

FACULDADE DE TECNOLOGIA SENAC GOIÁS
Tecnologia em Segurança da Informação



Aldo Filho
Jordan Hugs
Liniker Lettierre
Rony Carneiro

**LEVANTAMENTO E DESCRIÇÃO DE RECOMENDAÇÕES
PARA SEGURANÇA NOS SERVIÇOS IP**

Dinailton José da Silva

GOIÂNIA,
2016

Aldo Filho
Jordan Hugs
Liniker Lettierre
Rony Carneiro

LEVANTAMENTO E DESCRIÇÃO DE RECOMENDAÇÕES PARA SEGURANÇA NOS SERVIÇOS IP

Relatório apresentado como requisito parcial para obtenção de aprovação na disciplina de Serviços IP, no Curso de Tecnologia em Segurança da Informação, na Faculdade de Tecnologia Senac Goiás.

Dinailton José da Silva

GOIÂNIA,
2016

SUMÁRIO

1	INTRODUÇÃO.....	4
2	PROTOCOLO HTTP.....	4
2.1	O que é.....	4
2.2	Como instalar:.....	4
2.3	Configurando os hosts virtuais.....	6
2.4	Configurando o firewall.....	7
2.5	Resultado das configurações.....	7
3	PROTOCOLO SSH.....	8
3.1	O que é.....	8
3.2	Instalação e implementação.....	9
4	PROTOCOLO SMTP.....	13
4.1	O que é.....	13
4.2	Ferramentas necessárias para utilizar o protocolo.....	16
4.3	Implementação.....	16
5	PROTOCOLO DHCP.....	20
5.1	O que é:.....	20
5.2	Funcionamento:.....	21
5.3	Instalação e configuração Básica.....	21
6	CONCLUSÃO.....	24
7	REFERÊNCIAS BIBLIOGRÁFICAS.....	25

1 INTRODUÇÃO

A camada de aplicação é um termo utilizado em redes de computadores para designar a sétima camada do modelo OSI. É responsável por prover serviços para aplicações de modo a separar a existência de comunicação em rede entre processos de diferentes computadores. Também é a camada número cinco do modelo TCP/IP que engloba também as camadas de apresentação e sessão no modelo OSI. Ela contém os protocolos de nível mais alto como SMTP, SSH, HTTP, DHCP.

2 PROTOCOLO HTTP

2.1 O que é

HTTP é protocolo de transferência de hipertexto. É um protocolo de comunicação (na camada de aplicação, a partir do modelo OSI). É usado para transferir dados por intranet/extranets e pela World Wide Web. Normalmente, este protocolo é utiliza o porta 80 e é usado para a comunicação de sites, comunicando na linguagem HTML. Contudo, para haver comunicação com o servidor dos sites é necessário utilizar comandos adequados, que não estão em linguagem HTML.

Todos os Web Sites tem processos servidores que escutam a porta TCP 80, aguardando conexões dos clientes, após estabelecidas as conexões, o cliente envia uma solicitação e o servidor envia uma resposta, o protocolo que define as solicitações e respostas validas é chamado de Hyper Text Transfer Protocol (HTTP).

O protocolo HTTP não possui nenhum recurso de criptografia e, por consequência todo tráfego de rede gerado entre cliente e servidor poderia ser visualizado por um atacante. e deve ser substituído pelo HTTPS, que oferece conexões seguras Para aumentar a segurança de aplicações web é interessante habilitar o suporte a conexões cifradas através do SSL (Secure Socket Layer).

2.2 Como instalar:

1 Instalando os Softwares Necessários

#Todas as configurações foram feitas na distribuição CentOS 6.7

Para um servidor web com criptografia SSL você vai precisar satisfazer algumas dependências. Como root digite:

```
[root@localhost ~]# yum install mod_ssl openssl
```

2 Generate a self-signed certificate

Usando OpenSSL, vamos gerar um certificado auto assinado. Se você estiver usando isso em um servidor de produção, você provavelmente necessitará de uma chave de Autoridade de Certificação confiável, mas se você está apenas usando isso em um site

pessoal ou para fins de teste um certificado auto assinado é suficiente. Para criar a chave que você precisará ser root, assim que você pode fazer *su* para root ou usar *sudo* na frente dos comandos dependendo de sua configuração.

#Gerando a chave privada

```
[root@localhost ~]# openssl genrsa -out ca.key 2048
Generating RSA private key, 2048 bit long modulus
.....+++
..+++
e is 65537 (0x10001)
```

Gerando CSR

```
[root@localhost ~]# openssl req -new -key ca.key -out ca.csr
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) [XX]:br
State or Province Name (full name) []:Goias
Locality Name (eg, city) [Default City]:Goiania
Organization Name (eg, company) [Default Company Ltd]:TesteHTTPs
Organizational Unit Name (eg, section) []:TesteHTTPs
Common Name (eg, your name or your server's hostname) []:www.empresal.com.br
Email Address []:empresal@gmail.com.br

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:toor
An optional company name []:toor
```

Gerando a chave privada própria

```
[root@localhost ~]# openssl x509 -req -days 365 -in ca.csr -signkey ca.key
-out ca.crt
Signature ok
subject=/C=br/ST=Goias/L=Goiania/O=TesteHTTPs/OU=TesteHTTPs/CN=www.empresal.com.br/emailAddress=empresal@gmail.com.br
Getting Private key
```

Copiando os arquivos para os lugares corretos

```
[root@localhost ~]# cp ca.crt /etc/pki/tls/certs
[root@localhost ~]# cp ca.key /etc/pki/tls/private/ca.key
[root@localhost ~]# cp ca.csr /etc/pki/tls/private/ca.csr
```

#Agora você precisa atualizar o arquivo de configuração SSL do apache

```
[root@localhost ~]# vi +/SSLCertificateFile /etc/httpd/conf.d/ssl.conf
```

#Altere os caminhos de acordo com a localização dos arquivos

```
SSLCertificateFile /etc/pki/tls/certs/ca.crt
```

```
SSLCertificateKeyFile /etc/pki/tls/private/ca.key
```

Antes:

```
# Server Certificate:
# Point SSLCertificateFile at a PEM encoded certificate. If
# the certificate is encrypted, then you will be prompted for a
# pass phrase. Note that a kill -HUP will prompt again. A new
# certificate can be generated using the genkey(1) command.
SSLCertificateFile /etc/pki/tls/certs/localhost.crt

# Server Private Key:
# If the key is not combined with the certificate, use this
# directive to point at the key file. Keep in mind that if
# you've both a RSA and a DSA private key you can configure
# both in parallel (to also allow the use of DSA ciphers, etc.)
SSLCertificateKeyFile /etc/pki/tls/private/localhost.key
```

Depois:

```
# Server Certificate:
# Point SSLCertificateFile at a PEM encoded certificate. If
# the certificate is encrypted, then you will be prompted for a
# pass phrase. Note that a kill -HUP will prompt again. A new
# certificate can be generated using the genkey(1) command.
SSLCertificateFile /etc/pki/tls/certs/ca.crt

# Server Private Key:
# If the key is not combined with the certificate, use this
# directive to point at the key file. Keep in mind that if
# you've both a RSA and a DSA private key you can configure
# both in parallel (to also allow the use of DSA ciphers, etc.)
SSLCertificateKeyFile /etc/pki/tls/private/ca.key
```

#Salve o arquivo e reinicie o apache

```
[root@localhost ~]# /etc/init.d/httpd restart
Parando o httpd: [ OK ]
Iniciando o httpd: [ OK ]
```

2.3 Configurando os hosts virtuais

Assim como você configura Virtual Hosts para HTTP na porta 80, você faz para HTTPS na porta 443. A Virtual Host típico de um site na porta 80 se parece com isso:

```
<VirtualHost *:80>
    ServerAdmin webmaster@empresal.com.br
    DocumentRoot /var/www/empresal
    ServerName www.empresal.com.br
    ServerAlias empresal.com.br
    ErrorLog logs/empresal-error_log
    CustomLog logs/empresal-access_log common
</VirtualHost>
```

#Para adicionar um site na porta 443 precisamos adicionar isso no início de seu arquivo de configuração.

```
NameVirtualHost *:443
```

#E então um registro de Virtual Host será algo como isso:

```
<VirtualHost 192.168.1.6:443>

# General setup for the virtual host, inherited from global configuration
DocumentRoot "/var/www/empresa1"
ServerName www.empresa1.com.br:443

# Use separate log files for the SSL virtual host; note that LogLevel
# is not inherited from httpd.conf.
ErrorLog logs/ssl_error_log
TransferLog logs/ssl_access_log
LogLevel warn

#    SSL Engine Switch:
#    Enable/Disable SSL for this virtual host.
SSLEngine on
```

#Reinicie o apache usando

```
[root@localhost ~]# /etc/init.d/httpd restart
Parando o httpd: [ OK ]
Iniciando o httpd: [ OK ]
```

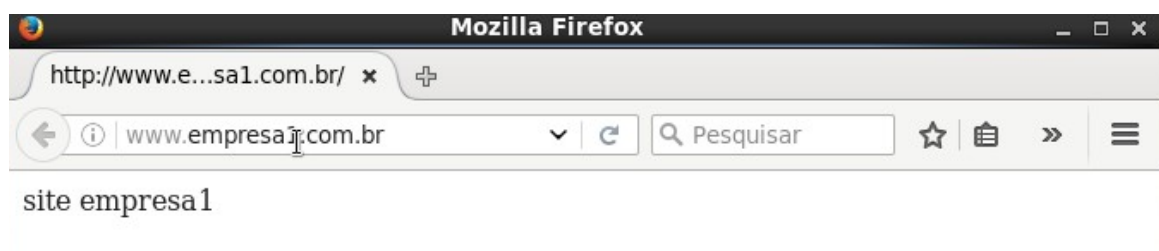
2.4 Configurando o firewall

Agora você deve ter um site que trabalha sobre HTTPS usando um certificado auto assinado. Se você não pode se conectar pode ser necessário abrir a porta do seu firewall. Para fazer isso alterar as suas regras de iptables:

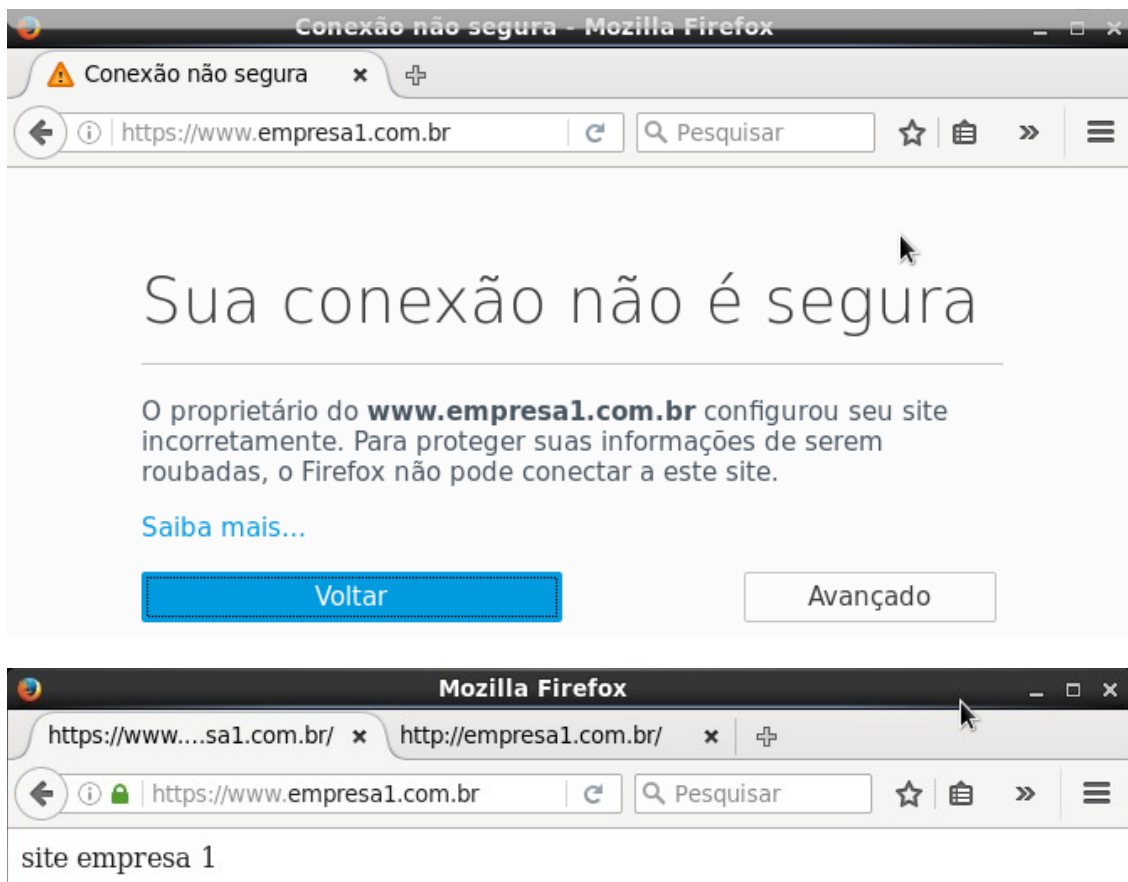
```
[root@localhost ~]# iptables -A INPUT -p tcp --dport 443 -j ACCEPT
[root@localhost ~]# /sbin/service iptables save
iptables: Salvando as regras do firewall no /etc/sysconfig/[ OK ]:
[root@localhost ~]# iptables -L -v
```

2.5 Resultado das configurações

Site antes da configuração do SSL:



Site depois da configuração do SSL:



3 PROTOCOLO SSH

3.1 O que é

SSH (Secure Shell) é uma ferramenta utilizada por muitos administradores de sistemas Unix por sua segurança e simplicidade em seu uso, podendo controlar via terminal, executar aplicativos gráficos, realizar transferências de arquivo também é capaz de encapsular de outros protocolos por exemplo o VNC através do tunelamento criptografado.

O SSH é capaz de fazer tudo o que o TELNET realiza, porém com uma importante diferença, todo o tráfego do SSH é criptografado mesmo que a rede local esteja comprometida a comunicação será feita de forma segura

O protocolo é preparado para responder a diferentes tipos de ataques já conhecidos, o SSH detecta por exemplo quando o servidor foi trocado por outra máquina com o objetivo de obter as credenciais e detecta injeções de dados na comunicação. Logo no primeiro acesso o cliente confirma o *Fingerprint* do servidor, isto permite que o cliente SSH detecta ataques de *MAN-IN-THE MIDDLE*.

A principal ideia é que mesmo que esteja conectado em uma rede pública seja praticamente impossível um atacante obter os dados que estão sendo trafegados pelo SSH, por mais que toda a comunicação seja criptografada a autenticação por meio de login e senha é suscetível ao ataque de *Brute Force* para proteger deste tipo de ataque é possível utilizar par de chaves no lugar das senhas comum

As chaves utilizada pelo SSH são assimétricas, neste método temos um par de chaves diferente da chave simétrica que é uma chave.

Uma chave é denominada de chave pública ela permite apenas encriptar os dados, a outra chave é denominada de chave privada, ela permite apenas descriptar as informações criptografada da primeira chave.

O SSH é dividido em dois módulos, o modulo SSHD é o servidor (a máquina que será acessada), e o modulo SSH é o cliente

3.2 Instalação e implementação

Para instalar o servidor SSHD é necessário instalar o seguinte pacote

```
# yum install openssh-server
```

Para instalar o cliente SSH:

```
# yum install openssh-client
```

Para que o SSH inicie junto com o sistemas:

```
# chkconfig sshd on
```

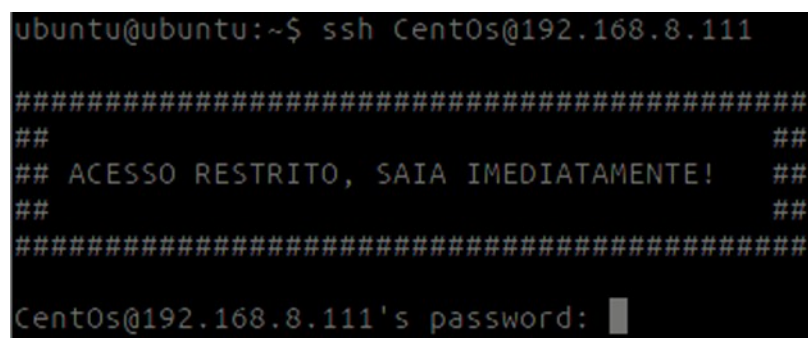
O arquivo de configuração do servidor SSH fica em /etc/ssh/sshd_config

Implementando as política de segurança:

1. Para colocar banner no SSH adicione a linha:

```
Banner = /etc/ssh/banner.txt
```

Resultado:



```
ubuntu@ubuntu:~$ ssh CentOS@192.168.8.111
#####
##                                     ##
## ACESSO RESTRITO, SAIA IMEDIATAMENTE! ##
##                                     ##
#####
CentOs@192.168.8.111's password: █
```

2. No arquivo de configuração descomente e modifique para a porta desejada:

Port 4432

Após isto para o cliente conectar no servidor será necessário informar a porta do SSH com o parâmetro -p 4432

Ex: ssh CentOS@192.168.8.111 -p 4432

3. No arquivo de configuração certificar de que o SSH está usando somente o Protocolo 2 encontre a opção “Protocol” no arquivo, ela deve estar desta maneira:

Protocol 2

4. Para desabilitar o login como root encontre a linha abaixo

PermitRootLogin yes

Mude o valor de yes para no

Resultado:

```
ubuntu@ubuntu:~$ ssh root@192.168.43.57 -p 4432
#####
##                                     ##
## ACESSO RESTRITO, SAIA IMEDIATAMENTE! ##
##                                     ##
#####

root@192.168.43.57's password:
Permission denied, please try again.
root@192.168.43.57's password:
Permission denied, please try again.
root@192.168.43.57's password:
Permission denied (publickey,gssapi-keyex,gssapi-with-mic,password).
```

5. Para permitir o acesso para usuários únicos adicione a opção AllowUser 'usuário'

AllowUser CentOS

Caso algum usuário que não está listado terá a Permissão negada

6. Para limitar as conexões não autenticadas adicione a linha:

MaxStartups 4:50:10

Para reduzir o tempo de espera do login adicione linha:

LoginGraceTime 15

O tempo é expresso em segundos

Quando o tempo de 15s é atingido a conexão é fechada

```
ubuntu@ubuntu:~$ ssh CentOS@192.168.43.57 -p 4432

#####
##                                     ##
## ACESSO RESTRITO, SAIA IMEDIATAMENTE! ##
##                                     ##
#####

CentOs@192.168.43.57's password:
Connection closed by 192.168.43.57
```

7. Uso de chaves assimétricas para autenticação no SSH:

Para gerar o par de chaves, no cliente use o seguinte comando:

```
#ssh-keygen -t rsa
```

```
ubuntu@ubuntu:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_rsa): PI
```

Primeiro será necessário informar o nome da chave.

```
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in PI.
Your public key has been saved in PI.pub.
```

Em seguida uma passphrase, é muito importante que ela seja uma frase complexa, ela é um componente da chave de encriptação,

Este comando gerará dois arquivos no diretório home “PI” e “PI.pub” são respectivamente a chave privada e a chave pública, para que funcione é obrigatório que o arquivo “.ssh/PI” esteja com permissão 600 para evitar que outros usuários possam lê-lo

```
#chmod 600 PI
```

Para instalar a chave no servidor permitindo que ela seja usada para realizar autenticação use o seguinte comando:

```
$ ssh-copy-id -i ~/.ssh/PI.pub CentOS@192.168.43.57 -p 4432
```

```

ubuntu@ubuntu:~$ ssh-copy-id -i ~/.PI.pub CentOS@192.168.43.57 -p 4432
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompt
ed now it is to install the new keys

#####
##                                     ##
## ACESSO RESTRITO, SAIA IMEDIATAMENTE! ##
##                                     ##
#####

CentOs@192.168.43.57's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh -p '4432' 'CentOs@192.168.43.57'"
and check to make sure that only the key(s) you wanted were added.

```

Para desativar o uso de senhas mude as opções “PasswordAuthetication” e “UsePAM” para no,

Ficando assim:

PasswordAuthetication no

UsePAM no

A opção “PasswordAuthetication” desativa logins por senhas, a opção UsePAM desativa qualquer outra forma de autenticação que não seja através de chave

Caso tente logar no SSH sem usar uma chave terá a permissão negada imediatamente, para utilizar a chave use o seguinte comando;

ssh CentOS@192.168.43.57 -p 4432 -i "PI"

```

ubuntu@ubuntu:~$ ssh CentOS@192.168.43.57 -p 4432 -i "PI"

#####
##                                     ##
## ACESSO RESTRITO, SAIA IMEDIATAMENTE! ##
##                                     ##
#####

Enter passphrase for key 'PI':
Last login: Tue Jun 14 18:20:24 2016 from 192.168.46.65

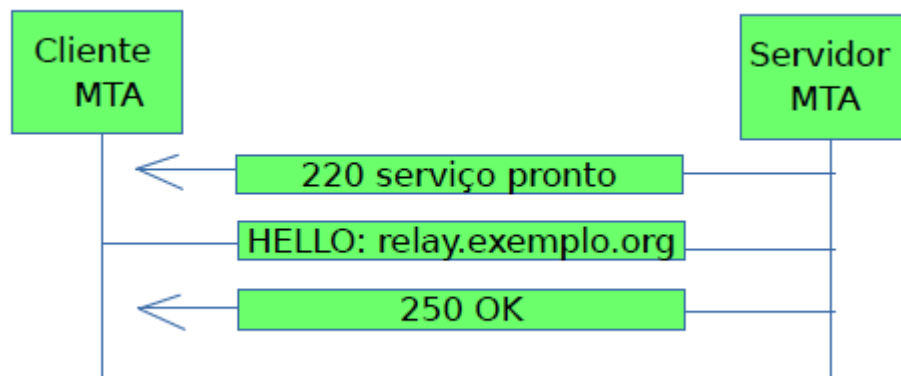
```

Será pedido a *passphrase*, atenção é a senha que foi usada no momento da geração da chave, e não a senha do usuário

4 PROTOCOLO SMTP

4.1 O que é

O SMTP (Simple Mail Transfer Protocol) é um MTA (Agente de Transferência de Mensagem) e protocolo padrão que permite a transferência de e-mail através da internet utilizando o TCP/IP. Dentro da internet, a mensagem de correio eletrônico é entregue quando a máquina de origem estabelece uma conexão TCP com a porta 25 da máquina de destino mas o SMTP utiliza também a porta 587 por medida do Comitê Gestor da Internet no Brasil (GCI.br) para evitar os spams. Após estabelecer a conexão, a máquina de transmissão, operando como cliente, aguarda a máquina receptora, operando como servidor, comunique-se primeiro.



O servidor inicia enviando uma linha de texto que contém sua identidade e informa que está pronto para receber as mensagens. Se o servidor não estiver disposto o cliente encerra sua conexão e tentará novamente mais tarde.

Caso o servidor esteja disponível para receber a mensagem o cliente anunciará de quem veio a mensagem e para quem ela está indo. O servidor realiza em si esse receptor existir no local de destino, o servidor dará ao cliente sinal para enviar mensagem. Em seguida, o cliente enviará e o servidor confirmará. Depois de todas as mensagens tiverem sido trocadas a conexão será encerrada.

O SMTP utiliza do código ASCII o que torna os protocolos mais fáceis de testar, desenvolver e depurar. Eles podem ser testados enviando comandos manualmente, e os registros das mensagens são fáceis de ler.

O SMTP usa comandos e respostas para transferir mensagens entre um cliente MTA e um servidor MTA.

S: 220 smtp.example.com SMTP Postfix

C: HELO relay.example.org

S: 250 Hello relay.example.org, I am glad to meet you

C: MAIL FROM:<bob@example.org>
S: 250 Ok
C: RCPT TO:<alice@example.com>
S: 250 Ok
C: RCPT TO:<theboss@example.com>
S: 250 Ok
C: DATA
S: 354 End data with <CR><LF>.<CR><LF>
C: From: "Bob Example" <bob@example.org>
C: To: "Alice Example" <alice@example.com>
C: Cc: theboss@example.com
C: Date: Tue, 15 January 2008 16:02:43 -0500
C: Subject: Test message
C:
C: Hello Alice.
C: This is a test message with 5 header fields and 4 lines in the message body.
C: Your friend,
C: Bob
C:.
S: 250 Ok: queued as 12345
C: QUIT
S: 221 Bye

Quadro 1 | Exemplo de transferência de mensagem de bob@example.org para alice@example.com e theboss@example.com

Os comandos são enviados do cliente ao servidor. Ele consiste em uma palavra-chave seguida de zero ou mais argumentos. O SMTP define 14 comandos. Os cinco primeiros comandos a seguir são obrigatórios.

Tabela de Comandos:

Palavras-chave	Argumento(s)
HELO	Nome do host remetente
MAIL FROM	Remetente da mensagem
RCPT TO	Destinatário da mensagem
DATA	Conteúdo da correspondência
QUIT	Encerra a sessão
RSET	Cancela a transação do correio atual
VRFY	Nome do destinatário a ser verificado
NOOP	Verifica o status do destinatário
TURN	Troca o nome do remetente com o do destinatário
EXPN	Lista de distribuição a ser verificado
HELP	Solicita ao destinatário informações sobre o comando enviado
SEND FROM	Envia a correspondência para o terminal do destinatário

As respostas são enviadas do servidor ao cliente. Uma resposta é um código de três dígitos que pode ser seguido por informações textuais adicionais. Os significados do primeiro dígito são os seguintes:

- 2yz(resposta de conclusão positiva). Se o primeiro dígito for 2 significa que o comando solicitado foi concluído com sucesso e pode-se iniciar um novo comando
- 3yz(resposta intermediária positiva). Se o primeiro dígito for 3, significa que o comando solicitado foi aceito, mas o destinatário precisa de mais algumas informações antes que a conclusão possa ocorrer.

- 4yz(resposta de conclusão negativa transitória). Se o primeiro dígito for 4, significa que o comando solicitado foi rejeitado, mas a condição de erro é temporária.
- 5yz(resposta de conclusão negativa permanente). Se o primeiro dígito for 5, significa que o comando solicitado foi rejeitado. O comando não pode ser enviado novamente

Este protocolo foi desenvolvido quando a internet era somente um pequeno grupo de computadores que era usado por poucas pessoas que se conheciam e confiavam umas às outras. O SMTP foi totalmente projetado na ideia de “cooperação” e “confiança” entre os servidores. E por não ter sido projetado para o uso global o SMTP possui graves problemas sobre a segurança, ele não implementa a autenticação facilitando o envio de spam e outra limitação é que as mensagens transmitidas não possuem criptografia ou seja suas mensagens são enviadas as claras.

Para resolvermos o problema de mensagens em texto vamos implementar então o SSL (Secure Socket Layer) que criará uma conexão criptografada com os servidores, outra solução que vamos implementar também será o envio de mensagens somente por pessoas autenticadas em nosso servidor.

4.2 Ferramentas necessárias para utilizar o protocolo

Na implementação do servidor o MTA que iremos utilizar será o Postfix. A ideia inicial do Postfix era a de produzir um substituto para o Sendmail que fosse “rápido, fácil de administrar e seguro”. Postfix é um projeto OpenSource patrocinado pela IBM, criado e mantido por Wietse Venema.

O Postfix se consolidou como uma alternativa ao Sendmail em razão de suas características técnicas tais como maior robustez, melhor desempenho e maior facilidade na manutenção e configuração.

4.3 Implementação

Primeiramente vamos instalar os pacotes postfix e o cyrus-sasl para autenticação do SMTP

```
yum install postfix cyrus-sasl
```

Após instalar os pacotes iremos configurar o postfix, o seu arquivo de configuração fica dentro da pasta /etc/postfix e os principais arquivos de configuração são o *main.cf* e o *master.cf*.

Abra o arquivo /etc/postfix/main.cf e edite-o


```
# The internet hostname of this mail system. Example: mail.example.com
myhostname = mail.seguranca3.com.br
# The internet domain name of this mail system. Example: example.com
mydomain = seguranca3.com.br
# The domain name that locally-posted mail appears to come from, and that
# locally posted mail is delivered to. Example: $mydomain
myorigin = $mydomain
# Optional external command that the local delivery agent should use for
# mailbox delivery. By default this should be empty.
mailbox_command =
# The list of "trusted" remote SMTP clients that have more privileges
# than "strangers" like relaying mail through Postfix.
mynetworks = 127.0.0.0/8 [::1]/128
# The network interface addresses that this mail system receives mail on.
# Specify "all" to receive mail on all network interfaces.
inet_interfaces = all
# The list of domains that are delivered via the $local_transport mail
# delivery transport. Default: $myhostname, localhost.$mydomain, localhost
mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain
# What destination domains and/or subdomains this system will relay mail to.
relay_domains =
# Lookup tables with all names or addresses of local recipients. The default
# (proxy:unix:passwd.byname $alias_maps) will cause postfix to access
# the /etc/passwd file.
local_recipient_maps =
```

Esta configuração faz com que o postfix tenha o básico para poder funcionar como o domínio e o hostname que o servidor de e-mail está instalado, as redes confiáveis e a rede em que o postfix executa.

Configuração de Autenticação para o SMTP

As seguintes configurações não estão no arquivo de configuração do postfix e precisam ser adicionadas ao arquivo de configuração */etc/postfix/main.cf*.

```
# SMTP-AUTH configuration
# The name of the Postfix SMTP server's local SASL authentication realm. (default: empty)
smtpd_sasl_local_domain =
# Enable SASL authentication in the Postfix SMTP server. By default, the
# Postfix SMTP server does not use authentication.
smtpd_sasl_auth_enable = yes
# The SASL plug-in type that the Postfix SMTP server should use for authentication.
smtpd_sasl_type = cyrus
# Postfix SMTP server SASL security options. noanonymous disallow methods
# that allow anonymous authentication.
smtpd_sasl_security_options = noanonymous
# Enable inter-operability with remote SMTP clients that implement an obsolete
# version of the AUTH command
broken_sasl_auth_clients = yes
# Do not report the SASL authenticated user name in the smtpd Received message header.
smtpd_sasl_authenticated_header = no
# Optional restrictions that the Postfix SMTP server applies in the context of
# a client RCPT TO command, after smtpd_relay_restrictions.
smtpd_recipient_restrictions = permit_sasl_authenticated,permit_mynetworks,reject_unauth_destination
```

Com estas configurações toda conta Linux é permitida ter login ao postfix que usa o cyrus sasl. O módulo sasl utiliza o PAM (Pluggable Authentication Module) para autenticar com o sistema operacional.

A configuração a seguir é usada para o suporte do TLS com o daemon do smtpd. Ela contém também um certificado SSL. Este certificado SSL deve ser requisitado por qualquer CA. Alguns oferecem certificados de graça como a StartSSL. Neste exemplo foi gerado um certificado local.

```
# TLS configuration
# With this, the Postfix SMTP server announces STARTTLS support to remote SMTP
# clients, but does not require that clients use TLS encryption.
smtpd_use_tls = yes
smtpd_tls_security_level = may
# Configures the server certificate file and key file as well as the CA's
# intermediate certificate file.
smtpd_tls_cert_file = /etc/postfix/smtpd.crt
smtpd_tls_key_file = /etc/postfix/smtpd.key
smtpd_tls_CAfile = /etc/postfix/cacert.pem
# Enable logging of summary message for TLS handshake and to include
# information about the protocol and cipher used as well as the client and
# issuer CommonName
smtpd_tls_loglevel = 0
smtpd_tls_received_header = yes
# Postfix SMTP server and the remote SMTP client negotiate a session, which
# takes some computer time and network bandwidth. SSL protocol versions other
# than SSLv2 support resumption of cached sessions.
smtpd_tls_session_cache_database = btree:/var/lib/postfix/smtpd_scache
# Cached Postfix SMTP server session information expires after a certain
# amount of time. RFC2246 recommends a maximum of 24 hours.
smtpd_tls_session_cache_timeout = 10800s
disable_vrfy_command = yes
```

Este último comando *disable_vrfy_command = yes* desabilita o VRFY, comando que torna possível a verificação de contas disponíveis no servidor, esse comando pode fazer com que um atacante descubra um usuário, então iremos desabilitar para evitar este tipo de risco.

Para testar a autenticação do SMTP conecte o TELNET com o postfix como o exemplo a seguir. Caso não tenha TELNET instale-o digitando `yum install TELNET`. Depois de instalado digite TELNET exemplo.com.br.

Se tudo estiver correto seu servidor de e-mail deverá atender sua requisição.

```
[root@seguranca3 postfix]# telnet seguranca3.com.br 25
Trying 192.168.1.7...
Connected to seguranca3.com.br.
Escape character is '^]'.
220 mail.seguranca3.com.br ESMTP Postfix
EHLO test
250-mail.seguranca3.com.br
250-PIPELINING
250-SIZE 10240000
250-ETRN
250-STARTTLS
250-AUTH PLAIN LOGIN
250-AUTH=PLAIN LOGIN
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
MAIL FROM: aldo@seguranca3.com.br
250 2.1.0 Ok
RCPT TO: teste2@seguranca3.com.br
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
From: aldo@seguranca3..com.br
To: teste2@seguranca3.com.br
Subject: Testando
Um, dois, três
.
250 2.0.0 Ok: queued as 6DA6F4550
QUIT
221 2.0.0 Bye
```

O que foi digitado pelo cliente na imagem a seguir foram os comandos:

EHLO test

MAIL FROM: <transmissor@dominio.com.br>

RCPT TO: <receptor@dominio.com.br>

DATA:

From: <transmissor>

To: <destinatario>

Subject: (assunto)

---- Mensagem---- (depois da mensagem, para encerrar, digite um ponto).

QUIT para encerrar a sessão

E por fim a mensagem enviada vai estar na caixa de e-mail do usuário, visualize com um *cat* em */var/spool/mail/usuario*.

```
From aldo@seguranca3.com.br Mon Jun 13 00:45:03 2016
Return-Path: <aldo@seguranca3.com.br>
X-Original-To: teste2@seguranca3.com.br
Delivered-To: teste2@seguranca3.com.br
Received: from test (seguranca3.com.br [192.168.1.7])
    by mail.seguranca3.com.br (Postfix) with ESMTP id 6DA6F4550
    for <teste2@seguranca3.com.br>; Mon, 13 Jun 2016 00:44:01 -0300 (BRT)
From: aldo@seguranca3.com.br
To: teste2@seguranca3.com.br
Subject: Testando
Message-Id: <20160613034409.6DA6F4550@mail.seguranca3.com.br>
Date: Mon, 13 Jun 2016 00:44:01 -0300 (BRT)

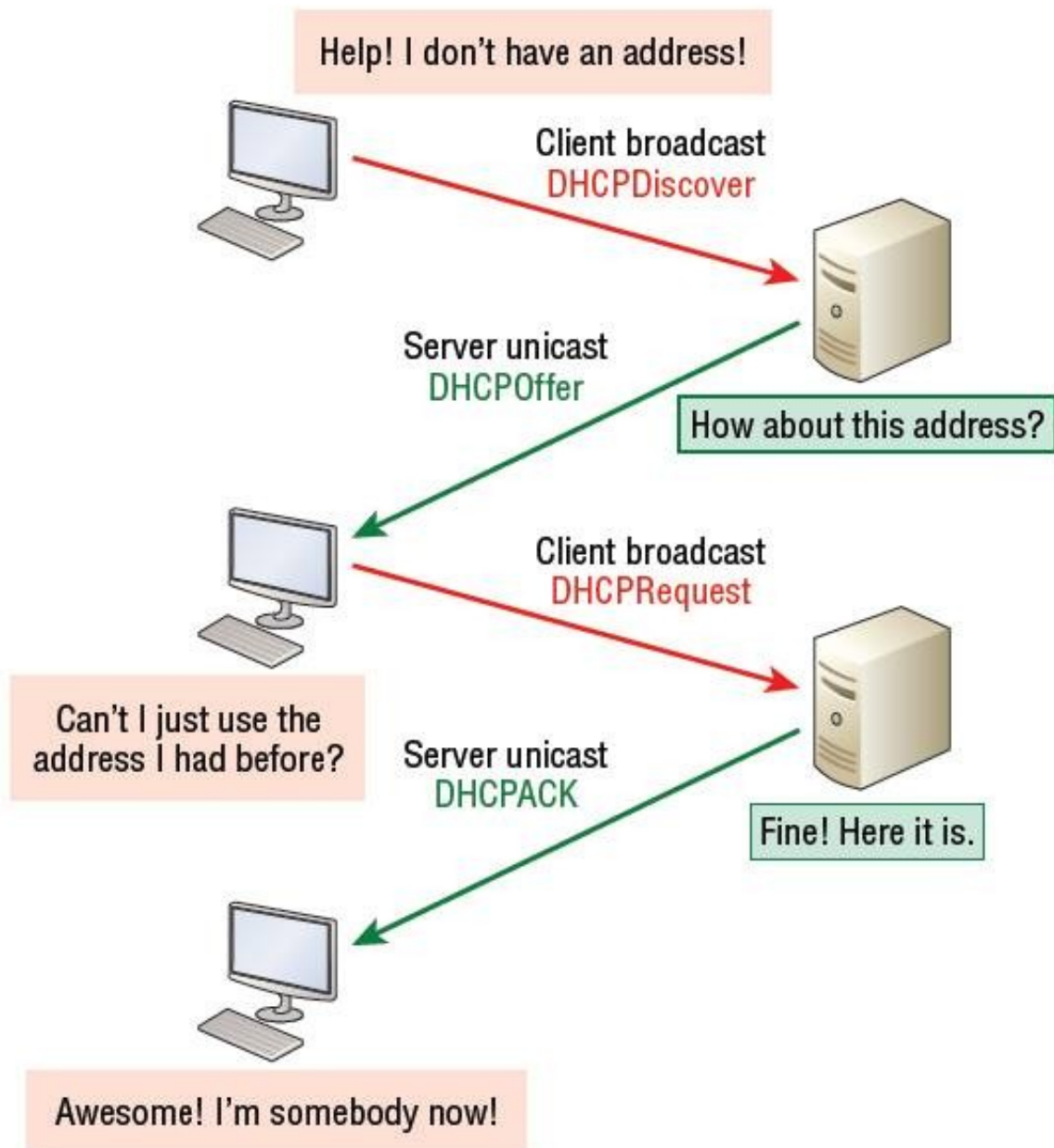
Um, dois, três
```

5 PROTOCOLO DHCP

5.1 O que é:

O DHCP (Dynamic Host Configuration Protocol) é um protocolo utilizado por dispositivos conectados a uma rede para obter as configurações de rede, tais configurações incluem endereço IP, máscara de sub-rede, gateway padrão, e os endereços IP dos servidores de DNS, tudo a partir de um servidor executando o software do servidor DHCP. Os endereços IP cedidos aos clientes são na realidade atribuídos a partir de um intervalo de endereços configurados no servidor.

5.2 Funcionamento:



Para a prevenção de ameaças em relação ao DHCP, se faz necessário conhecer seu mundo de instalação e configuração.

5.3 Instalação e configuração Básica

- Para instalar o serviço apenas digite o comando abaixo (Sem o parâmetro -y irá pedir confirmação de Download):

```
[root@localhost CentOs]# yum install dhcp -y
```

Use um editor de texto e entre em `/etc/dhcp/dhcpd.conf`. No arquivo `dhcpd.conf` coloque a configuração abaixo, adaptando o seu endereço IP em uso:

```
[root@localhost network-scripts]# ifconfig
eth2      Link encap:Ethernet  Endereço de HW 08:00:27:8C:DD:01
          inet end.: 192.168.0.5  Bcast:192.168.0.255  Masc:255.255.255
          endereço inet6: fe80::a00:27ff:fe8c:dd01/64  Escopo:Link
          UP BROADCASTRUNNING MULTICAST  MTU:1500  Métrica:1
          RX packets:64217 errors:0 dropped:0 overruns:0 frame:0
          TX packets:42456 errors:0 dropped:0 overruns:0 carrier:0
          colisões:0 txqueuelen:1000
          RX bytes:49570238 (47.2 MiB)  TX bytes:3981024 (3.7 MiB)

lo        Link encap:Loopback Local
          inet end.: 127.0.0.1  Masc:255.0.0.0
          endereço inet6: ::1/128  Escopo:Máquina
          UP LOOPBACKRUNNING  MTU:65536  Métrica:1
          RX packets:2201 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2201 errors:0 dropped:0 overruns:0 carrier:0
          colisões:0 txqueuelen:0
          RX bytes:193287 (188.7 KiB)  TX bytes:193287 (188.7 KiB)
```

```
[root@localhost network-scripts]# vim /etc/dhcp/dhcpd.conf
```

Segue abaixo o conteúdo de um arquivo de exemplo:

```
# DHCP Server Configuration file.
#   see /usr/share/doc/dhcp*/dhcpd.conf.sample
#   see 'man 5 dhcpd.conf'
#
authoritative;
default-lease-time 600;
max-lease-time 7200;
#
subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.1 192.168.1.50;
    option routers 192.168.1.1;
    option broadcast-address 192.168.1.255;
}

host maqui {
    hardware ethernet 08:00:27:8C:DD:01;
    fixed-address 192.168.0.15;
}
```

Salve o arquivo e feche (:wq)

Explicação:

-default-lease-time 600; controla o tempo de renovação dos endereços IP.

-Max-lease-time 7200; determina o tempo que cada máquina pode usar um determinado IP.

-Authoritative; Se um cliente fizer a requisição de um endereço que o servidor não conheça, ou

Seja, o endereço é incorreto para aquele segmento, o servidor não enviará um DHCPNAK, que diz

Para ao cliente parar de usar aquele endereço.

-Subnet 192.168.1.0 netmask 255.255.255.0; Define sua "sub-rede" 192.168.1.0 com a máscara

255.255.255.0 e abre uma faixa de IPs que o cliente poderá usar.

-Range 192.168.1.101 192.168.1.200; Faixa de IPs que o serviço irá disponibilizar.

-Option routers 192.168.1.1; Este é o gateway padrão (neste caso).

-Option broadcast-address 192.168.1.255; Essa linha é o endereço de broadcast (neste caso).

O servidor DHCP deve ser iniciado em apenas uma das interfaces, então temos que configurar o servidor para iniciar somente naquela interface. No /etc/sysconfig/dhcpd, adicione o nome da interface à lista de DHCPDARGS

```
[root@localhost sysconfig]# vim dhcpd
```

```
# Command line options here
DHCPDARGS=eth2
```

Agora vamos reiniciar o serviço com o comando *service dhcpd restart*

```
# service dhcpd restart
```

Para se ter mais segurança com acessos não autenticados, como o DHCP não é um protocolo autenticado, você deve fazer uma tabela ARP fixa, criando o arquivo /etc/ethers como no exemplo:

```
[root@localhost network-scripts]# vim /etc/ethers
```

```
# see man ethers for syntax
192.168.0.15 08:00:27:8C:DD:01
```

E executar o arp (pelo rc.local ou outro script de inicialização) dessa forma:

```
arp -f
```

IFCONFIG no terminal na máquina cliente, com o *IP* configurado:

```
[root@localhost CentOs]# ifconfig
eth2      Link encap:Ethernet  Endereço de HW 08:00:27:8C:DD:01
          inet end.: 192.168.0.15  Bcast:192.168.0.255  Masc:255.255.255.0
          endereço inet6: fe80::a00:27ff:fe8c:dd01/64 Escopo:Link
          UP BROADCASTRUNNING MULTICAST  MTU:1500  Métrica:1
          RX packets:6664 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4505 errors:0 dropped:0 overruns:0 carrier:0
          colisões:0 txqueuelen:1000
          RX bytes:5605078 (5.3 MiB)  TX bytes:692473 (676.2 KiB)

lo        Link encap:Loopback Local
          inet end.: 127.0.0.1  Masc:255.0.0.0
          endereço inet6: ::1/128 Escopo:Máquina
          UP LOOPBACKRUNNING  MTU:65536  Métrica:1
          RX packets:506 errors:0 dropped:0 overruns:0 frame:0
          TX packets:506 errors:0 dropped:0 overruns:0 carrier:0
          colisões:0 txqueuelen:0
          RX bytes:68146 (66.5 KiB)  TX bytes:68146 (66.5 KiB)
```

6 CONCLUSÃO

Tendo em vista os aspectos analisados, conclui-se que o protocolo de comunicação HTTP (Hyper Text Transfer Protocol) não é seguro pois não tem criptografia com base no ataque Man In The Middle, sendo assim todo tráfego na rede pode ser observado por atacantes, comprometendo a confidencialidade do serviço, e tem como solução a configuração do SSL habilitado da porta 80 HTTP para a porta 443 HTTPS.

O SSH (Secure Shell) é uma ferramenta com tráfego criptografado, que permite controles via terminal, execução de aplicativos gráficos, realização de transferências de arquivos e também é capaz de encapsular outros protocolos, por exemplo o VNC por tunelamento criptografado. O protocolo responde diferentes tipos de ataques já conhecidos, e detecta quando o servidor foi trocado por outra máquina com o objetivo de obter as credenciais e detectar injeções de dados na comunicação e também permite a detecção de ataques de MAN-IN-THE MIDDLE. É Suscetível a Brute-Force, porém, esse ataque é vetado devido ao uso de chaves assimétricas, uma chave é denominada de chave pública quando ela permite apenas encriptar os dados, a outra chave é denominada chave privada, que permite apenas descriptar as informações criptografadas da primeira chave.

O SMTP (Simple Mail Transfer Protocol) é um MTA (Agente de Transferência de Mensagem) é um protocolo padrão que permite que faz a transferência de e-mail através da internet utilizando o TCP/IP. Utiliza do código ASCII o que torna os protocolos mais fáceis de testar, desenvolver e depurar. Na internet, a mensagem de correio eletrônico é entregue

quando a máquina de origem estabelece uma conexão TCP com a porta 25 da máquina de destino, mas o SMTP utiliza também a porta 587.

Caso o servidor esteja disponível para receber a mensagem o cliente anunciará de quem veio a mensagem e para quem ela está indo. O servidor realiza e, se esse receptor existir no local de destino, o servidor sinaliza ao cliente para enviar mensagem. Em seguida, o cliente enviará e o servidor confirmará. Depois de todas as mensagens tiverem sido trocadas a conexão será encerrada.

O SMTP foi projetado na ideia de “cooperação” e “confiança” entre os servidores. E por não ter sido projetado para o uso global o SMTP possui graves problemas sobre a segurança, como não implementação da autenticação, facilitando o envio de spam e também que as mensagens transmitidas não possuem criptografia. Para resolução do problema de mensagens em texto se implementa o SSL (Secure Socket Layer) que criará uma conexão criptografada com os servidores, e outra solução é a implementação do envio de mensagens somente por pessoas autenticadas no servidor.

O DHCP (Dynamic Host Configuration Protocol) é um protocolo utilizado por dispositivos conectados a uma rede para obter as configurações de rede, configurações que incluem endereço IP, máscara de sub-rede, gateway padrão, e os endereços IP dos servidores de DNS, tudo a partir de um servidor executando o software do servidor DHCP.

Ao ligarem-se à rede, os clientes fazem um pedido para a rede, fazendo solicitação da informação de configuração. O pedido é recebido pelo servidor de DHCP, que responderá fornecendo a informação solicitada. Sendo um protocolo normalizado, clientes com diferentes sistemas operativos poderão interagir com este serviço, independentemente da plataforma na qual ele esteja implementado.

Como o DHCP não é um protocolo autenticado, faz-se a implementação de uma tabela ARP fixa, com IPS ligados a endereços MAC.

7 REFERÊNCIAS BIBLIOGRÁFICAS

DHCP Starvation

<https://brenn0.wordpress.com/2015/01/24/dhcp-starvation-ataque-de-negacao-de-servico-em-rede-local/>

Acesso em 15 de junho de 2016

Gabritech: Tipos de Ataques por camada

<http://gabritech.blogspot.com.br/2009/10/tipos-de-ataques-por-camada-camada-de.html>

Acesso em 14 de junho de 2016

Abusing, & Exploiting DHCP

<http://pt.slideshare.net/JamesHemmings/abusing-dhcp>

Acesso em 15 de junho de 2016

Blog LabCisco: DHCP Snooping na Mitigação de Servidores Falsos

<http://labcisco.blogspot.com.br/2013/01/dhcp-snooping-na-mitigacao-de.html>

Acesso em 15 de junho de 2016

Informações de segurança do DHCP

[https://technet.microsoft.com/pt-br/library/cc780347\(v=ws.10\).aspx](https://technet.microsoft.com/pt-br/library/cc780347(v=ws.10).aspx)

Acesso em 14 de junho de 2016

Configuração do cliente SSH - Dominando o SSH

<http://www.hardware.com.br/tutoriais/dominando-ssh/pagina3.html>

Acesso em 11 de junho de 2016

http e https – segurança com https

<https://juancarloskunha.wordpress.com/2009/06/02/http-e-https-seguranca-com-https-diferenca-entre-http-e-https/>

Acesso em 15 de junho de 2016

Cartilha de Segurança -- Uso seguro da Internet

<http://cartilha.cert.br/uso-seguro/>

Acesso em 15 de junho de 2016

Setup Postfix with SMTP-AUTH and TLS on CentOS

<https://blog.tinned-software.net/setup-postfix-with-smtp-auth-and-tls-on-centos/>

Acesso em 14 de junho de 2016

O que é protocolo HTTPS (Andamento) - Perguntas e Respostas HiperContas

<https://www.hipercontas.com.br/perguntas.php?&id=300>

Acesso em 14 de junho de 2016

HowTos/Https - CentOS Wiki

<https://wiki.centos.org/HowTos/Https>

Acesso em 15 de junho de 2016

Morimoto, Carlos Eduardo, Servidores Linux, Guia Prático/Carlos Eduardo Morimoto. – Porto Alegre: Sul Editores, 2015

Protocolo TCP/IP / Behrouz A. Forouzan, Sophia Chung Fegan; revisão técnica Flávio Soares Corrêa da Silva, Roerto Hirata Jr; tradução João Eduardo Nóbrega Tortello. – São Paulo: McGraw-Hill, 2008.

Redes de Computadores / Andrew S. Tanenbaum e David Wetherall ; tradução Daniel Vieira ; revisão técnica Isaías Lima. – São Paulo: Pearson Prentice Hall, 2011