PEDRO HENRIQUE RODRIGUES

PRÁTICAS DE SEGURANÇA PARA SERVIDORES WEB NAS NUVENS

TEÓFILO OTONI
FACULDADES UNIFICADAS DE TEÓFILO OTONI
2015

PEDRO HENRIQUE RODRIGUES

PRÁTICAS DE SEGURANÇA PARA SERVIDORES WEB NAS NUVENS

Monografia apresentada ao Curso de Sistemas de Informação das Faculdades Unificadas de Teófilo Otoni, como requisito parcial à obtenção do titulo de Bacharel em Sistemas de informação.
Área de Concentração: Segurança da Informação
Orientador: Prof. Amaury Gonçalves Costa

TEÓFILO OTONI
FACULDADES UNIFICADAS DE TEÓFILO OTONI
2015

AGRADECIMENTOS

Agradeço a primeiramente a Deus por me dar forças, em segundo minha mãe que faz de tudo por mim. Deixo um agradecimento especial ao professor Luciano da disciplina TCC2 por ter paciência comigo, ao meu orientador Amaury e meu amigo Matheus por terem me ajudado com as minhas duvidas. Não posso esquecer do professor Fabiano que me ofereceu uma boa experiência acadêmica e ao coordenador do curso Salim, também pela experiência acadêmica, pela paciência e por vários conselhos importantes.

RESUMO

Monografia de graduação do curso de Sistemas de Informação, ano 2015. A monografia tem como tema: Práticas de Segurança para Servidores Web nas Nuvens, com a área de concentração Segurança da Informação. Está monografia tem o objetivo de apresentar configurações para segurança de servidores web usando como metodologia análise de práticas já existentes através de testes para comprovação. Apresenta configurações de segurança para aplicações e servidores web, o funcionamento entre eles e principais ameaças que a falta dessas configurações de segurança possam ocasionar.

Palavras- chave: Segurança; Servidores; Nuvens; Aplicação; Configurações de Segurança; Banco de Dados; Redes de Computadores

ABREVIATURAS E SIGLAS

CN - Computação em Nuvem

TI - Tecnologia da Informação

SaaS - Software as a Service

PaaS - Plataforma as a Serveice

IaaS - Ifrastructure as a Service

HTML - HyperText Markup Language

CGI - Common Gateway Interface

PHP – Personal Home Page

DNS - Domain Name System

WWW - Word Wide Web

CSS - Cascading Style Sheets

URI - *Uniform Resource Identifier*

URL - Uniform Resource Locator

URN - Uniform Resource Name

HTTP - Hypertext Transfer Protocol

SGBD - Sistema de Gerenciamento de Banco de Dados

BD – Banco de Dados

DBA - Database administrator

SSH – Secure Shell

DOS - *Disk Operating System*

IP - Internet Protocol

SSL - Secure Socket Layer

HTTPS - *Hyper Text Transfer Protocol Secure*

LISTA DE ILUSTRAÇÕES

SUMÁRIO

INTRODUÇÃO

O presente Trabalho de conclusão de curso com o tema: Práticas de Segurança para Servidores Web nas Nuvens. Configura-se como requisito fundamental para a aprovação na disciplina Projeto TCC II e requisito para a obtenção do título em Bacharel em Sistemas de Informação das Faculdades Unificadas de Teófilo Otoni.

A pesquisa desenvolvida tem como objetivo o desenvolvimento do trabalho e alcançar ganho acadêmico respondendo ao problema levantado no projeto de pesquisa anteriormente elaborado “É possível configurar sistemas de segurança e utilizar boas práticas para proteger servidores web nas nuvens?”.

O método para o trabalho foi utilizar pesquisas buscando explorar vulnerabilidades dos servidores e realizar testes para comprovação da veracidade dos mesmos.

Cabe frisar que a pesquisa em questão apresenta uma alta relevância acadêmica visto que, se trata de apresentar soluções para proteger algo que tem um imenso valor comercial, social, governamental e intelectual, mediante que todos esses quesitos necessitam da tecnologia.

Os primeiros capítulos (um ao cinco) apresentam o “Referencial Teórico” que busca fazer um levantamento bibliográfico e discutir os principais itens que introduzem o objeto de estudo, para que o leitor entenda melhor a temática abordada. Foram apresentadas temáticas referentes às redes de computadores, serviços nas nuvens, servidores nas nuvens, arquitetura web e banco de dados. Todos esses capítulos apresentam o conceito e abordam brevemente o que precisa ser entendido para o bom entendimento prático do trabalho.

O capítulo seis aborda segurança de servidores web nas nuvens, que apresenta a importância do trabalho mostrando configurações de sistemas de

segurança para servidores web, como o SSH, Fail2Ban, Firewall, obtenção de titulo SSL+HTTPS. O capitulo aborda também problemas comuns e sistemas maliciosos como SQL Injection e ataques DoS, e é finalizado levantando boas práticas de segurança, que são rotinas simples que podem evitar problemas indesejáveis.

1 REDE DE COMPUTADORES

1.1 HISTÓRICO DA REDE DE COMPUTADORES

Antes de surgir as Redes de Computadores as maquinas operavam isoladamente, tendo como único meio de troca de informação o uso de disquetes, que tinham uma capacidade muito limitada para volumes de dados. As redes de computadores surgiram primeiramente a partir da necessidade de interligar os computadores entre si para trocas de informação em alta velocidade STILBEN (2005).

A rede de computadores surgiu graças à necessidade de adquirir maior volume e velocidade na troca de informações entre computadores. Com o tempo surgiram dispositivos como impressoras e atualmente pode-se utilizar interação com vários outros como aparelhos celulares.

1.2 DEFINIÇÃO E OBJETIVO DA REDE DE COMPUTADORES

O conceito de redes de computadores é basicamente um ou mais conjuntos de dispositivos (computadores, impressoras, celulares) conectados por pontos de conexão que permitem trafego de dados (SOUSA[1]).

Dentre esses dispositivos situam-se computadores, impressoras, base de dados, terminais, serviços de interconexão e periféricos computacionais (SOUSA[2]).

[1] Disponível em: <http://www.lanwan.com.br/Aulas_Senac/Tec_Redes_Final_Semana/Aula%20270310%20e%2010042010%20-%20Historia%20das%20redes.pdf>

Tanebaum (1994)[3] define os objetivos das interconexões de computadores autônomos como sendo:

- Compartilhamento de recursos: Fazer com que todos os programas, dados e equipamentos da rede estejam disponíveis a todos os usuários independentemente de sua localização física. Como exemplo, pode citar-se o compartilhamento de uma impressora por vários usuários;

- Economia: A substituição gradativa dos antigos mainframes para as redes de computadores de pequeno porte significou uma redução muito grande nos custos de manutenção dos sistemas de informação possibilitando uma verdadeira revolução nos CPDS. Esse fenômeno ficou conhecido mundialmente como Downsizing. Essas redes de computadores de pequeno porte possibilitam um aumento da capacidade de processamento a medida que a demanda cresce , ao contrário dos grandes mainframes, onde a sobrecarga só poderia ser solucionada com a substituição do mesmo por um mainframe de maior capacidade, a um custo geralmente muito elevado.

- Prover um meio de comunicação: As redes de computadores também são um poderoso meio de comunicação entre pessoas, possibilitando inclusive o trabalho em conjunto, mesmo estando a quilômetros de distância.

Tanebaum explana através dos objetivos apresentados a funcionalidade de redes, mostrando como explorar as vantagens evitando os pontos negativos.

1.3 CRESCIMENTO

Atualmente o conceito de comunicação mudou, qualquer um pode se comunicar facilmente através de seu celular. Tudo isso graças aos grandes avanços da tecnologia que iniciou sua acessão no século XX e não para de avançar até os dias atuais.

> Com o advento da globalização a partir do século XX, a demanda tornou-se extremamente capitalista, surgindo novas necessidades de conquistar qualidade e eficiência no serviço a custo baixo. A grande quantidade de dados ou informações a serem processadas e armazenas demonstram um volume incalculável. O sistema de redes surge neste contexto: para superar a necessidade do ser humano de controlar os dados em abundância com precisão e rapidez (COSTA; SILVA; CRUZ, 2012, p. 82).

Com o crescimento e aperfeiçoamento de processos e também o gigantesco volume de informações, surgiu grande necessidade de continua utilização da rede de computadores e consequentemente o aumento de seu desempenho para atender

---

[2]Disponível em: <http://www.lanwan.com.br/Aulas_Senac/Tec_Redes_Final_Semana/Aula%20270310%20e%2010042010%20-%20Historia%20das%20redes.pdf>

[3] TANEBAUM apud COSTA; SILVA; CRUZ, 2012, p. 81.

a demanda de serviços. Cada pessoa pode obter um espaço na nuvem, seja para hospedar sites, para realizar backup de arquivos, gerenciar um comercio eletrônico, entre outros. A demanda está em constante crescimento, escalando com requisitos cada vez mais sofisticados, ágeis e inovadores.

2 SERVIÇOS NAS NUVENS

2.1 A ORIGEM DA COMPUTAÇÃO EM NUVENS

Segundo Otero (2013, p.19) a computação em nuvem (CN) surgiu como um modelo representando grandes mudanças, conforme serviços tecnológicos da informação são gerados, desenvolvidos, implantados, escalados, atualizados, mantidos e pagos. O autor explica que assim pode ser mais prático e rentável, porque o acesso geral é sob demanda de recursos computacionais compartilhados, como rede, servidores, armazenamento, aplicações e serviços. Tais recursos podem ser providenciados de maneira ágil sendo liberados com mais facilidade de gerenciamento ou interação com o provedor de serviços.

> A primeira tecnologia de nuvem, denominada nuvem 1.0, surgiu da abstração das camadas TCP/IP, onde dispositivos se comunicavam entre si através de especificações de protocolos TCP/IP, sem saber exatamente a localização de cada um deles. A próxima tecnologia de nuvem, ou nuvem 2.0, refere-se à abstração de dados da World Wide Web, onde documentos podem ser inseridos e recuperados através daWeb sem a necessidade do usuário saber onde estes estão localizados. Finalmente, a nuvem 3.0, a era atual da nuvem, surge da abstração da complexidade da infraestrutura de servidores, aplicativos, dados e plataformas heterogêneas em que infraestruturas, servidores ou aplicativos podem ser utilizados sem que o cliente/usuário saiba onde eles estão localizados (OTERO, 2013, p.12).

Como mostrado através do exposto o histórico da Computação em Nuvens vem sendo ampliado e se aperfeiçoando através dos tempos. Seu conceito procede de técnicas de TI utilizadas anteriormente, gerando novos conceitos e técnicas que levaram à sua amplitude, tendo assim que oferecer cada vez mais uma melhor qualidade, disponibilidade e principalmente segurança.

“A CN é um paradigma de disponibilização de recursos das TI, num ambiente virtual, escalável e multiutilizador.” (GOMES, 2012, p.13). A autora deixa a entender que com crescente demanda de serviços, a CN oferece uma ilusão de recursos infinitos, gerando assim confiança. Por atender as necessidades dos clientes, acaba criando um paradigma que introduz um novo sistema de negócios e gestão, onde o utilizador usa os recursos de hardware, software e rede, através de serviços disponibilizados na Internet para multiusuários. Os custos são reduzidos por não ter componentes físicos e também há simplicidade em utilizá-los. “Fisicamente, a nuvem é constituída por servidores sustentados por uma infraestrutura orientada ao serviço, escalável e com componentes coesos e fracamente acoplados” (GOMES, 2012, p.13). O método se torna rentável e muito utilizado com diferentes níveis de desempenho e segurança.

2.2 SERVIÇOS NAS NUVENS

2.2.1 SaaS, PaaS, IaaS

A computação nas nuvens dispõe de serviços pré-estabelecidos para atender as necessidades de seus clientes. A seguir serão mostrados tais serviços e como funcionam.

- *Software as a Service* (SaaS) – Ou Software como Serviço, o cliente pode usar aplicativos disponibilizados e licenciados pelo provedor, rodando sobre a camada PaaS ou IaaS de uma plataforma de nuvem (OTERO, 2013, p.21). As aplicações são acessadas de vários dispositivos do cliente, podendo ser de interface cliente-servidor (thin client), ou uma interface de aplicação web. Nesse caso o cliente só tem acesso ao nível de aplicação sendo apenas a infraestrutura básica da nuvem, incluindo rede, servidores, sistemas operacionais, armazenamento, ou até mesmo recursos de aplicativos individuais (GOMES, 2012, p.24).
- *Platform as a Service* (PaaS) – Ou Plataforma como Serviço, o cliente tem disponível plataformas na Computação em Nuvem, usando linguagens de programação, bibliotecas, serviços e ferramentas suportadas pelo provedor para

desenvolvimento de suas aplicações (OTERO, 2013, p.22). Neste modelo, o consumidor não tem a capacidade de gerenciar ou controlar a infraestrutura de nuvem subjacente, incluindo rede, servidores, sistemas operacionais ou armazenamento, mas tem controle sobre as aplicações implantadas e pode fazer ajustes de configuração no seu ambiente de hospedagem (OTERO, 2013, p.22).

- *Infrastructure as a Service* (IaaS) – Ou Infraestrutura como Serviço, oferece ao cliente a aptidão de gerenciar processamento, armazenamento, rede e outros recursos computacionais fundamentais, em que o cliente pode implantar e executar qualquer software, até mesmo sistemas operacionais e aplicativos (OTERO, 2013, p.22). O cliente não administra ou controla a infraestrutura básica da Computação em Nuvem. Pode ter um controle limitado sobre alguns componentes de rede (Gomes, 2012, p.23).

2.2.2 Tipos de Nuvens

Cada tipo de nuvem é direcionada para consumidores específicos, para atender seus respectivos tipos de serviços.

As Nuvens Públicas geralmente são usadas por pessoas físicas e empresas de serviços públicos. A nuvem pública é de livre uso e qualquer um pode acessá-la, mas isso faz sua segurança ser mais frágil e vulnerável a ataques.

De acordo com Castro e Ferraz (2013, p.362) as Nuvens Privadas são utilizadas geralmente por empresas particulares ou pelo governo, porque precisam de seus dados seguros e isolados de acesso externo diminuindo riscos de ataques externos.

As Nuvens Híbridas mescla as funções de nuvens publica e privada. São usadas por serviços públicos do governo e empresas privadas que disponibilizam serviços gratuitos por exemplo (como facebook). As informações devem ser bem guardadas, porém pode-se ter acesso externos de pessoas físicas que tenham cadastro.

- Nuvem Pública (*Public Cloud*) – Este modelo em Nuvem é disposto para uso aberto do público de forma pay-as-you-go (algo como pague e use). Este modelo

pode ser gerenciado e operado por organizações corporativas, acadêmicas e governamentais, ou até combinação deles (GOMES, 2012, p.20).

- Nuvem Privada (*Private Cloud*) – Este modelo em Nuvem serve para uso exclusivo de uma única organização contratante, constituído de vários consumidores (clientes, unidades de negócios por exemplo). Pode ser de propriedade, gerenciada e operada pela organização, um terceiro, ou alguma combinação deles. Pode existir dentro ou fora das instalações da organização (GOMES, 2012, p.18).
- Nuvem Híbrida (*Hybrid Cloud*) – Este modelo em nuvem mescla infraestrutura de dois ou mais modelos de Computação em Nuvem distintos (publico, privado e comunitário) que continuam como únicas entidades, mas estão unidos por tecnologia padronizada ou proprietária que permite a comunicação de dados e aplicações. Ex: balanceamento de carga entre nuvens (GOMES, 2012, p.21).
- Nuvem Comunitária (Community Cloud) – Este modelo em nuvem é dedicado para uso único e exclusivo de consumidores de comunidade especifica de organizações que têm interesses comuns (missão, requisitos de segurança, política e considerações de conformidade). Pode ser de propriedade, gerenciada e operada por uma ou mais das organizações na comunidade, um terceiro, ou alguma combinação deles. Pode existir dentro ou fora das instalações das organizações (CASTRO; FERRAZ, 2013, p.362).

3 SERVIDORES WEB NAS NUVENS

Os servidores web são os pilares que sustentam a Internet, são eles que hospedam todas as páginas e suas bases de dados. No futuro, esta tendência deve se acentuar, com páginas web dinâmicas e aplicativos via web substituindo cada vez mais os aplicativos desktop[4]. Essa citação revela a importância dos servidores web, afinal são eles que armazenam um imenso volume de dados nas nuvens, provindo serviços para seus usuários. A seguir será mostrado algumas configurações básicas para deixar um servidor ativo.

3.1 RECURSOS E PACOTES DE DESENVOLVIMENTO WEB

Nos primórdios da internet, eram utilizadas apenas páginas HTML estáticas e scripts CGI. O Apache continua oferecendo suporte apenas a esses recursos básicos, mas pode ser expandido através de módulos, passando a suportar scripts em PHP, acessar bancos de dados MySQL, entre inúmeros outros recursos.[5] Com o passar do tempo os recursos do Apache tornaram-se limitados para acompanhar a quantidade de plataformas de desenvolvimento que foram aparecendo na web. Para cobrir algumas essas necessidades surgiram pacotes de ferramentas web que incluem recursos como o Apache2, Mysql, PHP, Perl, SQLtite, entre outros, facilitando a instalação do ambiente de desenvolvimento, estes pacotes são popularmente conhecidos como Wamp (windowns) ou Lamp (linux) que tem a capacidade de atender todo tipo de plataforma. Também é possível instalar os recursos citados individualmente, podendo escolhe apenas os recursos necessários para executar uma determinada aplicação.

[4]<http://www.hardware.com.br/livros/servidores-linux/capitulo-configurando-servidores-web.html>
[5] <http://www.hardware.com.br/livros/servidores-linux/capitulo-configurando-servidores-web.html>

3.2 DNS

O DNS (*Domain Name System* - Sistema de Nomes de Domínios) é um sistema de gerenciamento de nomes hierárquico e distribuído visando resolver nomes de domínios em endereços de rede IP (VERGARA, 2014).

A configuração do servidor DNS é quase sempre necessária, porque ele responderá pelo domínio. Aprender a configurar o DNS corretamente é de vital importância, caso contrário poderá surgir problemas ao enviar e-mails pela falta do DNS reverso, ou problemas mais graves com o registro do domínio.[6]

A configuração avançada do servidor DNS gira em torno de quatro arquivos, dos quais dois são existentes e dois tem de ser criados pelo usuário. A seguir será demonstrada a configuração dos arquivos existentes:

/etc/bind/named.conf.options - Contem as opções globais de configuração DNS.

/etc/bind/named.conf.local - Arquivo em branco, para designar qual a zona que o DNS irá responder.

Os dois arquivos criados pelo usuário tem o destino referenciado no arquivo */etc/bind/named.conf.local* com a função de arquivos de banco de dados (um normal e outro reverso) que irá conter toda a configuração de zona.

3.3 APACHE

O Apache pode ser dividido em duas grandes famílias: o Apache 2.x e o Apache 1.3 que, apesar de muito antigo, ainda é usado em muitos servidores. O Apache 2 trouxe muitas vantagens, sobretudo do ponto de vista do desempenho, além de oferecer novos módulos e mais opções de segurança, mas sua adoção foi retardada nos primeiros anos por um detalhe muito simples: o fato de ele ser incompatível com os módulos compilados para o Apache 1.3. Como os módulos são a base do servidor web, muitos administradores ficavam dependentes ao Apache 1.3 devido à falta de disponibilidade de alguns módulos específicos para o Apache 2.[7] Conclui-se que há muitos problemas quanto a versões do apache, que geram situações adversas para o consumidor que precisa se adequar aos módulos que

[6]<http://www.hardware.com.br/livros/servidores-linux/instalando-servidor-lamp.html>
[7]<http://www.hardware.com.br/livros/servidores-linux/instalando-apache.html>

vieram surgindo com tempo, o Apache 1.3 ainda se encontra em muitas instalações devido à inação natural que existe no ramo de servidores, mas não existem bons motivos para usá-lo em novas instalações, pois o Apache 2 é melhor em todos os quesitos.[8]

Então para obter um bom padrão de configuração para o servidor deve-se instalar o Apache 2 e também o pacote apache2-utils, que abrange diversos utilitários de gerenciamento. Para instalar os pacotes do apache, primeiro deve-se atualizar o repositório com comando *apt-get update*. [9]Logo após usa-se o comando *apt-get install apache2 apache2-utils* que é o comando para baixar e instalar o Apache 2 juntamente com pacote de utilidades.

3.4 CONFIGURAÇÃO DO AMBIENTE NA NUVEM

A Computação em Nuvem utiliza-se da virtualização para criar um ambiente (nuvem), onde aloca instâncias (sistemas operacionais virtualizados) de acordo com os recursos disponíveis (máquinas físicas). Essas instâncias de máquinas virtuais são alocadas de acordo com as máquinas físicas que compõem o parque de máquinas da nuvem (BACHIEGA, 2014). As instancias são usadas para simular maquinas operantes para diversas finalidades, como configuração ou testes.

As configurações utilizadas para servidor na nuvem nesse trabalho, foram o sistema operacional Debian7 fornecido pela DigitalOcean, com o apache2 instalado e utilizando o banco de dados MySQL.

[8]< http://www.hardware.com.br/livros/servidores-linux/instalando-apache.html>
9<http://www.hardware.com.br/livros/servidores-linux/instalando-apache.html>

4 ARQUITETURA WEB

4.1 WORLD WIDE WEB

A World Wide Web (WWW, ou simplesmente Web) é um espaço de informação em que os itens de interesse, referidas como recursos, são detectados por identificadores globais chamados identificadores de recursos uniforme (URI)[10].

A Web evoluiu e ocupou todos os espaços fazendo *jus* ao nome “World Wide” (ROCHA, 1999). As Páginas interligadas por hipertexto aumentaram exponencialmente, existindo inúmeras atualmente. Páginas cada vez mais dinâmicas e interativas, passando a se comportar como aplicações. Rocha (1999) diz que o browser acompanhou esta evolução comportando-se como interface universal, apto em prover ao usuário acesso interativo e uniforme a programas remotos em diversas plataformas. O autor conclui que a Web deixou de ser um mero serviço e passou a ser uma imensa plataforma que estimula criação de novas linguagens e tecnologias.

4.2 HTML E CSS

Segundo Carvalho[11], *HyperText Markup Language* (HTML) é a linguagem desenvolvida para “transmitir” a informação dos documentos que são disponibilizados na Web pelo protocolo HTTP (*hypertext transfer protocol*). O autor explica que a formatação do HTML é executada no layout de qualquer texto inserido,

[10] Architecture of the World Wide Web, Volume One <http://www.w3.org/TR/webarch/#acks>

[11] <http://digitool.fe.up.pt:1801/view/action/singleViewer.do?dvs=1445434789188~448&locale=pt_BR&metadata_object_ratio=25&side_by_side=false&VIEWER_URL=/view/action/singleViewer.do?&preferred_extension=pdf&DELIVERY_RULE_ID=5&frameId=1&usePid1=true&usePid2=true>

na inserção de imagens, na criação de links e atribuição de cores ou imagens para papel-de-parede.

A linguagem HTML é considerada fácil de aprender, pela simplicidade e por sua estrutura ser formada por marcação e não por comandos complexos. Sua dificuldade está na quantidade que se tem de escrever. O CSS basicamente serve para aplicar um formulário para várias paginas HTML, evitando assim perda de tempo e estabelecendo uniformidade.

De acordo com Rocha (1999) a linguagem HTML que foi um pivô para impulsão da web, mesmo sendo construída somente para estruturar páginas de hipertexto. Para tarefas como busca na web e trocas de e-mails, foram exploradas varias idéias. A mais bem aceita foi CGI (*Common Gateway Interface*) que ainda é popular atualmente. CGI é o método usado para permitir a interação entre o servidor e outros programas executados no sistema (OTSUKA)[12]. O método CGI pode fazer a conexão aplicação/servidor por que funciona de maneira dinâmica, ou seja, a interação é em tempo real.

Percebe-se que a evolução da plataforma Web é constante e atualmente existem páginas HTML que não são estáticas, podendo assim replicar interativamente ações do usuário.

4.3 URI, URL E URN

Desde a criação da web, um de seus principais objetivos é de construir uma comunidade global em que qualquer das partes possa compartilhar informações com qualquer outra parte. Para atingir este objetivo, a Web faz uso de um único sistema de identificação mundial: URI (Identificador de Recurso Universal). URIs proporcionam identificação que é comum em toda a Web. O escopo global de URIs promove em larga escala "efeitos de rede": o valor de um identificador aumenta quanto mais ele é usado de forma consistente.

O URI abrange URL(Localizador de Recurso Universal) e URN(Nome de Recurso Universal)[13]. URN qualifica um nome a um objeto na internet, como uma imagem, registro, etc. O URL é um caminho para um objeto da internet, basicamente

[12]<http://penta.ufrgs.br/edu/forms/cgi.html>
[13]<http://www.phpmais.com/url-urn-e-uri-que-confusao/>

o URI é o identificador que contém o URL e URN de algo na web. Por exemplo a URI: *http://www.exemplo.org/localizacao/URI/com/o/nome-do-recurso.html*[14].
URL: *http://www.exemplo.org/localizacao/URI/com/o/*
URN: *nome-do-recurso.html*

Através do exposto, é possível perceber que cada um destes endereços de recursos são específicos, e o URI é formado pela junção do URL e URN em sua estrutura.

4.4 PROTOCOLO HTTP

De acordo com Rocha (1999, p.8), quando o tema é plataforma web o HTTP (Hypertext Transfer Protocol) tem um papel importante por ser um prevalecente agente de comunicação entre browser e servidor Web. Lima[15] explica como funciona o protocolo dizendo que é uma interação simples de pedido e resposta chamado de transmissão web. O autor explica que as interações são feitas com o pedido do cliente ao servidor e a resposta do servidor para o cliente, podendo ter diferentes tipos de formatos, e nesse ponto o autor deixa claro que é necessário estabelecer o formato de resposta, pois nem sempre o cliente está preparado para receber respostas de qualquer tipo de formato.

HTTP é um protocolo de nível de aplicação para sistemas de hipermídia, colaborativos e distribuídos. É um protocolo genérico, sem estado e orientado a objetos que pode ser usado para diversas tarefas, tais como servidores de nomes e sistemas de gerenciamento de objetos distribuídos, através da extensão de seus métodos de requisição.[16]

Lima[17] (2003) afirma que a comunicação via HTTP é estabelecida normalmente em uma conexão TCP geralmente com a porta padrão (80). O autor relata que o protocolo HTTP não depende de nenhum procedimento prévio realizado pelo cliente e não guarda informações sobre requisições anteriores, não definindo

[14] <http://www.phpmais.com/url-urn-e-uri-que-confusao/>
[15] LIMA, 2003, p. 20.
[16] ROCHA apud W3C, 1999, p. 8.
[17] LIMA, 2003, p. 20.

assim o conceito de estados. Nesse ponto Rocha[18] da o exemplo de três requisições de diferentes clientes ao mesmo tempo, o que levará o servidor a não saber separar requisições por cliente considerando que as requisições não afetem suas subsequentes, concluindo que não é possível atender aplicações de estados que dependam de informações vindas de páginas especificas, sem utilizar mecanismos externos.

[18] ROCHA apud W3C, 1999, p. 8.

5 BANCO DE DADOS

5.1 SISTEMAS DE GERENCIAMENTO DE BANCO DE DADOS (SGBD)

O Sistema de Gerenciamento de Banco de Dados (SGBD) é um programa que implementa operações que visam à persistência de dados (KANASHIRO, 2011). O SGBD funciona como um pacote composto por um aglomerado de programas para gerenciar o Banco de Dados (BD). Existem vários SGBDs como o Oracle, FireBird e MySQL. Neste trabalho será abordado o MySQL, por ser um SGBD bem conceituado no mercado, de simples manuseio e pela facilidade em se obter.

Segundo Narciso (2003) O SGBD MySQL é um servidor de base de dados baseado na linguagem de consulta SQL, a qual é a mais popular e difundida linguagem de consulta a bancos de dados no mundo. O autor citado afirma que o SGBD MySQL é multiusuário, possuindo velocidade, robustez e facilidade de uso.

Existem diversas razões para realizar a segurança dos dados em um SGBD. Pode-se destacar a necessidade das organizações em proteger seus dados, realizar análises e auxiliar na tomada de decisões. Segundo Pazinatto (2010) as informações e conhecimentos contidos em banco de dados precisam auxiliar com segurança os processos decisórios.

De acordo com Elmasri (2005) apud Pazinatto (2010) a segurança em banco de dados é uma área muito ampla que se refere a várias questões. Essas questões podem ser: legais ou éticas referentes ao acesso a certas informações; questões políticas com a definição de quais informações não podem ser tornadas públicas; e questões relacionadas a sistemas, com definição de quais níveis do sistema as funções de segurança devem ser implementadas (PAZINATTO, 2010).

O autor Pazinatto (2010) faz referência a um profissional que gerencia o SGBD. Este profissional é chamado de Administrador de Banco de Dados (DBA) sendo este responsável pela segurança geral do sistema de banco de dados. Ainda segundo o autor supracitado, o DBA também realiza ações como cadastro de usuários/contas, concessão de privilégios e atribuição de níveis de segurança" (PAZINATTO, 2010).

Na literatura, são feitas comparações entre os SGBDs em vários quesitos (desempenho, velocidade de acesso, etc.). Porém segundo Narciso (2003), não se foca no quesito *backup* e recuperação sendo esta uma necessidade atual das grandes empresas e corporações.

5.2 MYSQL

O MySQL é um sistema de gerenciamento de banco de dados open source que ajuda os usuários a armazenar, organizar, e posteriormente, recuperar dados. Ele possui uma variedade de opções para conceder a usuários específicos permissões diferenciadas dentro de tabelas e bases de dados (SVERDLOV, 2014)[19]. O MySQL dispõe um controle sobre as contas dos usuários, podendo definir permissão de acesso conforme a aplicação que será utilizada. Percebe-se então que o usuário não terá como executar ações que não está capacitado e nem ter acesso a dados não obstante.

5.3 SEGURANÇA PARA BANCO DE DADOS

Um ponto crucial em SGBDs é a segurança. Segundo Ackermann (2003) segurança em SGBD refere-se à proteção dos dados contra a divulgação, alteração ou destruição não-autorizada. Os dados são a identidade da empresa, o autor cita que a segurança garante aos usuários permissão para fazer o que estiverem

[19]<https://www.digitalocean.com/community/tutorials/como-criar-um-novo-usuario-e-conceder-permissoes-no-mysql-pt>

tentando fazer (DATE,1999). “Uma das principais razões que motivaram o uso de SGBDs é o controle centralizado, tanto dos dados, quanto dos programas de acesso a esses dados” (ACKERMANN, 2003 apud OZSU,1999). O autor reforça que uma gestão centralizada de um SGBD, em ambiente cliente/servidor, provê vantagens de segurança.

Para estabelecer a segurança de um Banco de Dados (BD), existem procedimentos, como *backup* e gerenciamento de privilégios de usuários. Percebe-se então que é de vital importância seguir procedimentos para assegurar a integridade de um banco de dados. A seguir serão mostrados algumas medidas para cuidar do mesmo.

5.3.1 Backup

Backup é uma palavra que vem do idioma inglês, que significa cópia de segurança de um dispositivo de armazenamento a outro, para que os dados possam ser restaurados caso houver perda dos dados originais (Junior, 2011). De acordo com Sousa (EMBRAPA, 2009) quando o assunto é banco de dados, é imprescindível que se tenha um plano de manutenção e backup, assim como a garantia de que este sempre esteja disponível. O autor deixa clara a importância do backup e manutenção de um BD frisando a disponibilidade do mesmo, sendo que o usuário tenha disponibilidade de acessar os seus dados sempre que julgar necessário.

Segundo Sousa (EMBRAPA, 2009) tais operações têm custo operacional, logo há a necessidade de horários estabelecidos para sua execução. O autor diz ainda que os métodos tradicionais de backup (físico ou lógico) são extremamente eficazes, mas estão sempre com atraso na atualização por serem gerados de acordo com pré-agendamento e não em tempo real. Há mecanismos para sanar esse problema, mas não serão abordados por fugirem do foco deste trabalho.

5.3.2 Backup em Nuvens

Atualmente a forma mais segura de se manter dados é na nuvem, por disponibilizar grande volume de espaço, através da web garante disponibilidade e segurança. Segundo Junior (2011) é uma grande vantagem utilizar a computação em nuvem porque oferece ao usuário armazenamento de informações em seus servidores que podem ser sempre acessado ao invés de manter uma própria infra-estrutura. O autor afirma que a capacidade de fazer backup de banco de dados em nuvens é essencial para sua segurança, já que serão guardados em unidades externas seguras. E ainda reforça que backups armazenados em nuvens são mais acessíveis e na maioria das situações tem acesso mais ágil para se restaurar e bem mais confiável (JUNIOR, 2011 apud ORACLE3, 2008).

5.3.3 Privilégios de Usuários

A gestão de controle de acesso dos usuários do SGBD é estabelecida de acordo com a função que o usuário exerce no sistema. Segundo Siedler[20], no MySQL exite uma lista de privilégios para os usuários, privilégios que serão mostrados a seguir:

- ALL [PRIVILEGES]: Todos os privilégios exceto o GRANT OPTION.
- ALTER: Permite ao usuário alterar novas tabelas ou base de dados;
- CREATE: Permite ao usuário criar novas tabelas ou base de dados;
- CREATE TEMPORARY TABLES: Permite ao usuário criar tabelas temporárias;
- DELETE: Permite ao usuário deletar linhas das tabelas;
- DROP: Permite ao usuário deletar novas tabelas ou base de dados;
- EXECUTE: Permite ao usuário executar stored procedures (MySQL 5.0);

[20]<http://187.7.106.14/marcelo/si/permissao.pdf>

- FILE: Permite executar SELECT ... INTO OUTFILE e LOAD DATA INFILE;
- INDEX: Permite executar CREATE INDEX e DROP INDEX;
- INSERT: Permite ao usuário inserir linhas das tabelas;
- LOCK TABLES: Permite ao usuário "trancar" tabelas que o mesmo tenha privilégio SELECT;
- PROCESS: Permite ao usuário executar SHOW FULL PROCESSLIST, para ter acesso a lista de processos.
- REFERENCES: Ainda não esta implementado;
- RELOAD: Permite ao usuário executar FLUSH;
- REPLICATION CLIENT: Permite ao usuário obter a localização do Master ou Slave;
- REPLICATION SLAVE: Necessário para a replicação Slave (leitura dos eventos do log binário do Master);
- SELECT: Permite ao usuário utilizar o comando SELECT para ler bases de dados;
- SHOW DATABASES: Permite ao usuário visualizar todos os bancos de dados;
- SHUTDOWN: Permite executar mysqladmin shutdown;
- SUPER: Permite executar CHANGE MASTER**,** KILL , PURGE MASTER LOGS e SET GLOBAL. Permite conectar-se ao servidor uma vez, mesmo que o max_connections tenha sido atingido;
- UPDATE: Permite ao usuário atualizar as linhas das tabelas;
- USAGE: Sinônimo para "no privileges";
- GRANT OPTION: Permite ao usuário disponibilizar ou remover privilégios de outros usuários.

De acordo com Siedler, o usuário é criado, mas não tem permissão nenhuma na base de dados, por isso é necessário gerenciar as permissões com os comandos GRANT e REVOKE. Esses comandos permitem aos administradores do sistema criar usuários e conceder e revogar direitos aos usuários do MySQL em quatro níveis de privilégios.

Siedler exemplifica os quatro níveis de privilégios como globais que se aplicam para todos os bancos de dados em um determinado servidor (Nivel Global). Privilégios de bancos de dados aplicam-se a todas as tabelas em um determinado banco de dados (Nível dos bancos de dados). Privilégios de tabelas que se aplicam a todas as colunas em uma determinada tabela (Nível das Tabelas). Privilégios de colunas são aplicados a uma única coluna em uma determinada tabela (Nível das colunas). Os níveis são específicos e deixam a gestão do Banco de dados menos confusas, concluindo assim que com o controle de privilégios da segurança e organização ao banco de dados.

6 SEGURANÇA DE SERVIDORES WEB NAS NUVENS

Quando se diz que algo está “nas nuvens” no sentido de TI é bem mais complexo do que dizer que os dados estão simplesmente voando. As informações armazenadas nas nuvens estão armazenadas em algum dispositivo físico. Esse dispositivo é o servidor web, os servidores são à base da internet porque eles disponibilizam o espaço para o armazenamento de toda informação que há na grande web. Por isso configurar a segurança de servidores web é de vital importância.

O ambiente utilizado para todas as configurações deste trabalho foi o Debian7, disponibilizados pela DigitalOcean.

6.1 CONFIGURAÇÃO DO SSH()

O SSH - Secure Shell ou Acesso Remoto Seguro - é um protocolo que permite acesso virtual ao servidor como um terminal de comando semelhante ao prompt de comando do DOS, por exemplo[21].

A transmissão de dados através do SSH é totalmente criptografada, logo não existem riscos de alguém ter acesso ao que o usuário está fazendo no servidor. Quando o usuário conecta via terminal remoto com seu servidor, ele está controlando aquele servidor a partir de seu próprio sistema operacional. Qualquer comando do usuário é executado no servidor e estabelecido de acordo com os parâmetros de comandos do servidor[22].

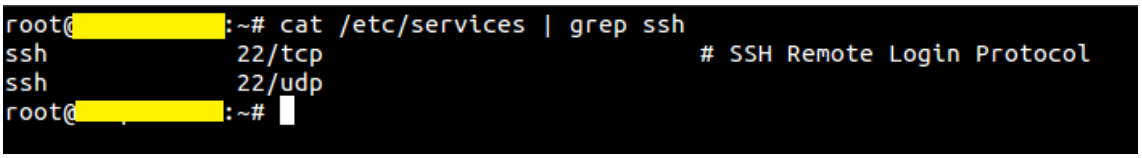
[21]< https://www.vandyke.com/solutions/ssh_overview/ssh_overview.pdf>

[22] <https://www.vandyke.com/solutions/ssh_overview/ssh_overview.pdf>

Para todos os testes deste trabalho foram usados o sistema operacional Debian7 fornecido pela DigitalOcean[23].

Uma opção interessante para melhorar a segurança do SSH é mudar a porta de acesso. Por padrão o SSH aceita conexões na porta 22 (comprovado na imagem 1), porém a maioria dos softwares de escaneamento de portas e serviços não analisam portas acima de 1024[24].

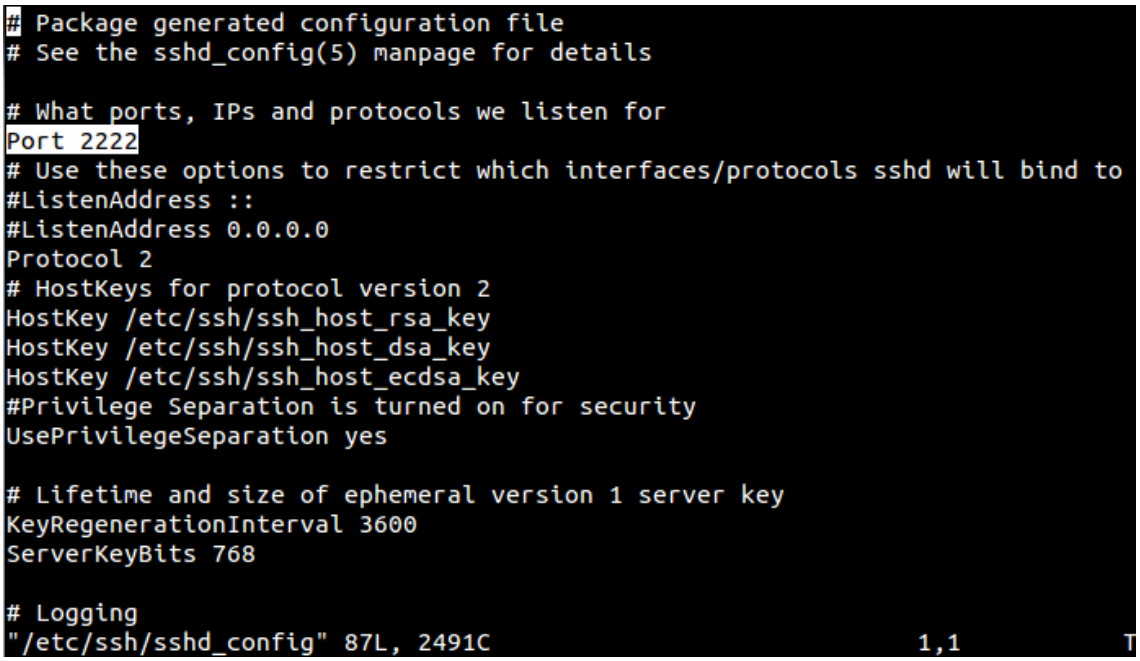
Imagem 1: Captura das portas do protocolo SSH



Fonte: Pedro Henrique Rodrigues

Na imagem 2 está aberto o arquivo */etc/ssh/sshd_config* de configuração do SSH executado por um editor de texto. Pode-se ver que a porta foi alterada para 2222, assim as chances de softwares de escaneamento de portas e serviços terem sucesso são diminuídas consideravelmente.

Imagem 2: Arquivo de configuração do SSH



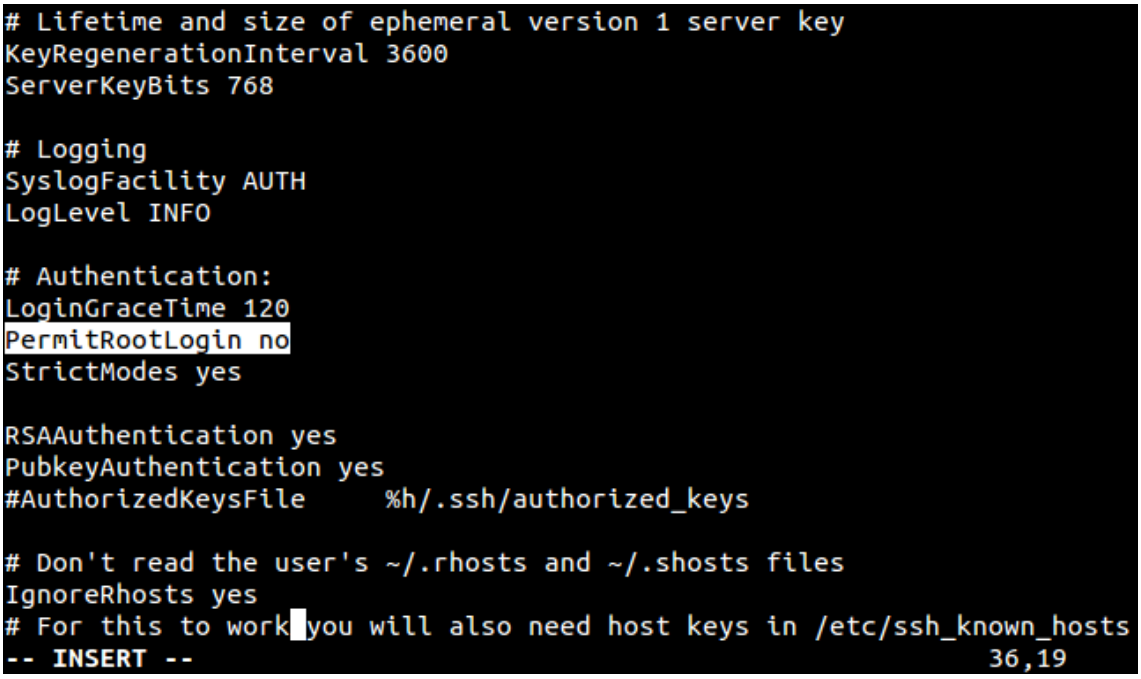
Fonte: Pedro Henrique Rodrigues

Não é aconselhável deixar o usuário *root* se conectar diretamente via SSH, porque o usuário que fizer isso terá privilégios de administrador e poderá fazer quaisquer alterações de configuração, podendo assim, este usuário estar mal

[23]< http://es.tldp.org/Tutoriales/doc-ssh-intro/introduccion_ssh-0.2.pdf>
[24] <http://es.tldp.org/Tutoriales/doc-ssh-intro/introduccion_ssh-0.2.pdf>

intencionado ou até mesmo ser inexperiente, colocando assim em risco o servidor. Então é viável configurar o SSH para negar o acesso do usuário *root* como na imagem 3[25]:

Imagem 3: Bloqueio do usuário *root* com funções de administrador



Fonte: Pedro Henrique Rodrigues

Cuidar da porta SSH é extremamente importante, pois ela pode deixar o servidor web exposto para acesso remoto. Usuários mal intencionados podem tentar diferentes modos para acessar ou prejudicar de alguma forma o servidor. O SSH é um protocolo muito útil, mas se não ter uma boa configuração de segurança pode se tornar um problema[26].

6.2 CONFIGURAÇÃO FAIL2BAN

O fail2ban é uma aplicação que verifica o arquivo de *log* relacionado com a autenticação de usuários e analisa falhas de *login* bloqueando em seguida hosts com falhas sucessivas[27].

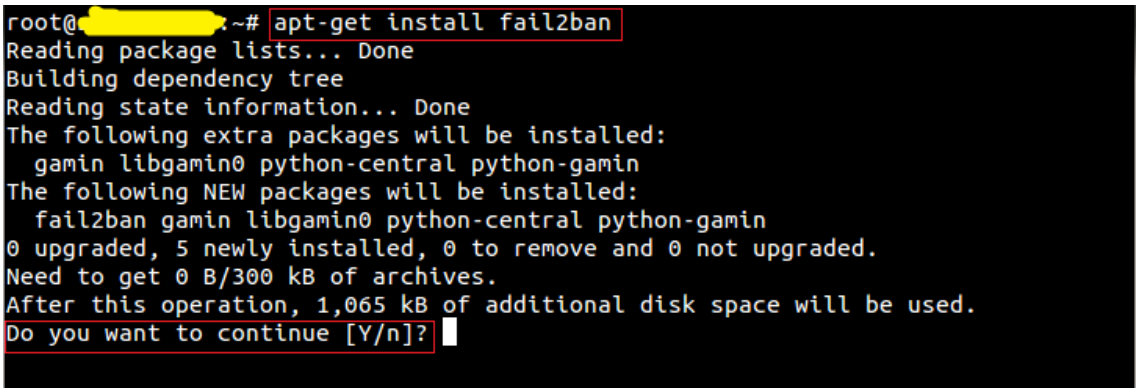
[25] <http://www.if.usp.br/pub/unix/security/ssh2-adminguide.pdf>
[26] <http://www.if.usp.br/pub/unix/security/ssh2-adminguide.pdf>
[27] <https://media.readthedocs.org/pdf/fail2ban-kwirk/latest/fail2ban-kwirk.pdf>

O Fail2Ban é extremamente eficaz na prevenção de ataques de força bruta e de negação de serviço (DoS)[28].

O Fail2Ban é uma ferramenta de segurança que pode prevenir ataques que poderiam ter resultados catastróficos. Os sistemas DoS tentam forçar a entrada através de várias tentativas de acesso consecutivas, a utilidade da ferramenta é exatamente bloquear os IPs que falham no *login* algum numero de vezes (numero pré-definido na configuração do fail2ban).

A imagem 4 mostra em destaque como é instalado o Fail2Ban pelo comando *apt-get install fail2ban* e na parte inferior da mesma, também em destaque solicita a confirmação para o usuário. O usuário então tem que aceitar para a instalação prosseguir automaticamente.

Imagem 4: Instalação Fail2Ban

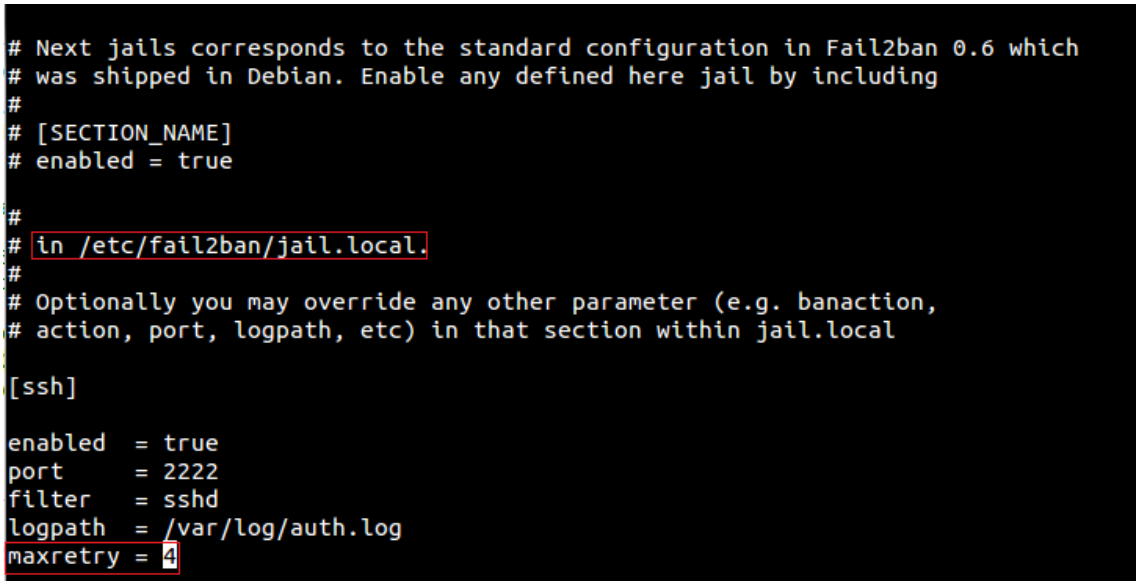


Fonte: Pedro Henrique Rodrigues

O Fail2Ban pode ser configurado para o tipo de auditoria necessária ao servidor. Todas as configurações necessárias se encontram no arquivo */etc/fail2ban/jail.local* mostrado na imagem 5.

[28] < https://media.readthedocs.org/pdf/fail2ban-kwirk/latest/fail2ban-kwirk.pdf >

Imagem 5: Arquivo de configuração do Fail2Ban



Fonte: Pedro Henrique Rodrigues

As primeiras configurações a serem feitas são os endereços (IPs) que não serão afetados pelo programa, também quanto tempo os IPs barrados permanecerão bloqueados e após quantas tentativas falhas o IP persistente será bloqueado. Na parte inferior da Imagem 5 mostra a quantidade de tentativas antes do bloqueio "*maxretry = 4",* ou seja, o IP tem quatro tentativas validas de acesso, a quinta resulta no bloqueio do endereço atacante[29].

As exceções de endereços são adicionadas escrevendo no arquivo mencionado */etc/fail2ban/jail.local* da seguinte maneira:

ignoreip = 127.0.0.1 192.168.1.0/24

E o tempo que do endereço é restringido é escrito da forma a seguir:

bantime = 1800

o tempo é definido por segundos, no exemplo são estabelecidos 1800 segundos, logo o endereço é banido por 30 minutos[30].

O administrador do sistema define nível dessa segurança, podendo tornar a ferramenta abordada mais flexível ou rigorosa. Isso depende de vários fatores, perfil de usuários que podem acessar informação, nível de relevância dos dados entre outros.

Pode-se definir também o endereço de email que serão enviados os alertas de ataque, através da sintaxe "*destemail = pedro@example"* e a ação para se

[29] <https://en.ucloudbiz.olleh.com/manual/Security_fail2ban.pdf>
[30] <https://en.ucloudbiz.olleh.com/manual/Security_fail2ban.pdf>

realizar em caso de ataque. No exemplo a seguir o invasor é banido e o administrador do sistema é contatado por e-mail[31].

banaction = iptables-multiport

action = %(action_mwl)s

É muito interessante essa função, pois mantém o administrador do sistema a par de ataques para o mesmo tomar as medidas necessárias, ou averiguar se realmente o IP bloqueado apresenta algum perigo.

Enfim deve-se editar no ficheiro JAILS os parâmetros de serviços a serem monitorados, como estão destacados na imagem 5[32]:

Enable = true	(Ativação Fail2ban)
port = 2222	(Porta utilizada pelo SSH, que o Fail2Ban irá agir)
filter = sshd	(“Filtra” algumas configurações do SSH)
logpath = /var/log/auth.log	(Caminho do arquivo de log)
maxretry = 4	(Numero de Tentativas)

Basicamente são essas as configurações necessárias para execução do Fail2Ban no servidor. Após as configurações deve-se reiniciar o serviço para o mesmo ser efetivado. É compreendido que está ferramenta de segurança é muito importante e bastante eficaz.

6.3 CONFIGURAÇÃO FIREWALL

O nome firewall significa “barreira de fogo” que nas associações é inserido para que os usuários da Internet não acessem dados das Intranets, limitando o caminho das informações, como as permissões de cada um, ou seja, entre duas redes ele é a barreira que permite ou não o acesso de dados (BAQUI, 2012).

De acordo com Baqui (2012, p17) o firewall tem como concepção moderar a comunicação que a Internet dispõe. O autor implica que isso enfraquece a imagem de “conectividade sem limites”. Essa afirmação deixa a entender que conectividade é importante, porem os usuários não podem ter liberdade de fazer qualquer coisa, como roubo de informações, gerar riscos para dados importantes ou outros tipos de crimes virtuais.

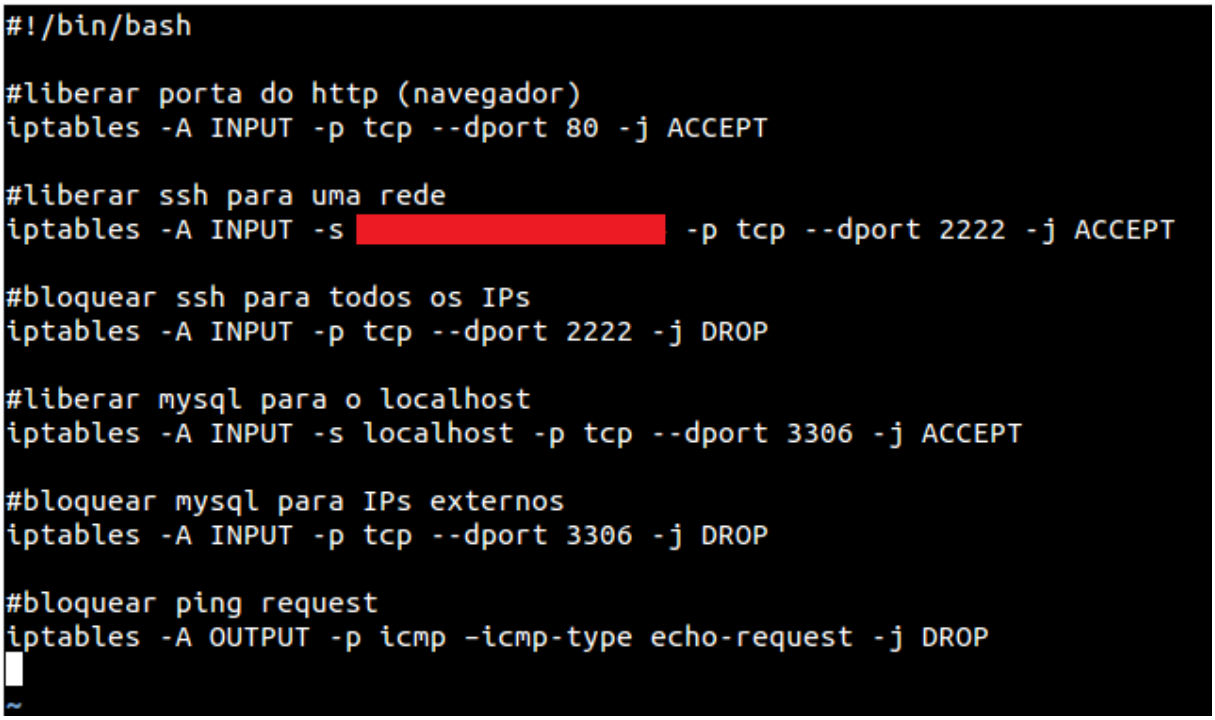
[31] <https://en.ucloudbiz.olleh.com/manual/Security_fail2ban.pdf>
[32] <https://en.ucloudbiz.olleh.com/manual/Security_fail2ban.pdf>

Baqui (2012, p18) pressupõe o firewall como um sistema, ou conjunto de sistemas que influência um plano de segurança em uma empresa e prováveis usuários externos, principalmente os de origem de Internet. A idéia é criar um bloqueio inteligente, que só pode atravessar tráfego autorizado.

A solução certa para a construção de um firewall é dificilmente formada de uma única técnica (BAQUI, 2012). O autor presume que um firewall deve ser um conjunto de técnicas para solucionar diferentes tipos de problemas, problemas estes que devem ser sanados dependendo dos serviços que a empresa planeja disponibilizar e riscos que a mesma possa aceitar. As técnicas para solução destes problemas dependem do tempo, recursos financeiros e conhecimento técnico disponível na empresa.

Para um servidor web é indispensável à configuração de firewall para estabelecer segurança, a imagem 6 aborda algumas configurações de firewall para servidores web.

Imagem 6: Configurando Firewall



Fonte: Pedro Henrique Rodrigues

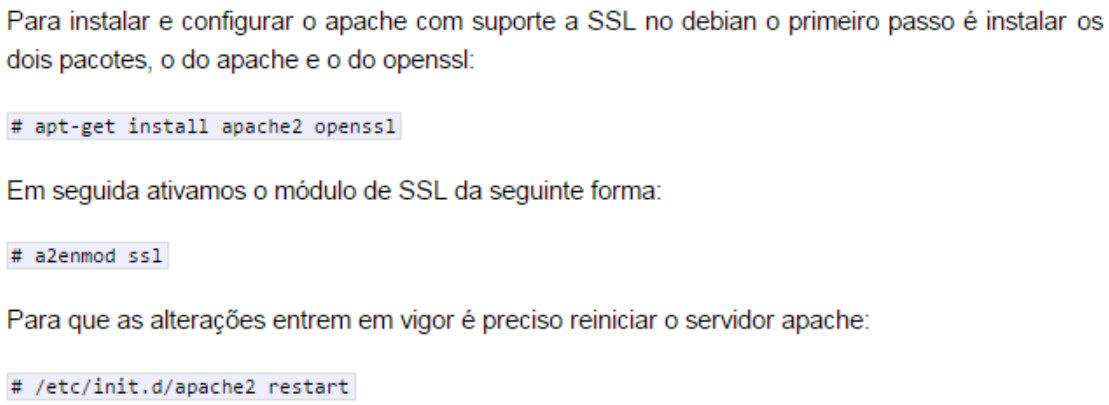
A imagem é auto-explicativa, basicamente mostra a ação e abaixo como é o comando no *bash*(interpretador de comando do Shell) para que esta tenha efeito.

6.4 CONFIGURAÇÃO HTTP+SSL

Secure Socket Layer (SSL) é um padrão global em tecnologia de segurança desenvolvida pela Netscape em 1994. Ele cria um canal criptografado entre um servidor web e um navegador (browser) para garantir que todos os dados transmitidos sejam sigilosos e seguros (ROCHA; PEDROSO; JUNIOR, 2003). Ou seja, a comunicação/transmissão (pedido e resposta) entre usuário e servidor, é segura, e não pode ser compreendida em caso de interceptação por parte de um terceiro individuo com intuito de roubo de informação por exemplo.

A segurança que o SSL oferece graças à criptografia é apreciada e muito utilizada principalmente por sites de comércio eletrônicos, garantindo assim transações seguras com cliente. Neste trabalho foi feito um teste, com esse protocolo seguindo um uma pesquisa realizada[33] e usando imagens de um blog de tutorial de instalação do SSL, posteriormente adquirindo o certificado HTTPS.

Imagem 7: Instalação do pacote Apache2


Para instalar e configurar o apache com suporte a SSL no debian o primeiro passo é instalar os dois pacotes, o do apache e o do openssl:

```
# apt-get install apache2 openssl
```

Em seguida ativamos o módulo de SSL da seguinte forma:

```
# a2enmod ssl
```

Para que as alterações entrem em vigor é preciso reiniciar o servidor apache:

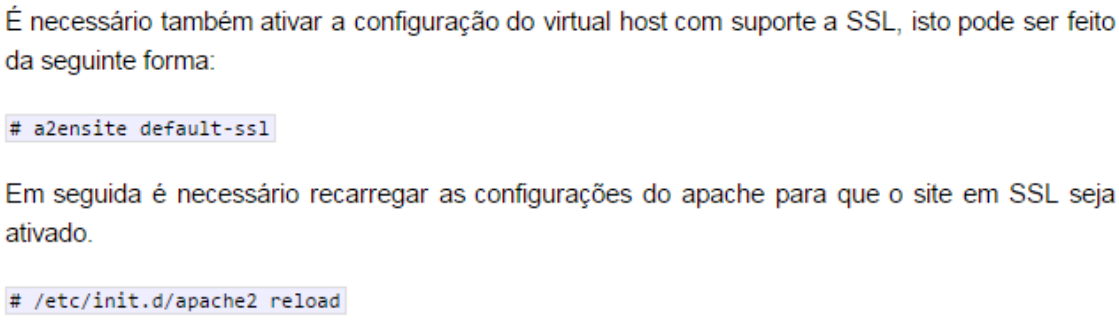
```
# /etc/init.d/apache2 restart
```

Fonte: (http://blog.felipemunhoz.com/configurando-o-apache-para-utilizar-ssl-no-debian/)

O apache é requisito para funcionamento do SSL, que contem diversas utilidades para funcionamento do sistema. A Imagem 7 insere comandos de instalação do apache2, comando de ativação do SSL para o mesmo ser configurado e a reinicializarão do apache para que as alterações entrem em vigor.

33 <http://www.ic.unicamp.br/~rdahab/cursos/mp202/Welcome_files/trabalhos/SSL/texto/SSL_Texto.pdf/>

Imagem 8: Suporte SSL para o Apache



É necessário também ativar a configuração do virtual host com suporte a SSL, isto pode ser feito da seguinte forma:

```
# a2ensite default-ssl
```

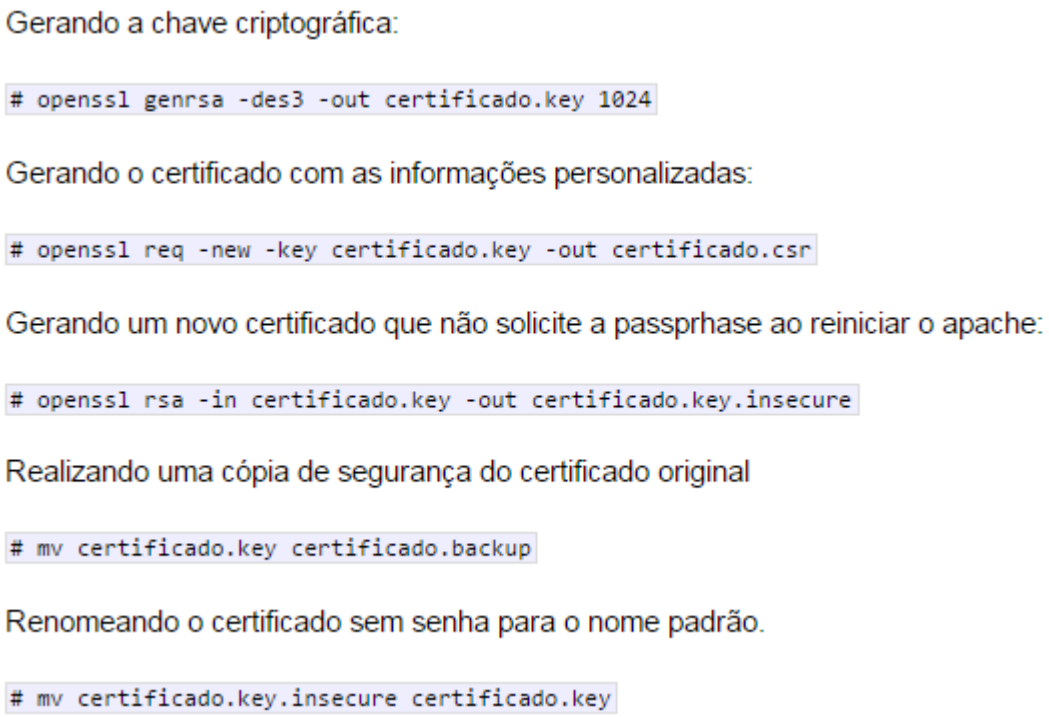
Em seguida é necessário recarregar as configurações do apache para que o site em SSL seja ativado.

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

Fonte: (http://blog.felipemunhoz.com/configurando-o-apache-para-utilizar-ssl-no-debian/)

Na Imagem 8 é instruído ao administrador do sistema que ative o virtual host com o suporte ao SSL, e implica que é preciso recarregar o apache para o site em SSL seja ativado.

Imagem 9: Gerando chave e certificado SSL



Gerando a chave criptográfica:

```
# openssl genrsa -des3 -out certificado.key 1024
```

Gerando o certificado com as informações personalizadas:

```
# openssl req -new -key certificado.key -out certificado.csr
```

Gerando um novo certificado que não solicite a passprhase ao reiniciar o apache:

```
# openssl rsa -in certificado.key -out certificado.key.insecure
```

Realizando uma cópia de segurança do certificado original

```
# mv certificado.key certificado.backup
```

Renomeando o certificado sem senha para o nome padrão.

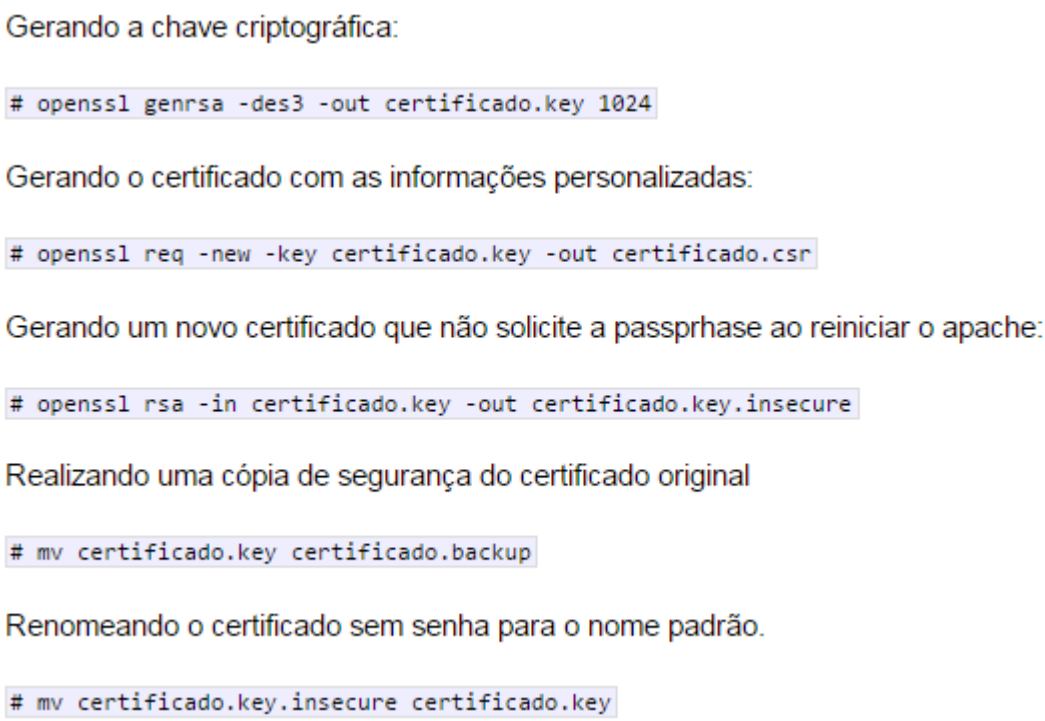
```
# mv certificado.key.insecure certificado.key
```

Fonte: (http://blog.felipemunhoz.com/configurando-o-apache-para-utilizar-ssl-no-debian/)

A imagem 9 apresenta todos os comandos para gerar: chave criptográfica, informações personalizadas, novo certificado, copia de segurança do certificado. Essas configurações são essenciais para o funcionamento do SSL. Por ultimo a imagem mostra como dar um novo nome padrão para o certificado, que não tem senha.

Nesse ponto já é possível acessar o servidor por HTTPS, porém o certificado criado é padrão. Munhoz[34] ressalta que o certificado não é gerado por uma autoridade certificadora conhecida, mas é viável que contenha informações da empresa ou organização dona do site. O autor infere que é preciso utilizar o OpenSSL para gerar uma chave criptográfica e o seu respectivo certificado. As próximas imagens continuam o tutorial[35] para gerar a chave criptográfica e o certificado.

Imagem 10: Gerando chave e certificado SSL final



Fonte: (http://blog.felipemunhoz.com/configurando-o-apache-para-utilizar-ssl-no-debian/)

Gerando o certificado final que possui a extensão '.crt' com o prazo para expiração e a chave criada anteriormente que possui a extensão '.key' ambos serão utilizados na configuração do servidor apache, como a seguir[36]:

[34]<http://www.ic.unicamp.br/~rdahab/cursos/mp202/Welcome_files/trabalhos/SSL/texto/SSL_Texto.pdf>
[35]<http://www.ic.unicamp.br/~rdahab/cursos/mp202/Welcome_files/trabalhos/SSL/texto/SSL_Texto.pdf>
[36]<http://www.ic.unicamp.br/~rdahab/cursos/mp202/Welcome_files/trabalhos/SSL/texto/SSL_Texto.pdf/>

Imagem 11: Chave e certificado SSL autenticados

```
# openssl x509 -req -days 365 -in certificado.csr -signkey certificado.key -out
certificado.crt
```

Fonte: (http://blog.felipemunhoz.com/configurando-o-apache-para-utilizar-ssl-no-debian/)

Na imagem 12 constata os comandos para copiar os arquivos do certificado *.crt* e da chave *.key* para o local definitivo.

Imagem 12: Diretório para chave e certificado SSL.

```
# cp certificado.crt /etc/ssl/certs/

# cp certificado.key /etc/ssl/private/
```

Fonte: (http://blog.felipemunhoz.com/configurando-o-apache-para-utilizar-ssl-no-debian/)

Para finalizar o autor diz que é preciso editar o arquivo */etc/apacha2/sites-enabled/default-ssl* para apontar para o certificado e a chave criados.

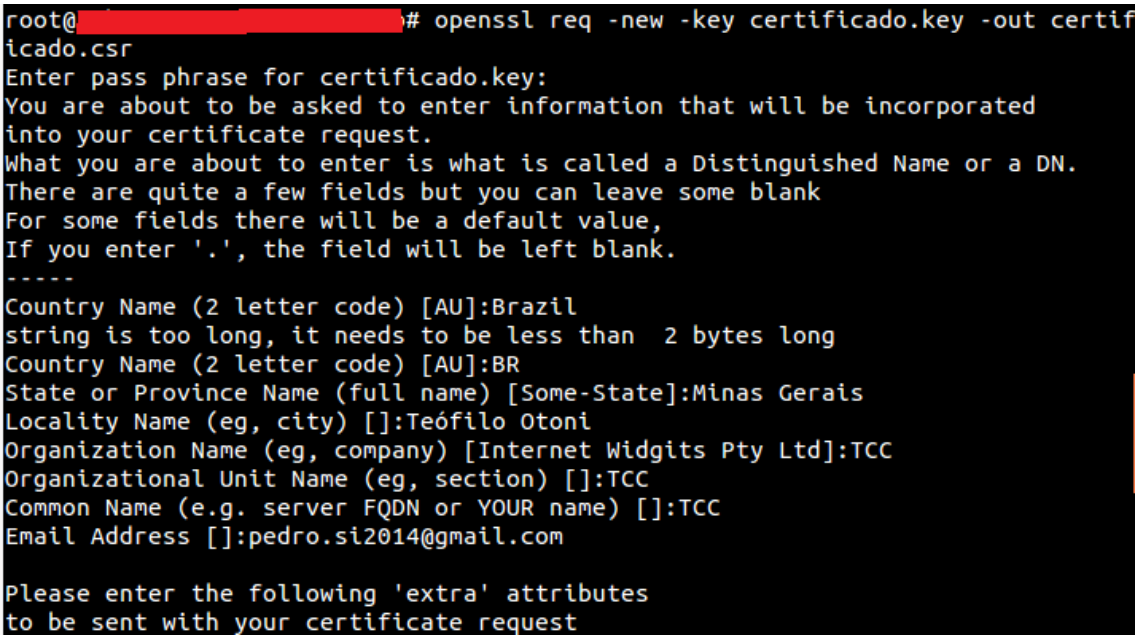
De acordo com o autor deve-se alterar as seguintes configurações:

SSLCertificateFile /etc/ssl/certs/certificado.crt

SSLCertificateKeyFile /etc/ssl/private/certificado.key

Então é só reiniciar o servidor apache e aferir se o novo certificado funcionou como esperado, com o comando */etc/init.d/apache2 restart*. O autor prognostica que se surgir algum problema depois do reinicio do servidor apache, deve-se verificar os arquivos de log do apache no destino */var/log/apache2* para verificar o problema.

Imagem 13: SSL configurado

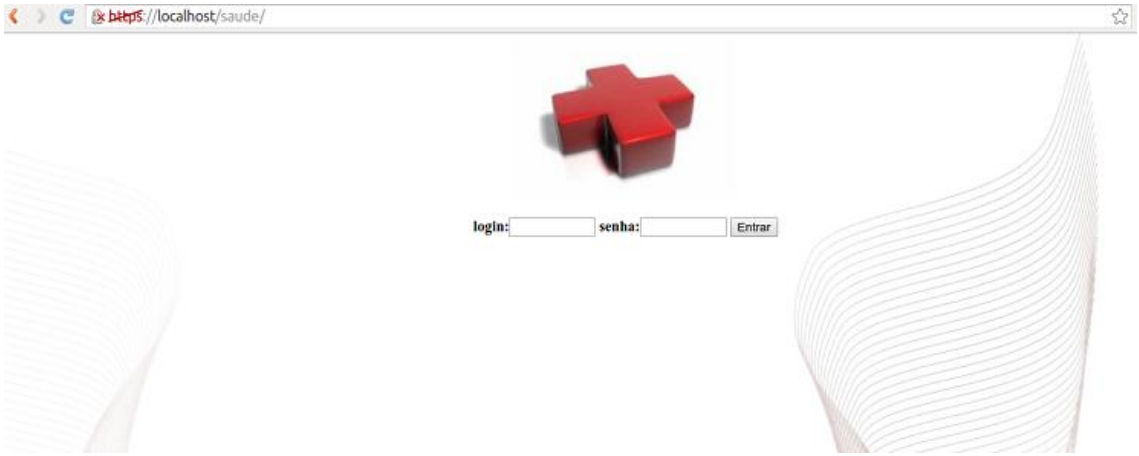


Fonte: Pedro Henrique Rodrigues

A imagem13 apresenta o final da configuração do SSL, todos os comandos já foram realizados e bem sucedidos, nela observa-se o comando *openssl -new -key certificado.key -out certificado.crs* que basicamente gera o certificado final, e a chave criada para serem usados na configuração do servidor apache. Não foi utilizada uma licença paga nesse trabalho, então não há um período de expiração como na imagem 11.

Na imagem 14 a seguir, mostra o site já com a licença ativa. Através do browser foi acessada uma página criada como exemplo através do endereço localhost.

Imagem 14: Pagina protegida pelo protocolo HTTPS



Fonte: Pedro Henrique Rodrigues

O exemplo mostra uma tela de login(autenticação). Na barra de navegação pode-se observar que o site está certificado, o símbolo *HTTPS* antes do endereço comprova que o SSL esta ativo. O símbolo permanece vermelho por que a licença não é paga, mas mesmo assim oferece a criptografia, como é exposto na imagem 15 a seguir.

Imagem 15: Criptografia das senhas do Banco de Dados

login	senha	nome_user
[redacted]	81dc9bdb52d04dc20036dbd8313ed055	[redacted]
[redacted]	81dc9bdb52d04dc20036dbd8313ed055	[redacted]
[redacted]	81dc9bdb52d04dc20036dbd8313ed055	[redacted]
[redacted]	ee14c41e92ec5c97b54cf9b74e25bd99	[redacted]
[redacted]	8762d574aa219edd6e2b9b11421e91cb	[redacted]

Fonte: Pedro Henrique Rodrigues

A imagem 15 mostra os usuários do banco de dados, percebe-se que a senha está criptografada, provando o êxito do apresentado.

6.5 SQL IJECTION

De acordo com Dougherty[37] SQL Injection são comandos SQL que usam a barra de navegação do browser para roubar informações do banco de dados. O autor em questão relata que com o aumento da tecnologia da informação,

[37] <https://www.us-cert.gov/sites/default/files/publications/Practical-SQLi-Identification.pdf>

dispositivos móveis, facilidade de acesso à internet e a migração dos meios de pagamento para web, os roubos de dados de usuários se tornaram mais frequentes.

Inserindo esses comandos no navegador o atacante pode obter informações cruciais, dando ao mesmo acesso a dados restritos, ou oportunidades de ações maliciosas como tomar controle da base de dados e até prejudicar a integridade dos dados do usuário.

Dougherty[38] explica que pelo fato de utilizar instruções SQL, o SQL Injection não depende da linguagem de programação utilizada na arquitetura de um sistema. Assim caso o programa não estiver preparado, as chances de um ataque desses ser eficaz é muito alto. O autor alega que há métodos e ferramentas de segurança capazes de conter e revelar falhas na segurança. O autor ainda cita uma ferramenta chamada *SQL INJECTION ME* que é um plugin do navegador Firefox que acusa o uso de instruções SQL.

6.6 AUTENTICAÇÃO DE USUÁRIOS

Quando se fala em segurança, a primeira coisa que muitas pessoas pensam é sobre autenticação. Seja em sites de comercio, contas bancos, redes sociais, sistemas customizados, etc. É necessário que o usuário tenha um cadastro e que o mesmo possa confirmar sua identidade. Para tal fim existe a autenticação, o usuário confirma através de um login e uma senha correspondente a veracidade de sua identidade.

A importância da autenticação é a base da segurança da informação e o inicio de quase toda aplicação ou operação na web, se estendendo até diretrizes mais simples, como o login de um sistema operacional em computadores.

Este capítulo demonstrará as configurações chaves para se criar um login simples para um sistema validado por senha criptografada contra uma tabela no banco de dados e armazenando os dados na sessão, utilizando as linguagens de

[38] <https://www.us-cert.gov/sites/default/files/publications/Practical-SQLi-Identification.pdf>

programação PHP na versão 5.2.9 e o MySQL na versão 5.0.5. Tendo como base um artigo publicado no site linhadecódigo pelo autor Thiago Belem[39].

Belem[40] deixa entender que o primeiro passo é executar um código MySQL para criar a tabela de usuários contendo sete campos: id, nome, usuario, senha, niveis, ativo e cadastro.

```
CREATE TABLE IF NOT EXISTS `usuarios` (
        `id` INT(11) UNSIGNED NOT NULL AUTO_INCREMENT,
        `nome` VARCHAR( 50 ) NOT NULL ,
        `usuario` VARCHAR( 25 ) NOT NULL ,
        `senha` VARCHAR( 40 ) NOT NULL ,
        `email` VARCHAR( 100 ) NOT NULL ,
        `nivel` INT(1) UNSIGNED NOT NULL DEFAULT '1',
        `ativo` BOOL NOT NULL DEFAULT '1',
        `cadastro` DATETIME NOT NULL ,
        PRIMARY KEY (`id`),
        UNIQUE KEY `usuario` (`usuario`),
        KEY `nivel` (`nivel`)
    ) ENGINE=MyISAM ;
```

Com o comando exposto, será criado no banco de dados uma nova tabela chamada *usuarios* (não se usa acento na programação de BD) com os campos estabelecidos para receber as informações dos usuários, para gerar assim seus cadastros. O código acima também define a chave primaria(PRIMARY KEY) que é a principal forma de localização do usuário cadastrado na tabela e a chave única(UNIQUE KEY) que é a representação da tabela.

6.7 BOAS PRÁTICAS DE SEGURANÇA

6.7.1 Manter programas atualizados

A atualização de sistemas tem várias funções, como adicionar

[39] <http://www.linhadecodigo.com.br/artigo/3577/php-sistema-de-login-com-niveis-de-acesso.aspx>
[40] <http://www.linhadecodigo.com.br/artigo/3577/php-sistema-de-login-com-niveis-de-acesso.aspx>

funcionalidades, acompanhar plataformas, segurança, correção de bugs, etc. Todas abrangem segurança, porque se um sistema não funciona corretamente ele pode correr riscos. É extremamente importante atualizar periodicamente todos os programas, pois principalmente as atualizações sanam as falhas e vulnerabilidades da versão anterior, diminuindo consideravelmente a margem de riscos.

6.7.2 Senhas fortes

Como abordado no capitulo de Autenticação, as senhas são a confirmação do cadastro/login de um usuário em algum sistema. Usuários/clientes precisam escolher bem suas senhas, é comum rackers tentarem acessar contas de usuários através de datas de aniversários, apelidos, nomes e senhas fáceis de lembrar (12345, 123123). Senhas fortes devem conter bastante algarismos e de preferência alternados entre números, letras e caracteres especiais.

6.7.3 Criptografia de Senhas (MD5,RSA,ETC)

Belem (2009) [41] explica que a md5 é uma forma de criptografia muito comum e o mesmo é um algoritmo de hash de 128 bits. Ele explana que o md5 gera uma string alfa-numérica de 32 caracteres (128 bits), e que não importa a quantidade de caracteres que o usuário solicitar, o md5 gerado sempre vai ter 32 caracteres. Ele basicamente compara o md5 digitado no campo de senha com o que foi definido no banco, estando certo a autenticação é feita.

```
<?php
$string = “exemplo”;
$codificada = md5($string);
echo “Resultado da codificação usando md5: “ . $codificada;
```

Acima é mostrado um exemplo de como a senha fornecida pelo usuário é codificada, um parâmetro do tipo *string* recebe os caracteres do usuário, na linha seguinte é chamado o método md5 passando o parâmetro *string* e a ultima linha apenas mostra como ficou a codificação.

[41] <http://blog.thiagobelem.net/criptografia-no-php-usando-md5-sha1-e-base64/>

CONCLUSÃO

Atualmente a internet está muito presente na vida das pessoas, se tornando indispensável como ferramenta para sociedade. A tecnologia está em constante evolução e cada vez mais os dispositivos, softwares, hardwares, servidores, e assim por diante, vêm buscando melhorias, sempre um adaptando se adaptando ao outro.

O pilar que sustenta tudo isso é a segurança da informação, que protege todo isso de todos os tipos de riscos. Neste trabalho foram pesquisados métodos e sistemas de segurança para proteger servidores e aplicações web.

Após a pesquisa sobre configurações de segurança importantes para a proteção das aplicações e servidores web, foram feitos alguns testes e comprovado o funcionamento e a qualidade e também analisados metodologias que não puderam ser testadas consentindo com a hipótese H0:

- H0 – Não foi possível aplicar todos os testes por falta de infraestrutura indispensável para realizar-los.

Mesmo com a falta de equipamentos por pesquisas e alguns testes realizados, foi possível validar as hipóteses H1 e H2:

- É possível proteger Servidores Web nas nuvens de ataques configurando sistemas de segurança e utilizando boas práticas de segurança.

- Foi demonstrado técnicas e boas praticas para prevenir riscos e comprovado que o mesmo não necessita de tecnologias de alto custo e configurações muito complexas.

Foi concluído que é indispensável ter boas configurações de segurança para servidores e aplicações web, para proteger o mesmo de todo tipo de ameaça, de maneira simples e com baixo custo.

REFERÊNCIAS

ALBUQUERQUE, Fernando. *Acesso a base de dados intranet*. 4 f. Artigo Academico – Departamento de Ciência da Computação, Universidade de Brasília. Disponível em: <file:///B:/TCC/Referencias/aplicacao%20web%20acesso%20bd.pdf > Acesso em: 12 de outubro de 2015.

ALMEIDA, José Henrique Monteiro De. *PHP com MySQL*, 67 f. Apostila de PHP com MySQL. Disponível em: <http://www.cin.ufpe.br/~ags/2464_php_com_mysql.pdf> Acesso em 05 de novembro de 2015.

BELEM, Thiago. *Criptografia no PHP usando md5*. 2009. Disponível em: <http://blog.thiagobelem.net/criptografia-no-php-usando-md5-sha1-e-base64/> Acessado em 05 de novembro de 2015.

CASTRO, Simone Metello de Mattos; FERRAZ, Fernando Toledo. *A COMPUTAÇÃO EM NUVEM E A INDÚSTRIA DE TELECOMUNICAÇÕES*. 2013. 11 f. Revista Eletrônica Sistemas & Gestão. Disponível em: < http://www.revistasg.uff.br/index.php/sg/article/download/V8N4A4/SGV8N4A4> Acesso em: 13 de outubro de 2015.

COSTA, Johnatan da silva; SILVA, Jovina Da; CRUZ, Maria Auxiliadora Pereira da. *SEGURANÇA DE REDES DE COMPUTADORES NA INTERNET*. Revista Inova Ação, Teresina, v. 1, n. 2, art. 6, p. 77-88. Disponível em: <http://www4.fsanet.com.br/revista/index.php/inovaacao/article/download/480/pdf> Acesso em: 13 de outubro de 2015.

DOUGHERTY, Chad. *PRACTICAL INDENTIFICATION OF SQL INJECTION VULNERABILITIES*. US-CERT, Carnegie Mellon University, 2012. Disponível em: <https://www.us-cert.gov/sites/default/files/publications/Practical-SQLi-Identification.pdf>.

GOMES, Carina Nobre. *Estudo do Paradigma Computação em Nuvem*. 2012. 109 f. Área Departamental de Engenharia Eletrônica e Telecomunicações e de Computadores - INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA. Disponível em: <repositorio.ipl.pt/bitstream/10400.21/2375/1/Dissertação.pdf> Acesso em 12 de outubro de 2015.

OTERO, Lenin Ernesto Abadié. *Uma Arquitetura para a Implantação Automática de Serviços em Infraestruturas de Nuvem*. Cento de Informática UFPE, RECIFE, 2013.

ROCHA, Cláudio Ap.; PEDROSO, Edson Tessarini; JUNIOR, Eduardo Tarciso Soares. *O PROTOCOLO SECURE SOCKETS LAYER(SSL)*. Universidade Estadual de Campinas Instituto de Computação, Mestrado Profissional em Computação, 2003. Disponível em: <http://www.ic.unicamp.br/~rdahab/cursos/mp202/Welcome_files/trabalhos/SSL/texto/SSL_Texto.pdf>

SMALDONE, Javier. *Introducción a Secura Shell*. 2014. Disponível em: <http://es.tldp.org/Tutoriales/doc-ssh-intro/introduccion_ssh-0.2.pdf>.

DEVIN. Trabalhando com PHP e MySQL: Uma Introdução. Disponível em: <http://www.devin.com.br/intro_php/> Acesso em: 06 de novembro de 2015.

Fail2Ban Developers' Documentation. Disponível em: <https://media.readthedocs.org/pdf/fail2ban-kwirk/latest/fail2ban-kwirk.pdf>

LINHADECÓDIGO. PHP: Sistema de Códigos com Linhas de Acesso. Disponível em: <http://www.linhadecodigo.com.br/artigo/3577/php-sistema-de-login-com-niveis-de-acesso.aspx> Acesso em: 04 de novembro de 2015.

Method of security enhancement on VM access – fail 2 ban installation and setting. Disponível em: <https://en.ucloudbiz.olleh.com/manual/Security_fail2ban.pdf>

MUNHOZ, Felipe. Configurando o apache para utilizar SSL no Debian, 2011. Disponível em: <http://blog.felipemunhoz.com/configurando-o-apache-para-utilizar-ssl-no-debian/> Acesso em: 04 de novembro de 2015.

O que é CGI. Disponível em: <http://penta.ufrgs.br/edu/forms/cgi.html> Acesso em: 12 de outubro de 2015.

PHP+. Entendendo o que é URI, URN e URL, Disponível em: <http://www.phpmais.com/url-urn-e-uri-que-confusao/> Acesso em: 22 de outubro de 2015.

SIEDLER, Marcelo. Gerenciamento de Permissão em Base de dados. Disponível em: <http://187.7.106.14/marcelo/si/permissao.pdf> Acesso em: 12 de outubro de 2015.

SVERDLOV, Etel. Como criar um Novo Usuário e Conceder Permissões no MySQL, DigitalOcean, 2014. Disponível em: <https://www.digitalocean.com/community/tutorials/como-criar-um-novo-usuario-e-conceder-permissoes-no-mysql-pt> Acesso em: 12 de outubro de 2015.

Servidor Linux, guia prático, Cap. 6: Configurando servidores web. disponível em: <http://www.hardware.com.br/livros/servidores-linux/capitulo-configurando-servidores-web.html>. Acesso em: 24 de setembro 2015.

Servidor Linux, guia prático, Instalando o Apache. Disponível em: <http://www.hardware.com.br/livros/servidores-linux/instalando-servidor-lamp.html>. Acesso em: 24 de setembro de 2015.

Servidor Linux, guia prático, Instalando um servidor LAMP. Disponível em: <http://www.hardware.com.br/livros/servidores-linux/instalando-servidor-lamp.html>. Acesso em: 24 de setembro de 2015.

SIGNIFICADOS. Significados de HTML. Disponível em: <http://www.significados.com.br/html/> Acesso em: 12 de outubro de 2015>.

SSH COMMUNICATIONS SECURY CORP, *SSH Secure Shell for UNIX Servers Administrator's Guide*. Finland, 2000. Disponível em: <http://www.if.usp.br/pub/unix/security/ssh2-adminguide.pdf>.

VANDYKE SOFTWARE, An Overview of the Secure Shell(SSH). Disponível em: <https://www.vandyke.com/solutions/ssh_overview/ssh_overview.pdf>.

Wide Web, Volume One, 2014. Disponível em: <http://www.w3.org/TR/webarch/#acks> Acesso em: 12 de outubro de 2015.
Disponivel em: <http://digitool.fe.up.pt:1801/view/action/singleViewer.do?dvs=1445434789188~448&locale=pt_BR&metadata_object_ratio=25&side_by_side=false&VIEWER_URL=/view/action/singleViewer.do?&preferred_extension=pdf&DELIVERY_RULE_ID=5&frameId=1&usePid1=true&usePid2=true> Acesso em: 21 de outubro de 2015.