ROTEIRO DE EXECUÇÃO

EXERCÍCIOS PRÁTICOS DO MODULO 01 – REDES PARA OPERADORES NOC/SOC

RESUMO DA ATIVIDADE

CÓDIGO	DATA DO RELATÓRIO	PREPARADO POR
004	12/09/2022	Rafael Menezes

FILE TRANSFER PROTOCOL (FTP)

O FTP (File Transfer Protocol) ou Protocolo de Transferência de Arquivos, é um protocolo usado para compartilhar arquivos por redes de computadores. O protocolo funciona em modo cliente e servidor, no qual o servidor FTP é responsável por hospedar os arquivos e o cliente realiza o acesso destes arquivos diante uma autenticação previa.

Apesar do protocolo ter mais de 50 anos e não implementar por padrão mecanismos de segurança em criptografia, softwares de servidor FTP atuais oferecem o recurso para adicionar uma camada extra de segurança ao protocolo FTP baseada em criptografia.

O acesso a um servidor FTP pode ser gerenciado de duas maneiras:

- Anônimo
- Autenticado

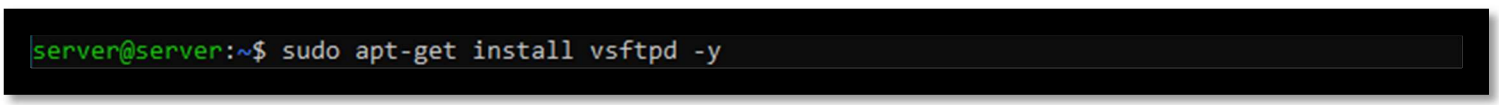
Acesos em modo Anônimo é muito insegura e não deve ser usada exceto em circunstâncias especiais. O acesso autenticado do usuário aos diretórios e arquivos do servidor FTP depende das permissões definidas para a conta usada no login.

Instalação e Configuração

Abaixo demonstramos como é realizada a instalação do servidor vsftpd para distribuições Ubuntu Linux. Algumas configurações de segurança e boas praticas também foram documentadas.

A instalação do servidor FTP vsftpd no Linux ubuntu pode ser realizada por meio do gerenciador de pacotes nativo como mostrado abaixo.

```
# Comando de instalação servidor vsftpd
~$ sudo apt-get install vsftpd -y
```



A configuração do servidor de FTP pode variar de ambiente para ambiente, nestas práticas vamos abordar uma configuração genérica com algumas boas praticas voltadas a proteção de dados e segurança cibernética. O principal arquivo de configuração se encontra em /etc/vsftpd.conf, abaixo demonstramos as algumas de suas principais variáveis de configuração.

Utilizando o editor de texto nano, vamos abrir o arquivo de configuração do vsftpd e analisar algumas variáveis de configuração.

Comando para abrir o arquivo de configuração.
~$ sudo nano /etc/vsftpd.conf

```
server@server:~$ sudo nano /etc/vsftpd.conf
```

local_enable – Permitir login com usuários locais (do sistema operacional).

```
# Uncomment this to allow local users to log in.
local_enable=YES
```

anonymous_enable - Desativar configuração de FTP anônimo.

```
# Allow anonymous FTP? (Disabled by default).
anonymous_enable=NO
```

write_enable – Desativar a permissão que usuários autenticados façam uploads ou substituam arquivos no servidor.

```
# Uncomment this to enable any form of FTP write command.
write_enable=NO
```

chroot_local_user – Mantem os usuários com acesso somente ao diretório local do usuário.

```
chroot_local_user=YES
allow_writeable_chroot=YES
```

ftp_banner – Desativar banner. Por padrão, o servidor FTP envia uma mensagem de boas vindas em texto para o cliente a que se conecta ao servidor. Um atacante pode explorar vulnerabilidades direcionadas ao saber a versão especifica do serviço em execução.

```
# You may fully customise the login banner string:
ftpd_banner=none
```

user_sub_token – Recebe $USER (nome do usuário que se conecta ao seu servidor FTP)

```
user_sub_token=$USER
```

local_root: Indica o diretório dos usuários de FTP. Neste exemplo, é o diretório /ftp na /home do usuario. Por exemplo, se você vinculou um Storage a maquina e deseja usá-lo, use local_root=/mnt/storage/. Para este caso vamos manter o local_root sendo local_root=/home/$USER/ftp

```
local_root=/home/$USER/ftp/
```

Apos as configurações realizadas, reinicie o serviço vsftpd para que estas configurações entrem em vigor.

Comando para reiniciar o serviço vsftpd

~$ sudo systemctl restart vsftpd.service

Comando verificar o status serviço vsftpd

~$ sudo systemctl status vsftpd.service

```
server@server:~$ sudo systemctl restart vsftpd.service
server@server:~$ sudo systemctl status vsftpd.service
● vsftpd.service - vsftpd FTP server
     Loaded: loaded (/lib/systemd/system/vsftpd.service; enabled; vendor preset: enabled)
     Active: active (running) since Wed 2022-10-26 03:23:25 UTC; 7s ago
    Process: 2347 ExecStartPre=/bin/mkdir -p /var/run/vsftpd/empty (code=exited, status=0/SUCCES
```

Firewall Liberar Portas para Acesso Externo

Liberar porta 20 usada na transferência de dados.

~$ sudo ufw allow 20/tcp

```
server@server:~$ sudo ufw allow 20/tcp
Skipping adding existing rule
Skipping adding existing rule (v6)
```

Liberar porta 21 usada para o controle da conexão.

~$ sudo ufw allow 21/tcp

```
server@server:~$ sudo ufw allow 21/tcp
Skipping adding existing rule
Skipping adding existing rule (v6)
```

Usuários e Controle de Acesso

Vamos criar primeiro um novo usuário e apos isto inserir o nome do usuário no arquivo vsftpd.userlist para permitir o acesso ao serviço.

Adiciona ao sistema o usuário ftp_user, com senha password.

~$ sudo adduser ftp_user

```
server@server:~$ sudo adduser ftp_user
```

Crie o diretório compartilhado.

~$ sudo mkdir /home/ftp_user/ftp

```
server@server:/$ sudo mkdir /home/ftp_user/ftp
```

Remova a permissão de escrita e execução no diretório compartilhado.

~$ sudo chmod a-w /home/ftp_user/ftp

```
server@server:/$ sudo chmod a-w /home/ftp_user/ftp
```

Transfira a propriedade do diretório para o usuario ftp_user.

~$ sudo chown ftp_user:ftp_user /home/ftp_user/ftp

```
server@server:/$ sudo chown ftp_user:ftp_user /home/ftp_user/files
```

Agora crie o arquivo vsftpd.userlist no qual vamos gerenciar os nomes dos usuários com permissão de acesso ao serviço FTP.

~$ sudo touch /etc/vsftpd.userlist

```
server@server:~$ sudo touch /etc/vsftpd.userlist
```

Por padrão, todos os usuários do sistema podem se conectar com o servidor FTP, para que isso não ocorra, habilite a opção userlist_enable, seguido da variável userlist_deny no arquivo de configuração /etc/vsftpd.userlist. Tais variáveis são responsáveis pela lógica operacional da política a ser aplicada, ou seja, se será uma política permissiva ou restritiva. Neste caso vamos usar uma política restritiva na qual somente os usuários que constarem na userlist poderão ter acesso ao FTP.

userlist_enable=YES

userlist_deny=NO

```
userlist_enable=YES
userlist_deny=NO
```

Adicione a variável userlist_file com caminho do arquivo contendo os nomes de usuários que vão ter acesso ao serviço FTP.

userlist_file=/etc/vsftpd.userlist

```
userlist_file=/etc/vsftpd.userlist
```

Adiciona o nome do usuário (ftp_user) ao arquivo de acesso ao serviço FTP.

~$ sudo bash -c "echo ftp_user >> /etc/vsftpd.userlist"

```
server@server:~$ sudo bash -c 'echo ftp_user >> /etc/vsftpd.userlist'
```

Comunicação FTP Criptografada Utilizado o Protocolo TLS

Com intuito de evitar o acesso não autorizado dos dados trafegados durante a transferência entre o servidor e o cliente FTP, é necessário habilitar a criptografia desses dados utilizando o protocolo TLS.

ssl_enable – Ativa a comunicação FTP criptografada utilizado o protocolo TLS

```
ssl_enable=YES
```

Gerar certificados SSL/TLS

~$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/vsftpd.key -out /etc/ssl/private/vsftpd.pem

```
server@server:~$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/vsftpd.key -out /e
tc/ssl/private/vsftpd.pem
```

Especificar no arquivo /etc/vsftpd.conf as variáveis de certificados para atender os caminhos das chaves geradas anteriormente como apresentado:

rsa_cert_file=/etc/ssl/private/vsftpd.pem

rsa_private_key_file=/etc/ssl/private/vsftpd.key

```
#
# This option specifies the location of the RSA certificate to use for SSL
# encrypted connections.
rsa_cert_file=/etc/ssl/private/vsftpd.pem
rsa_private_key_file=/etc/ssl/private/vsftpd.pem
```

Em seguida, restringiremos o tipo de conexão ao TLS, que é mais seguro que o SSL. Faremos isso permitindo explicitamente o TLS e negando o uso de SSL no arquivo de configuração /etc/vsftpd.conf:

ssl_tlsv1=YES

ssl_sslv2=NO

ssl_sslv3=NO

```
ssl_tlsv1=YES
ssl_sslv2=NO
ssl_sslv3=NO
```

Alguns paramentos extras como os mostrados abaixo podem ser adicionados para garantir o uso da criptografia.

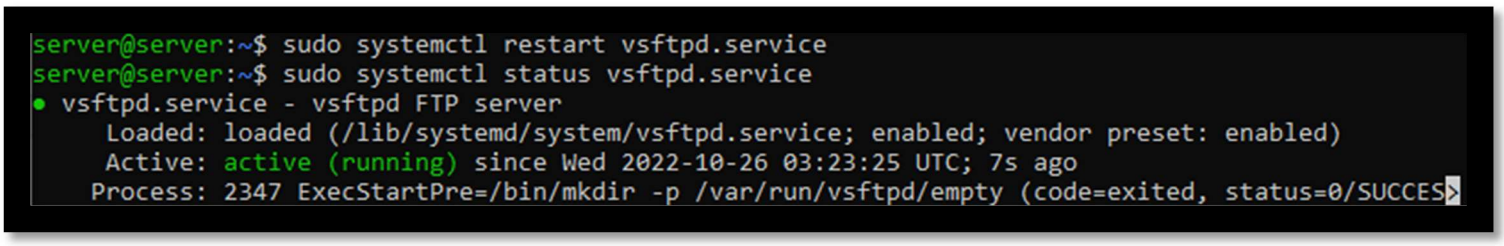
force_local_data_ssl=YES

force_local_logins_ssl=YES

```
force_local_data_ssl=YES
force_local_logins_ssl=YES
```

```
# Comando para reiniciar o serviço vsftpd
~$ sudo systemctl restart vsftpd.service

# Comando verificar o status serviço vsftpd
~$ sudo systemctl status vsftpd.service
```



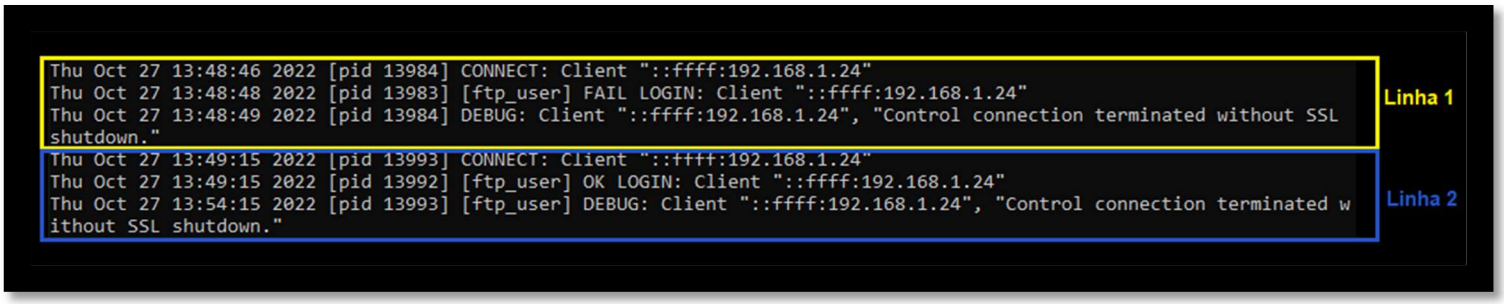
Logs do servidor FTP (vsftpd)

Para habilitar os logs do servidor vsftpd, ajuste a variável de configuração xferlog_file no arquivo /etc/vsftpd.conf como mostrado abaixo:

xferlog_file – Configura o caminho dos logs do serviço, por padrão mantemos o caminho pré-definido sendo /var/log/vsftpd.log

```
# You may override where the log file goes if you like. The default is shown
# below.
xferlog_file=/var/log/vsftpd.log
```

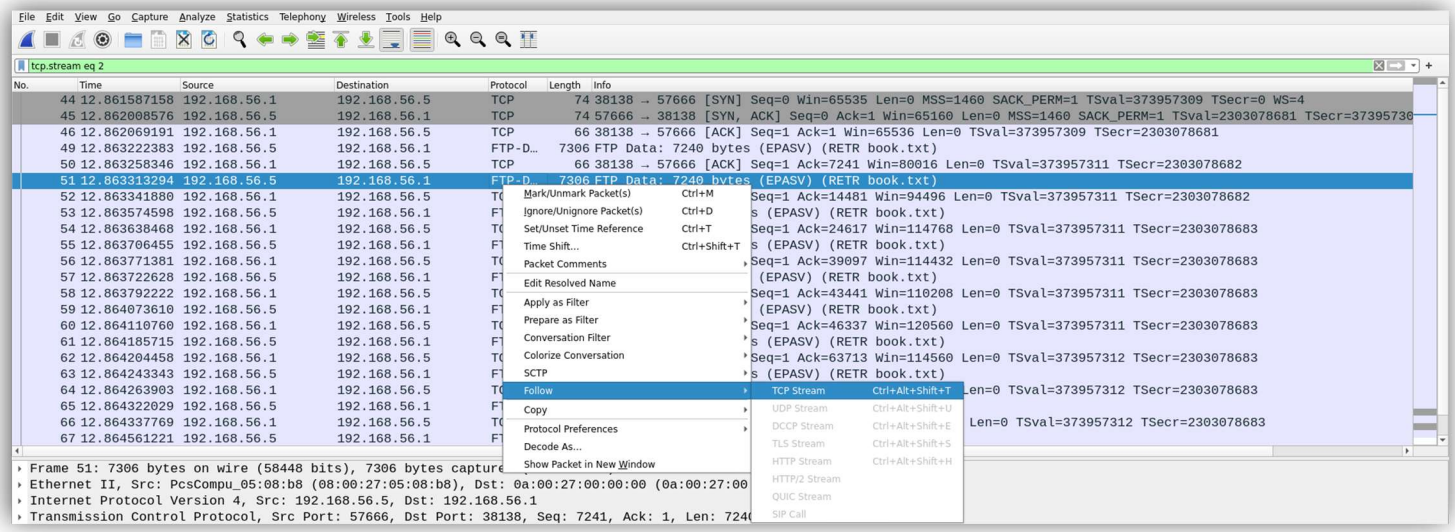
Um trecho do arquivo de log real FTP é apresentado com dois exemplos frequentemente encontrados em logs deste serviço. A imagem esta dividida em duas linhas de log sendo a primeira linha demonstrando uma tentativa de conexão ao servidor FTP que resultou em falha na autenticação devido as credencias de acesso. A segunda linha apresenta uma conexão bem-sucedida com o serviço FTP utilizando o canal de comunicação de dados criptografado.



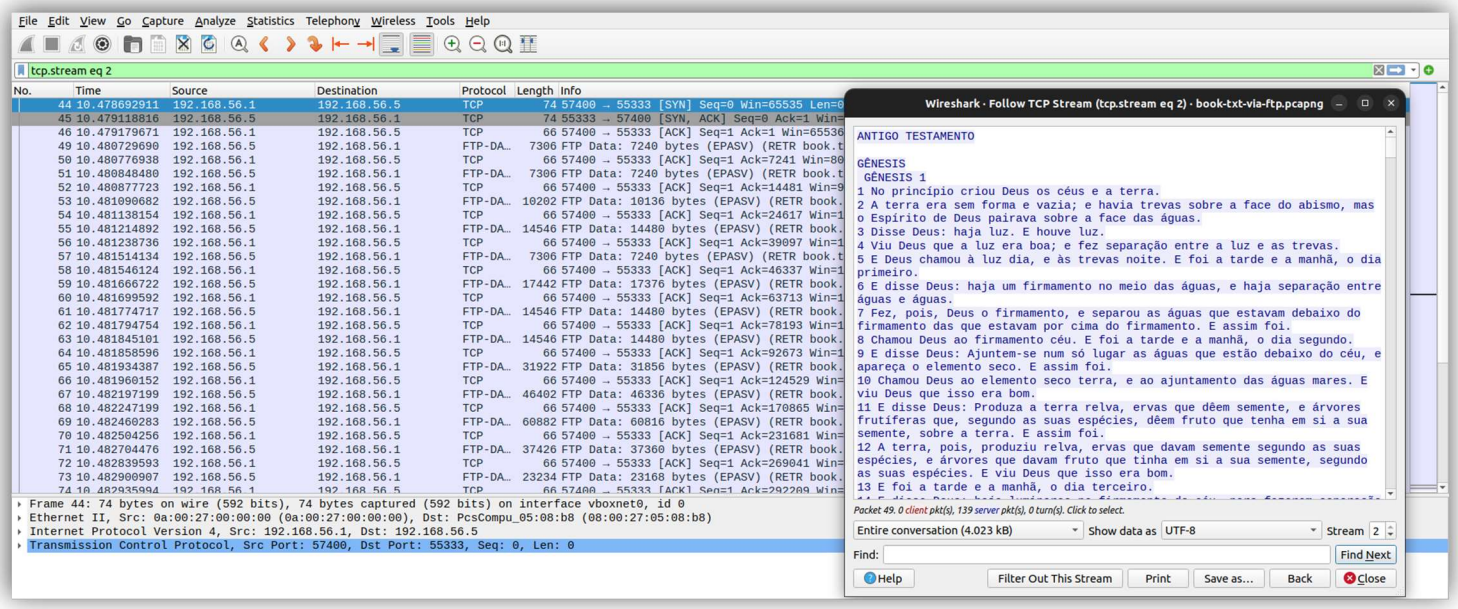
Demonstração de Tráfego de rede FTP Seguro

Para demonstra a fragilidade de protocolos que não utilizam criptografia de dados, assim como o FTP em seu modo parão, foi realizado uma prática de transferência de um arquivo TXT (não criptografado) entre duas máquinas na mesma rede.

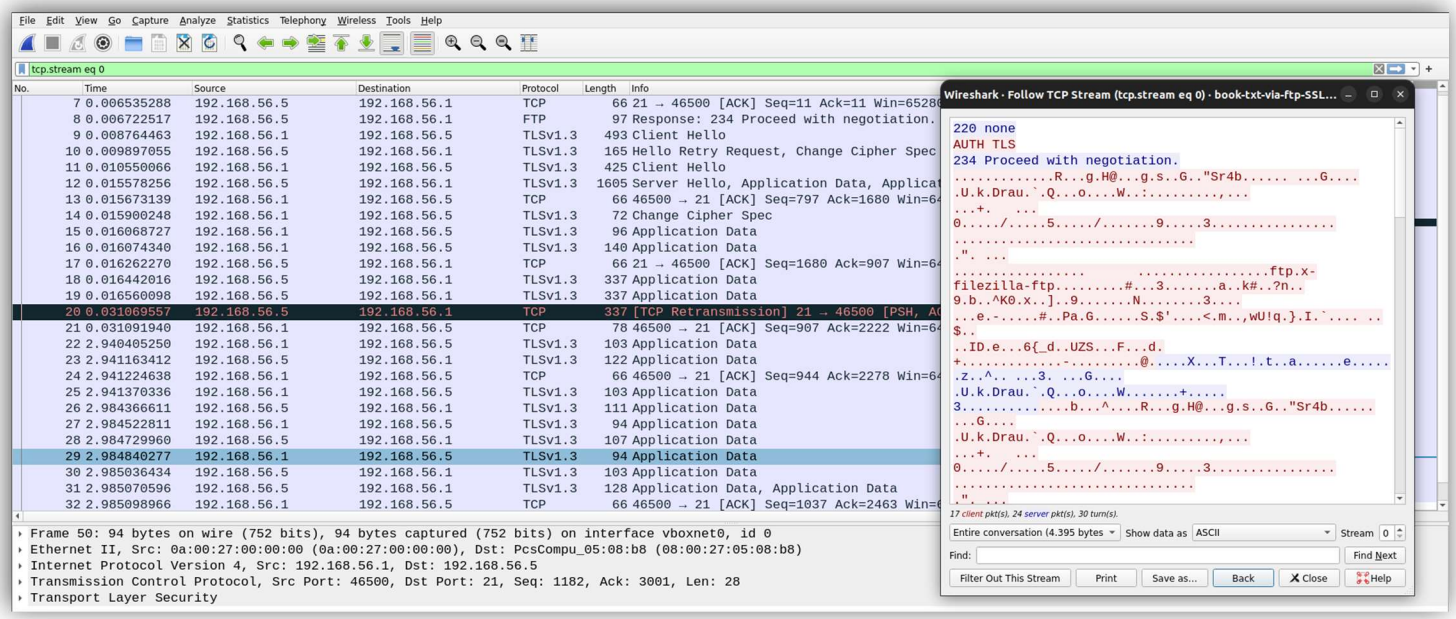
Uma terceira máquina na mesma rede em uma porta espelhada capturou o tráfego de dados FTP e visualizou o conteúdo do mesmo, como aprestando nas imagens abaixo. A imagem abaixo é possível ativar a opção TCP Stream e acompanhar o fluxo de comunicação cliente e servidor.



Com TCP Stream visualizamos todo tráfego de dados em claro que foi trafegado entre as partes.



Realizado o mesmo procedimento de conexão FTP e transferência do dado porem utilizando criptografia TLS. Nota-se que a mensagem esta criptografada e não é possível visualizar seu conteúdo através da intercetação do tráfego de rede.



O tráfego de rede e materiais extras desta prática foram capturados e disponibilizados nos links:

Link do livro em TXT: https://raw.githubusercontent.com/rmmenezes/services-linux/main/ftp-config/book.txt

Link do tráfego de rede FTP: https://raw.githubusercontent.com/rmmenezes/services-linux/main/book-txt-via-ftp.pcapng

Link do tráfego de rede FTPS: https://raw.githubusercontent.com/rmmenezes/services-linux/main/book-txt-via-ftp-SSL.pcapng

Conteúdo Extra	
Lista de Comandos CLI Comumente Aceitos por servidores e clientes FTPs	
!	Executa o comando na máquina local.
?	Semelhante a HELP.
APPEND	Adiciona dados a um arquivo existente.
ASCII	Configura o tipo de transferência de arquivos para ASCII.
BELL	Emite um bip quando um comando é executado.
BINARY	Configura o tipo de transferência de arquivos para binário.
BYE	Encerra a sessão FTP.
CD	Seguido de caminho/diretório muda para o diretório informado.
DELETE	Apaga um arquivo. Para mais de um arquivo usa se MDELETE.
DEBUG	Estabelece a modalidade de depuração.
DIR	Mostra o conteúdo do diretório servidor atual.
DISCONECTED	Semelhante a BYE.
GET	Obtém um arquivo do servidor. Para mais de um arquivo usa se MGET.
GLOB	Seleciona a expansão para nomes de arquivo.
HASH	Demonstra cada bloco do arquivo durante a transferência.
HELP	Lista sumariamente todos os comandos disponíveis.
LITERAL	Permite enviar comandos arbitrários.
LS	Mostra uma abreviada do conteúdo do diretório servidor.
MKDIR	Cria um diretório ou subdiretório no servidor.
PROMPT	Ativa/desativa o modo interativo.
PUT	Envia um arquivo ao servidor. Para enviar mais de um arquivo usa se MPUT.
PWD	Mostra o diretório de trabalho.
QUIT	Finaliza a sessão FTP.
QUOTE	Envia subcomandos do servidor FTP, como se encontram no servidor.
RECV	Similar ao GET.
REMOTEHELP	Solicita ajuda do servidor FTP remoto.
RENAME	Renomeia um arquivo.
SEND	Semelhante ao PUT.
STATUS	Obtém informações de estado do servidor.
TRACERT	Demonstra o caminho percorrido pelo arquivo na transferência.
TYPE	Especifica o tipo de representação.
USER	Iniciar a sessão no servidor.
VERBOSE	Ativa/desativa a modalidade literal.

O HTTP (Hypertext Transfer Protocol ou Protocolo de Transferência de Hipertexto) possibilita a obtenção de informações, tais como documentos no formado HTML (sites). Ele é um protocolo cliente-servidor, ou seja, um cliente faz uma requisição (geralmente um navegador web) e o servidor responde com as informações requeridas.

O HTTPS (Hypertext Transfer Protocol Secure) é uma versão do protocolo HTTP criptografado. Utiliza os protocolos SSL/TLS para criptografar a comunicação realizada entre cliente e servidor, permitindo a troca de informações de forma segura.

Instalação e Configuração do servidor WEB Apache

O Apache é um servidor Web de código aberto, mantido pela Apache Software Foundation, tendo como função disponibilizar páginas Web e todos os seus recursos na internet, como um site, por exemplo, por meio da comunicação entre um servidor e um navegador. Ele é instalado, configurado e executado em um servidor, tendo como função processar as requisições solicitadas pelo navegador e devolver o conteúdo solicitado pela rede.

Abaixo demonstramos como é realizada a instalação do servidor apache versão 2.4.52 para distribuições Ubuntu Server 22.04 LTS. Algumas configurações de segurança e boas praticas também foram documentadas.

A instalação do servidor Apache no Linux ubuntu pode ser realizada por meio do gerenciador de pacotes nativo como mostrado abaixo.

```
# Comando de instalação servidor Apache.
~$ sudo apt-get install apache2 -y
```

A configuração do servidor de Apache pode variar de ambiente para ambiente, nesta prática vamos abordar uma configuração genérica com algumas boas praticas voltadas a segurança cibernética. As pastas que contém os arquivos de configuração do apache estão localizados nos diretórios:

- /var/www/
- /etc/apache2/
- /etc/apache2/sites-available/

Após a instalação do Apache, deve-se criar o diretório responsável por armazenar todo código fonte da aplicação WEB.

```
~$ sudo mkdir /var/www/cursoceape
```

Adicione uma página html simples para seguir com a configuração.

~$ sudo nano /var/www/cursoceape/index.html


```
<!DOCTYPE html>
	<html>
	<body>
		<h1>WEB Application</h1>
		<p>CEAPE</p>
	</body>
</html>
```

É sempre uma boa pratica manter os níveis de permissão de acesso aos diretórios bem configurados para não permitir outros grupos e usuários do sistema possam obter acesso indevido. Portanto, vamos restringir o acesso de leitura, escrita e execução para usuários e grupos como mostramos abaixo:

~$ sudo chmod -R 755 /var/www/cursoceape/

Para indicar ao serviço Apache a localização exata do diretório da aplicação web, é necessário configurar um arquivo de configuração em /etc/apache2/sites-available/ como demonstrado:

~$ sudo nano /etc/apache2/sites-available/cursoceape.conf


```
<VirtualHost *:80>
	ServerAdmin	webmaster@localhost
	ServerName	cursoceape
	ServerAlias	www.cursoceape
	DocumentRoot	/var/www/cursoceape
	ErrorLog	${APACHE_LOG_DIR}/error.log
	CustomLog	${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

```
#HTTP na porta 80
<VirtualHost *:80>
        ServerAdmin             webmaster@localhost
        ServerName              cursoceape
        ServerAlias             www.cursoceape
        DocumentRoot            /var/www/cursoceape
        ErrorLog                ${APACHE_LOG_DIR}/error.log
        CustomLog               ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

É necessário habilitar o arquivo de configuração criado utilizando o comando a2ensite que realiza links simbólicos em /etc/apache2/sites-enabled.

~$ sudo a2ensite cursoceape.conf

```
server@server:~$ sudo a2ensite cursoceape.conf
Enabling site cursoceape.
To activate the new configuration, you need to run:
  systemctl reload apache2
```

Da mesma forma, comando a2dissite desabilita um site removendo os links simbólicos. Portanto, desabilite o arquivo de configuração padrão do apache 000-default.conf pois não será mais usado nesta atividade.

~$ sudo a2dissite 000-default.conf

```
server@server:~$ sudo a2dissite 000-default.conf
Site 000-default disabled.
To activate the new configuration, you need to run:
  systemctl reload apache2
```

Insira a variável de configuração ServerName final do arquivo /etc/apache2/apache2.conf. Tal procedimento pode ser realizado apenas executando a linha de comando abaixo.

~$ sudo bash -c 'echo "ServerName localhost" >> /etc/apache2/apache2.conf'

```
server@server:~$ sudo bash -c 'echo "ServerName localhost" >> /etc/apache2/apache2.conf'
```

Realize o teste de syntax para verificar se não há nenhum erro de configuração nos arquivos. O comando anterior deve resultar na mensagem "Syntax OK", conforme demonstrado:

~$ sudo apache2ctl configtest

```
server@server:~$ sudo apache2ctl configtest
Syntax OK
```

Comando para iniciar o serviço do Apache junto com sistema operacional:

~$ sudo systemctl enable apache2

```
server@server:~$ sudo systemctl enable apache2
```

Firewall Liberar Portas para Acesso Externo

Considerando que servidor Ubuntu possuí um firewall instalado e ativado, é necessário liberar o acesso externo para o serviço Apache, com os seguintes comandos:

Para HTTP

~$ sudo ufw allow 80/tcp

```
server@server:~$ sudo ufw allow 80/tcp
Rules updated
Skipping adding existing rule (v6)
```

Para HTTPS

~$ sudo ufw allow 443/tcp

```
server@server:~$ sudo ufw allow 443/tcp
Rules updated
Rules updated (v6)
```

Desativar Listagem de Diretórios

Entre novamente no arquivo de configuração cursoceape.conf para adicionar a configuração referente a listagem de diretórios.

~$ sudo nano /etc/apache2/sites-available/cursoceape.conf

```
server@server:~$ sudo nano /etc/apache2/sites-available/cursoceape.conf
```

Com o arquivo aberto em modo de edição, adicione as seguintes linhas as configurações já existentes como mostra a imagem a seguir:

<Directory /var/www/cursoceape>

Options -Indexes

AllowOverride None

</Directory>

```
# HTTP na porta 80
<VirtualHost *:80>
    ServerAdmin     webmaster@localhost
    ServerName      cursoceape
    ServerAlias     www.cursoceape
    DocumentRoot    /var/www/cursoceape
    ErrorLog        ${APACHE_LOG_DIR}/error.log
    CustomLog       ${APACHE_LOG_DIR}/access.log combined
        <Directory /var/www/cursoceape>
                Options -Indexes
                AllowOverride None
        </Directory>
</VirtualHost>
```

Reiniciar o Apache.
~$ sudo systemctl restart apache2

```
server@server:~$ sudo systemctl restart apache2
```

Desativar Banner

Esta pode ser uma falha de segurança no seu servidor pois disponibiliza informações das versões do servidor WEB e sistema operacional. Tal informação pode ser útil para quem deseja conseguir um acesso indevido via ataques direcionados.

Para impedir que Apache de mostre o Banner do serviço com as versões, devemos alterar as variáveis de configurações como mostrado abaixo:

~$ sudo nano /etc/apache2/conf-available/security.conf

Alterar as variáveis para:
ServerTokens Prod
ServerSignature Off

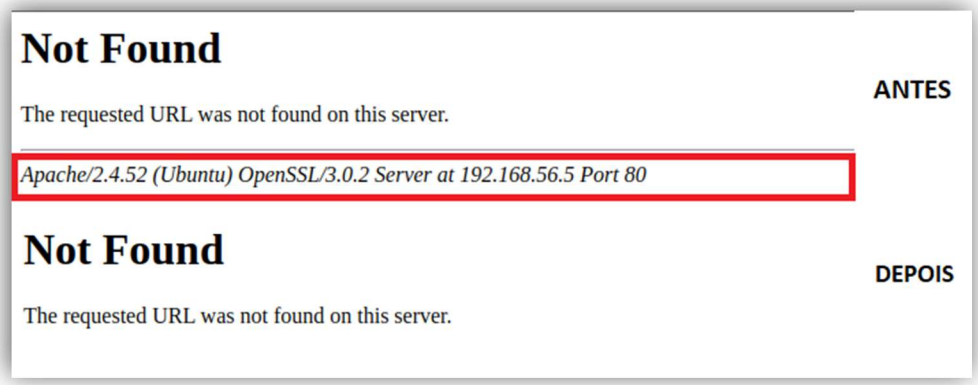
```
ServerTokens Prod
```

```
ServerSignature Off
```

Reiniciar o Apache.
~$ sudo systemctl restart apache2

```
server@server:~$ sudo systemctl restart apache2
```

Com a desativação deste recurso, atacantes terão maior dificuldade em detetar a exata versão da aplicação Apache e do sistema operacional do servidor.



Configuração do Servidor HTTPS Apache com Conexão Criptografada

Considerando as boas práticas de segurança, no que se refere a acesso a páginas web, se faz necessário a implementação de camadas de segurança na comunicação entre o browser e o servidor WEB. Nesse contexto, é fundamental a implementação do protocolo HTTPS (Hypertext Transfer Protocol Secure), que consiste em uma versão HTTP, com a adição de uma camada de segurança utilizando protocolos TLS/SSL.

SSL significa “Secure Sockets Layer”, que consiste em um protocolo de segurança que cria um link criptografado entre o servidor Web e o browser, garantindo a proteção dessa comunicação.

Inicialmente deve-se instalar o OpenSSL e habilitar o módulo de segurança SSL no Apache:

~$ sudo apt install openssl

```
server@server:~$ sudo apt install openssl
```

~$ sudo a2enmod SSL

```
server@server:~$ sudo a2enmod ssl
```

~$ sudo systemctl restart apache2

```
server@server:~$ sudo systemctl restart apache2
```

Agora será realizado a criação da chave privada e do certificado digital para que o serviço de HTTPS possa funcionar corretamente:

~$ sudo mkdir /etc/apache2/certificate

```
server@server:~$ sudo mkdir /etc/apache2/certificate
```

~$ sudo openssl req -new -newkey rsa:4096 -x509 -sha256 -days 365 -nodes -out /etc/apache2/certificate/apache-certificate.crt -keyout /etc/apache2/certificate/apache.key

```
server@server:~$ sudo openssl req -new -newkey rsa:4096 -x509 -sha256 -days 365 -nodes -out /etc/apach
e2/certificate/apache-certificate.crt -keyout /etc/apache2/certificate/apache.key
```

Nesse momento deve ser editado o arquivo de configuração relativo a as variáveis da aplicação web adicionando as informações:

~$ sudo nano /etc/apache2/sites-available/cursoceape.conf

```
server@server:~$ sudo nano /etc/apache2/sites-available/cursoceape.conf
```

```
<VirtualHost *:443>
        ServerAdmin             webmaster@localhost
        ServerName              localhost
        ServerAlias             www.cursoceape
        DocumentRoot            /var/www/cursoceape
        ErrorLog                ${APACHE_LOG_DIR}/error.log
        CustomLog               ${APACHE_LOG_DIR}/access.log combined
        SSLEngine               on
        SSLCertificateFile      /etc/apache2/certificate/apache-certificate.crt
        SSLCertificateKeyFile   /etc/apache2/certificate/apache.key
</VirtualHost>
```

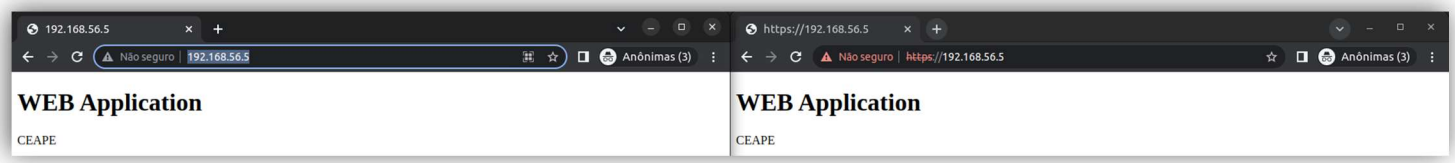
```
#HTTPS na porta 443
<VirtualHost *:443>
    ServerAdmin             webmaster@localhost
    ServerName              cursoceape
    ServerAlias             www.cursoceape
    DocumentRoot            /var/www/cursoceape
    ErrorLog                ${APACHE_LOG_DIR}/error.log
    CustomLog               ${APACHE_LOG_DIR}/access.log combined
    SSLEngine               on
    SSLCertificateFile      /etc/apache2/certificate/apache-certificate.crt
    SSLCertificateKeyFile   /etc/apache2/certificate/apache.key
</VirtualHost>
```

Concluindo esses passos, resta somente reiniciar o serviço do Apache e testar o acesso ao site utilizando o HTTPS.

~$ sudo systemctl restart apache2

```
server@server:~$ sudo systemctl restart apache2
```

A imagem abaixo apresenta o acesso via Web Bowser da aplicação em HTTP e em HTTPS.



Logs do servidor Apache

Em relação aos logs do servidor Apache, durante a criação do arquivo /etc/apache2/sites-available/cursoceape.conf nas variáveis ErrorLog e CustomLog, é habilitado e direcionado os logs para os seguintes diretório:

- /var/log/apache2/error.log
- /var/log/apache2/access.log

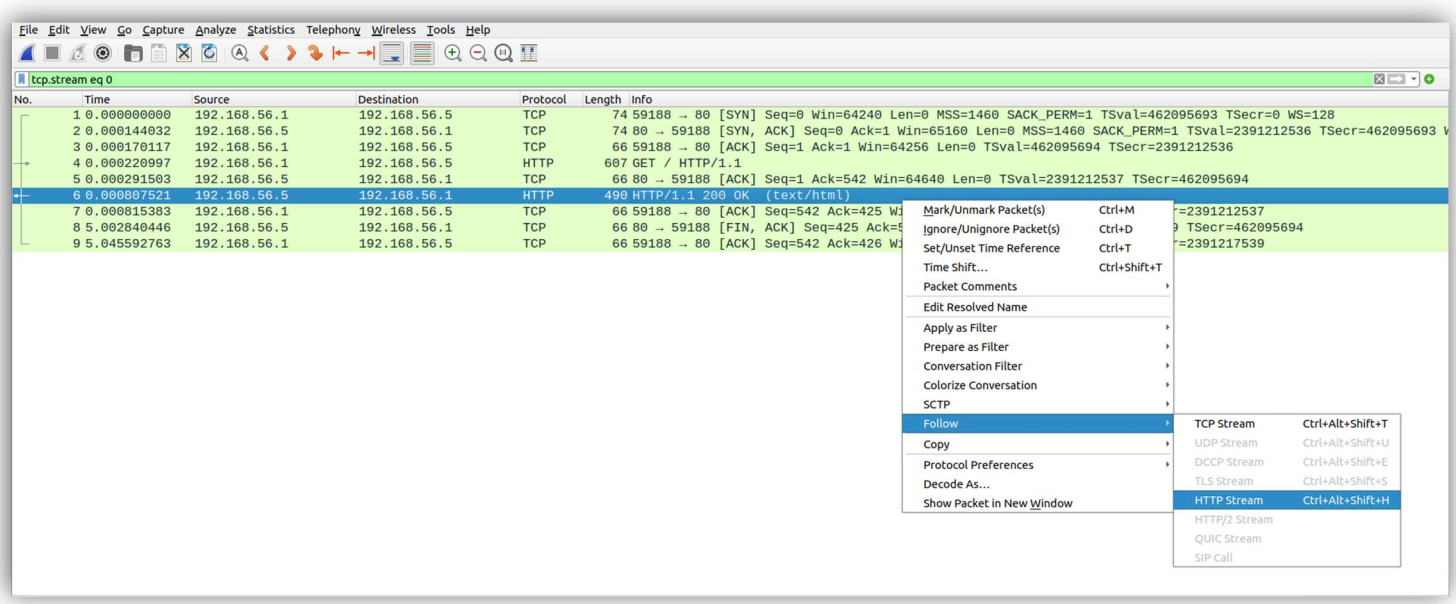
A pasta contendo os logs do apache são armazenadas no caminho /var/log/apache2/. As informações contidas nesses arquivos de logs, podem ser visualizadas conforme mostrado a seguir:

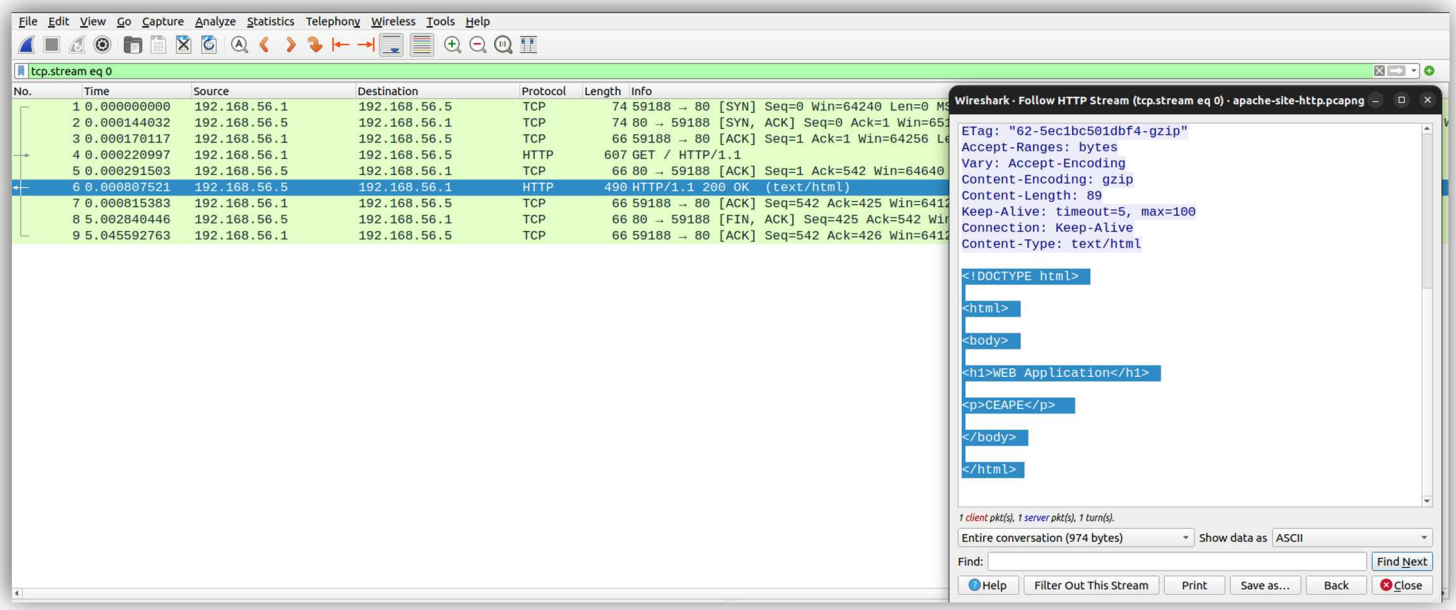
```
server@server:~$ tail -2 /var/log/apache2/error.log
[Fri Oct 28 17:50:07.756911 2022] [mpm_event:notice] [pid 26512:tid 139821431195520] AH00489: Apache/2
.4.52 (Ubuntu) OpenSSL/3.0.2 configured -- resuming normal operations
[Fri Oct 28 17:50:07.756923 2022] [core:notice] [pid 26512:tid 139821431195520] AH00094: Command line:
 '/usr/sbin/apache2'
server@server:~$ tail -2 /var/log/apache2/access.log
192.168.56.1 - - [28/Oct/2022:17:44:01 +0000] "GET / HTTP/1.1" 200 424 "-" "Mozilla/5.0 (X11; Linux x8
6_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/106.0.0.0 Safari/537.36"
192.168.56.1 - - [28/Oct/2022:17:44:08 +0000] "GET / HTTP/1.1" 200 3190 "-" "Mozilla/5.0 (X11; Linux x
86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/106.0.0.0 Safari/537.36"
```

Demonstração de Tráfego de rede HTTP e HTTPS

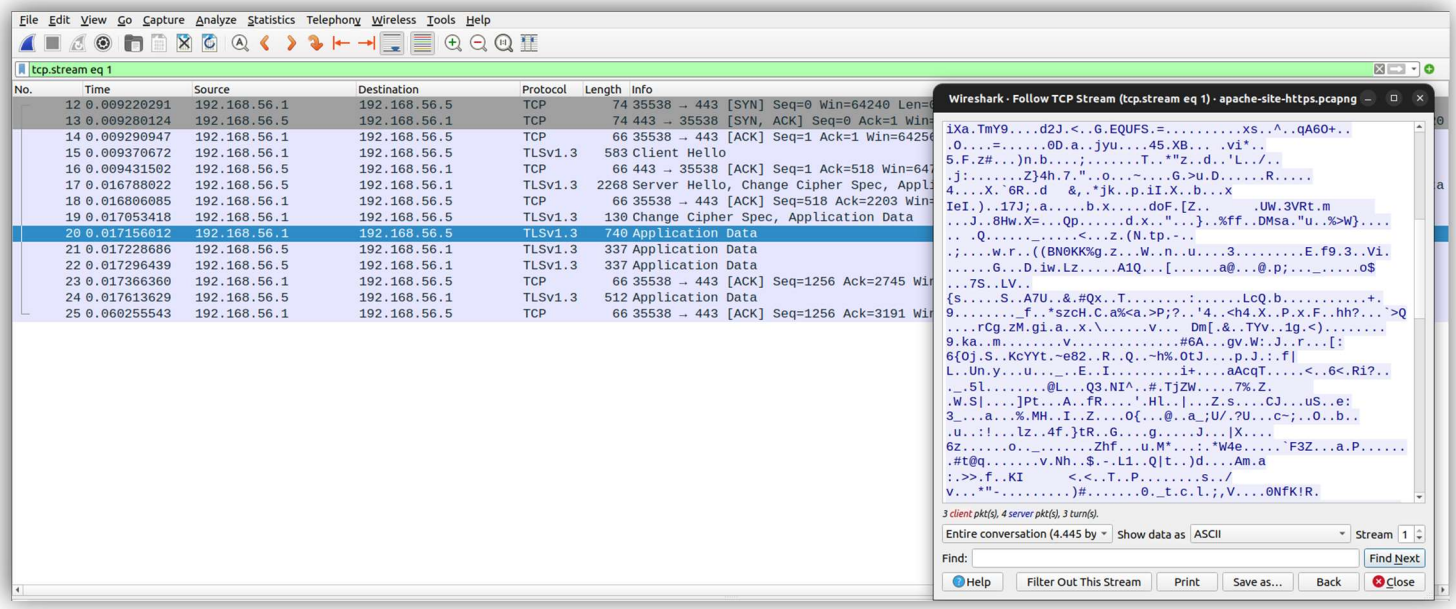
Com intuito de demonstrar de forma pratica o funcionamento dos protocolos HTTP e HTTPS, foi elaborada esta demonstração na qual apresenta a comunicação dentre duas máquinas cliente/servidor em ambos protocolos.

Uma terceira máquina que poderia ser um atacante na rede intercepta o tráfego de rede destas máquinas e pode verificar que na utilização do protocolo HTTP, é possível visualizar todo conteúdo do que se é trafegado entre as partes.





Todavia, ao capturar o tráfego criptografado (HTTPS), o terceiro na rede não pode visualizar o conteúdo das mensagens dos pacotes oferecendo assim maior segurança e privacidade entre cliente e servidor.



O tráfego de rede desta prática foi capturado e disponibilizado nos links:

Link do tráfego de rede Apache HTTP: https://raw.githubusercontent.com/rmmenezes/services-linux/main/apache-site-http.pcapng

Link do tráfego de rede Apache HTTPS: https://raw.githubusercontent.com/rmmenezes/services-linux/main/apache-site-https.pcapng