1. OBJETIVO

Este documento apresenta um breve resumo das informações discutidas em sala de aula na disciplina de Administração de Sistemas Operacionais de Redes, alunos do 3º termo do curso de Tecnólogo em Segurança da Informação da FATEC Ourinhos.

Mesmo sendo este documento consistido de material que será utilizado durante o curso, ele deverá ser utilizado apenas como material de consulta prática, ficando o aluno ciente que as fontes de informação corretas cobradas posteriormente em avaliação são as apresentadas na ementa da disciplina.

2 – Servidor WEB (Apache – http://httpd.apache.org)

2.1 - Por que o Apache?

Bom, mas por que o Apache? Pelo fato de ter licenciamento livre, amplamente utilizado pelo mundo todo (imagino que isso não seja por acaso, vejam os dados apresentados na Netcraft (https://news.netcraft.com/archives/category/web-server-survey/), possibilitar as configurações necessárias tanto no fornecimento dos serviços quanto em segurança, etc.

Maiores detalhes sobre o Apache podem ser observadas em http://httpd.apache.org.

2.2 – Instalação

```
apt-get install apache2
```

Podemos testar o funcionamento do apache abrindo a página padrão do servidor em um browser. Fazendo isso, deve aparecer a mensagem "**It works!**" no seu navegador. Considerando que o IP do nosso servidor Debian é 192.168.0.1, basta ir à VM Windows e digitar no browser este IP.

2.3 – Virtual Host

Com esta configuração padrão, os arquivos HTML lidos pelo apache para apresentação em browsers de máquinas clientes ficam no diretório **/var/www/html/**. Podemos conferir isto com o comando abaixo:

```
ls -l /var/www/
-rw-r--r--     1 root root    45        Nov  6 14:48   index.html
```

Se quisermos fazer uma troca simples do site default do apache, de forma que o conteúdo apresentado quando se digita o IP do servidor no browser da máquina seja outro, basta editarmos ou até mesmo trocarmos o arquivo **/var/www/html/index.html** para deixá-lo com as configurações necessárias.

Apesar de podermos mudar o conteúdo do site padrão do apache da forma que quisermos, na maioria dos casos, os servidores apache não armazenam apenas um site, e sim um conjunto de sites geralmente relacionados a domínios diferentes. Neste caso podemos utilizar uma *feature* do Apache conhecida como Virtual Host.

Virtual Host nada mais é do que criarmos outras bases de informação para o Apache interpretar. Localmente estas bases de informação são diretórios que podem armazenar sites inteiros.

Caso a configuração do seu servidor tenha sido feita acompanhando corretamente nosso material, você terá configurado em sua máquina o domínio **segfatecou.edu.br**. Digitando o endereço **www.segfatecou.edu.br** no browser de uma máquina também teremos acesso ao site default do Apache.

Abaixo segue um "passo-a-passo" de como configurar um Virtual Host. Segue:

1 – criar o arquivo "**segfatecou.conf**" no diretório "**/etc/apache2/sites-available/** com o seguinte comando:

```
vi  /etc/apache2/sites-available/segfatecou.conf
```

Colocar o seguinte conteúdo nesse arquivo:

```
<VirtualHost  *:80>
     ServerAdmin    webmaster@segfatecou.edu.br
     ServerName     www.segfatecou.edu.br
     DocumentRoot   /var/www/http/segfatecou/
     ErrorLog       /var/log/apache2/error_segfatecou.log
     LogLevel       warn
     CustomLog     /var/log/apache2/access_segfatecou.log combined
</VirtualHost>
```

Segue abaixo uma explicação dos parâmetros de configuração utilizados neste arquivo.

<VirtualHost *:80> e **</VirtualHost>** – Início de final da configuração do Virtual Host, informando qual porta o servidor estará recebendo conexões para este VirtualHost.
ServerAdmin – email que será informado em mensagens de erro aos usuários do site.
ServerName – nome do servidor. Caso seu servidor possua um alias, você pode colocar o nome correspondente ao seu site neste campo.
DocumentRoot – localização do diretório que será a base para os arquivos do site. Caso não exista previamente, deve ser criado manualmente.
ErrorLog – arquivo de log dos erros de o servidor possa encontrar na execução do site, como por exemplo, links quebrados de imagens etc.
LogLevel – controla os tipos de mensagens que serão enviados ao log. Os tipos são: emerg, alert, crit, error, warn, notice, info e debug.
CustomLog – informa o que foi acessado no servidor (site).

Mais detalhes sobre os parâmetros de configuração do Virtual Host devem ser observados em **http://httpd.apache.org/docs/2.2/mod/core.html** .

Agora criamos o diretório onde vai ficar o site com o comando:

```
mkdir -p  /var/www/http/segfatecou
```

Depois, criamos um arquivo chamado **index.html** lá dentro com alguma mensagem de teste ou ir o diretório repositório do site como no caminho informado em seu VirtualHost

(no nosso exemplo **/var/www/http/segfatecou/**) e colocar lá dentro todas as URLs do seu site.

Agora habilitarmos o Virtual Host para ser trabalhado pelo Apache. Para isso, precisamos digitar o seguinte comando:

```
a2ensite segfatecou
```

e desabilitar o site padrão do Apache com o seguinte comando:

```
a2dissite 000-default
```

e testamos se a configuração do apache está correta, com o comando:

```
apache2ctl configtest
```

Onde o resultado esperado é:

```
Syntax OK
```

Caso você encontre a mensagem: "*AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1. Set the 'ServerName' directive globally to suppress this message"*

Insira a seguinte linha no final do arquivo /etc/apache/apache2.conf

ServerName 127.0.0.1

Se estiver tudo certo, recarregamos as configurações do Apache para leitura das novas configurações com o comando:

```
/etc/init.d/apache2 reload
```

ou

```
systemctl reload apache2.service
```

Após estas configurações, se você digitar **www.segfatecou.edu.br** em seu browser abrirá o site existente em **/var/www/http/segfatecou**. O site padrão, existente em **/var/www/index.html** não será mais acessível pelo Apache.

Caso seu servidor DNS responder para outros domínios ou mesmo máquinas do mesmo domínio (segfatecou.edu.br), basta criar outros arquivos VirtualHost para as outras configurações seguindo o mesmo conceito do anterior.

3. Simulando erros na configuração do Apache

Suponhamos que o usuário cometeu um erro na configuração do arquivo do VirtualHost. Ao fazer o reload do apache vamos ver a seguinte mensagem:

```
/etc/init.d/apache2 reload
[FAIL] Reloading web server config: apache2 failed!
```

Neste caso a melhor coisa é usar o seguinte comando:

```
apache2ctl configtest
```

```
Syntax error on line 3 of /etc/apache2/sites-enabled/segfatecou:
Invalid command 'ServeName', perhaps misspelled or defined by a module
not included in the server configuration
Action 'configtest' failed.
The Apache error log may have more information.
```

Repare na mensagem em negrito, onde diz que há um erro de sintaxe na linha 3 do arquivo **segfatecou.** E logo depois, a mensagem diz: **Invalid command 'ServeName'.** Isso quer dizer que o usuário digitou na linha 3 a opção ServerName incorretamente (faltou o **r**). Feita a correção é só dar um reload no Apache novamente.

4 – Servidor PHP

Sem dúvida existem muitos sites e sistemas desenvolvidos em PHP, mas para que esses sites e sistemas possam funcionar corretamente é necessário que o servidor que os armazenem possua a capacidade de interpretar a codificação feita. O Apache pode ser "estendido" para fazer este serviço com a instalação do servidor PHP na máquina. Este capítulo traz esta configuração.

4.1 – Instalação

```
apt-get install php7 libapache2-mod-php7
```

```
vi /var/www/http/segfatecou/test.php
```

digitar e salvar o seguinte conteúdo no arquivo aberto:

```
<?php phpinfo(); ?>
```

salvar e fechar o arquivo.

acessar: **http://www.segfatecou.edu.br/test.php**

Aparecerá um site em PHP apresentando todas as features existentes no seu PHP. Estas informações são muito úteis para você saber exatamente o que tem habilitado e desabilitado no seu servidor PHP. O ideal é salvar estas informações para posterior consulta, por exemplo para saber se um determinado sistema a ser implantado está totalmente compatível com seu servidor e evitar sufocos desnecessários. Depois é aconselhável apagar este site, já que seu conteúdo pode ser aproveitado por usuários mal intencionados, pois, conhecendo as features habilitadas num servidor, um usuário mal intencionado pode tentar se aproveitar de alguma vulnerabilidade para prejudicar seu ambiente.

Para modificar alguma feature do seu servidor PHP basta configurar o arquivo **/etc/php5/apache2/php.ini** de acordo com suas necessidades. Configurações de "hardening" do PHP podem ser obtidas em **http://www.hardened-php.net/** .

5– SSL (https)

O servidor acima possibilita o tráfego de informações web como a maioria dos servidores de sites que acessamos pela Internet. O problema, é que com estas configurações um usuário mal intencionado, e que consiga capturar o tráfego da máquina cliente para o servidor e vice-versa, acaba tendo possibilidade de obter informações particulares, como usuários e senhas. Para resolver este problema, podemos utilizar as features do SSL para criptografar os dados trafegados em sessões HTTP, bem como autenticar servidor e cliente.

Os detalhes deste protocolo serão abordados profundamente na disciplina competente, no curso de Segurança da Informação da FATEC, mas aqui teremos uma noção de seu funcionamento para possibilitar a configuração do servidor.

5.1 – Apache com suporte a SSL no Debian

Para instalar o SSL no Debian basta digitar o seguinte comando:

```
apt-get install openssl ssl-cert
```

Depois de instalado o OpenSSL, você deverá gerar o certificado do servidor, habilitar o SSL e fazer as configurações necessárias nos VirtualHosts.

5.2 – O que é o HTTPS?

De forma bastante simples, o HTTPS (HTTP com suporte SSL) garante a autenticação das máquinas envolvidas e a criptografia dos dados da seguinte forma.

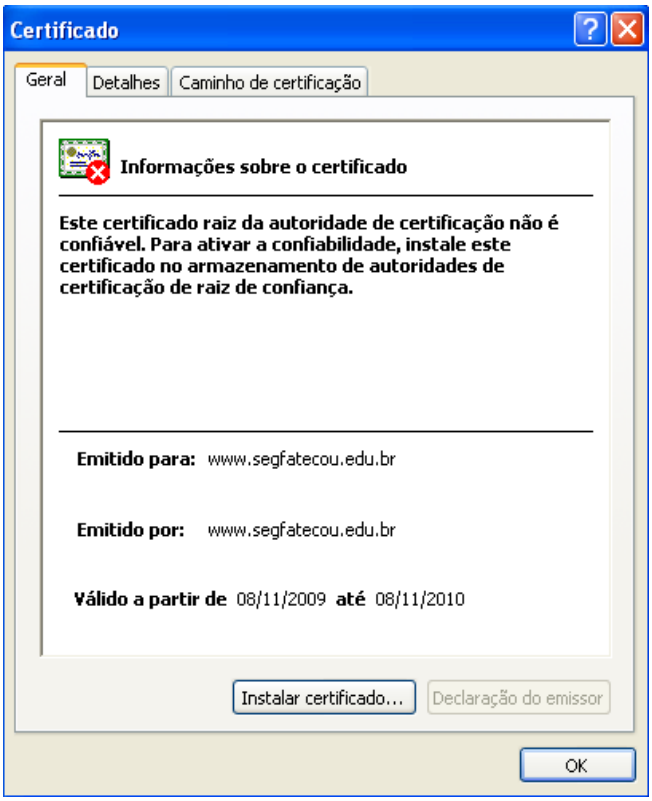
Quando uma sessão SSL é fechada entre um cliente e um servidor, o servidor inicia enviando sua chave pública ao cliente. O envio da chave pública é feita sem qualquer preocupação com a segurança do envio, já que, como o próprio nome diz, esta chave é PÚBLICA e qualquer pessoa pode tê-la em mãos.

A partir deste momento, o cliente gera uma chave aleatória de 46 bytes que é criptografada com a chave pública do servidor e enviado a ele. Somente o detentor da chave privada, par desta chave pública, terá condições de decriptar estes dados, desta forma, apenas o servidor poderá conhecer a chave de 46 bytes enviadas pelo cliente.

O próximo passo agora é gerar a estrutura de chaves RC4 para criptografia definitiva do tráfego da sessão estabelecida entre o servidor e o cliente.

5.3 – Certificado X.509

O certificado X.509 é utilizado para autenticar o servidor e até mesmo o cliente, quando este possuir um.



A máquina que deseja se autenticar, calcula o hash de uma sequência de dados trafegados na sessão e criptografa este hash com sua chave privada. Envia este hash criptografa à outra ponta que utiliza a chave pública do emissor para decriptar os dados. Esta mesma ponta da comunicação também calcula o hash do mesmo conteúdo e compara com os valores. Se conferirem então a máquina é quem diz ser.

5.4 – Configuração

5.4.1 – Gerando a Chave Privada e Certificado

Primeiro vamos criar uma pasta para guardar a chave com segurança:

```
mkdir -p /etc/apache2/ssl/segfatecou
```

Para gerar o conjunto de Chaves Privada e Pública e Certificado do servidor Debian, utilize o seguinte comando (em uma linha apenas):

```
openssl req -new -x509 -days 365 -nodes -out /etc/apache2/ssl/segfatecou/apache.csr -keyout /etc/apache2/ssl/segfatecou/apache.key
```

Onde:

openssl req: comando para gerar o certificado.
-new -x509 -days 365: novo certificado, do tipo X.509 e validade de 365 dias a contar deste momento.
-nodes -out <arquivo> -keyout <arquivo>: criar a chave privada nestes arquivos sem criptografá-los. Este comando gera dois arquivos separados, o "apache.key", que contém

a chave criptográfica e o "apache.csr", que contém o certificado que será fornecido aos clientes

Com o comando acima, será apresentado ao administrador a possibilidade de criar os dados do certificado. Abaixo segue um exemplo do output para o comando acima:

```
Generating a 1024 bit RSA private key
.........++++++
.............................++++++
writing new private key to '/etc/apache2/ssl/segfatecou/apache.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:BR
State or Province Name (full name) [Some-State]:Sao Paulo
Locality Name (eg, city) []:Ourinhos
Organization Name (eg, company) [Internet Widgits Pty Ltd]:FATEC Ourinhos
Organizational Unit Name (eg, section) []:Rede
Common Name (eg, YOUR name) []:www.segfatecou.edu.br
Email Address []:webmaster@segfatecou.edu.br
```

Podemos observar que a chave privada RSA criada é de 1024 bits e ela será armazenada no arquivo **/etc/apache2/ssl/segfatecou/apache.key**.

Os dados acima escritos em negrito e itálico são as informações que devemos preecher para que o servidor crie o certificado com os dados corretos. Destes dados, apenas o **Common Name (CN)** deve ser preenchido com total critério, já que ele será utilizado pelas browsers mais avançados na validação do certificado. Deste campo deve ser digitado o nome da URL do seu site, para não correr o risco do browser do cliente identificar seu certificado como sendo inválido por estar relacionado a outra URL.

Após criado, podemos conferir a criação dos arquivos **apache.key e apache.csr** no caminho informado:

```
ls /etc/apache2/ssl/segfatecou
apache.csr  apache.key
```

Podemos conferir o conteúdo do arquivo com o comando **cat**. Dessa forma veremos que ele se refere a da chave privada e o certificado do servidor.

debian:/etc/apache2/ssl/segfatecou# cat apache.key

-----BEGIN RSA PRIVATE KEY-----
MIICXAIBAAKBgQCv7+Feria6bmzY1R8jgDockCR98a9dXMwr41ZbxQsYIEbatLRz
pWdv7dxhdhJPG7tO8+zR/bWgRlmwlBlh8Ei5mAGifAN76Kl1oSNP+YmzZhNzZW5b
go7yEv2Z0vVn3F560akMzZfkp/C6Mx/ige7ppLKOd6/1xAibbaZ0Wm+whwlDAQAB
AoGBAI01ArVzlTJQnpwJscxUhk+WgTN0X0OXz2cKN63AlH9kqHmwv4RyJOr5BgBT
O66ym0trC5AfCT9yN1o5/bOvny3A2iqOJ971tegOLOsaBUr/dCwVonmpMMpZmvl5
//DhPzy2PNB3ywod6k07J65Vqh5FzVuxOfkgldyzqhP5nHCBAkEA5Ur+x+Hm8L4W
5dqrcV82UW5EdPAekTecUbk+fVqpHhp7T/TwKEEhErdHcfReVe0zGOFQoecbH7lo
97NVL3U3zwJBAMRt7jlvHcKX3VRaeAiwjrv0XFRKnPp6odKvLT/yJBYQdSSGGLra
ZSj2bNlJN3tuwHmd6t7oJYqACQatqa6T8ckCQETJm+fBOLQugtYQJPDNHJGPYBAo
3TK5mTbXOQ0lPsYeMbJKDCY3a8GepFtmcUqb74FeTv9TQsrgp8Hv1fV31/cCQAUb
dp43rWZp3G9dCtHvBUcNy3jFl99LYPnBZEPNX/LS6rjQY8Q3+XlfLuVXagHjl02S
NZEmKBO50lJD8a2AW0ECQFe+usW6PmHhdfRkSmXXl8fEmiEHLe11+urULvdA0exZ
04LlpiFvDFXvS7ffBVMXBKKbH+bsySNMGXKpmaH0Vls=
-----END RSA PRIVATE KEY-----

debian:/etc/apache2/ssl/segfatecou# cat apache.csr
-----BEGIN CERTIFICATE-----
MIIDpTCCAw6gAwIBAgIJAKjRhHG1kl9oMA0GCSqGSIb3DQEBBQUAMIGUMQswCQYD
VQQGEwJCUjESMBAGA1UECBMJU2FvIFBhdWxvMRMwEQYDVQQHEwpPdXJpbmhvc3Nz
MQ4wDAYDVQQKEwVGQVRFQzENMAsGA1UECxMEUmVkZTEeMBwGA1UEAxMVd3d3LnNl
Z2ZhdGVjb3UuZWR1LmJyMR0wGwYJKoZIhvcNAQkBFg5zZGFzZEBzZWdmYXRlYzAe
Fw0xMTA2MTEwMjUyMzZaFw0xMjA2MTAwMjUyMzZaMIGUMQswCQYDVQQGEwJCUjES
MBAGA1UECBMJU2FvIFBhdWxvMRMwEQYDVQQHEwpPdXJpbmhvc3NzMQ4wDAYDVQQK
EwVGQVRFQzENMAsGA1UECxMEUmVkZTEeMBwGA1UEAxMVd3d3LnNlZ2ZhdGVjb3Uu
ZWR1LmJyMR0wGwYJKoZIhvcNAQkBFg5zZGFzZEBzZWdmYXRlYzCBnzANBgkqhkiG
9w0BAQEFAAOBjQAwgYkCgYEAr+/hXq4mum5s2NUfl4A6HJAkffGvXVzMK+NWW8UL
GCBG2rS0c6Vnb+3cYXYSTxu7TvPs0f21oEZZsJQSlfBluZgBonwDe+ipdaEjT/mJ
s2YTc2VuW4KO8hL9mdL1Z9xeetGpDM2X5KfwujMf4oHu6aSyjnev9cQlm22mdFpv
slcCAwEAAaOB/DCB+TAdBgNVHQ4EFgQUKwl3kLpRBSuSNJ/Nu7XptmUGD38wgckG
A1UdIwSBwTCBvoAUKwl3kLpRBSuSNJ/Nu7XptmUGD3+hgZqkgZcwgZQxCzAJBgNV
BAYTAkJSMRIwEAYDVQQIEwlTYW8gUGF1bG8xEzARBgNVBAcTCk91cmluaG9zc3Mx
DjAMBgNVBAoTBUZBVEVDMQ0wCwYDVQQLEwRSZWRlMR4wHAYDVQQDExV3d3cuc2Vn
ZmF0ZWNvdS5lZHUuYnlxHTAbBgkqhkiG9w0BCQEWDnNkYXNkQHNlZ2ZhdGVjggkA
qNGEcbWSX2gwDAYDVR0TBAUwAwEB/zANBgkqhkiG9w0BAQUFAAOBgQBU6hjlbmnS
FR510342QJilmTz15HSlK+CX1hcj81axBlJyJMVr/UC1/HCwQwEGLBuBuPkwC8Km
8LxLvx75VnrGSVk9LFnouNqdC/jKqdjl6tatU5xF6kHuqXM7Y4+gyBK0Czymb/nA
YC4hsdVO6PD09Lba0kbFoGPzoXuLZuqU3w==
-----END CERTIFICATE-----

OBS: obviamente, a chave privada de um servidor não poderia ficar assim exposta para todos.

Justamente pelo fato da chave privada de um servidor ser a “alma” da criptografia utilizando a infraestrutura de chaves, então, deve-se verificar se o arquivo que contém a chave privada está com as permissões corretas. Segue abaixo a permissão correta para o aquivo:

```
ls -l /etc/apache2/ssl/segfatecou/apache.key
-rw------- 1 root root 2323 Nov  8 10:43 apache.key
```

Note que em negrito estão as permissões do arquivo, as quais são apenas leitura e escrita para o dono, no caso o root. Todas as outras permissões estão negadas.

Caso seja necessário alterar as permissões do arquivo **apache.key**, por diferença nas permissões, faça da seguinte forma:

```
chmod 600 /etc/apache2/ssl/segfatecou/apache.key

ls -l /etc/apache2/ssl/segfatecou/apache.key
-rw------- 1 root root 2323 Nov  8 10:43 apache.key
```

5.4.2 – Configurando o VirtualHost

Para habilitar o SSL para uso nos Virtuais Hosts, devemos reconfigurar os arquivos encontrados em **/etc/apache2/sites-available/** e adicionar as seguintes linhas:

```
<VirtualHost *:443>
     ServerAdmin     webmaster@segfatecou.edu.br
     ServerName      www.segfatecou.edu.br
     DocumentRoot    /var/www/http/segfatecou/
     ErrorLog        /var/log/apache2/error_segfatecou_ssl.log
     LogLevel        warn
     CustomLog       /var/log/apache2/access_segfatecou_ssl.log combined
     SSLEngine       on
     SSLCertificateFile      /etc/apache2/ssl/segfatecou/apache.csr
     SSLCertificateKeyFile   /etc/apache2/ssl/segfatecou/apache.key
</VirtualHost>
```

Estas configurações informam que este VirtualHost agora passará a trabalhar na porta 443, HTTPS, possui suporte ao SSL e utiliza o arquivo apache.key para como base para infraestrutura de chaves.

5.4.3 – Habilitando o suporte ao SSL no servidor

Como dito acima, para habilitar o uso de criptografia nos servidores web Apache com suporte a SSL devemos gerar o certificado do servdor (feito), habilitar o SSL e fazer as configurações necessárias nos VirtualHosts.

O próximo passo é habilitar a utilização do SSL pelo servidor, para isso, basta digitar o seguinte comando:

```
a2enmod ssl
```

Será pedido para que o administrador reinicie o Apache, porém precisamos fazer os testes se as configurações estão corretas.

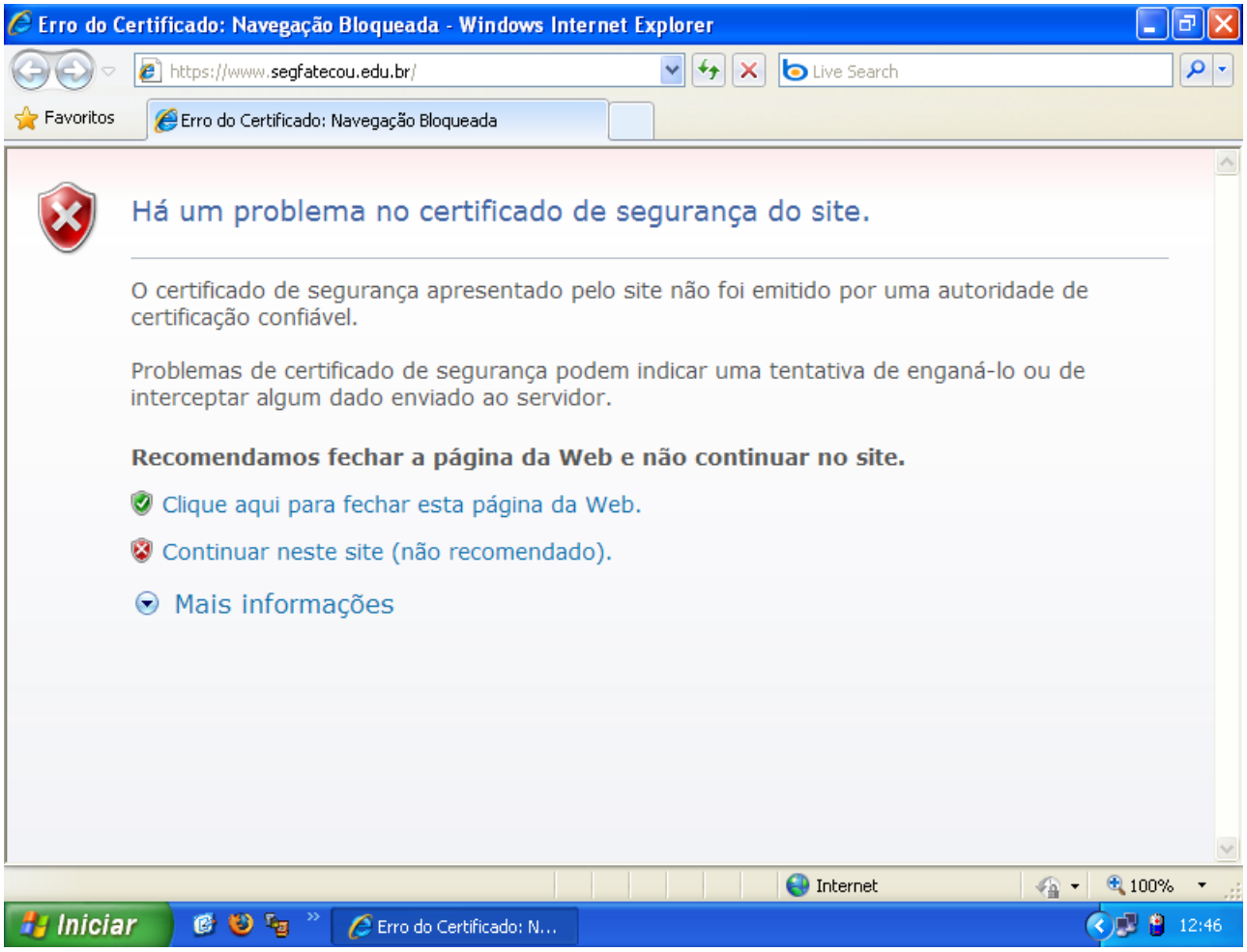
```
apache2ctl configtest
```

Se o resultado for **Syntax Ok**, podemos fazer o restart:

```
/etc/init.d/apache2 restart
```

6 – Utilização

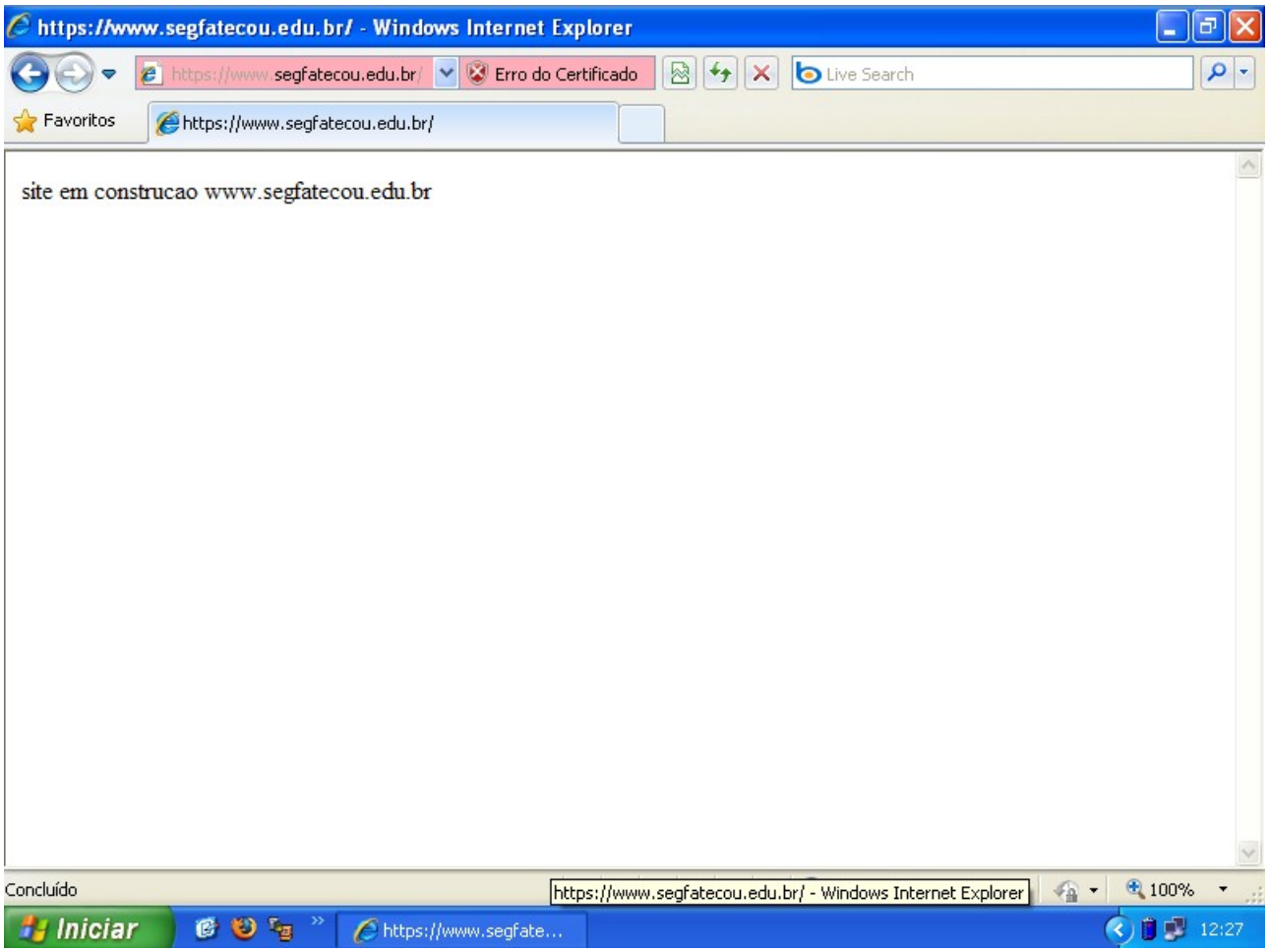
Para acessarmos o site **www.segfatecou.edu.br** haverá a necessidade de colocarmos o **https://** no início do endereço na URL do browser, como apresentado na figura a seguir:



Desta forma acessaremos o servidor e requisitaremos um site criptografado. Bem, veja que em vez de recebermos o site que esperávamos, acabamos recebendo uma informação de erro na tela. Isso se deve ao fato de nosso certificado (criado por nós) não possui uma assinatura de um CA (Certificate Authority) válida.

Certificate Authority são entidades que garantem a veracidade de certificados digitais pela Internet, de forma a possibilitar que um browser reconheça automaticamente um certificado digital. Neste caso, como não somos reconhecidos por uma CA, devemos aceitar o aviso do browser e aceitar o certificado.

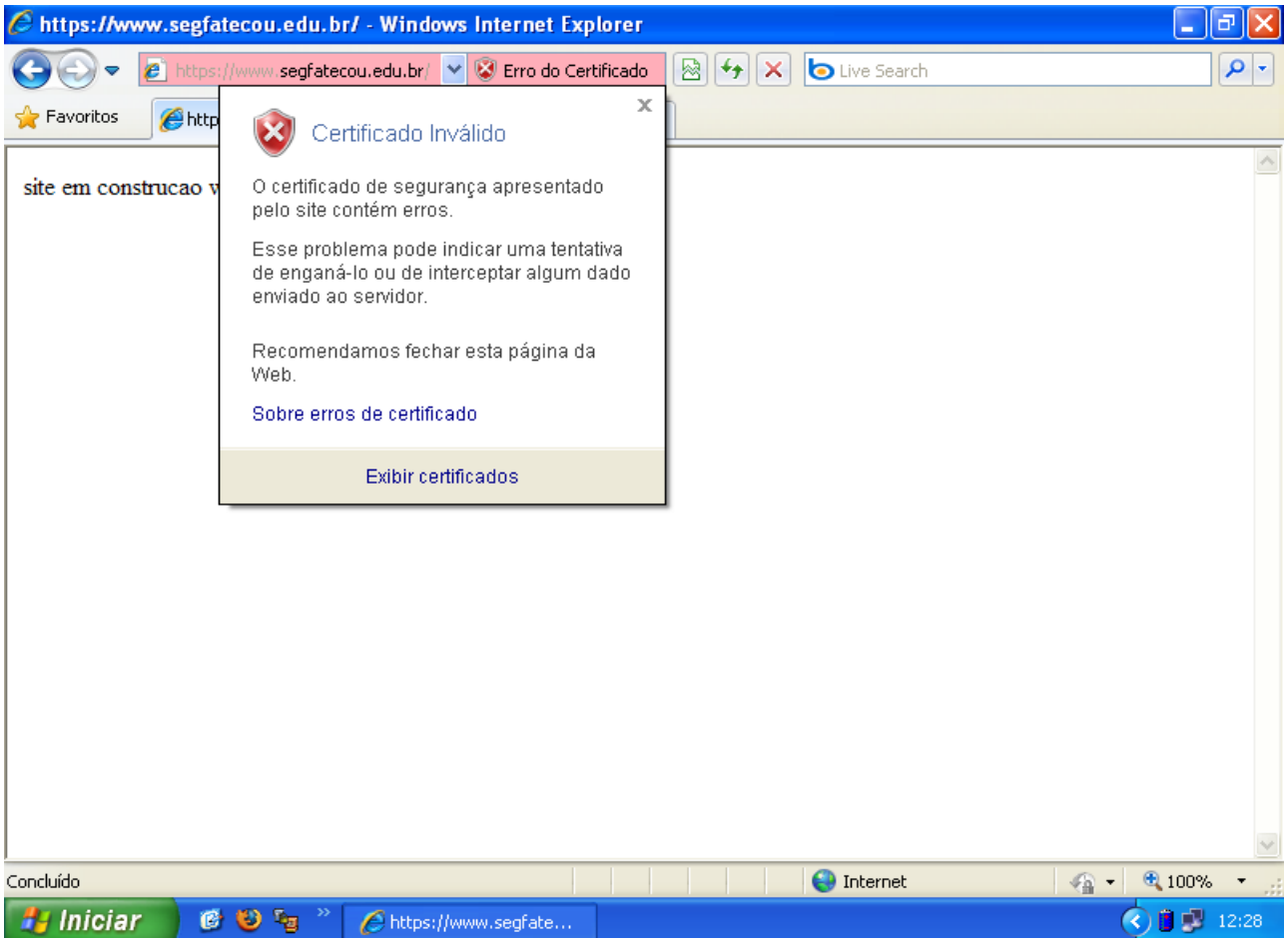
Depois de aceito o certificado, o site é aberto em nosso navegador como apresentado na figura abaixo:



Observem que antes do endereço **www.segfatecou.edu.br** existe a informação do protocolo utilizado, **HTTPS**, isso significa que nosso site agora tem comunicação criptografada e autenticada com o servidor.

Podemos conferir os dados do certificado navegando nele como apresentado nas figuras a seguir.

Notem também que tanto a URL quanto as informações do certificado no browser aparecem em vermelho, identificando que este site (certificado) foi aceito, porém não é confiável para o browser por não possuir um CA responsável por este certificado.


https://www.segfatecou.edu.br/ - Windows Internet Explorer
https://www.segfatecou.edu.br/
Erro do Certificado
Live Search
Favoritos
site em construcao w
Certificado Inválido
O certificado de segurança apresentado pelo site contém erros.
Esse problema pode indicar uma tentativa de enganá-lo ou de interceptar algum dado enviado ao servidor.
Recomendamos fechar esta página da Web.
Sobre erros de certificado
Exibir certificados
Concluído
Internet
100%
Iniciar
https://www.segfate...
12:28