در تمامی تصاویر از ترمینال، ساعت و تاریخ در پایین ترمینال (قسمت سبز رنگ) مشخص است.

# راه‌اندازی سرور HTTP با احراز هویت کاربری

## نصب وب‌سرور Apache

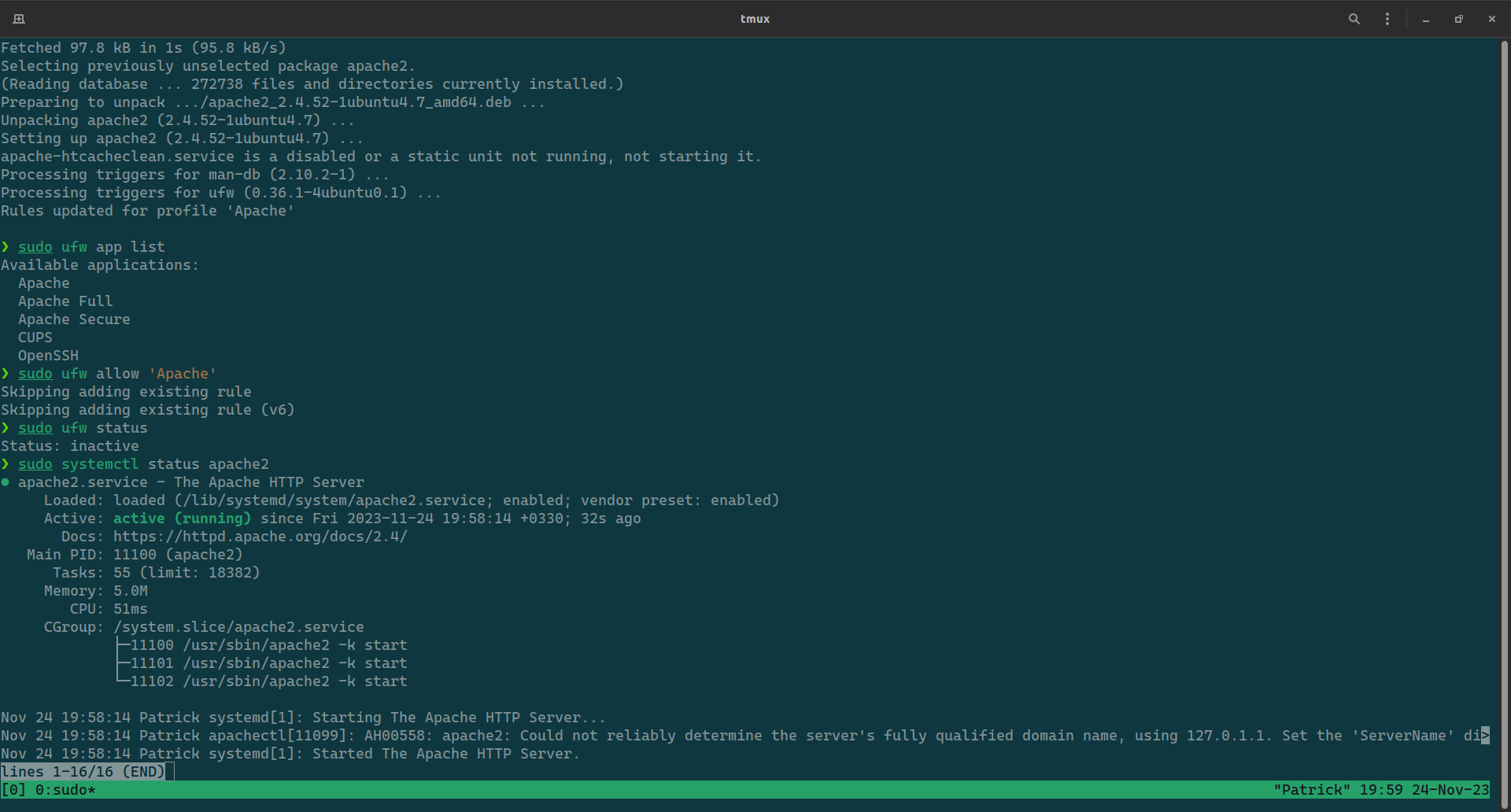
برای این‌کار، مراحل زیر را طی می‌کنیم:

sudo apt update  
sudo apt install apache2

با این کار، نرم‌افزار apache بر روی سرور نصب می‌شود.

sudo ufw app list  
sudo ufw allow 'Apache'  
sudo ufw status  
sudo systemctl status apache2

با انجام این دستورات، تنظیمات firewall را تغییر داده و اجازه دسترسی به apache (برای پورت 80) را می‌دهد و سپس، وضعیت سرویس apache را مشاهده می‌کند.

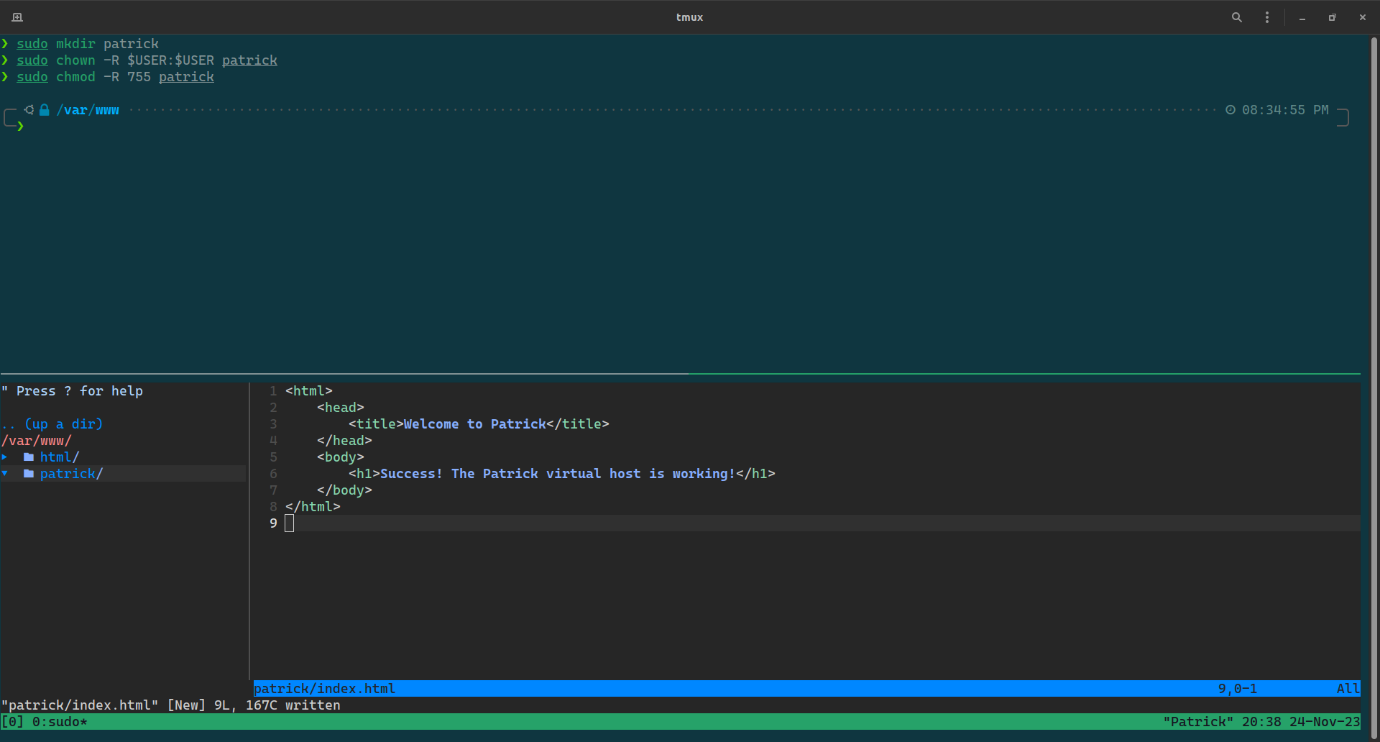


حال یک virtual host به نام patrick را به کانفیگ apache اضافه می‌کنیم.

sudo mkdir /var/www/patrick  
sudo chown -R $USER:$USER /var/www/patrick  
sudo chmod -R 755 /var/www/patrick  
sudo nvim /var/www/patrick/index.html

و داده زیر را در آن قرار می‌دهیم:

<html>  
 <head>  
 <title>Welcome to Patrick!</title>  
 </head>  
 <body>  
 <h1>Success! The Patrick virtual host is working!</h1>  
 </body>  
</html>

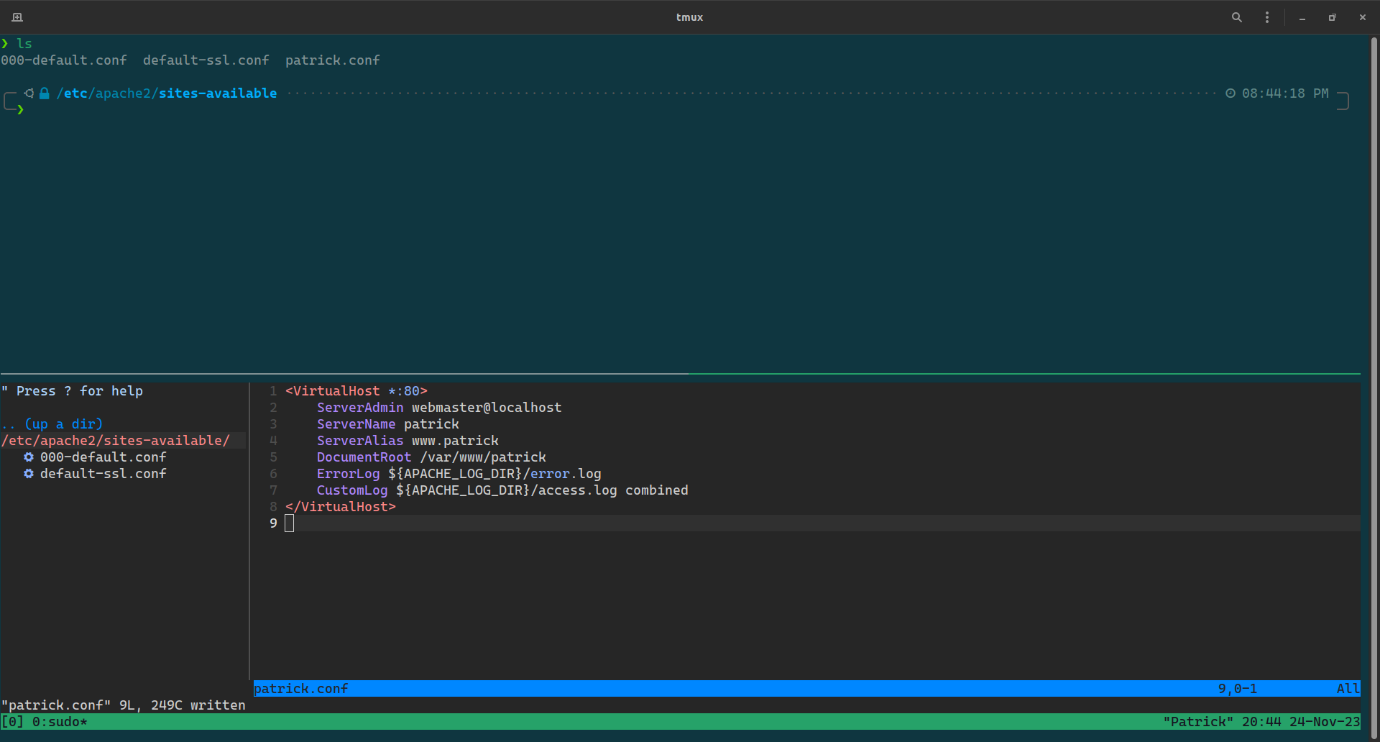


حال به کمک دستور و مقادیر زیر، این virtual host را کانفیگ می‌کنیم:

sudo nvim /etc/apache2/sites-available/patrick.conf

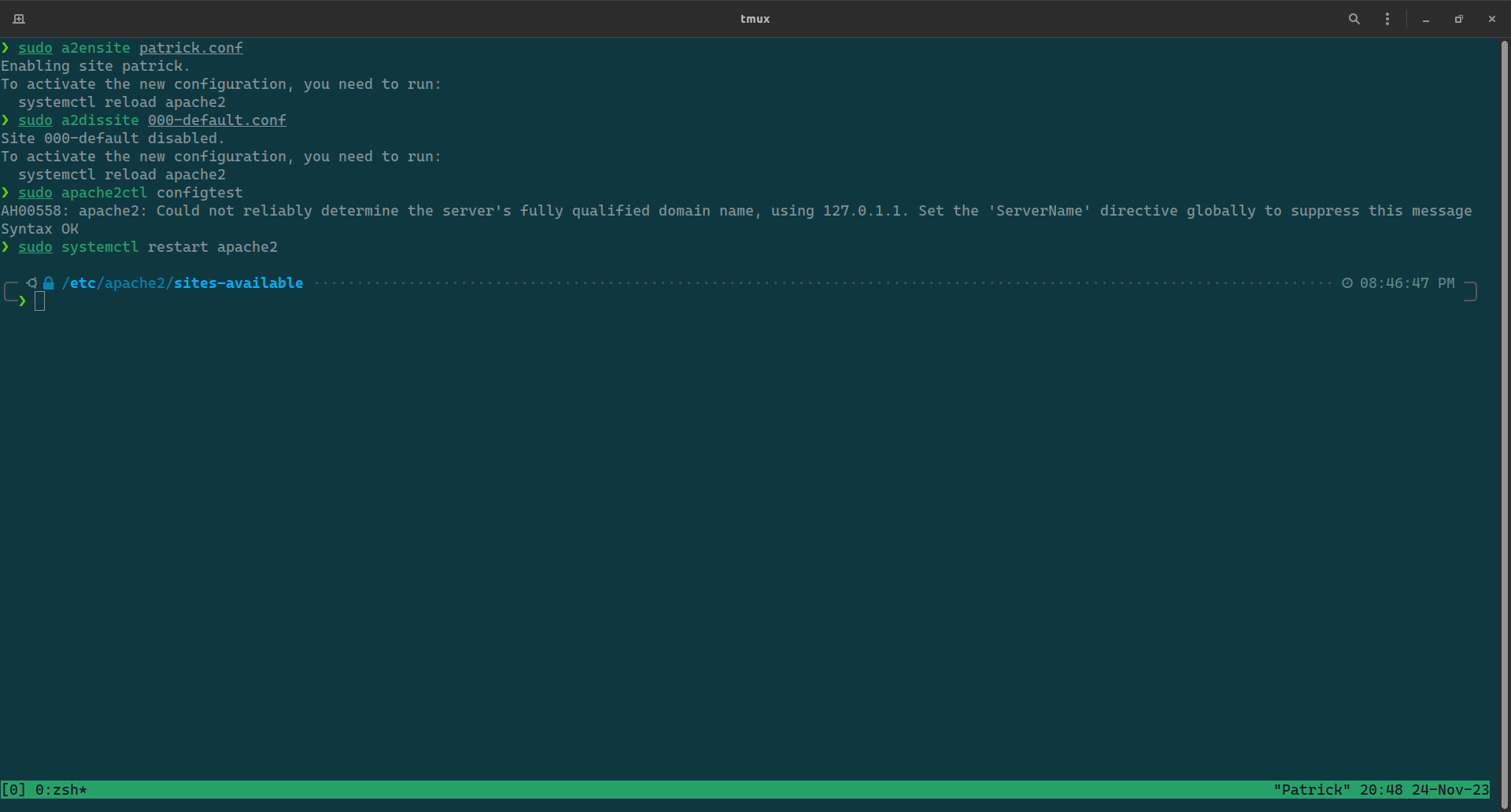
مقدار زیر را در آن قرار می‌دهیم:

<VirtualHost \*:80>  
 ServerAdmin webmaster@localhost  
 ServerName patrick  
 ServerAlias www.patrick  
 DocumentRoot /var/www/patrick  
 ErrorLog ${APACHE\_LOG\_DIR}/error.log  
 CustomLog ${APACHE\_LOG\_DIR}/access.log combined  
</VirtualHost>



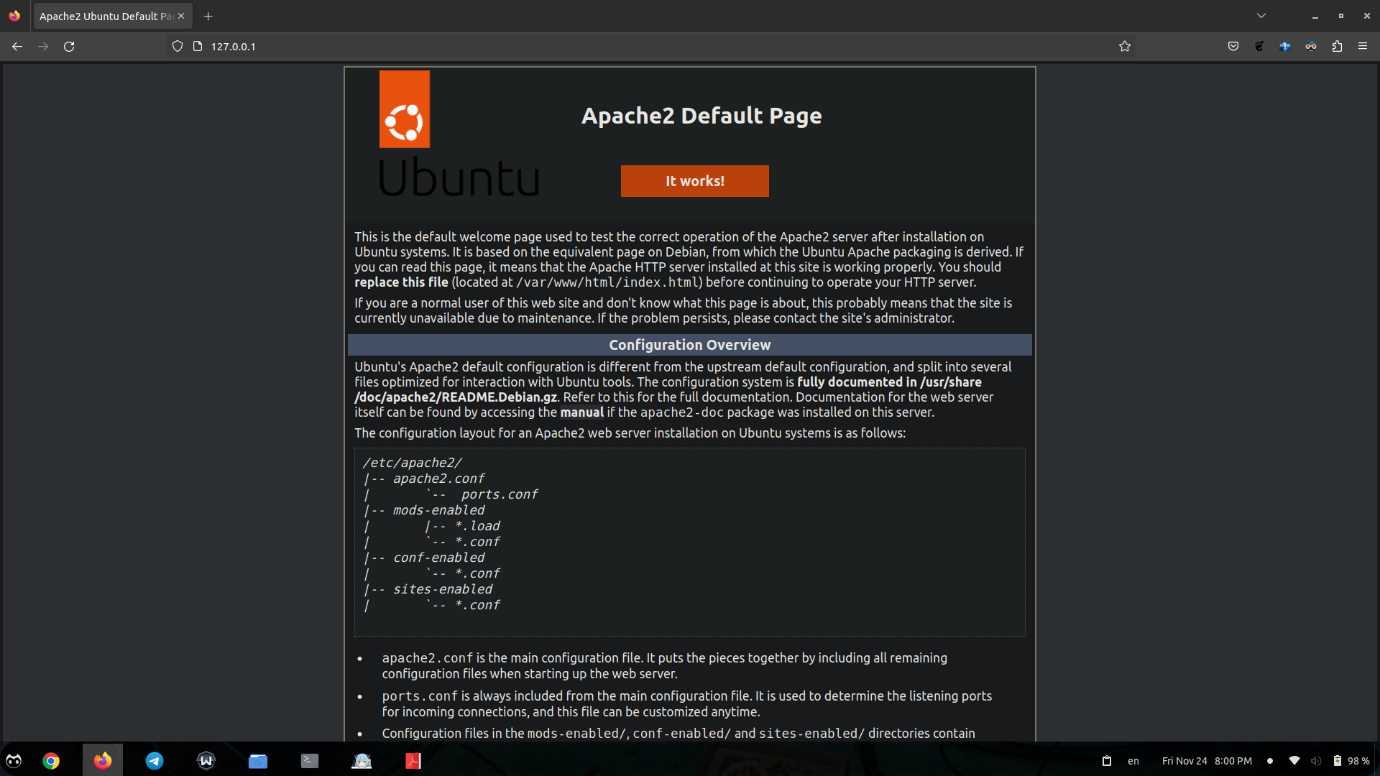
حال با استفاده از دستورات زیر، ابتدا سایت قبلی را غیر فعال کرده، سایت جدید را فعال کرده، درستی کانفیگ را تست کرده و سرویس apache را ری‌استارت می‌کنیم:

sudo a2ensite patrick.conf  
sudo a2dissite 000-default.conf  
sudo apache2ctl configtest  
sudo systemctl restart apache2

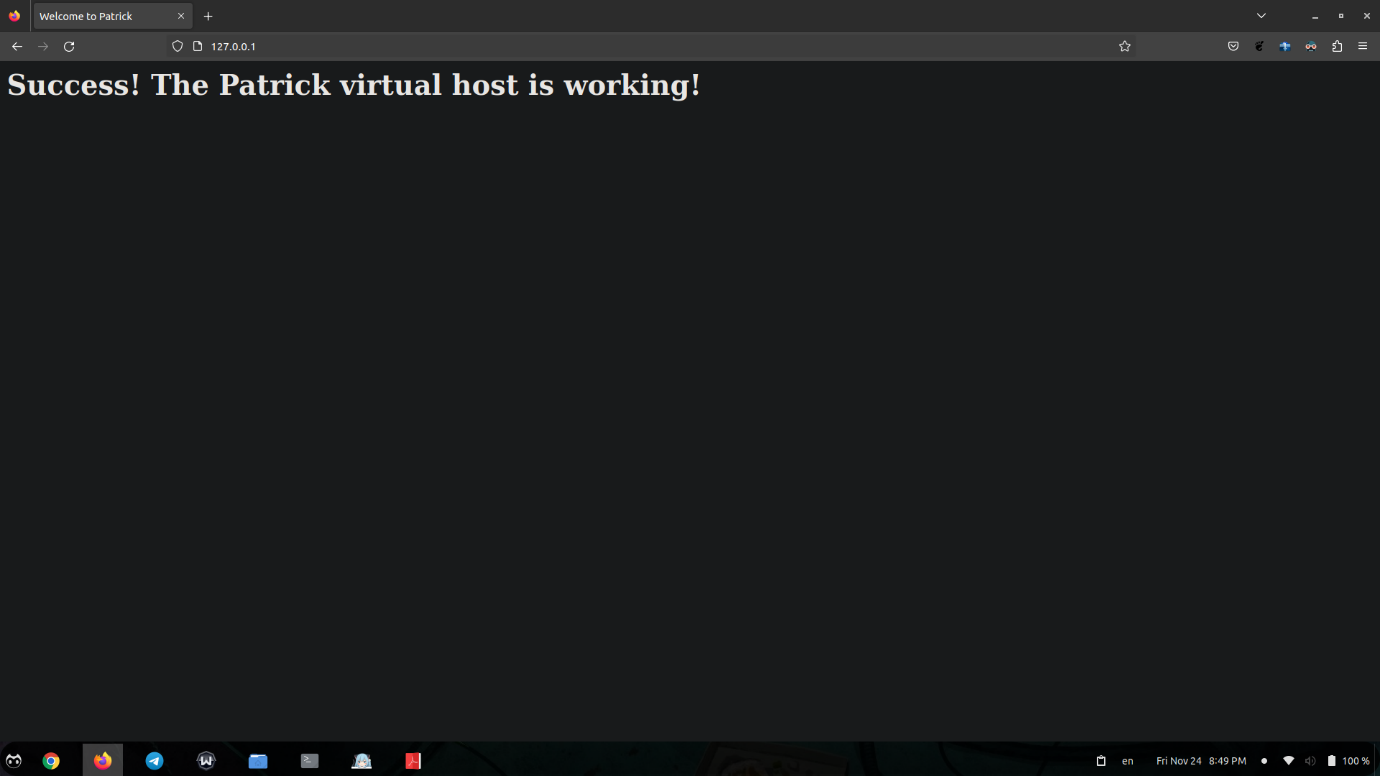


## مشاهده صفحه سایت

پیش از تنظیم virtual host، صفحه نمایش داده شده به صورت زیر است:



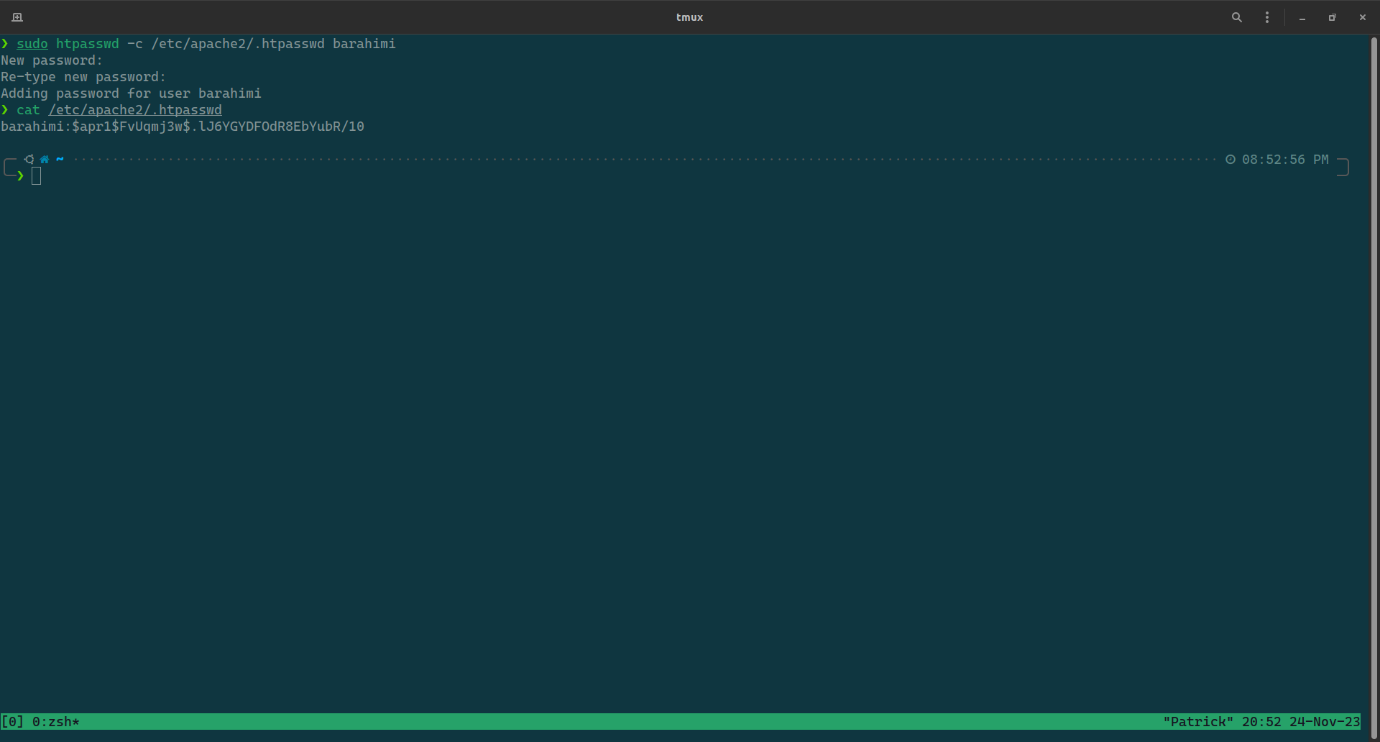
همچنین پس از تنظیم سایت patrick، صفحه به این صورت تغییر می‌کند:



## تنظیم احراز هویت برای سرویس

ابتدا با استفاده از دستور زیر، یک یوزر جدید تعریف کرده و سپس مقدار قرار داده شده در فایل را مشاهده می‌کنیم. در این بخش، یوزرنیم برابر با barahimi و پسورد برابر با 1234 است.

sudo htpasswd -c /etc/apache2/.htpasswd barahimi  
cat /etc/apache2/.htpasswd



حال باید کانفیگ قبلی را طوری عوض کرده که از این فایل برای authorization استفاده کند.

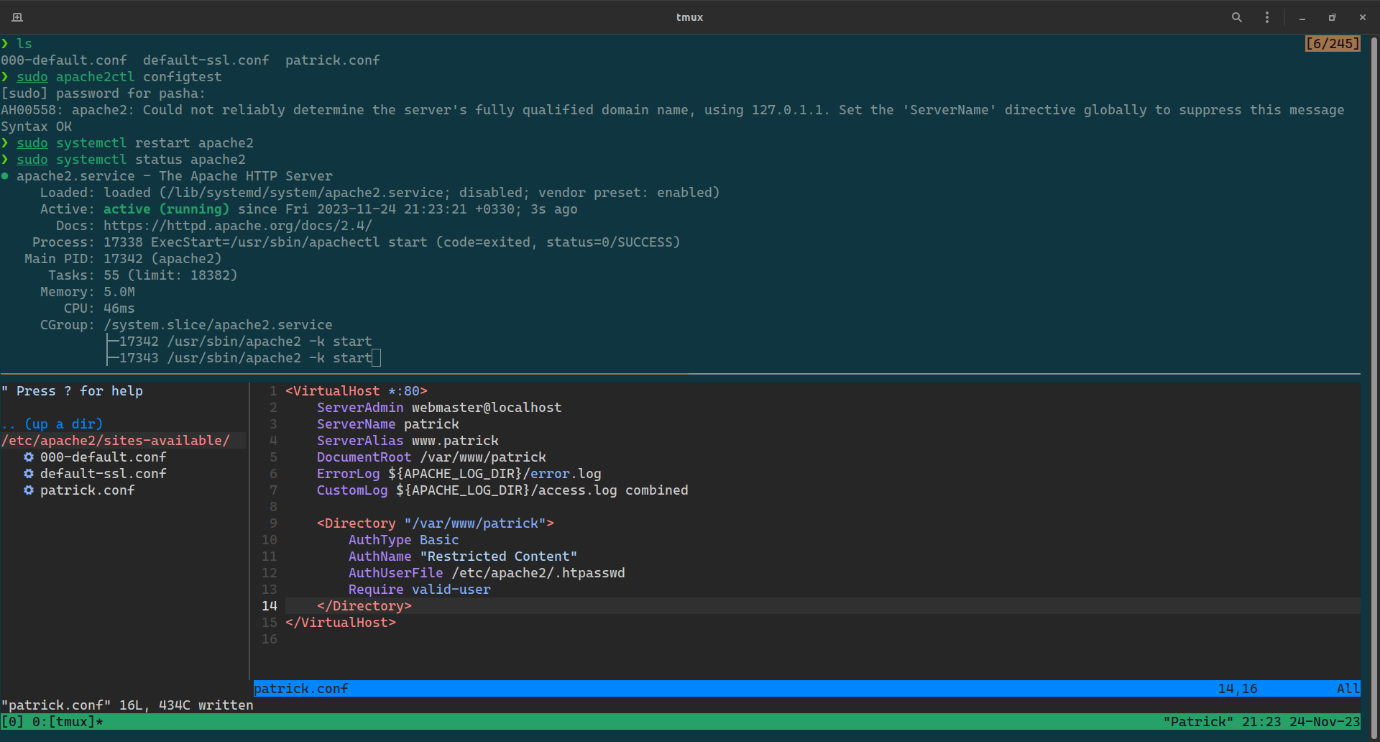
sudo nvim /etc/apache2/sites-available/patrick.conf

مقدار زیر را به این فایل اضافه می‌کنیم:

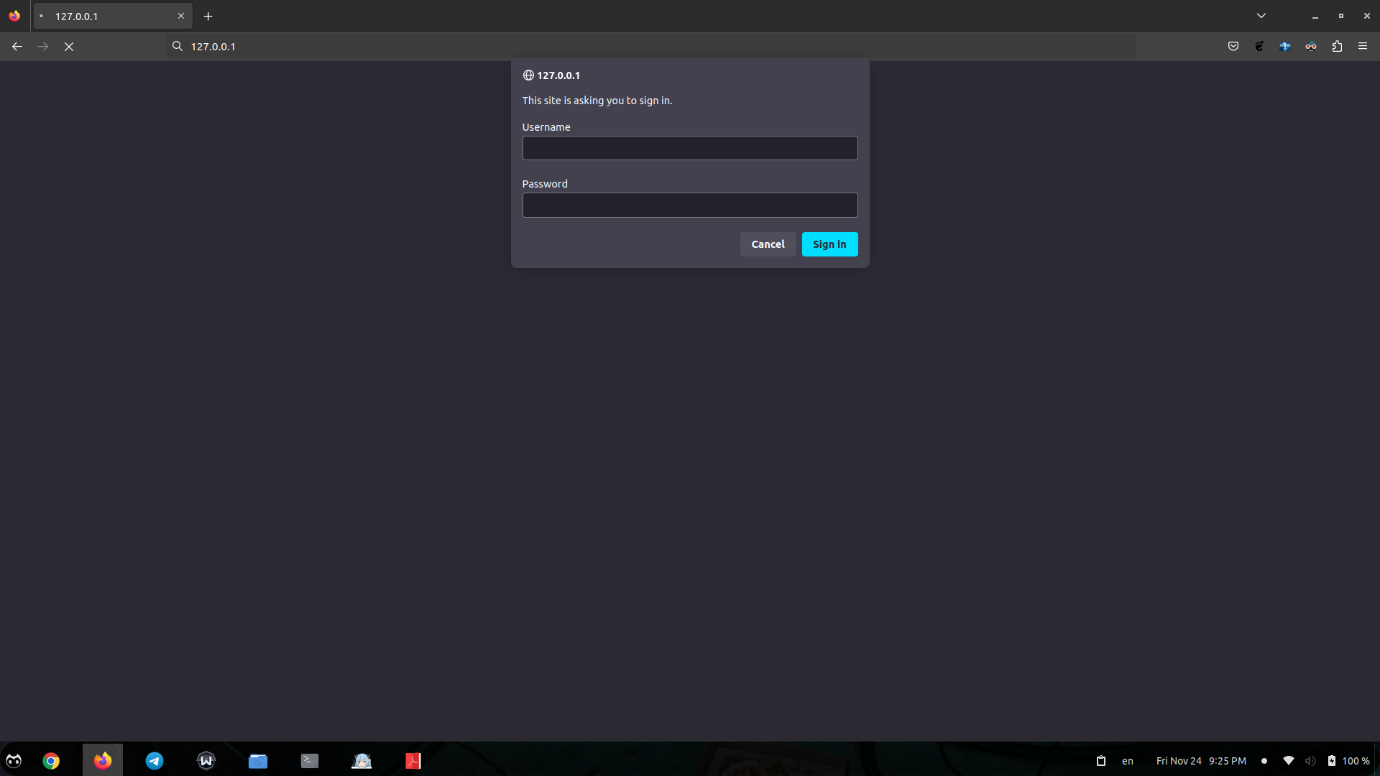
<Directory "/var/www/patrick">  
 AuthType Basic  
 AuthName "Restricted Content"  
 AuthUserFile /etc/apache2/.htpasswd  
 Require valid-user  
</Directory>

سپس دستورات زیر را اجرا می‌کنیم:

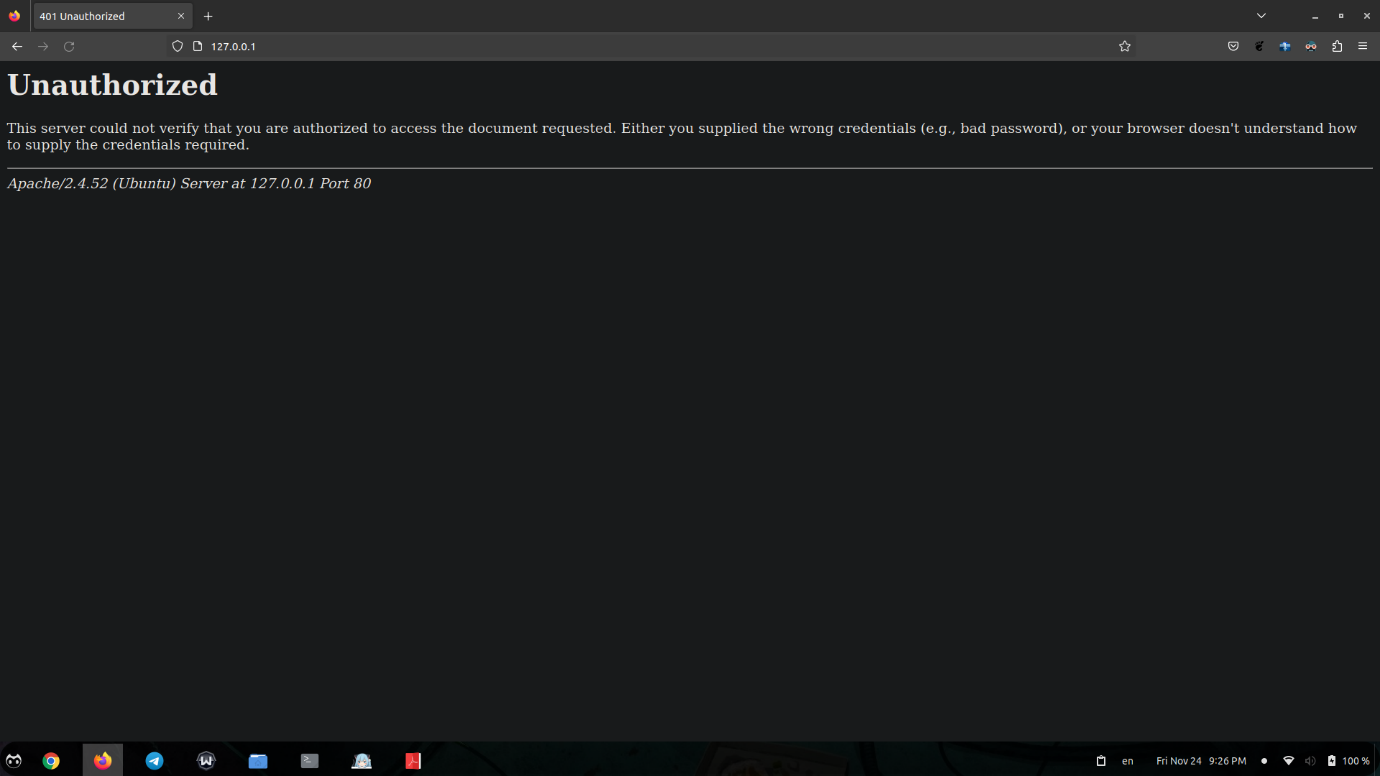
sudo apache2ctl configtest  
sudo systemctl restart apache2  
sudo systemctl status apache2



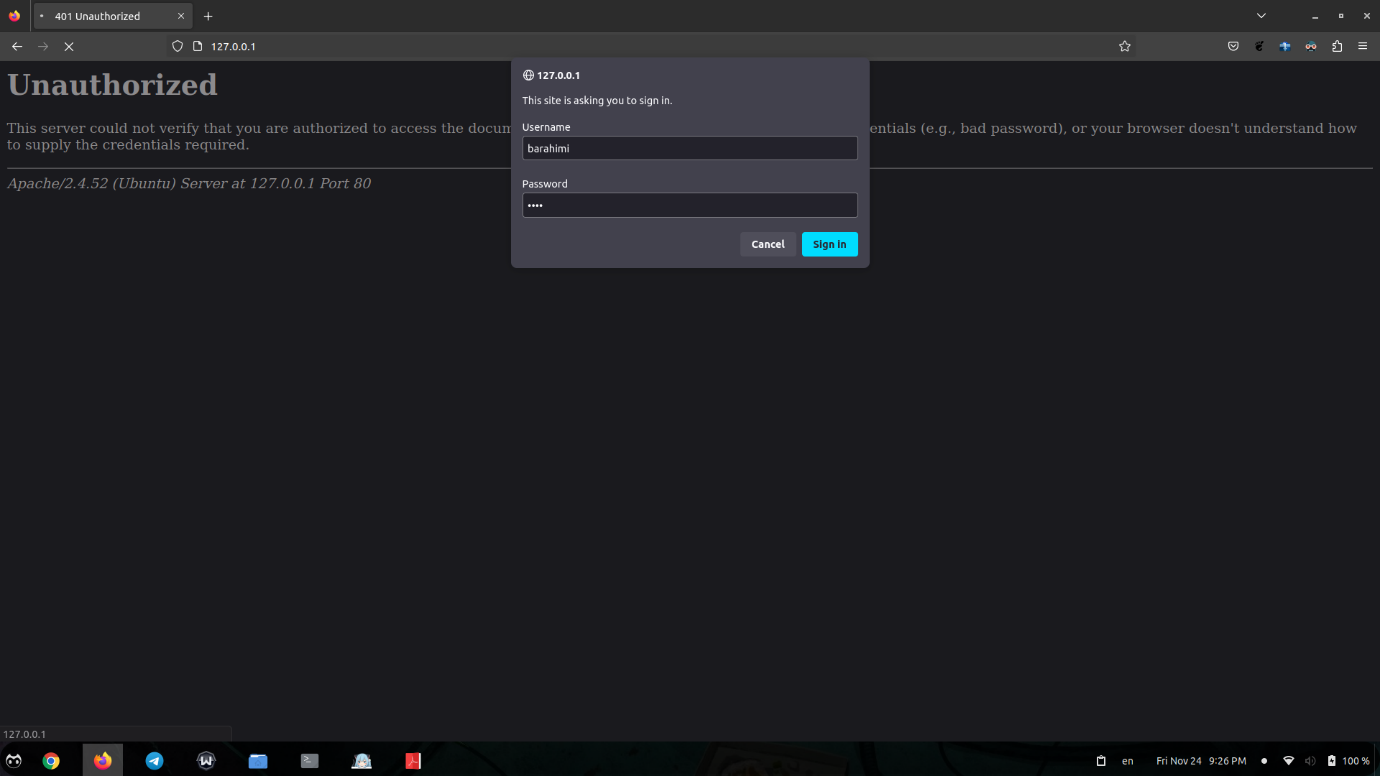
حال اگر صفحه را ریلود کنیم، مشاهده می‌کنیم که یوزرنیم و پسورد از ما خواسته می‌شود:



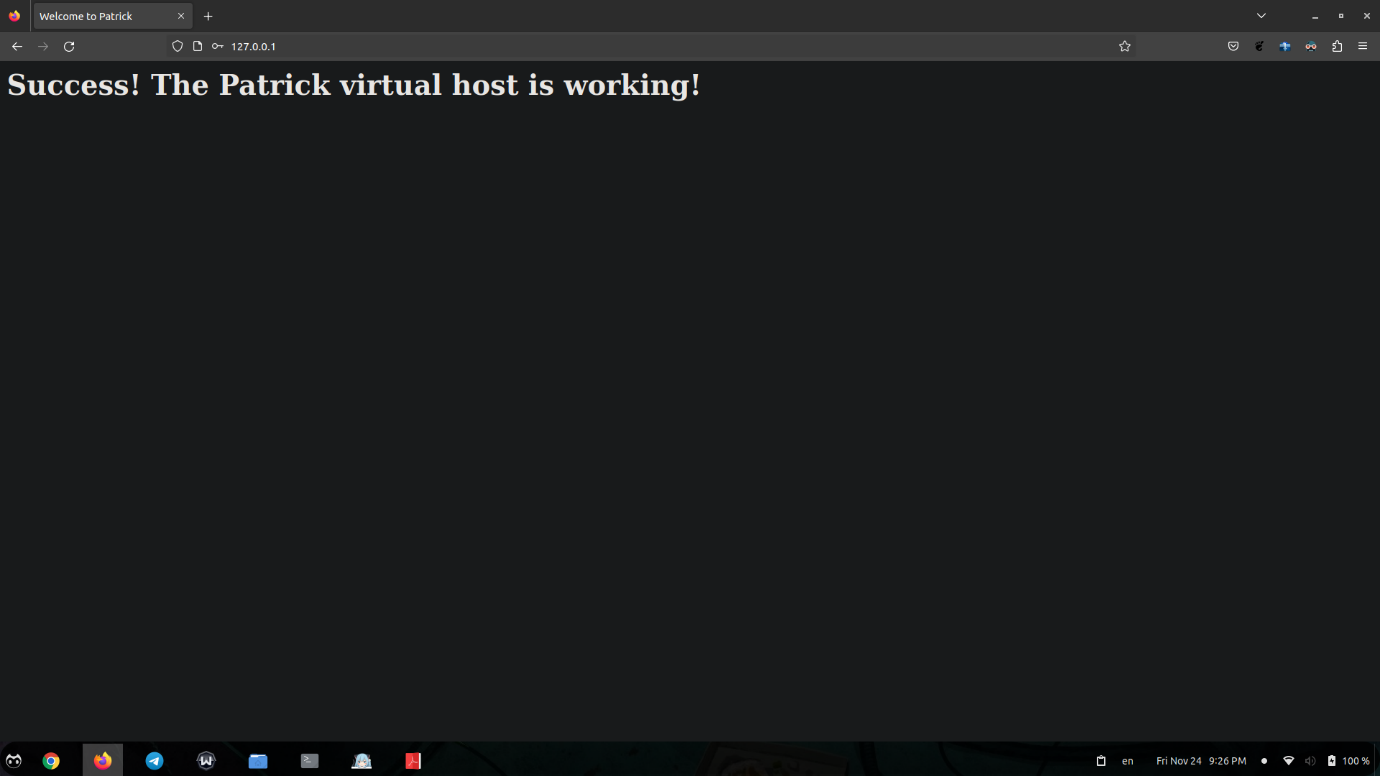
اگر این موارد را وارد نکنیم و یا اشتباه وارد کنیم، با صفحه زیر مواجه می‌شویم:



حال مجددا یوزرنیم و پسور را وارد می‌کنیم:

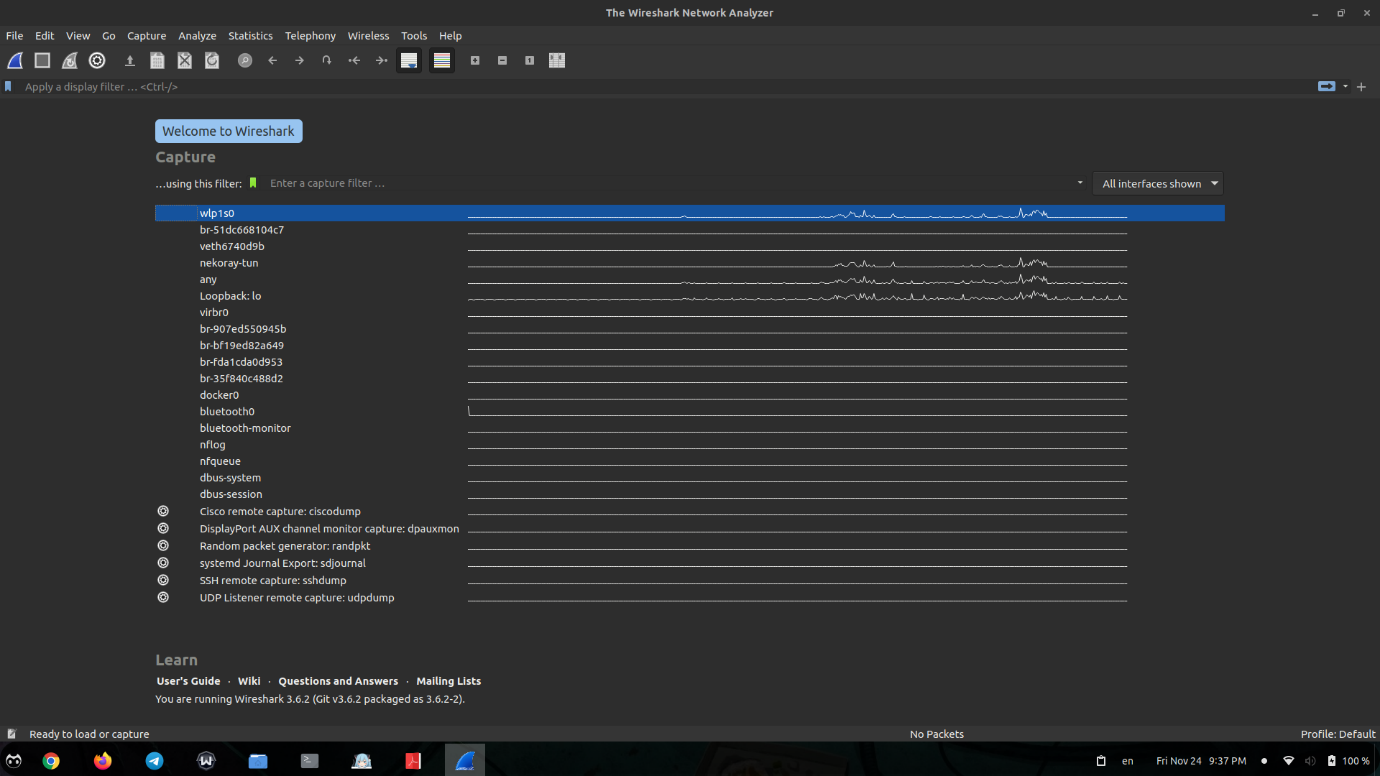


و با صفحه زیر مواجه می‌شویم:



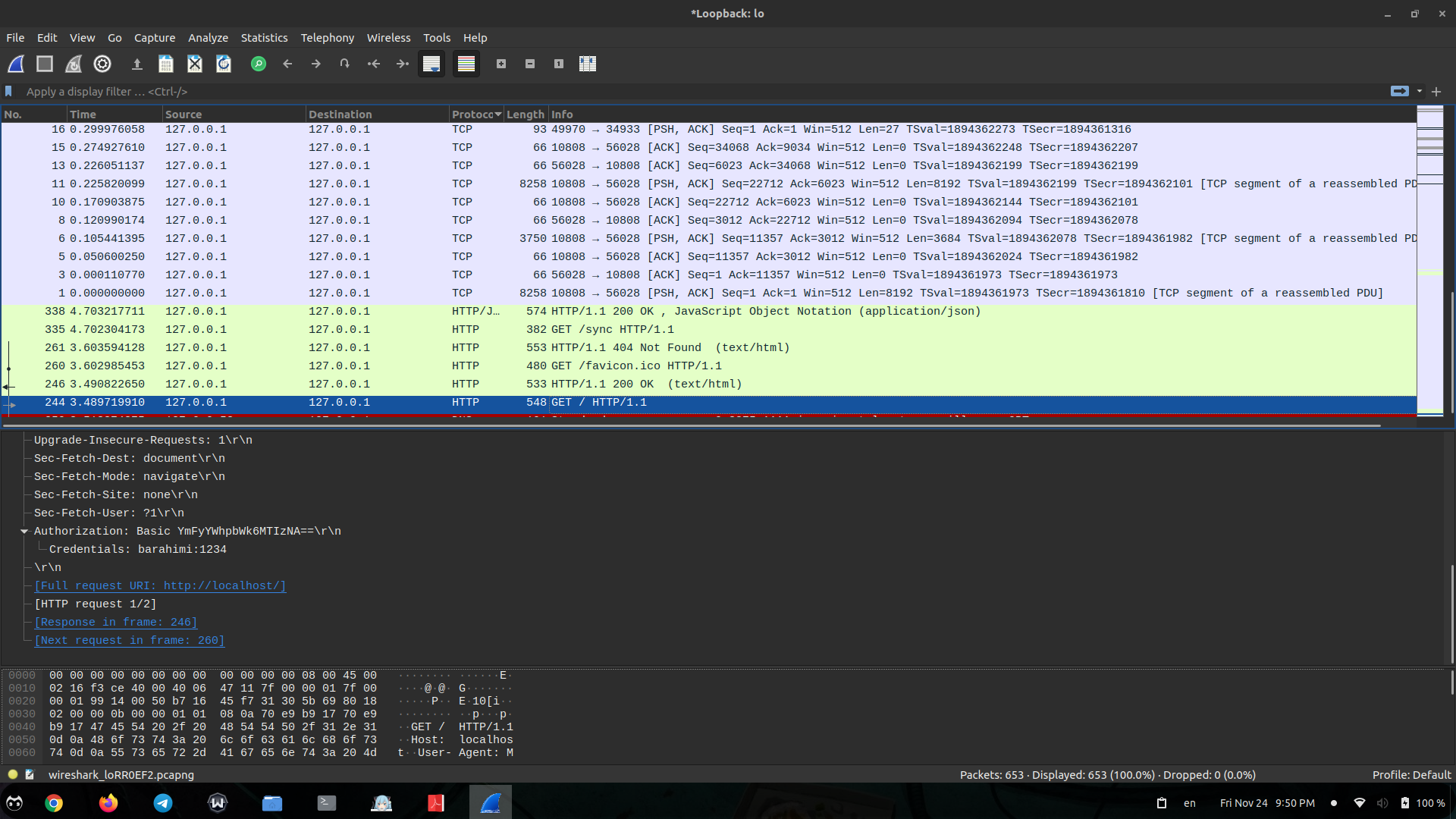
## نصب wireshark

در این مرحله اقدام به نصب wireshark می‌کنیم. با توجه به اینکه این برنامه از پیش روی سیستم من نصب بوده است، تنها به گذاشتن تصویری از این نرم‌افزار بسنده می‌کنم:



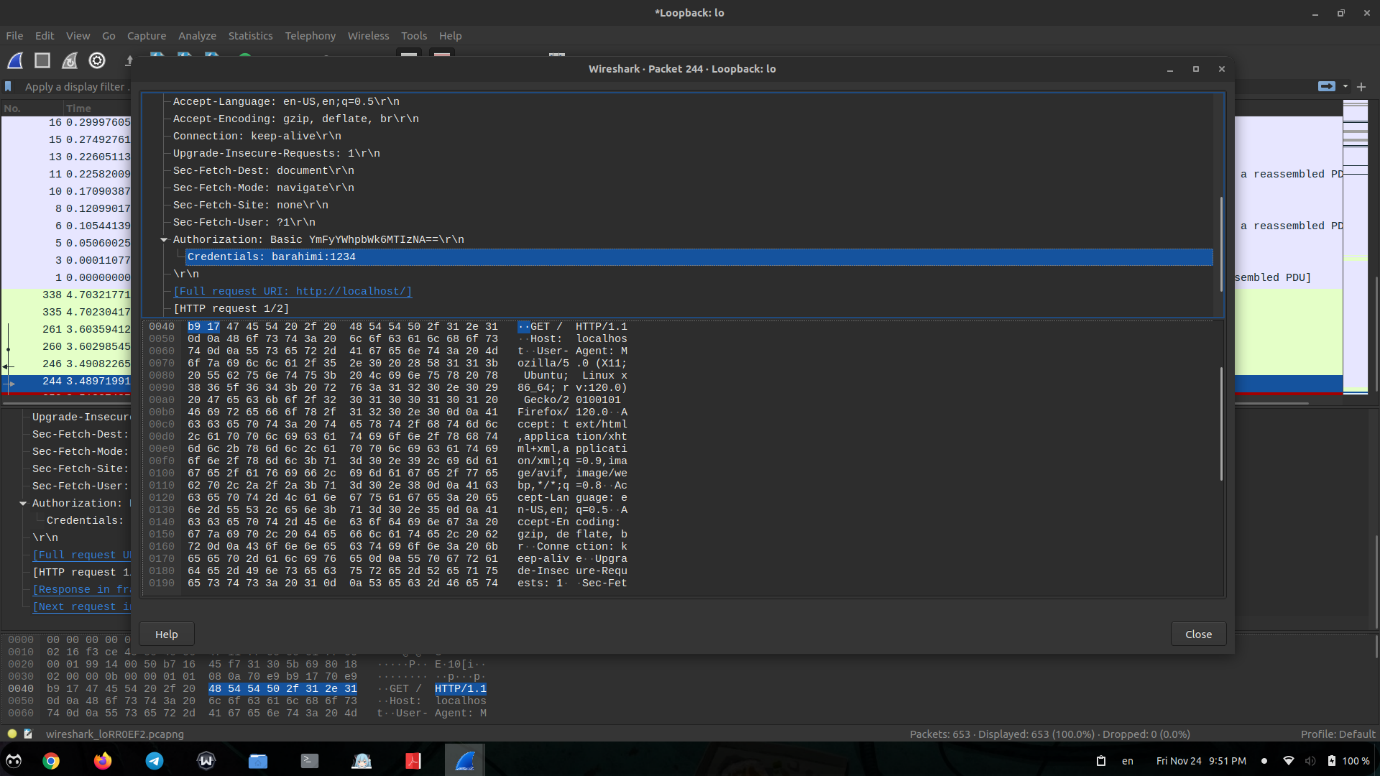
## ضبط بسته‌های HTTP

در این زمان wireshark را روی loobback (lo) قرار داده و ضبط را شروع می‌کنیم. مجددا صفحه سایت را ریلود کرده و یوزرنیم و پسورد را وارد می‌کنیم و سپس، ضبط wireshark را متوقف می‌کنیم. برخی از بسته‌های دریافت شده به صورت زیر هستند:



## مشاهده اطلاعات بسته ضبط شده

حال یکی از بسته‌های HTTP را باز کرده و بخش Authorization را مشاهده می‌کنیم:



همانطور که مشاهده می‌شود، در فیلد Credentials، عبارت barahimi:1234 قرار داده شده و یوزرنیم و پسورد به راحتی قابل مشاهده هستند.

## علت امکان شنود اطلاعات احراز هویت کاربری

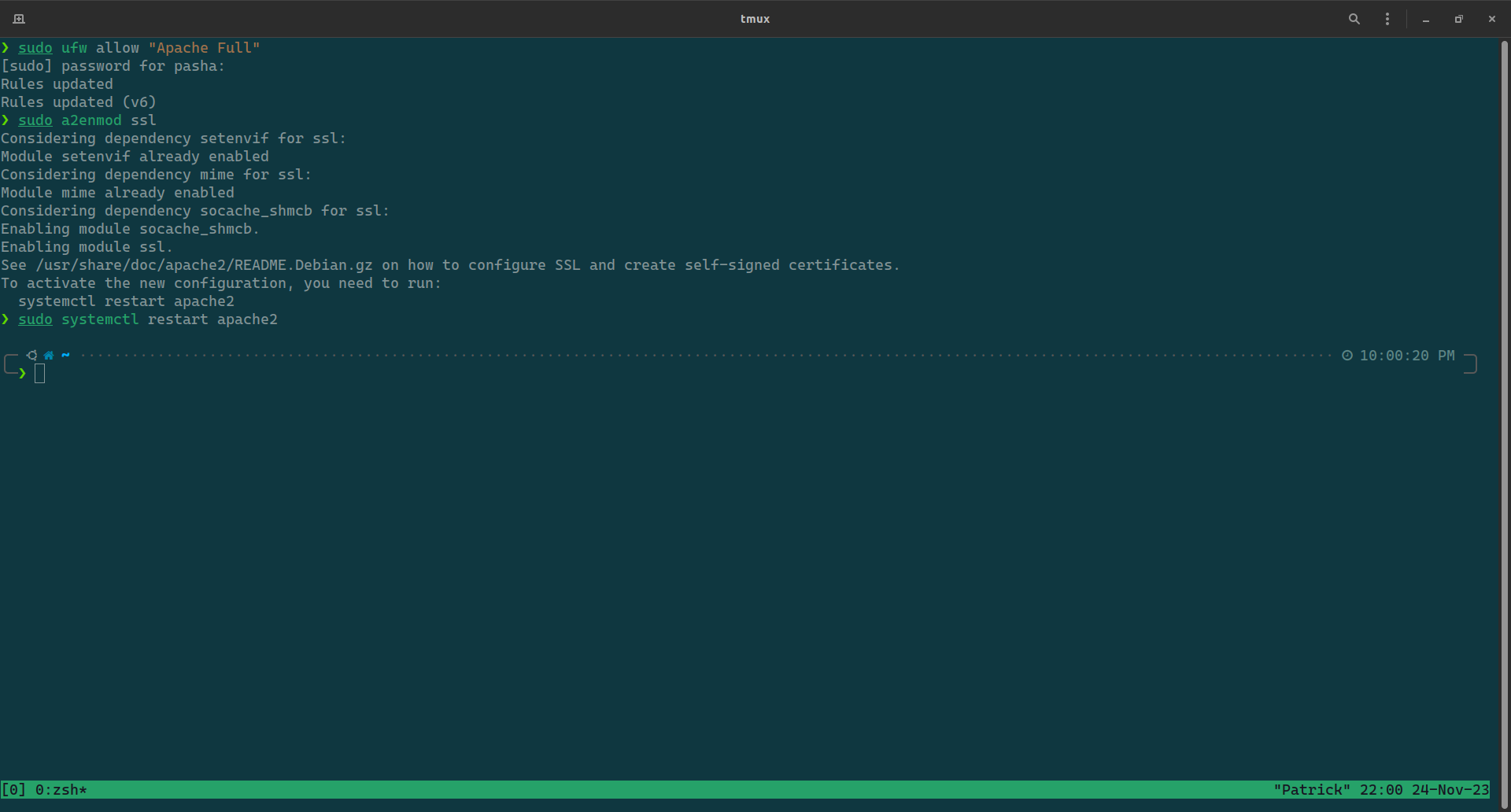
می‌دانیم که پسورد کاربر به صورت hash شده در سرور ذخیره شده و فرآیند انجام این hash را خود سرور انجام می‌دهد. در نتیجه لازم است کلاینت سوزرنیم و پسورد را به صورت خام برای سرور ارسال کند تا سرور بتواند با بررسی پسورد، اجازه ورود دهد. از طرفی در پروتکل HTTP هیچ‌گونه رمزنگاری‌ای وجود ندارد و کل داده به صورت plain text برای سرور ارسال می‌شود. به همین دلیل، اگر کسی در میانه راه بتواند بسته را شنود کند، بدون نیاز به انجام کاری می‌تواند کل داده رد و بدل شده را بخواند که در بین این داده‌ها، یوزرنیم و پسورد کاربر هم وجود دارد.

# راه‌اندازی سرور HTTPS با احراز هویت کاربری

## ایجاد گواهی SSL

ابتدا با استفاده از دستورات زیر، دسترسی apache به پورت 443 را آزاد کرده، ماژول ssl را فعال کرده، و سرویس apache را ری‌استارت می‌کنیم:

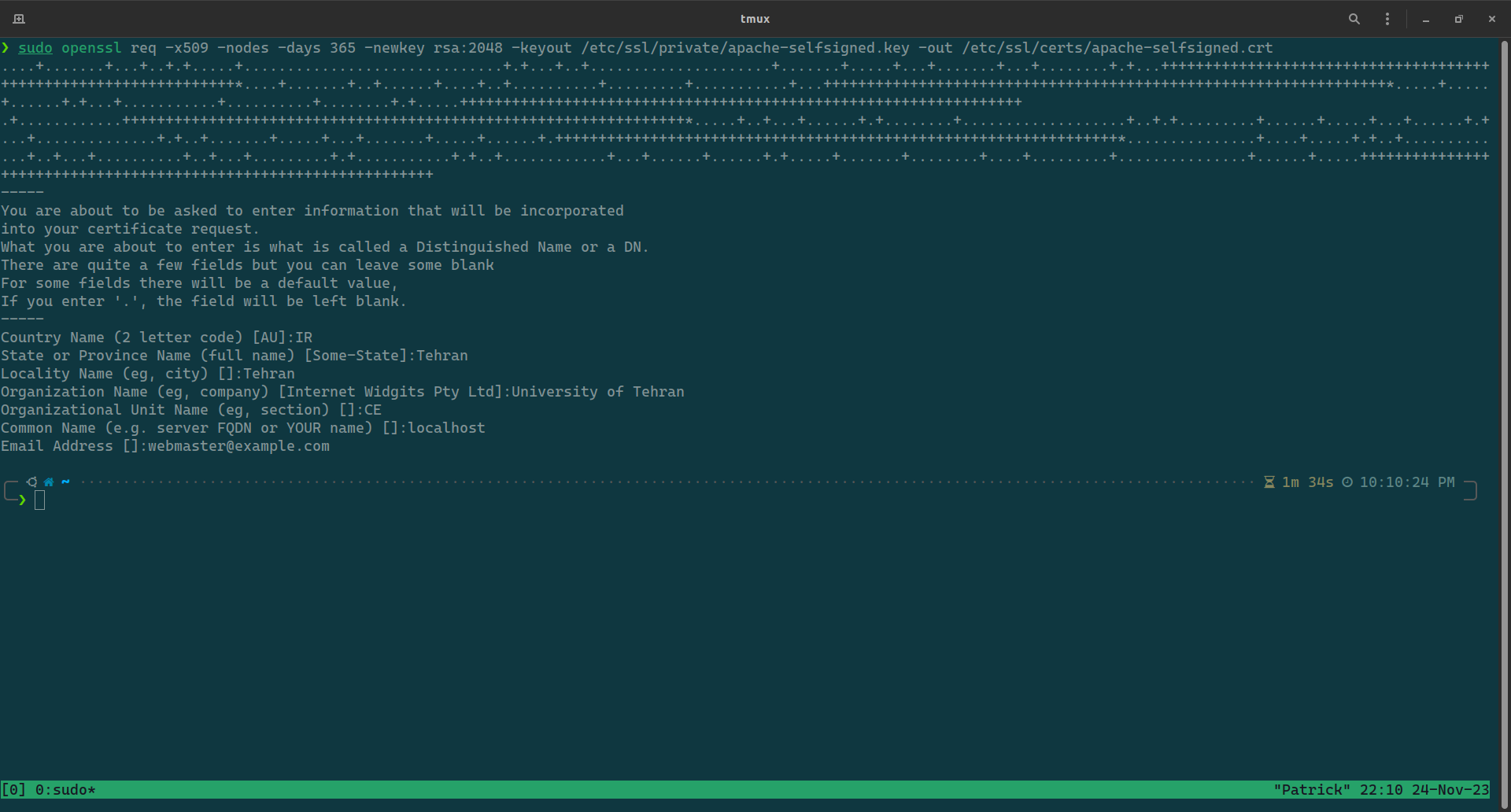
sudo ufw allow 'Apache Full'  
sudo a2enmod ssl  
sudo systemctl restart apache2



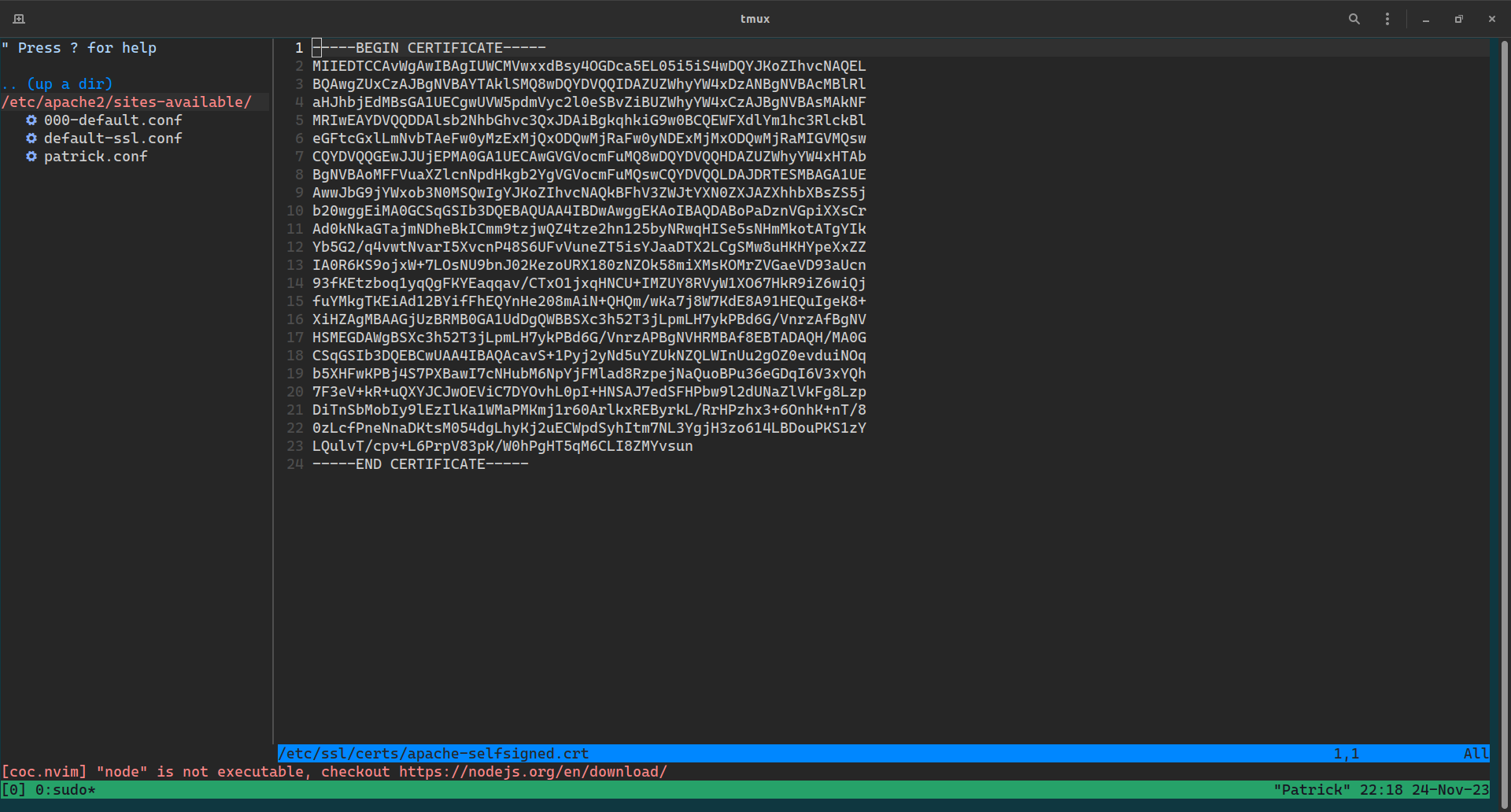
حال با استفاده از دستور زیر، یک گواهی SSL به صورت Self-Signed می‌سازیم:

sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.crt

این دستور یک گواهی SSL به همراه کلیدش در آدرس‌های ذکر شده تولید می‌کند. مقدار common name در ادامه این دستور، برابر با localhost قرار داده شده است.



این گواهی در تصویر زیر نمایش داده شده است:



حال باید این گواهی را به کانفیگ apache اضافه کنیم. برای این کار از دستور زیر استفاده کرده و فایل کانفیگ را ادیت می‌کنیم:

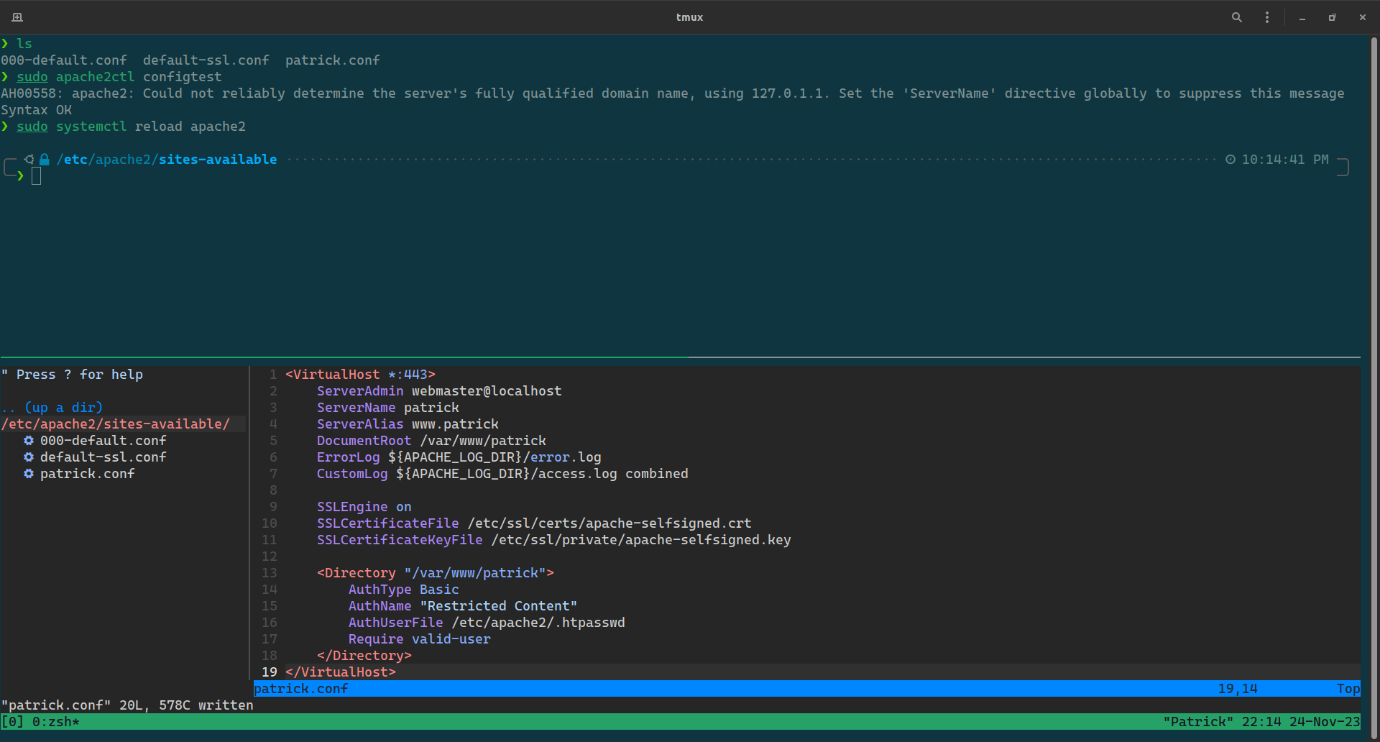
sudo nvim /etc/apache2/sites-available/patrick.conf

و سپس فایل را طوری تغییر داده که مقادیر زیر را داشته باشد:

<VirtualHost \*:443>  
 ServerName patrick  
 DocumentRoot /var/www/patrick  
  
 SSLEngine on  
 SSLCertificateFile /etc/ssl/certs/apache-selfsigned.crt  
 SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key  
</VirtualHost>

و با استفاده از دستورات زیر، کانفیگ را چک کرده و سرویس را ریلود می‌کنیم:

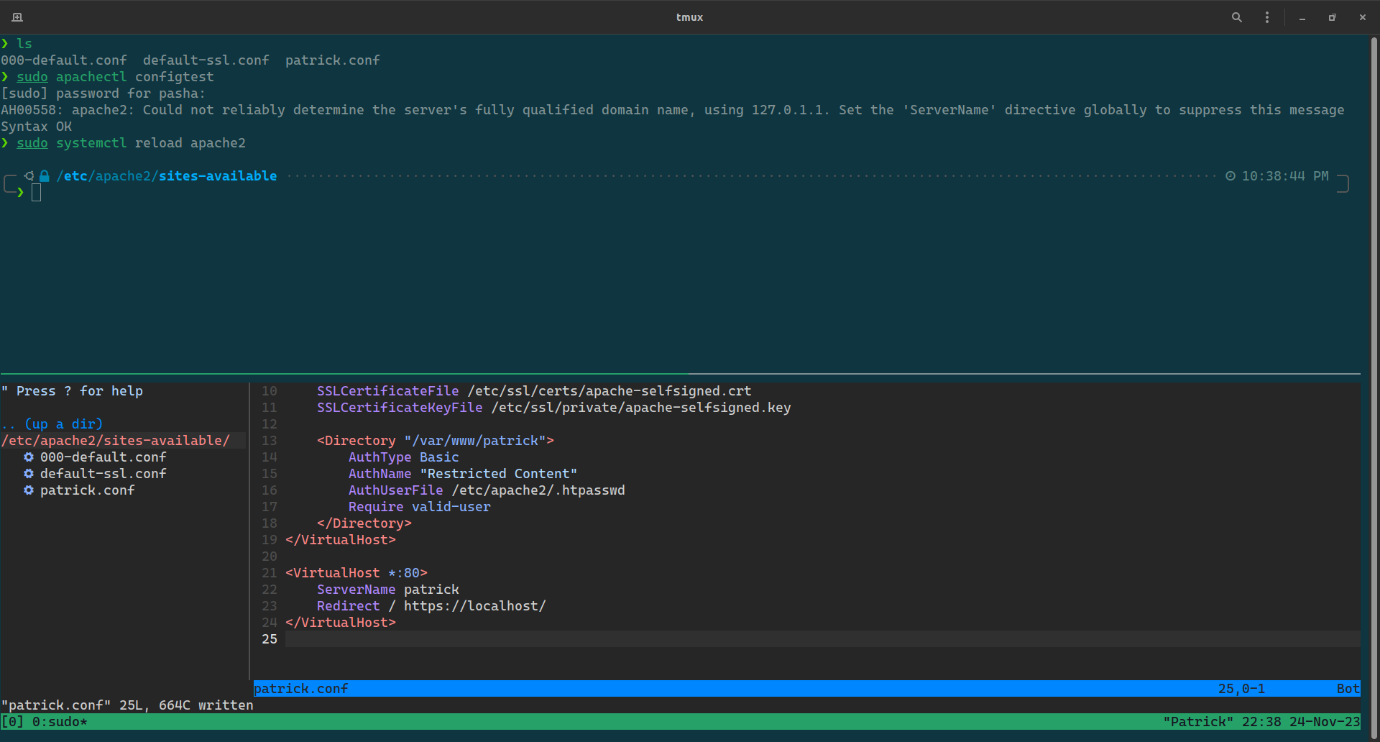
sudo apache2ctl configtest  
sudo systemctl reload apache2



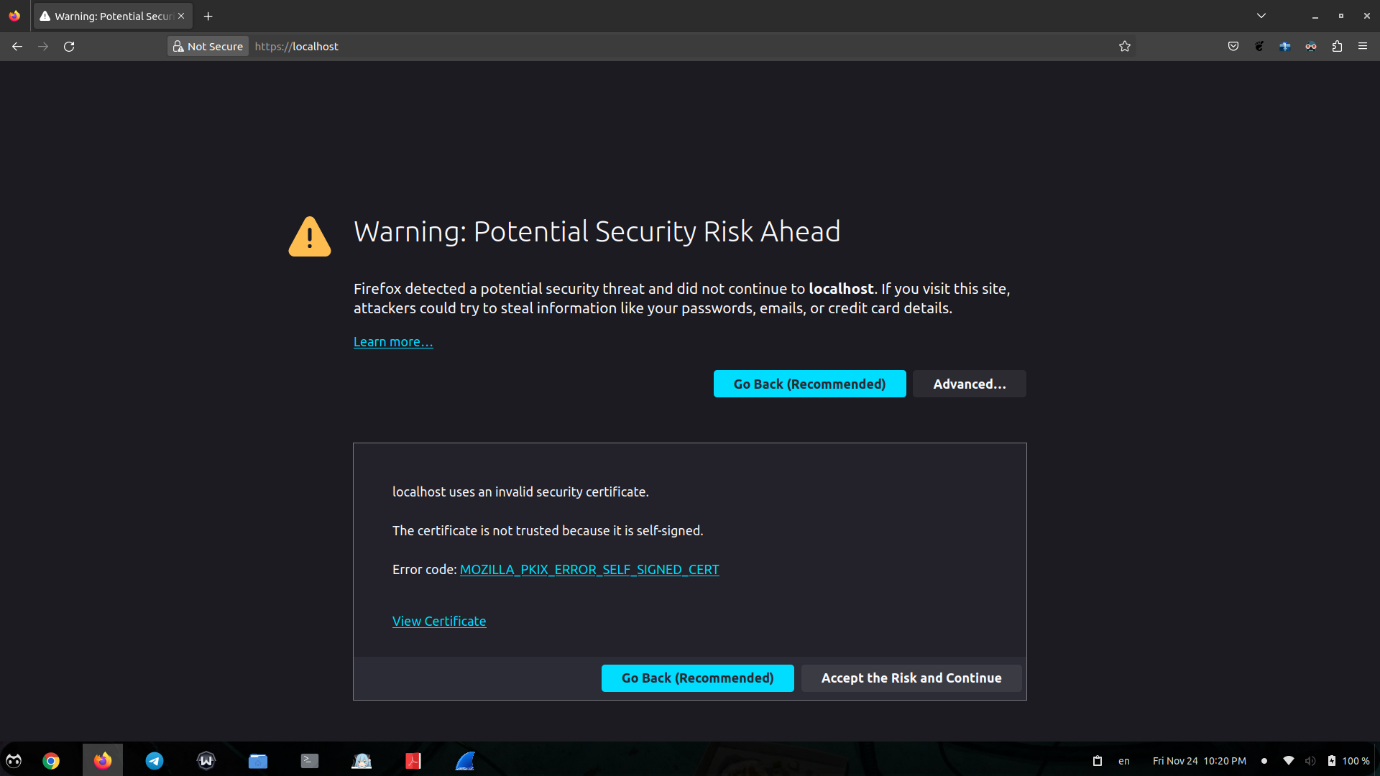
حال اگر سایت را با HTTP بررسی کنیم، می‌بینیم که صفحه اولیه apache لود شده و صفحه ما لود نمی‌شود. این به این دلیل است که سایت جدید فقط بر روی HTTPS است و برای HTTP تنظیم نشده است. به همین دلیل، مجددا کانفیگ را تغییر داده و HTTP را به HTTPS ری‌دایرکت می‌کنیم. برای این کار، مقدار زیر را به فایل کانفیگ اضافه می‌کنیم:

<VirtualHost \*:80>  
 ServerName patrick  
 Redirect / https://localhost/  
</VirtualHost>

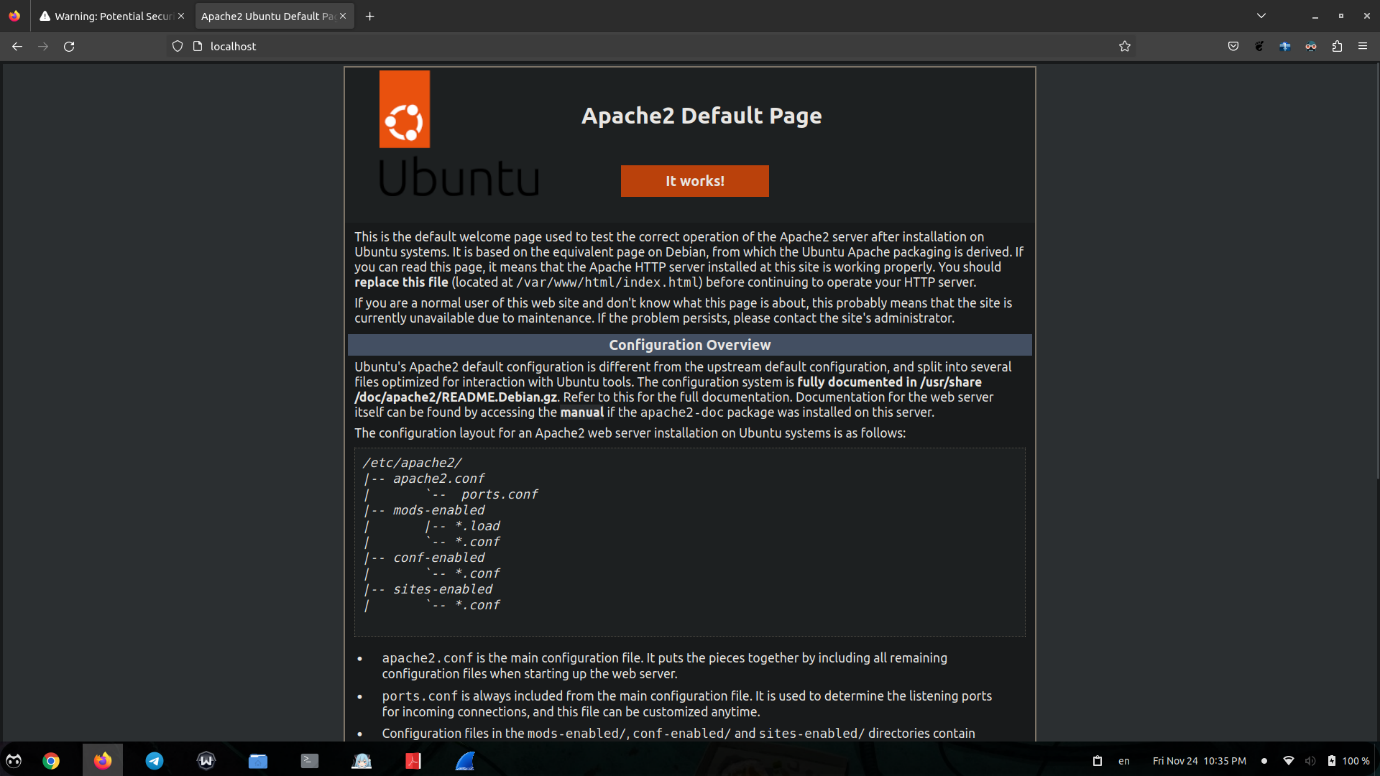
سپس مجددا کانفیگ را ریلود می‌کنیم:



## صفحه سایت در حالت HTTPS

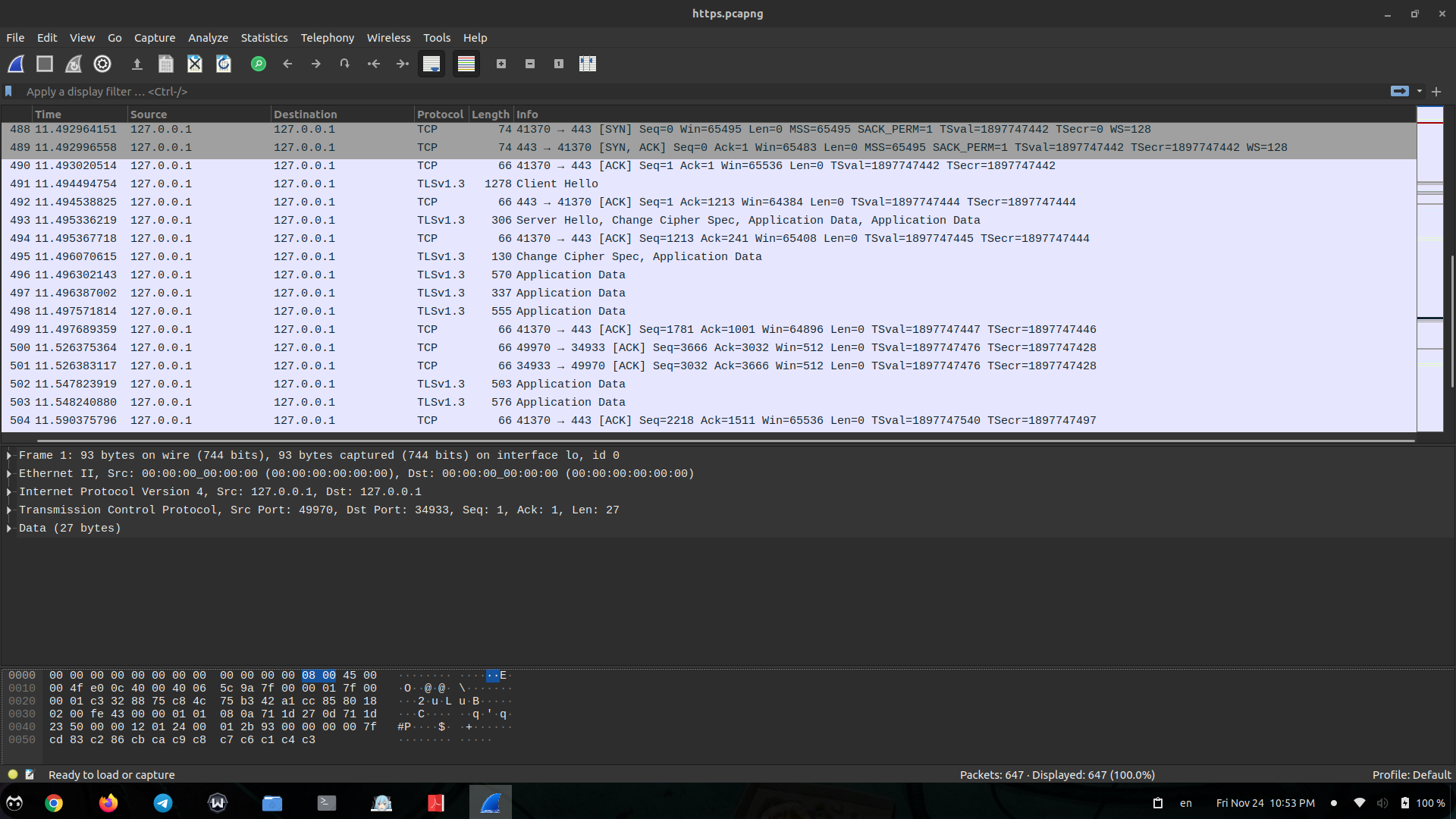


همچنین پیش از تنظیم Redirection، در حالت HTTP به صورت زیر باز می‌شد:



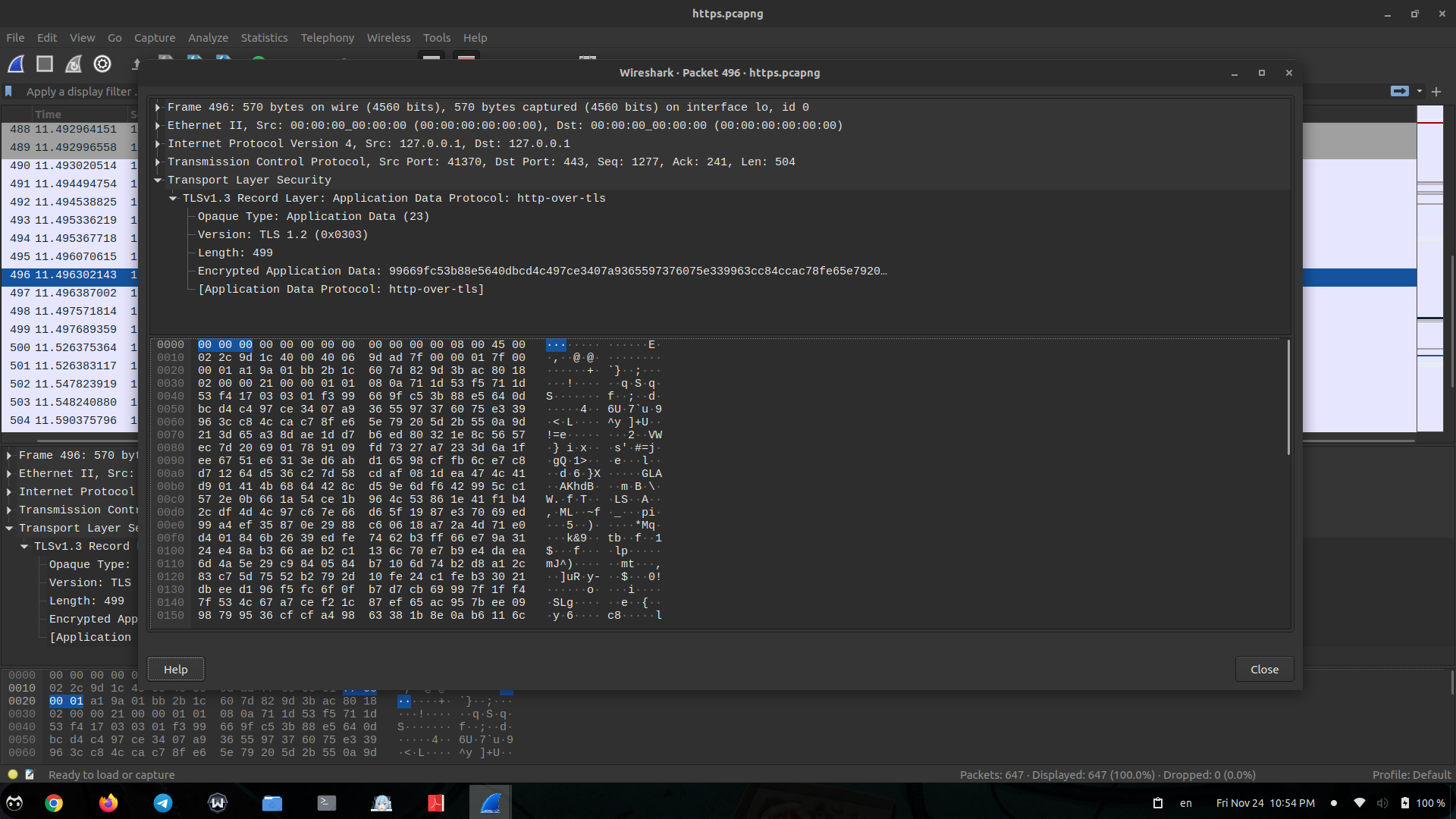
## ضبط بسته‌های HTTPS توسط wireshark

پس از شروع ضبط wireshark و وارد کردن یوزرنیم و پسورد، صفحه wireshark به صورت زیر است:



مشاهده می‌کنیم که دیگر پکت‌های HTTP وجود ندارند و تنها پکت‌های TCP و TLS را داریم.

حال یکی از این بسته‌ها را مشاهده می‌کنیم:



طبق تصویر، می‌بینیم که هیچ اطلاعات Authorizaton-ای وجود ندارد و تنها یک فیلد Encrypted Application Data داریم و اطلاعات به صورت رمزنگاری شده ارسال می‌شوند و کسی نمی‌تواند یوزرنیم و پسورد را مشاهده کند.

## علت عدم امکان شنود اطلاعات احراز هویت کاربری

زمانی که از HTTPS استفاده می‌کنیم، ابتدای اتصال به کمک یک الگوریتم نامتقارن، کلاینت و سرور یک کلید رمزنگاری متقارن را با همدیگر تبادل می‌کنند. از طرفی سرور certificate خود را برای کلاینت ارسال کرده و کلاینت با بررسی صحت آن، از اینکه فرد دیگری خود را به جای سرور جا نزده مطمئن می‌شود. مراحل کامل انجام این کار در بخش بعدی توضیح داده شده است.

از این پس، تمام اطلاعاتی که بین سرور و کلاینت رد و بدل شده به صورت رمز شده (توسط کلید متقارن انتخاب شده) ارسال می‌شوند و فردی نمی‌تواند با خواندن بسته، اطلاعات آن را بدست آورد زیرا رمزگشایی آن نیازمند دانستن کلید رمزنگاری متقارن است که چون این کلید به صورت plain text منتقل نشده و توسط یک الگوریتم نامتقارن رد و بدل شده است، این امکان برای آن فرد وجود نداشته و حمله MITM خنثی می‌شود.

## مراحل handshake و بسته‌های رد و بدل شده

این مراحل به صورت زیر هستند:

### Client Hello

کلاینت این بسته را برای سرور ارسال کرده که حاوی اطلاعات زیر است:

* ورژن‌های TLS که ساپورت می‌کند
* لیست cipher suite-هایی که ساپورت می‌شوند
* یک مقدار به نام client random

### Server Hello

سرور پس از دریافت بسته Client Hello این بسته با محتویات زیر را ارسال می‌کند:

* ورژن انتخاب شده TLS
* Cipher suite انتخاب شده
* یک مقدار رندوم به نام Server Random

### Change Cipher Spec

این پکت به این معنی است که از سرور از کلاینت می‌خواهد ارتباط را به حالت رمزنگاری شده تغییر دهد.

### Multiple Application Datas

این پکت‌ها شامل Certificate سرور و Server Finished هستند که به صورت رمزنگاری شده ارسال می‌شوند.

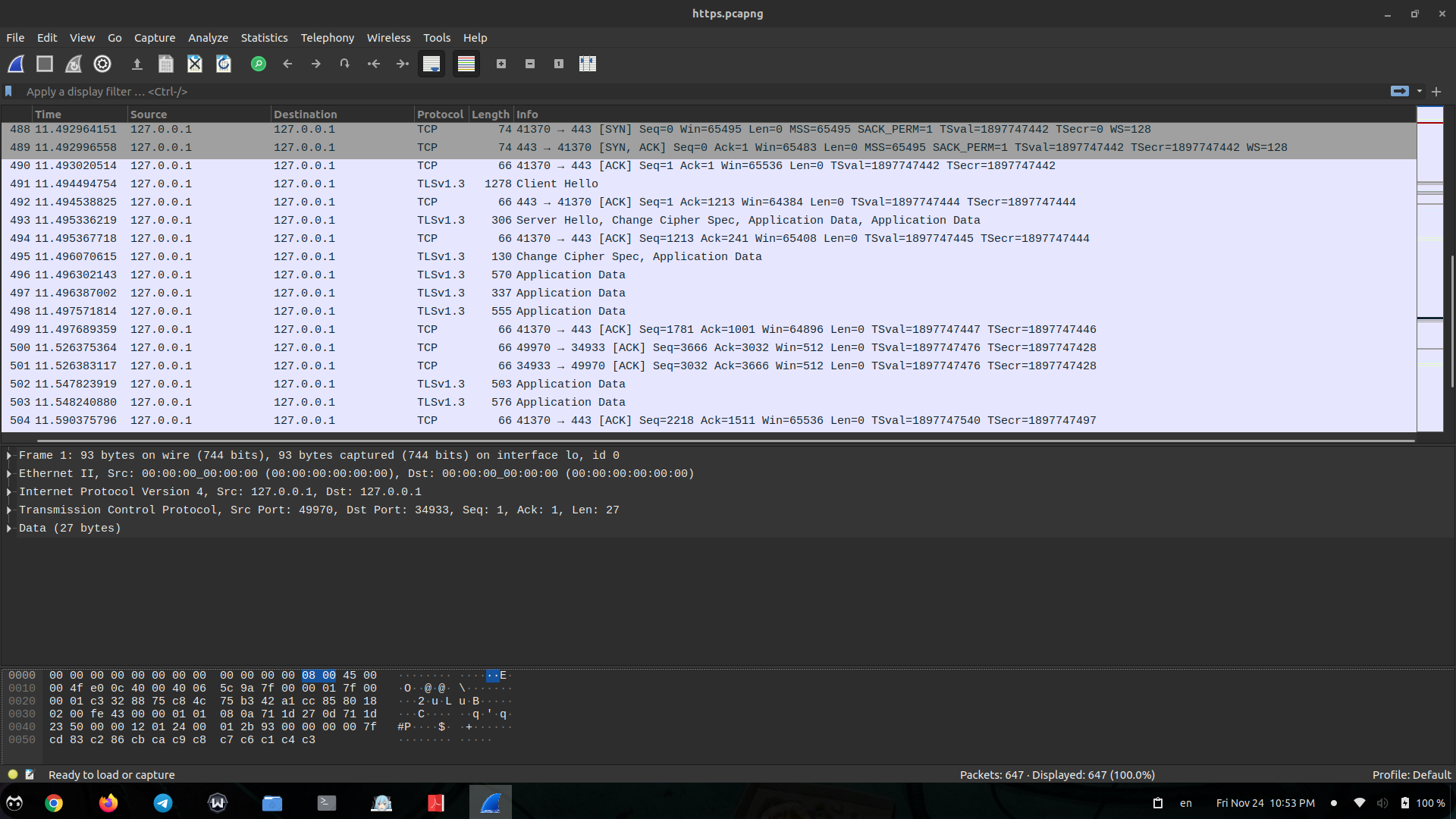
### Change Cipher Spec

این پکت از طرف کلاینت به سرور ارسال می‌شود و از او می‌خواهد که ارتباط رمزنگاری شود.

### Application Data

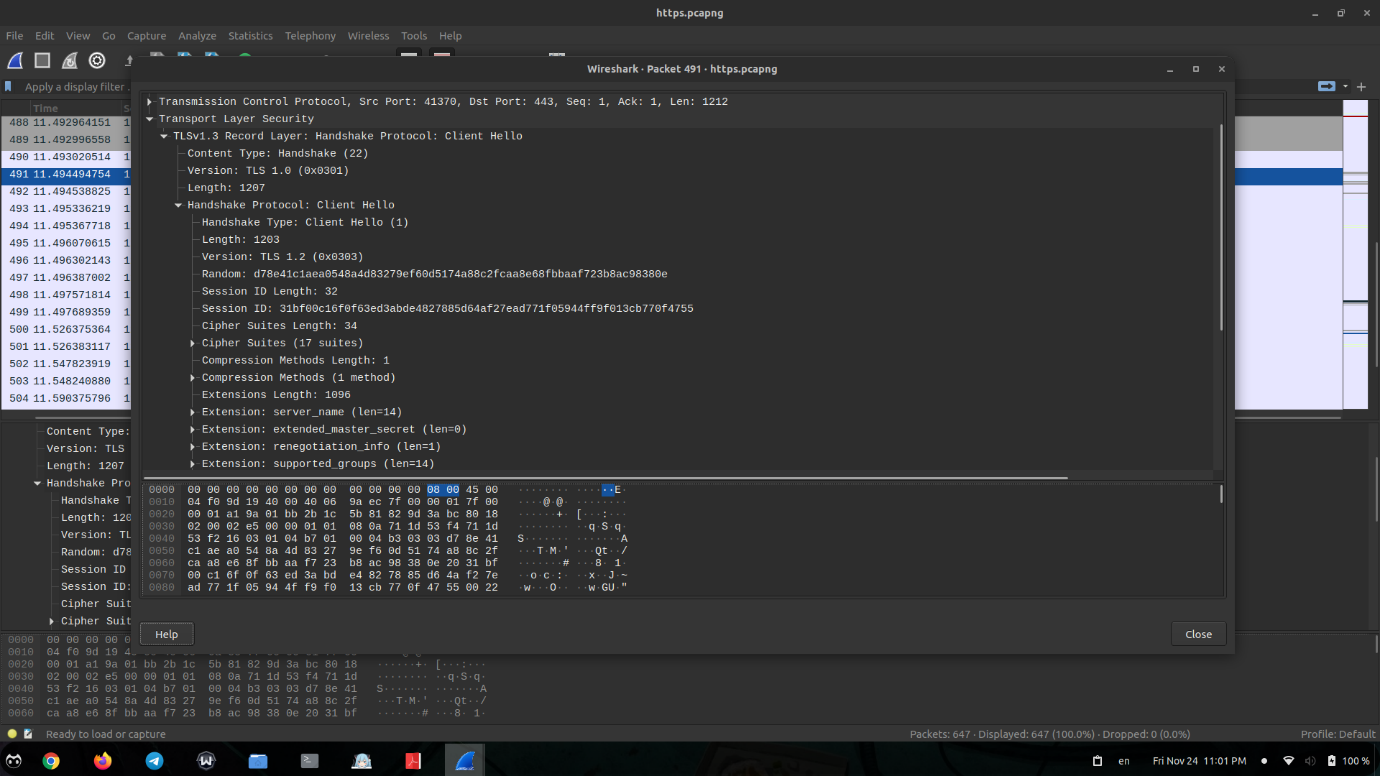
این پکت مجددا شامل Client Finished است که به صورت رمزنگاری شده ارسال می‌شود.

حال این مراحل را در wireshark می‌بینیم:

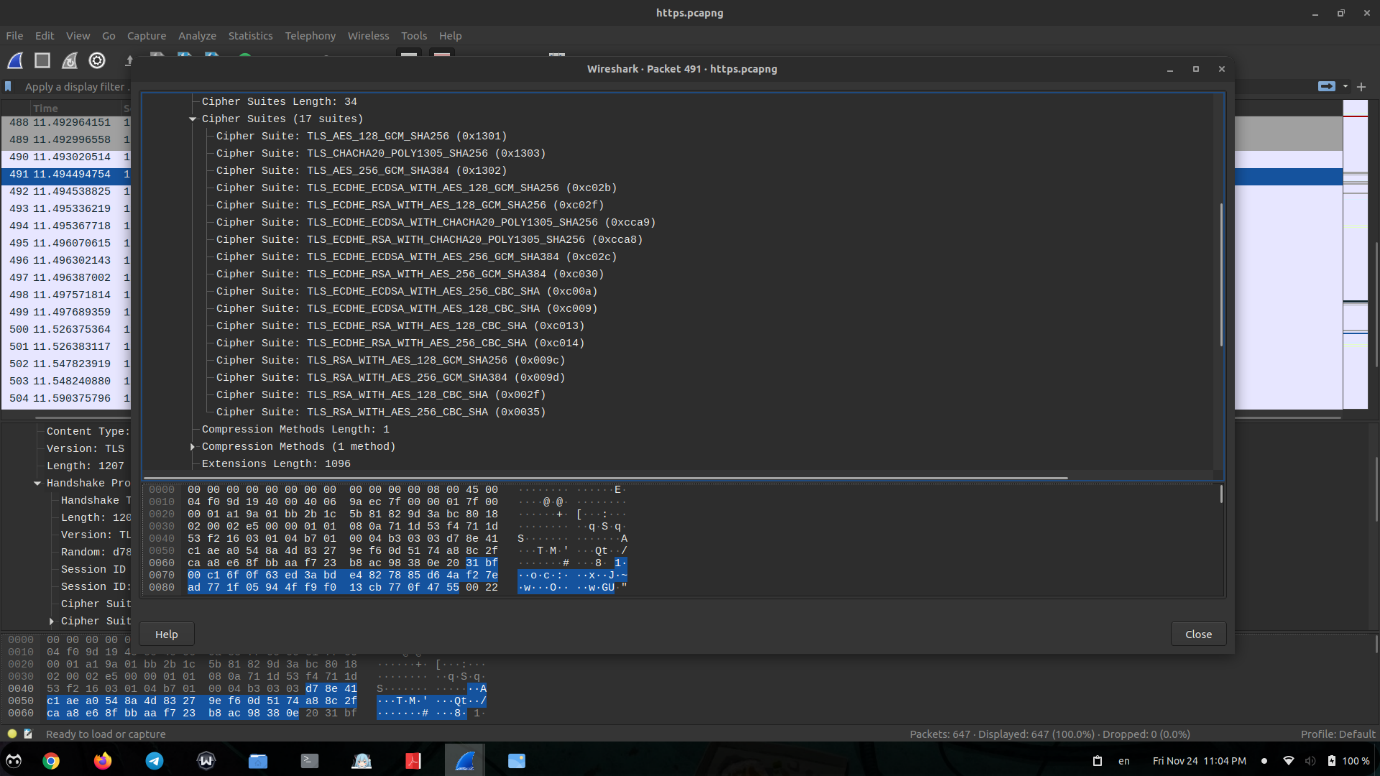


می‌بینیم که ابتدا پکت‌های Syn، Syn-Ack و Ack را برای برقراری ارتباط TCP داریم.

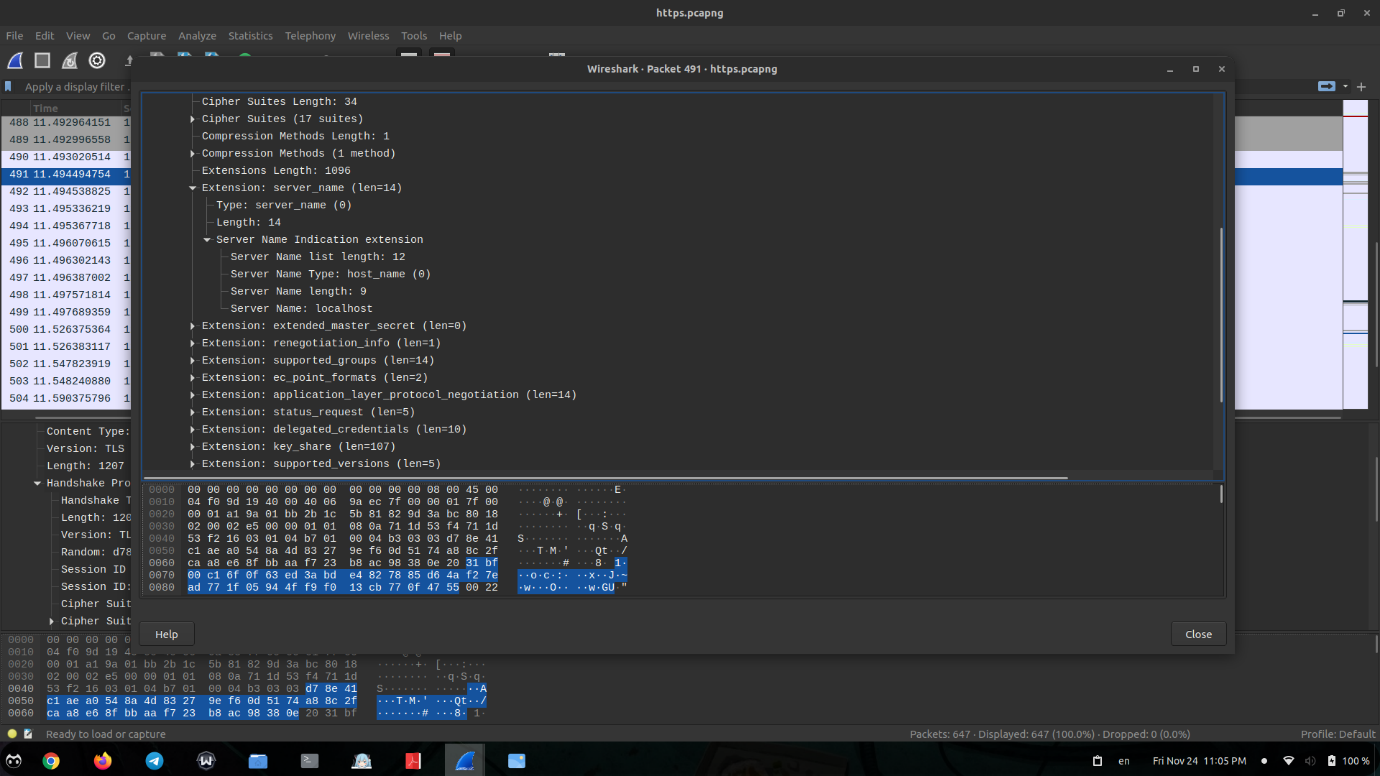
پس از آن، به پکت Client Hello می‌رسیم:



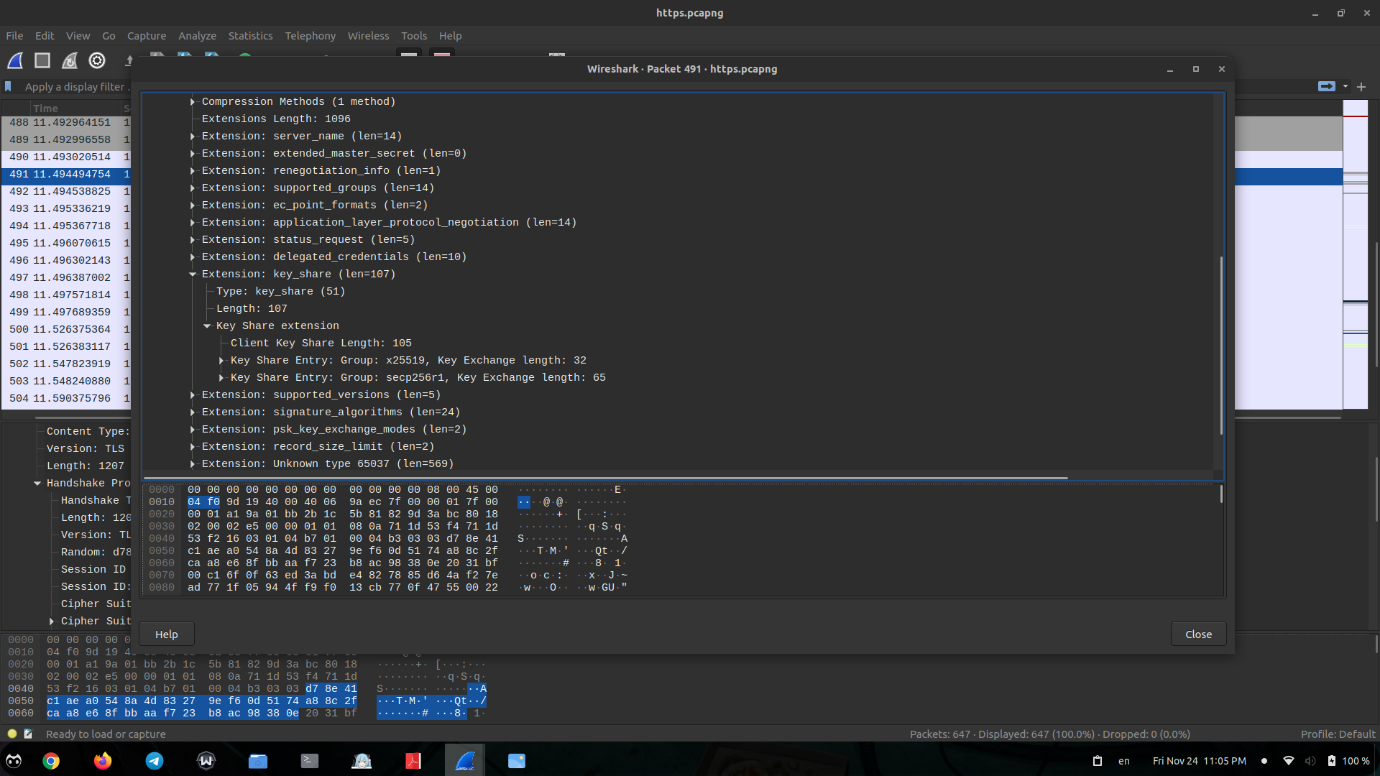
در این بخش می‌بینیم که ورژن TLS برابر با 1.2 انتخاب شده که صرفا برای Backward Compatibility است. پس از آن یک مقدار Random داریم که با عنوان Client Random می‌شناسیم. سپس تعدادی Cipher Suite مورد پذیرش داریم که در ادامه نشان داده شده‌اند:



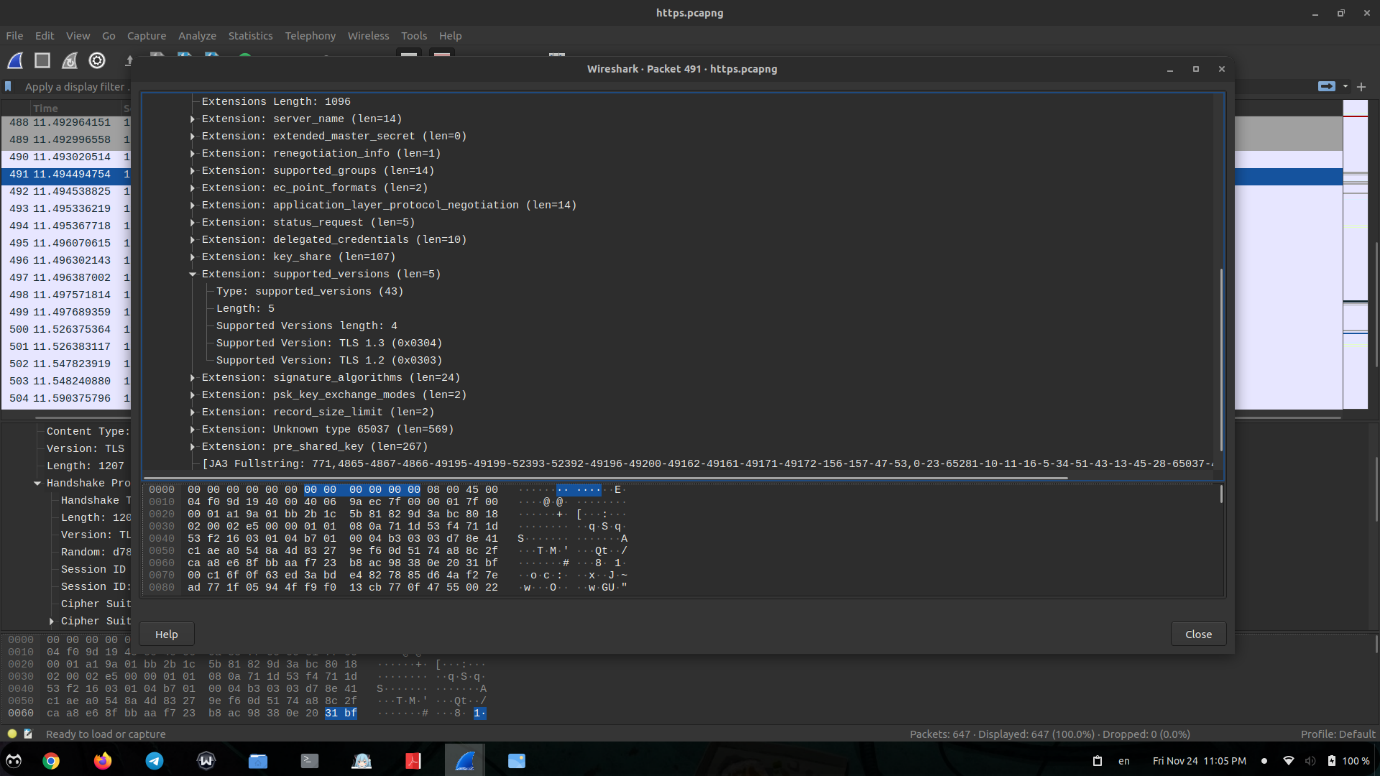
پس از آن، server\_name را خواهیم داشت که در این بخش، localhost است.



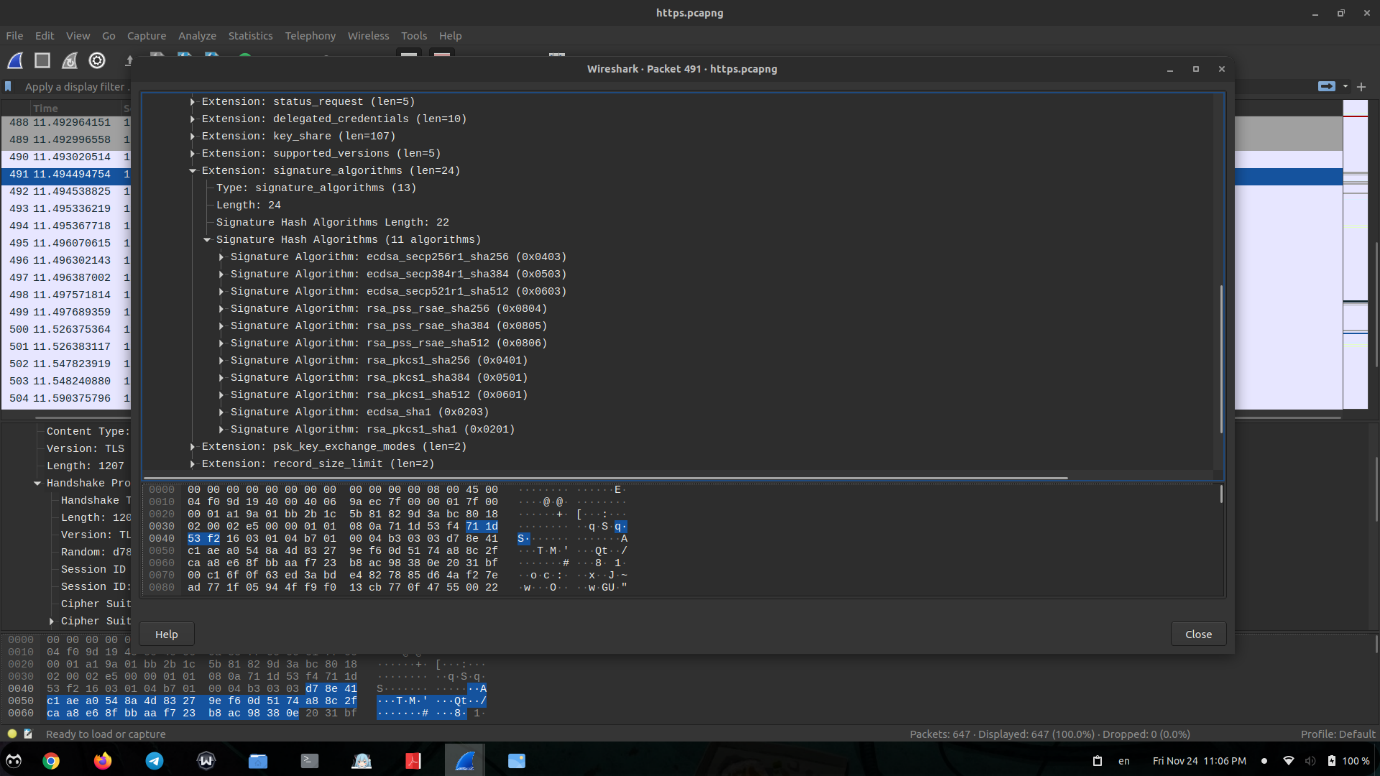
سپس key\_share را خواهیم داشت که برای ساخت رمز متقاون استفاده می‌شود.



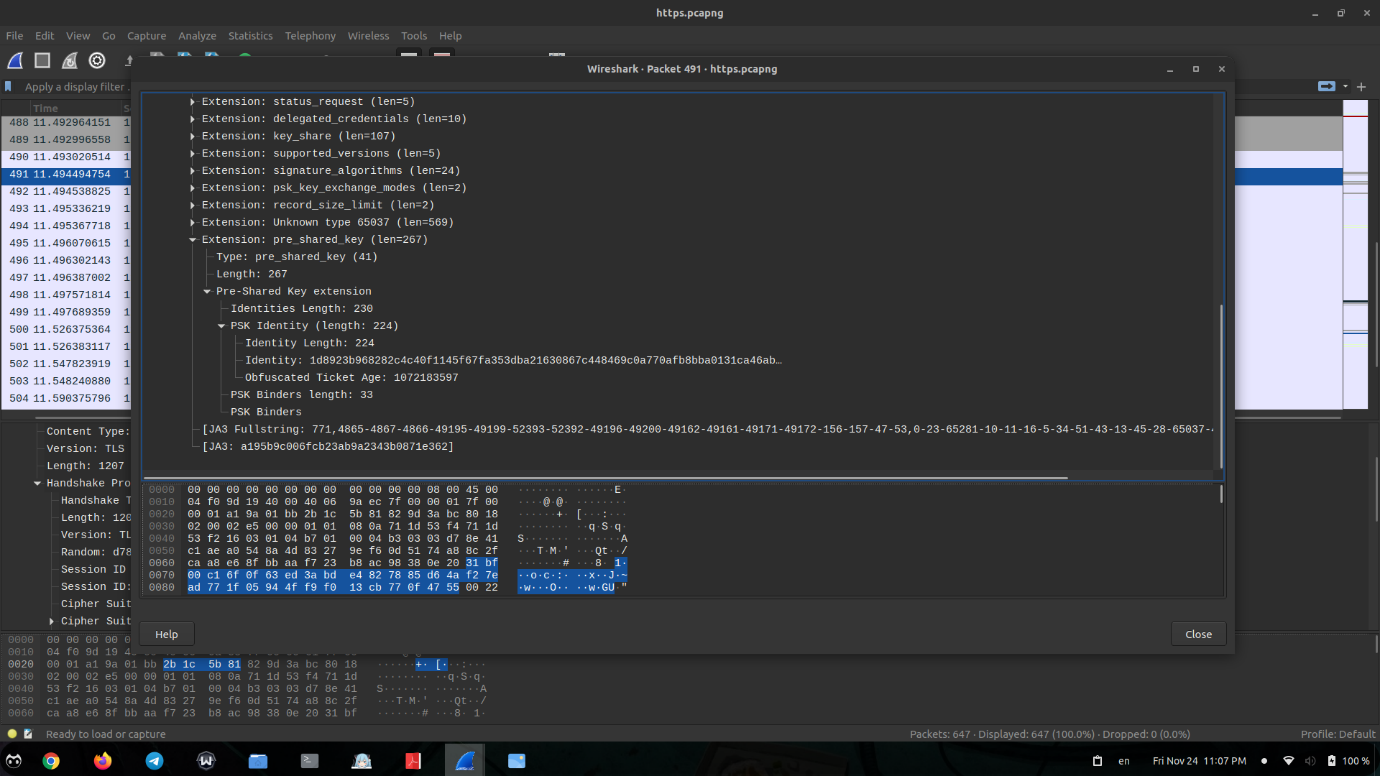
سپس ورژن‌های TLS که ساپورت می‌شوند:



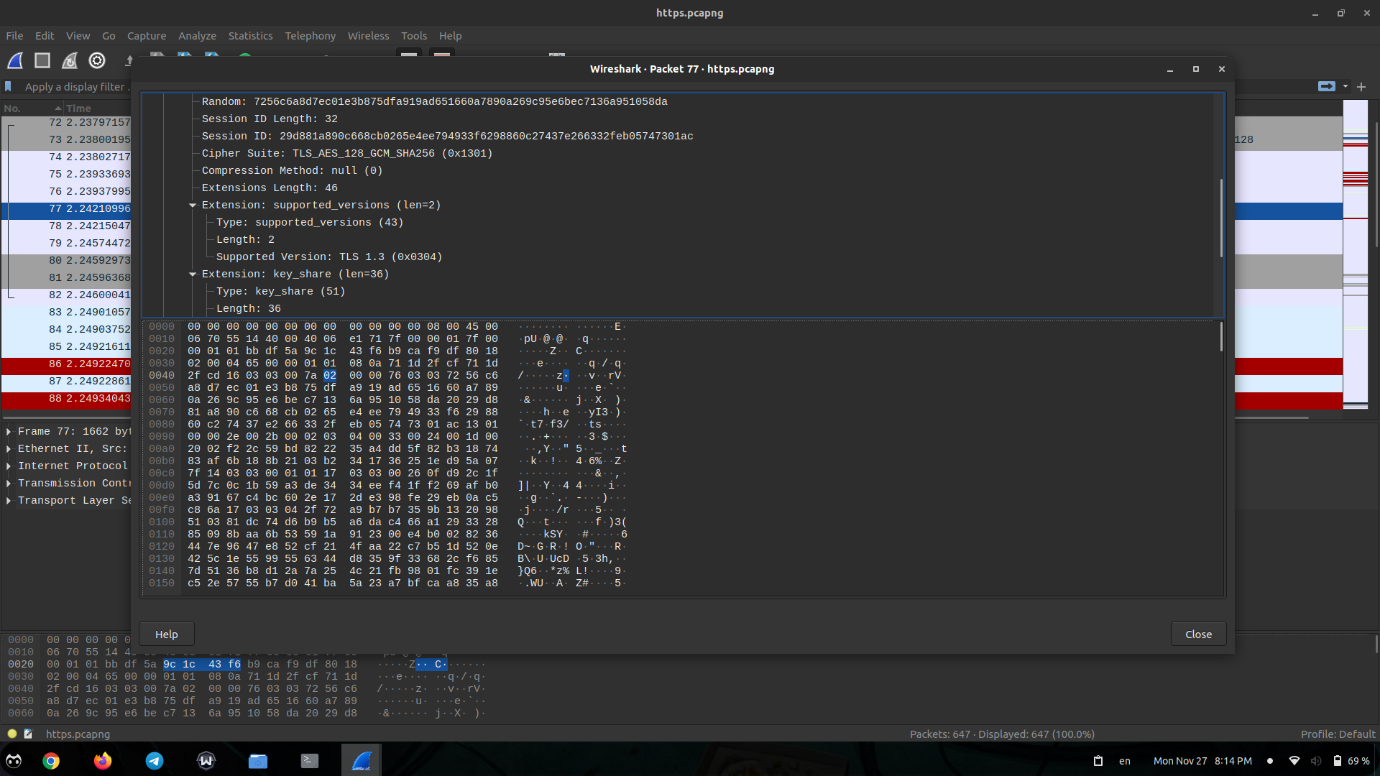
و پس از آن، الگوریتم‌های امضای دیجیتال که ساپورت می‌شوند:



در نهایت PreSharedKey را خواهیم داشت که بخشی از الگوریتم Key Exchange است.



حال به بسته Server Hello می‌رسیم:



می‌بینیم که مقادیر Random و Cipher Suite انتخاب شده و TLS Version ساپورت شده (1.3) و key\_share را داریم.

پس از آن، پکت‌های Change Cipher Spec و Finished را از هر دو طرف می‌بینیم.

همانطور که مشاهده کردیم، تمام بسته‌های لازم ارسال شده و در wireshark ضبط شده‌اند.

در واقع سرور و کلاینت با مقادیری که در بسته‌های hello دریافت کرده‌اند و private key-های خودشان، می‌توانند به یک کلید متقارن یکسان برسند و پیام‌های خود را رمزنگاری کنند.

در بخش Server Hello پس از ChangeCipherText تعداد چهار Application Data انتقال یافته که دو مورد از آن‌ها ذکر شده و از بین بقیه، می‌تواند شامل Certificate Request (به صورت اختیاری) باشد.