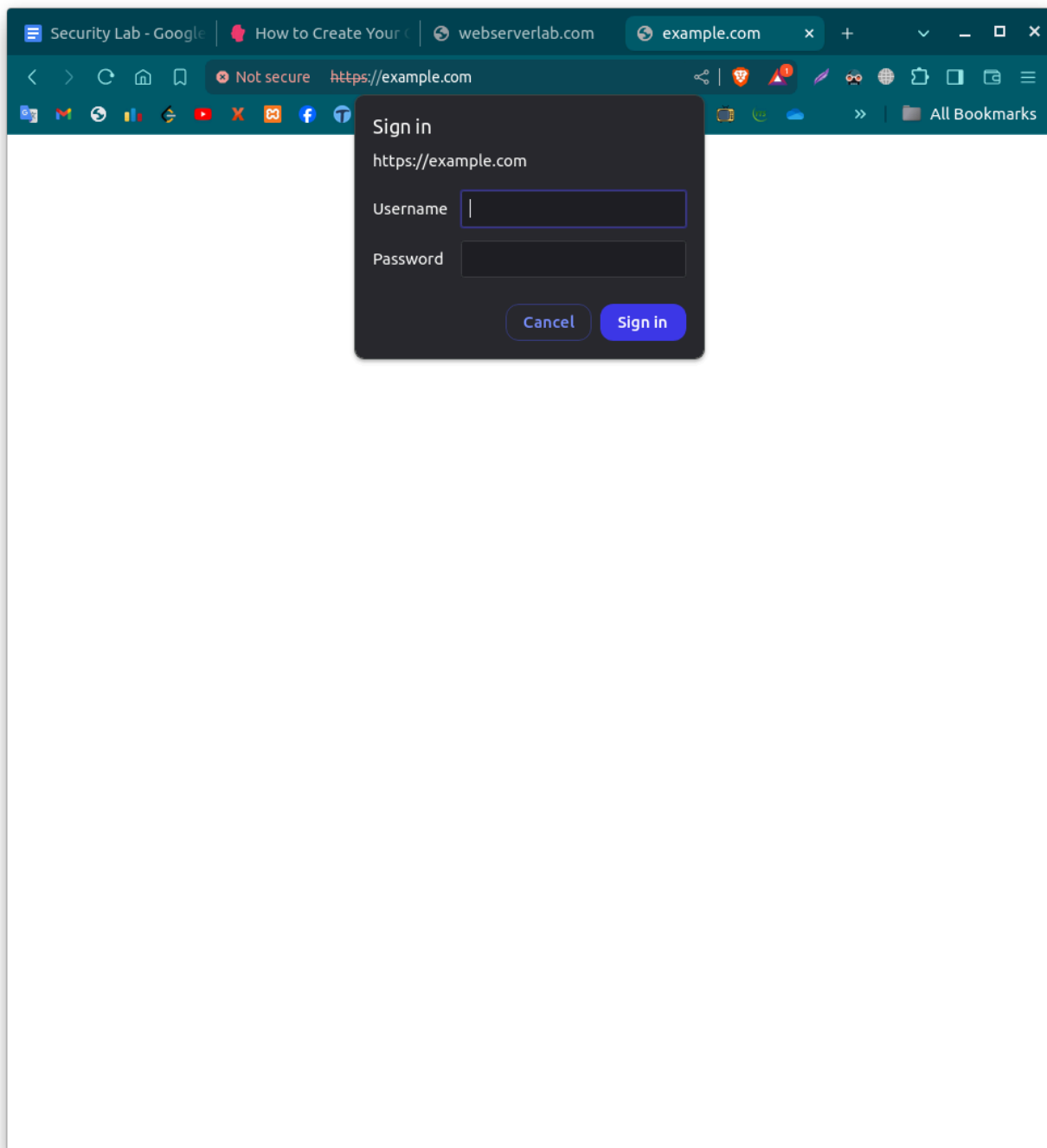
+      bijon@cyborg: /etc/apache2/sites-enabled
GNU nano 6.2      example.com.conf
#   number which specifies how deeply to verify the certificate
#   issuer chain before deciding the certificate is not valid.
#SSLVerifyClient require
#SSLVerifyDepth 10

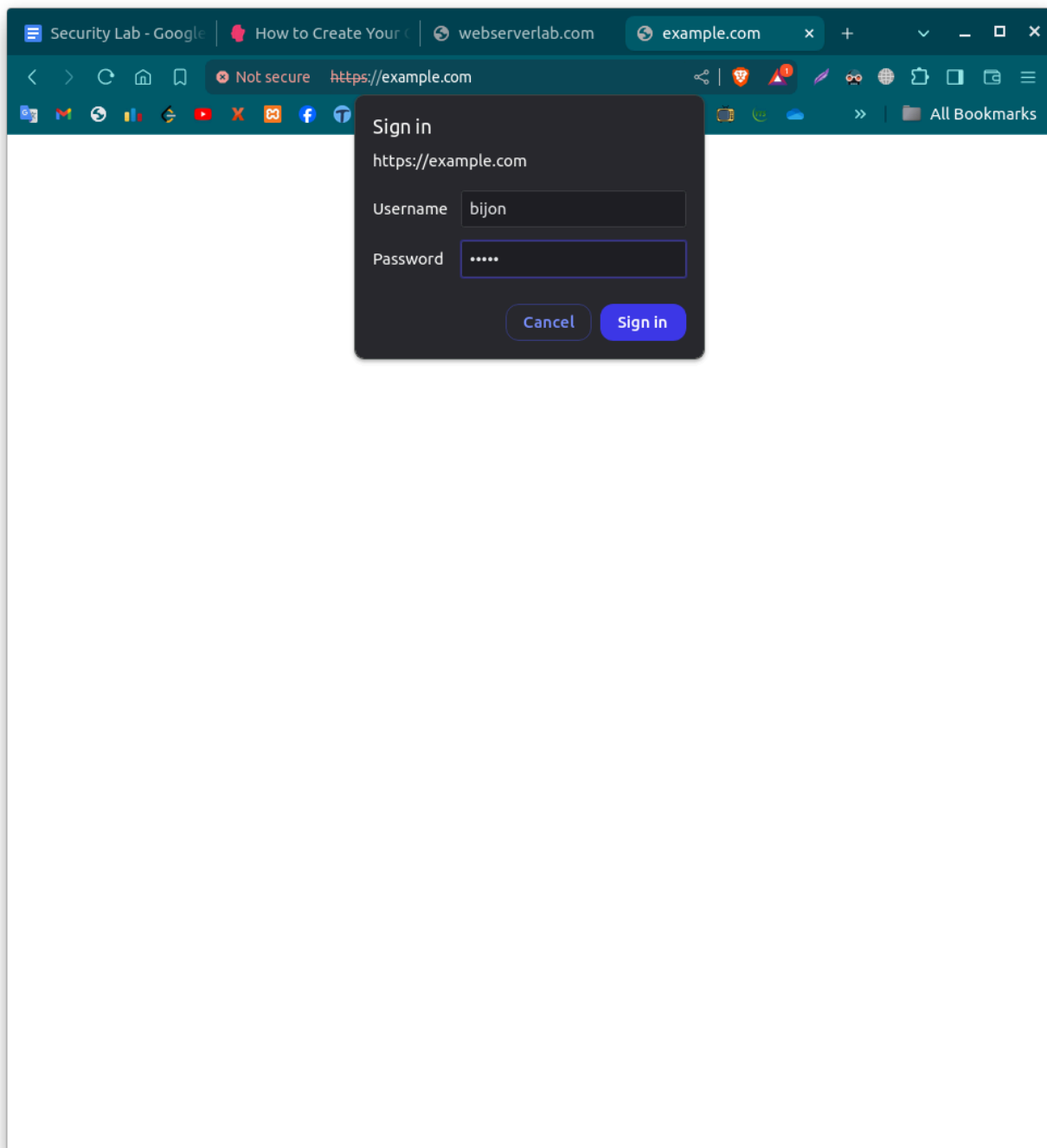
#   SSL Engine Options:
#   Set various options for the SSL engine.
#   o FakeBasicAuth:
#       Translate the client X.509 into a Basic Authorisation. This means that
#       the standard Auth/DBMAuth methods can be used for access control. The
#       user name is the 'one line' version of the client's X.509 certificate.
#       Note that no password is obtained from the user. Every entry in the user
#       file needs this password: 'xxj31ZMTZzkVA'.
#   o ExportCertData:
#       This exports two additional environment variables: SSL_CLIENT_CERT and
#       SSL_SERVER_CERT. These contain the PEM-encoded certificates of the
#       server (always existing) and the client (only existing when client
#       authentication is used). This can be used to import the certifica
<Directory "/var/www/example.com/html">
AuthType Basic
AuthName "Restricted Content"
AuthUserFile /etc/apache2/.htpasswd
Require valid-user
</Directory>

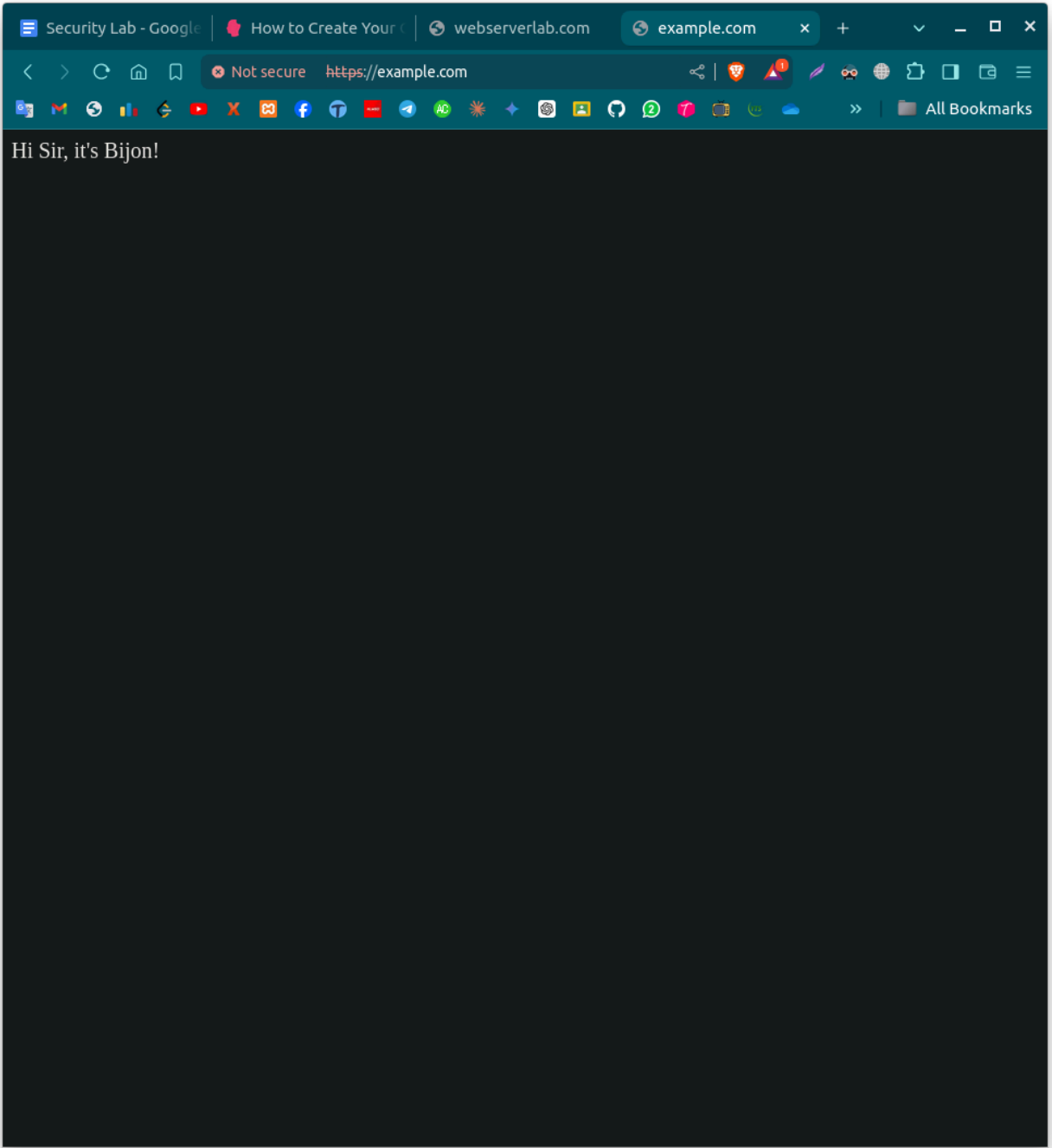
<FilesMatch "\.(cgi|shtml|phtml|php)$">
    SSLOptions +StdEnvVars
</FilesMatch>
<Directory /usr/lib/cgi-bin>
    SSLOptions +StdEnvVars
</Directory>

#   SSL Protocol Adjustments:
#   The safe and default but still SSL/TLS standard compliant shutdown
#   approach is that mod_ssl sends the close notify alert but doesn't wait for
#   the close notify alert from client. When you need a different shutdown
#   approach you can use one of the following variables:
#   o ssl-unclean-shutdown:
#       This forces an unclean shutdown when the connection is closed, i.e. no
#       SSL close notify alert is send or allowed to received. This violates
#       the SSL/TLS standard but is needed for some brain-dead browsers. Use
#       this when you receive I/O errors because of the standard approach where
#       mod_ssl sends the close notify alert.
#   o ssl-accurate-shutdown:
#       This forces an accurate shutdown when the connection is closed, i.e. a
#       SSL close notify alert is send and mod_ssl waits for the close notify

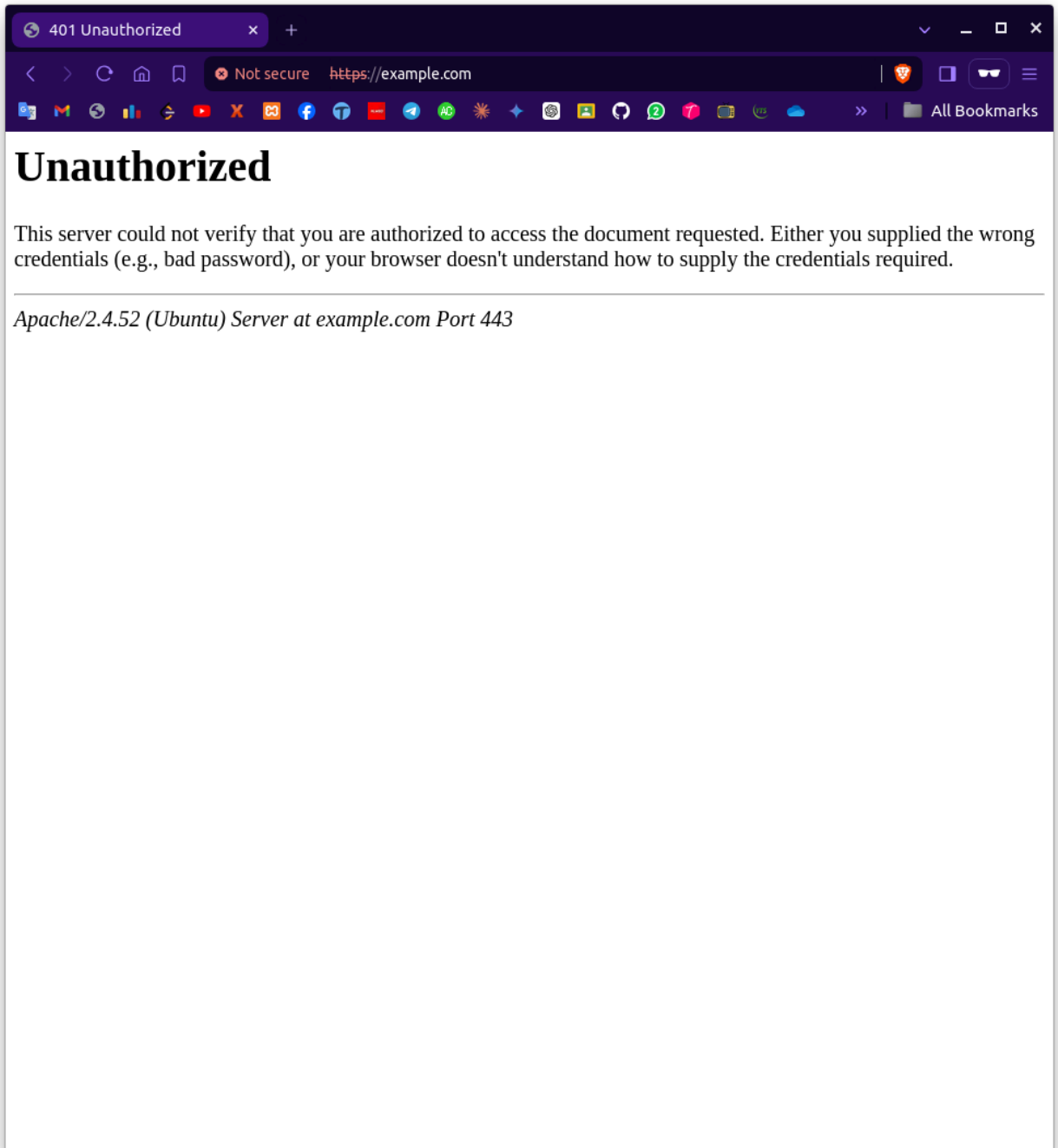
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line  M-E Redo
```











```
+      bijon@cyborg: /etc/apache2/sites-enabled
Connecting to MySQL using a blank password.

VALIDATE PASSWORD COMPONENT can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD component?

Press y|Y for Yes, any other key for No: n

Skipping password set for root as authentication with auth_socket is used by default.
If you would like to use password authentication instead, this can be done with the "ALTER_USER" command.
See https://dev.mysql.com/doc/refman/8.0/en/alter-user.html#alter-user-password-management for more information.

By default, a MySQL installation has an anonymous user,
allowing anyone to log into MySQL without having to have
a user account created for them. This is intended only for
testing, and to make the installation go a bit smoother.
You should remove them before moving into a production
environment.

Remove anonymous users? (Press y|Y for Yes, any other key for No) : n

... skipping.

Normally, root should only be allowed to connect from
'localhost'. This ensures that someone cannot guess at
the root password from the network.

Disallow root login remotely? (Press y|Y for Yes, any other key for No) : n

... skipping.

By default, MySQL comes with a database named 'test' that
anyone can access. This is also intended only for testing,
and should be removed before moving into a production
environment.

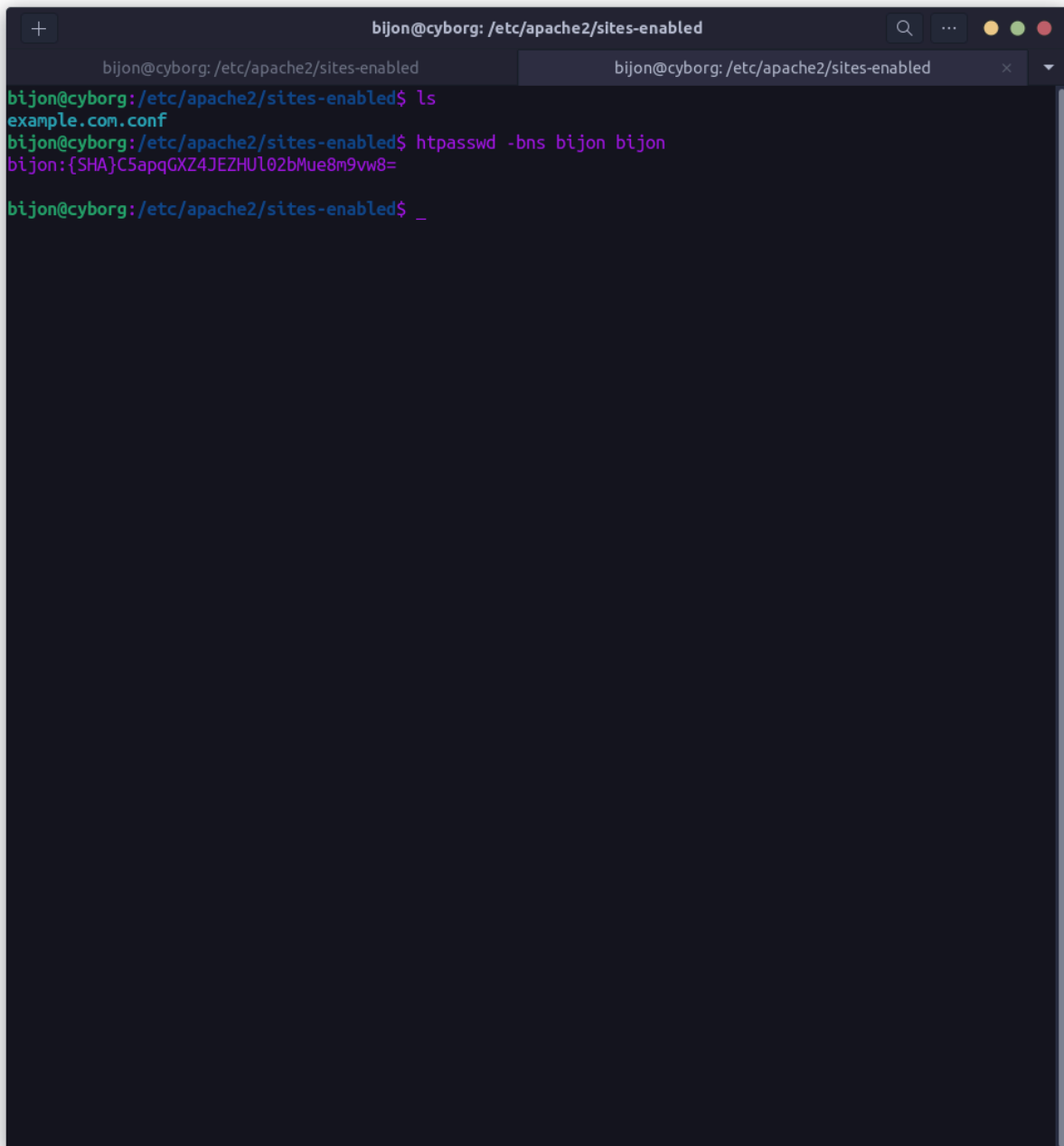
Remove test database and access to it? (Press y|Y for Yes, any other key for No) : n

... skipping.

Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.

Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y
Success.

All done!
bijon@cyborg: /etc/apache2/sites-enabled$
```



A terminal window titled "bijon@cyborg: /etc/apache2/sites-enabled" with two tabs. The active tab shows the following commands and output:

```
bijon@cyborg: /etc/apache2/sites-enabled$ ls
example.com.conf
bijon@cyborg: /etc/apache2/sites-enabled$ htpasswd -bns bijon bijon
bijon:{SHA}C5apqGXZ4JEZHU102bMue8m9vw8=
bijon@cyborg: /etc/apache2/sites-enabled$ _
```

```
bijon@cyborg: /etc/apache2/sites-enabled
Remove test database and access to it? (Press y|Y for Yes, any other key for No) : n
... skipping.
Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.

Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y
Success.

All done!
bijon@cyborg:/etc/apache2/sites-enabled$ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.37-0ubuntu0.22.04.3 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE apache;
Query OK, 1 row affected (0.01 sec)

mysql> use apache;
Database changed
mysql> CREATE TABLE users (username VARCHAR(30) PRIMARY KEY,password VARCHAR(512)
-> NOT NULL);
Query OK, 0 rows affected (0.05 sec)

mysql> {SHA}C5apqGXZ4JEZHUL02bMue8m9vw8=
-> INSERT INTO users VALUES('bijon', '{SHA}C5apqGXZ4JEZHUL02bMue8m9vw8=');
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server vers
ion for the right syntax to use near '{SHA}C5apqGXZ4JEZHUL02bMue8m9vw8=
INSERT INTO users VALUES('bijon', '{SHA}C5apqG' at line 1
mysql> INSERT INTO users VALUES('bijon', '{SHA}C5apqGXZ4JEZHUL02bMue8m9vw8=');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO users VALUES('bijon', 'INSERT INTO users VALUES('bijon', '{SHA}C5apqGXZ4JEZHUL02bMue8m9vw8=');');
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server vers
ion for the right syntax to use near 'bijon', '{SHA}C5apqGXZ4JEZHUL02bMue8m9vw8=');' at line 1
mysql> INSERT INTO users VALUES('saha', '{SHA}v13EeJ5MN4R4VlGhCNGuNGe+U0=');
Query OK, 1 row affected (0.01 sec)

mysql>
```

```
bijon@cyborg: /etc/apache2/sites-enabled
bijon@cyborg: /etc/apache2/sites-enabled$ ls
example.com.conf
bijon@cyborg: /etc/apache2/sites-enabled$ htpasswd -bns bijon bijon
bijon: {SHA}C5apqGXZ4JEZHUL02bMue8m9vw8=
bijon@cyborg: /etc/apache2/sites-enabled$ htpasswd -bns saha saha
saha: {SHA}v13EeJ5MN4R4VLLGhCNGuNGe+U0=
bijon@cyborg: /etc/apache2/sites-enabled$ ^C
bijon@cyborg: /etc/apache2/sites-enabled$ INSERT INTO users VALUES('bijon',
bash: syntax error near unexpected token '('
bijon@cyborg: /etc/apache2/sites-enabled$ INSERT INTO users VALUES('bijon', '{SHA}C5apqGXZ4JEZHUL02bMue8m9vw8=');
bash: syntax error near unexpected token '('
bijon@cyborg: /etc/apache2/sites-enabled$ ^Csert INTO users VALUES('bijon', '{SHA}C5apqGXZ4JEZHUL02bMue8m9vw8=');
bijon@cyborg: /etc/apache2/sites-enabled$ sudo a2enmod dbd
[sudo] password for bijon:
Sorry, try again.
[sudo] password for bijon:
Enabling module dbd.
To activate the new configuration, you need to run:
    systemctl restart apache2
bijon@cyborg: /etc/apache2/sites-enabled$ systemctl restart apache2
Failed to restart apache2.service: Access denied
See system logs and 'systemctl status apache2.service' for details.
bijon@cyborg: /etc/apache2/sites-enabled$ systemctl restart apache2
bijon@cyborg: /etc/apache2/sites-enabled$ sudo a2enmod authn_dbd
Considering dependency dbd for authn_dbd:
Module dbd already enabled
Enabling module authn_dbd.
To activate the new configuration, you need to run:
    systemctl restart apache2
bijon@cyborg: /etc/apache2/sites-enabled$ systemctl restart apache2
bijon@cyborg: /etc/apache2/sites-enabled$ sudo a2enmod socache_shmcb
Module socache_shmcb already enabled
bijon@cyborg: /etc/apache2/sites-enabled$ sudo a2enmod authn_socache
Enabling module authn_socache.
To activate the new configuration, you need to run:
    systemctl restart apache2
bijon@cyborg: /etc/apache2/sites-enabled$ systemctl restart apache2
bijon@cyborg: /etc/apache2/sites-enabled$ _
```

