UNIVERSIDADE DO SAGRADO CORAÇÃO

BRUNO JACINTO PARPAGNOLI

CRIPTOGRAFIA SSL: UM COMPARATIVO E ANÁLISE DAS VANTAGENS NA UTILIZAÇÃO EM DOMÍNIOS WEB

BAURU
2015

BRUNO JACINTO PARPAGNOLI

CRIPTOGRAFIA SSL: UM COMPARATIVO E ANÁLISE DAS VANTAGENS NA UTILIZAÇÃO EM DOMÍNIOS WEB

Trabalho de conclusão de curso apresentado ao Centro de Ciência Exatas e Sociais Aplicadas da Universidade do Sagrado Coração, como parte dos requisitos para obtenção do título de bacharel em Ciência da Computação, sob a orientação do Prof. Me. Henrique Pachioni Martins.

BAURU
2015

Parpagnoli, Bruno Jacinto

P2579c

Criptografia SSL: um comparativo e análise das vantagens na utilização em domínios web / Bruno Jacinto Parpagnoli. -- 2015.

74f. : il.

Orientador: Prof. Me. Henrique Pachioni Martins.

Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) – Universidade do Sagrado Coração – Bauru – SP.

1. Internet. 2. Segurança. 3. SSL. 4. Criptografia. 5. Comparativo. I. Martins, Henrique Pachioni. II. Título.

BRUNO JACINTO PARPAGNOLI

CRIPTOGRAFIA SSL: UM COMPARATIVO E ANÁLISE DAS VANTAGENS NA UTILIZAÇÃO EM DOMÍNIOS WEB

Trabalho de Conclusão de Curso apresentado ao Centro de Ciências Exatas e Sociais Aplicadas da Universidade do Sagrado Coração como parte dos requisitos para obtenção do título de Bacharel em Ciência da Computação, sob a orientação do Prof. Me. Henrique Pachioni Martins.

Banca examinadora:

_____
Prof. Me. Henrique Pachioni Martins
Universidade do Sagrado Coração

_____
Prof. Esp. Alex Setolin Beirigo
Universidade do Sagrado Coração

_____
Prof. Me. Patrick Pedreira Silva
Universidade do Sagrado Coração

Bauru, 7 de dezembro de 2015.

AGRADECIMENTOS

Agradeço aos meus amigos, Rodrigo Cesar Rodrigues, Simone Moreto pelo auxilio na conclusão do mesmo e ao meu pai, Hugo Parpagnoli.

Agradeço ao meu professor orientador Henrique Pachioni Martins pela orientação e disponibilidade e aos professores Patrick Pedreira Silva e Alex Setolin Beirigo por participarem da banca e contribuírem com aprimoramento do mesmo.

E finalmente a bibliotecária Laudeceia Machado pela ajuda na correção.

RESUMO

Com a popularização da Internet ao final da década de 80 o e-commerce e o Internet Banking começam a ser utilizados em maior escala por pessoas do mundo todo, o que posteriormente gerou uma grande demanda nos meios de comunicação através das redes necessitando assim de segurança, visando a autenticidade e integridade das informações trafegadas nas redes. Este trabalho avaliou a performance de segurança de uma solução apresentada para suprir a demanda de segurança desde a década de 80, que ainda é muito utilizada nos dias de hoje. Portanto, utilizou o protocolo *Secure Socket Layer* (SSL), que utiliza funções de criptografia e *hash* para primeiramente criar uma conexão segura e, posteriormente utilizar-se da mesma para realizar a troca de informações de forma segura. Com a utilização de ferramentas, técnicas de invasão e fraude, foi realizado um comparativo das principais vulnerabilidades e vantagens encontradas que demonstrou a efetividade do protocolo em proteger as informações trocadas entre duas entidades em uma rede e do perigo de trafegar na rede sem a utilização do mesmo devido à ausência de recursos de segurança padrões.

Palavras-Chave: Internet. Segurança. SSL. Criptografia. Comparativo.

ABSTRACT

Due the popularization of the Internet at the end of the 80s, e-commerce and internet banking are beginning to be used on a larger scale by people around the world, which later generated a great demand in communication security across networks as well, aiming the authenticity and integrity of information trafficked in networks. This research aims to evaluate the safety performance of a given solution to meet the security demand from the 80s, which is still widely used nowadays. Therefore, it is intended to use the SSL (Security Socket Layer) protocol, which uses encryption and hash functions to first create a secure connection and subsequently used to perform the exchange in information securely. With the use of tools, hacking techniques and fraud, there has been made a comparison of key vulnerabilities and advantages found, proving the efectiveness of said protocol in protected and cypher information that has been through the web and showing the dangers in using the web without the SSL protocol.

Keywords: Internet, Security, SSL, Encryption, Comparison.

LISTA DE ILUSTRAÇÕES

LISTA DE ABREVIATURAS E SIGLAS

3DES - Triple DES

CERT – Centro de Estudo, Resposta e Tratamento de Incidentes de Segurança no Brasil

CNPJ – Cadastro Nacional da Pessoa Jurídica

CPF – Cadastro de Pessoa Física

DES – Data Encryption Standard

DoS – Denial of Service

e-CAC – Centro Virtual de Atendimento

e-CNPJ – Certificado Digital para Pessoa Jurídica

e-CPF – Certificado Digital para Pessoa Física

FTP – File Transfer Protocol

GLP – General Public License

HP – Hewlett-Packard

HTTP – Hyper Text Transfer Protocol

IBM – International Business Machines

IETF – Internet Engineering Task Force

IP – Internet Protocol

ISO – International Organization for Standardization

KEA – Keyphrase Extraction Algorithm

LAN – Local Area Network

MAC – Message Authentication Code

MAN – Message Authentication Code

MAN – Metropolitan Area Network

MDC – Modification Detection Code

MD5 – Message Digest 5

NF-e – Nota Fiscal Eletrônica

NIST – National Institute of Standards and Technology

NRC – National Research Council

NSA – National Security Agency

OSI – Open System Interconnection

RSA – Rivest-Shamir-Adleman

SHA – Secure Hash Algorithm

SPB – Sistema de Pagamento Brasileiro

SMTP – Simple Mail Transfer Protocol

SSL – Secure Socket Layer

TCP – Transmission Control Protocol

VM – Virtual Machine

VMM – Virtual Machine Monitor

WAN – Wide Area Network

SUMÁRIO

1 INTRODUÇÃO

Com sua popularização ao final dos anos 80, a Internet, que se tratava apenas de um sistema de informações baseada em páginas de *hypertexto,* começa a ser utilizada por grandes empresas para o comércio de seus produtos, o que ficou conhecido como *e-commerce*. Mais tarde passou também a ser utilizada para o acesso de contas bancárias, através de portais dos grandes bancos (*Internet Banking)*.

Segundo informações publicadas pela Exame.com da Editora Abril S. A. no ano de 2015, o *e-commerce* brasileiro registrou um crescimento de 24% em 2014 comparado com o ano de 2013, com faturamento total de R$35,8 bilhões. A tendência desses valores é aumentar, dado que a *Internet* está sendo acessada cada vez mais e, consequentemente, o *e-commerce* e *o Internet Banking* serão utilizados em maior escala. (MORENO, 2015).

Sabendo-se disso, tornou-se imprescindível a criação de meios que possibilitem a comunicação entre duas entidades através da rede, em total segurança, visando a autenticidade e integridade das informações trocadas.

Dentre várias soluções apresentadas para garantir tal segurança, será destacado neste trabalho o Procotolo *Secure Socket Layer* (SSL), criado pela Netscape Corp. que atua entre as camadas de aplicação e transporte dentro do protocolo TCP/IP (protocolo padrão dos *browsers* mais utilizados atualmente*),* primeiramente criando uma conexão segura e autenticada entre duas entidades e, posteriormente, responsável pelo tráfego das informações contidas na mesma. Para suprir essa demanda de segurança, o SSL utiliza inúmeros tipos de criptografias (do grego “*kryptos*”, secreto e “*grapho”*, escrita), uma técnica da área da segurança da informação que consiste em escrever secretamente, em outras palavras, comunicar-se de maneira em que apenas o remetente e o destinatário sejam capazes de ler o conteúdo original da mensagem, tendo-se em mente que somente ambos saibam a chave secreta (informação necessária para a encriptação e desencriptação da mensagem). Em adição, este protocolo utiliza funções de criptografia destinadas a garantir a autenticidade da mensagem, as funções *hash*, que resultam em uma mensagem única, de tamanho fixo e independente do tamanho original, que será anexada junto a mensagem original e comparada posteriormente pelo destinatário, onde o mesmo também aplicará uma função *hash* na mensagem recebida. Caso os

valores sejam idênticos, pode se dizer que a mensagem é autêntica, caso contrário significa que a mesma foi adulterada por intermédio de uma terceira entidade.

Sabe-se que o número de fraudes virtuais vem aumentando em larga escala e que segundo informações publicadas no ano de 2015 pelo Centro de Estudo, Resposta e Tratamento de Incidentes de Segurança no Brasil (CERT.br) o total de incidentes reportados no ano de 2014 foi de 1047031 (um milhão quarenta e sete mil e trinta e um), três vezes maior que no ano de 2013. (CENTRO DE ESTUDO, RESPOSTA E TRATAMENTO DE INCIDENTES DE SEGURANÇA NO BRASIL, 2015)

Com base nessas informações, este trabalho tem como foco analisar o desempenho de segurança de domínios web que possuem o protocolo SSL a diferentes tipos de técnicas de invasão e fraude de informações, posteriormente, realizando os mesmos testes com domínios que não possuem essa segurança. Ao final, os resultados obtidos serão usados para gerar um comparativo, apontando as principais vulnerabilidades e vantagens de sua utilização.

2 OBJETIVOS

2.1 OBJETIVO GERAL

Realizar um comparativo de domínios Web com e sem a segurança do protocolo Secure Socket Layer (SSL) e demonstrar as vantagens de segurança encontradas

2.2 OBJETIVOS ESPECÍFICOS

a) Levantar conhecimento sobre o funcionamento do protocolo SSL e criptografia;
b) Implementar um ambiente de protótipo para a realização de testes;
c) Realizar uma simulação de ataques utilizando a ferramenta Wireshark encontrada na VM Kali Linux a dois domínios locais, onde apenas um possuirá o procotolo SSL;
d) Estabelecer um comparativo apontando as principais vulnerabilidades encontradas;

3 REDE DE COMPUTADORES

Uma rede de computadores consiste em dois ou mais computadores conectados seja via fio de cobre, fibra ótica, ondas de rádio e também via satélite, que possuem a finalidade de compartilhar informações, dados, recursos, serviços e outras. Existem classificações para as mesmas, sendo que as mais comumente utilizadas são: *Local Area Network* (LAN), "*Metropolitan Area Network"* (MAN) e "*Wide Area Network"* (WAN). (SOARES, 1995).

a) LANs (Local Area Network): Redes de menor porte e privadas, geralmente contidas em um único prédio ou em um campus universitário com geralmente poucos quilômetros de extensão. Usadas para conectar estações de trabalhos ou computadores pessoais, permitindo a troca de informações e recursos. Diferem-se dos demais tipos de redes em seu tamanho, tecnologia de transmissão e topologia (TANEMBAUM, 1997).
b) MANs (Local Area Network): Uma rede metropolitana é, na verdade, uma versão ampliada das LANs e possue tecnologias semelhantes a mesma. Uma MAN pode abranger desde escritórios vizinhos a uma cidade inteira podendo também ser privada ou publica. Possuem a capacidade de transmitir dados de voz, sendo assim associada a redes de televisão. Sua estrutura simplificada ocorre devido a ausência de elementos de comutação, possuindo apenas dois cabos. (TANEMBAUM, 1997).
c) WANs (Wide Area Network): São as redes que abrangem uma determinada área geográfica, ou seja, interligando países, continentes e outros através de satélites e circuitos integrados.

> Uma rede geograficamente distribuída, ou WAN, abrange uma grande área geográfica, com frequência um país ou continente. Ela contém um conjunto de máquinas cuja finalidade é executar os programas (ou seja, as aplicações) do usuário. Seguiremos a tradição e chamaremos essas máquinas de host. O termo end system também é utilizado na literatura específica. Os hosts estão conectados por uma sub-rede de comunicação ou, simplificando, uma sub-rede. A tarefa da sub-rede é transportar mensagens de um host para outro, exatamente como um sistema de telefonia transporta as palavras da pessoa que fala a que ouve. Essa estrutura de rede é altamente simplificada, pois separa os aspectos da comunicação pertencentes à rede (a sub-rede) dos aspectos de aplicação (os hosts) (TANEMBAUM, 1997, p. 12).

3.1 PROTOCOLOS E PADRÕES

Assim como no relacionamento humano existem protocolos de convivência, desde um simples “com licença ou, por favor” para que se tenha um retorno desejável de comunicação, as máquinas também possuem protocolos de software e hardware, afim de que a autenticidade e integridade de informações sejam satisfeitas. Segundo Forouzan (2008), os protocolos são conjuntos de regras que controlam as comunicações de dados. Um protocolo define o que é comunicado, como isso é comunicado e quando deve ser comunicado.

A Cisco (2013) frisa a descrição de comunicações utilizando-se de camadas, as quais possuem protocolos específicos, sendo as camadas superiores focadas no conteúdo da mensagem e as inferiores, relacionadas a movimentação e transportes de dados.

Tais protocolos são divididos em duas categorias: de facto e de jure. O padrão de facto (“de fato”, em português), segundo Tanenbaum (2003) são aqueles que se consagram naturalmente, sem nenhum plano formal, enquanto padrões de jure (“por lei” em português), “são padrões legais e formais adotados por uma instituição de padronização legalizada”.

Forouzan (2008, p. 19), também faz uma definição sobre os padrões, sendo que estes

> [...] são essenciais na criação e na manutenção de um mercado aberto e competitivo para fabricantes de equipamentos e na garantia de interoperabilidade nacional e internacional de dados e de tecnologia de telecomunicações e processos.

Dentre os protocolos destacam-se dois modelos que servem como referência: *Open Systems Interconnection* (OSI) e o *Transmission Control Protocol / Internet Protocol (*TCP/IP).

3.2 MODELO OSI

Este modelo criado pela ISO é composto por sete camadas, sendo que cada uma delas recebe seus dados de sua camada superior, incluindo seus dados e assim enviando para a camada abaixo, repetindo este mesmo processo até que se chegue à camada mais inferior onde os dados se encontrarão prontos para serem transmitidos (ULBRICH; VALLE, 2004). A Figura 1 exemplifica as camadas OSI, que são: aplicação, apresentação, transporte, rede, enlace e física.

“A arquitetura de segurança OSI enfoca ataques, mecanismos e serviços de segurança” (STALLING, 2008, p. 5). Abaixo, de acordo com a Figura 1, a estrutura das camadas citadas.

Figura 1 - Modelo OSI

Fonte: Tanenbaum (2003).

a) camada física: responsável pela transmissão de bits por um canal de comunicação;
b) camada de enlace: transforma a transmissão bruta em uma linha livre de erros originados pela transmissão;
c) camada de rede: responsável pelo controle e operação da sub-rede;

d) camada de transporte: tem a finalidade de receber dados de uma camada superior, dividir em unidades, repassar essas e assegurar a integridade até outro ponto;
e) camada de sessão: permite o relacionamento de diferentes usuários e máquinas através de sessões;
f) camada de apresentação: gerencia a estrutura de dados e permite a relação a um nível mais alto, isto é, sintaxe e semântica;
g) camada de aplicação: é responsável pelas interações com os usuários, é comumente utilizado no HTTP (*Hyper Text Transfer Protocol*).

3.3 MODELO TCP/IP

O modelo TCP/IP possui apenas quatro camadas segundo Tanenbaum (2003), são elas: aplicação, transporte, inter-redes e host/rede. Este modelo é altamente difundido em todas redes de computadores geograficamente distribuídas, ou seja, internet mundial. Abaixo a Figura 2 exemplifica as camadas:

Figura 2 - Modelo TCP/IP

Fonte: Tanenbaum (2003)

a) camada de inter-redes: é baseada em uma comutação de pacotes que se relacionam sem interconexões;
b) camada de transporte: tem o objetivo de permitir que a origem e o destino inter-relacionem entre si, isto é, comuniquem-se;

c) camada de aplicação: é nela que estão estabelecidos todos os protocolos de nível mais alto (FTP–*File Transfer Protocol*, SMTP–*Simple Mail Transfer Protocol*, HTTP–*HyperText Transfer Protocol*, etc.);
d) camada de host / rede: onde host se conecta com a rede permitindo o envio e a recepção de pacotes IP (*Internet Protocol*).

> Os modelos de referência OSI e TCP/IP têm muito em comum. Os dois se baseiam no conceito de uma pilha de protocolos independentes. Além disso, as camadas têm praticamente as mesmas funções. Em ambos os modelos, por exemplo, estão presentes as camadas que englobam até a camada de transporte. Nesses modelos, são oferecidos aos processos que desejam se comunicar um serviço de transporte fim a fim independente do tipo de rede que está sendo usado. Essas camadas formam o provedor de transporte. Mais uma vez ambos os modelos, as camadas acima da camada de transporte dizem respeito aos usuários orientados à aplicação de serviço de transporte. (TANEMBAUM, 1997, p. 42).

3.4 INTERNET

A internet pode ser comparada à uma gigantesca WAN, que permite o compartilhamento de milhões de dispositivos.

Segundo Tanembaum (2003), a internet é um conjunto de redes interconectadas por milhões de dispositivos computacionais, dispositivos móveis, computadores pessoais e servidores.

> Sistemas finais são conectados por links (enlaces de comunicação) que podem transmitir dados em diferentes taxas de transmissão. Essas sequências de enlaces de comunicação e comutadores de pacotes transitam desde sua origem até o respectivo destino e são conhecidos como rotas ou caminhos. Normalmente esses tipos de serviços são fornecidos por Provedores de Serviços de Internet (Telefônica, Embratel, NET, etc.) que são meio entre a Internet e os sistemas finais.
> Estes sistemas finais possuem dispositivos e equipamentos que executam protocolos que controlam a recepção e envio de dados na internet, e normalmente utilizam o TCP/IP e são padronizados pela IETF (Internet Engineering Task Force). (BERNARDINELLI; AMERICANA, 2010, p. 22).

4 SEGURANÇA

A segurança é um fator importantíssimo na área de computação, aplicando-se desde a parte física até a lógica. Em uma rede cabeada, o único modo de ser quebrada é por algum dispositivo que esteja conectado á rede, mas no caso das redes não cabeadas como a informação trafega pelo ar, suas vulnerabilidades aumentam e assim a necessidade de uma segurança confiável é maior. As principais características que classificam os tipos de segurança são:

a) confidencialidade:

> Somente o remetente e o destinatário pretendido devem poder entender o conteúdo da mensagem transmitida. O fato de abelhudos poderem interceptar a mensagem exige, necessariamente, que seja cifrada de alguma maneira (que seus dados sejam disfarçados) para impedir que uma mensagem interceptada seja decifrada (entendida) por um interceptador. Esse aspecto de confidencialidade é, provavelmente, o significado mais comumente percebido na expressão comunicação seguro. Note, contudo, que essa não é apenas uma definição limitada de comunicação segura, mas também uma definição bastante restrita de confidencialidade. (KUROSE; ROSS, 2006, p. 513).

b) integridade:

> “Mesmo que o remetente e o destinatário consigam se autenticar reciprocamente, eles também querem assegurar que o conteúdo de sua comunicação não seja alterado, por acidente ou por má intenção, durante a transmissão. Extensões das técnicas de soma de verificação que encontramos em protocolos de transporte e de enlace confiáveis podem ser utilizadas para proporcionar integridade à mensagem” (KUROSE; ROSS, 2006, p. 513).

c) autenticação:

> O remetente e o destinatário precisam confirmar a identidade da outra parte envolvida na comunicação – confirmar que a outra parte realmente é quem alega ser. A comunicação pessoal entre seres humanos resolve facilmente esse problema por reconhecimento visual. Quando entidades comunicantes trocam mensagens por um meio pelo qual não podem ver a outra parte, a autenticação não é assim tão simples. Por que, por exemplo, você deveria acreditar que o e-mail veio a um amigo seu realmente veio daquele amigo? Se alguém o chama ao telefone dizendo ser de seu banco e perguntando qual é o número de sua conta, sua senha e saldo bancário, alegando finalidades de verificação, você daria essas informações? Esperamos que não. (KUROSE; ROSS, 2006, p. 513).

d) disponibilidade:

> A necessidade imperiosa de segurança na rede ficou dolorosamente óbvia nos últimos anos devido a numerosos ataques de recusa de serviço (denial of service – DoS) que inutilizaram uma rede, um hospedeiro, ou qualquer outro componente da infraestrutura de rede, para seus usuários legítimos; o mais notório desses ataques DoS talvez tenha sido o cometido contra os sites Web de inúmeras empresas de alta visibilidade. Portanto, um requisito fundamental para comunicação segura deve ser, antes de mais nada, que ela possa ocorrer – que os "bandidos" não possam ser legítimos, enquanto outros não levam, naturalmente, á noção de controle de acesso, para garantir que entidades que procuram obter acesso a recursos possam fazê-lo somente se tiverem os direitos de acesso apropriados e realizarem seus acessos de uma maneira bem definida. (KUROSE; ROSS, 2006, p. 514).

Em suma, essas características dependem necessariamente umas das outras, assim dizendo pode-se compará-las a uma estrutura onde cada uma delas seria uma das quatro colunas sustentadoras, sendo assim caso uma delas seja fraca, todo o contexto de segurança envolvido pode ser desestruturado.

4.1 CRIPTOGRAFIA

A National Research Council (NRC) cita que a criptografia hoje é, provavelmente, o aspecto mais importante da segurança de comunicações e está se tornando cada vez mais importante como um componente básico para a segurança do computador.

> A palavra criptografia é originária dos termos gregos kryptós, que quer dizer oculto, e graph, escrever. Em dicionários da língua portuguesa, pode-se encontrar a seguinte definição para palavra criptografia: escrita secreta por meio de abreviaturas ou de sinais convencionados de modo a preservar a confidencialidade da informação. (AQUINO JUNIOR et al., 2008, p. 13).

A criptografia é um ramo da matemática que faz o estudo de princípios e técnicas pelas quais uma informação clara pode ser transformada em uma informação ilegível e que apenas através de uma "chave" pode passar pelo processo reverso, tendo como objetivo de que apenas emissor e o receptor da mensagem possam ter acesso à informação original.

Como dito, o resultado de uma criptografia gera o chamado de texto cifrado, que é feito através de algoritmos matemáticos.

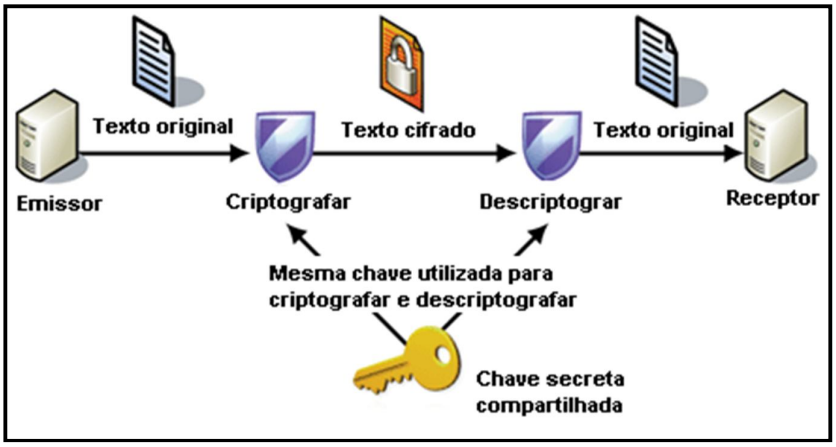
Stallings (2008) frisa que, com certeza, a ferramenta automatizada mais importante para a segurança da rede e das comunicações é a criptografia. Duas formas de criptografia são usadas normalmente: criptografia convencional, ou simétrica, e a por chave pública, ou assimétrica.

4.1.1 Criptografia Simétrica

Também conhecida como “Criptografia de Chave Secreta”, é concebida através da codificação e decodificação com a utilização de apenas uma chave, a “chave secreta”, ou seja, apenas os portadores da mesma terão acesso à informação original. Os algoritmos desenvolvidos para este tipo de criptografia são baseados em substituição, onde cada elemento do texto plano é substituído por outro, e transposição, que é a reorganização dos elementos do texto plano (AQUINO JUNIOR et al., 2008).

Stallings (2008), diz que a criptografia simétrica é baseada na utilização de cinco elementos, que são: texto plano, algoritmo de criptografia, chave secreta, o texto cifrado e o algoritmo de decriptografia. A Figura 3 mostra o fluxo de uma criptografia simétrica.

Figura 3 - Fluxo de criptografia simétrica



Fonte: Sampaio (2013).

4.1.2 Criptografia Assimétrica

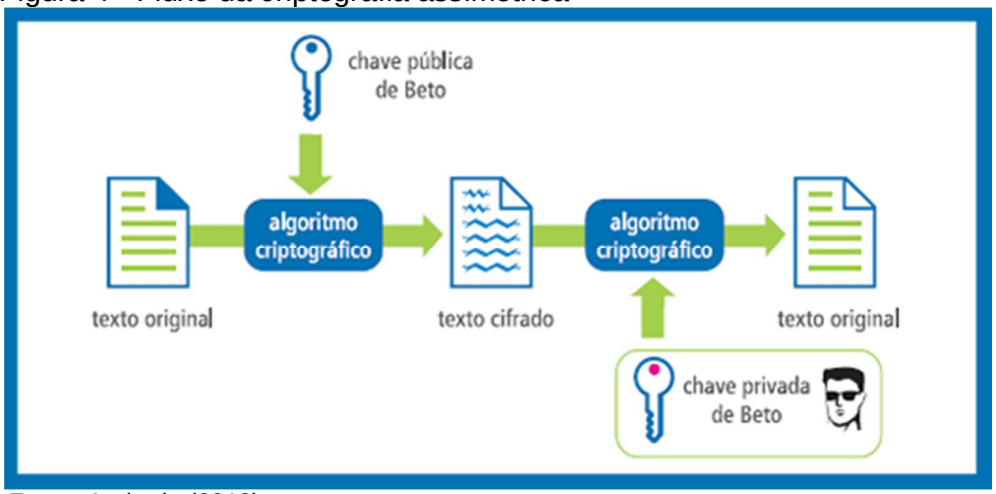
Também conhecida como criptografia de chave pública, esta se caracteriza pela utilização de algoritmos que geram duas chaves. Stallings (2008), afirma que cada usuário gera um par de chaves, dentre as quais uma delas se torna pública, enquanto a outra permanece privada.

Neste caso, para que um usuário A se comunique com um usuário B, um tem que possuir a chave pública do outro. Após receber a mensagem cifrada o respectivo destinatário usa uma chave privada própria que somente ele deve saber, sendo esse o único meio de decifrar a mensagem recebida.

> Com essa técnica, todos os participantes têm acesso ás chaves públicas. As chaves privadas são geradas localmente por cada participante e, portanto, nunca precisam ser distribuídas. Desde que a chave privada de um usuário permaneça protegida e secreta, a comunicação que chega está protegida. (STALLING, 2008, p. 183).

A Figura 4 demonstra o fluxo da informação e seus processos de criptografia.

Figura 4 - Fluxo da criptografia assimétrica



Fonte: Andrade (2013).

4.1.3 Principais Algoritmos de Criptografia e Autenticação

Existem inúmeros algoritmos famosos, cada um deles possui suas finalidades e propósitos, porém como o foco deste trabalho é o protocolo SSL serão exemplificados apenas os que fazem referência e o uso do mesmo.

4.1.3.1 Algoritmos De Criptografia

a) DES

Utiliza duas entradas de 64 bits, texto plano e chave. Usa o mesmo algoritmo para criptografria e descriptografia.

Segundo Peterson e Davie (2004), apesar de a chave ser de 64 bits, somente 56 bits são utilizados para o processo em si, pois a cada 8 bytes 1 byte é o byte de paridade.

Este processo consiste em três fases que devem ser realizados, os quais estão representadas na Figura 5. Primeiramente os 64 *bits* do texto plano de entrada são permutados, passando então para a segunda fase.

Após os *bits* serem permutados, os *bits* utilizam uma cifra, conhecida como Feistel, durante dezesseis rodadas. Cada rodada neste algoritmo será representada pela letra “i”.

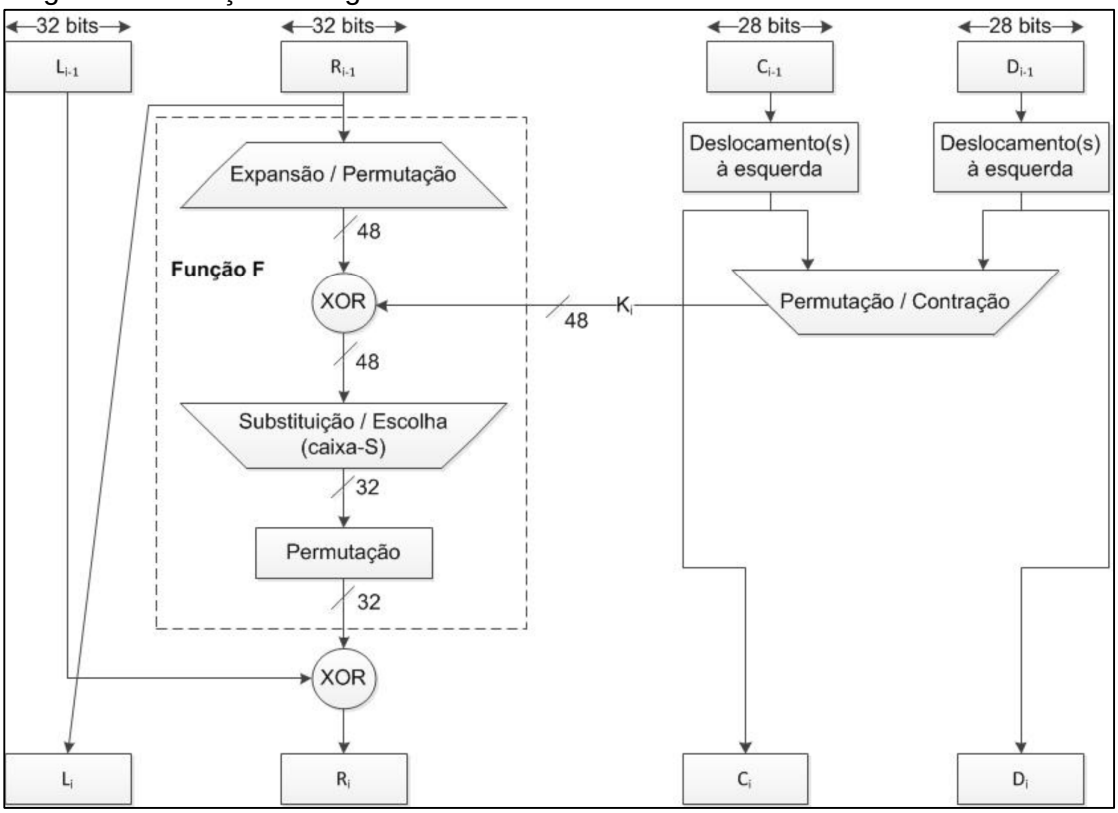
De acordo com Erickson (2009), os *bits* são divididos em dois grupos L (*Left*, ou esquerda) e R (*Right*, ou direita), e a cada rodada o novo L é definido para ser igual à antiga metade direita (Ri-1), e o novo R é definido para ser igual à antiga metade esquerda (Li-1) com a operação lógica XOR (OU exclusivo) sobre a saída de uma função F, que usa a antiga metade direita (Ri-1) e a subchave para a volta chamada de Ki.

A geração desta subchave Ki também se inicia com uma permutação nos 56 *bits* iniciais e logo após é dividida em grupos de 28 *bits,* e um deslocamento esquerdo circular a estes de um ou dois *bits*, dependendo da rodada. Logo após o deslocamento, é feito uma permutação e contração, as quais não serão tratadas em detalhes neste trabalho, para retornar um valor de 48 *bits* que serve como entrada para a função F. (STALLINGS, 2008).

Um ponto importante a ser declarado é que a função F precisa que Ri-1 também tenha 48 *bits* para que possa ser calculado o XOR entre estes *bits* e os *bits* da subchave. Para isso também é feito uma expansão e permutação dos *bits* contidos em Ri-1 e para que então seja realizado o XOR. Este resultado será de 48 *bits*, e deve ser reduzido aos 32 *bits* utilizados tanto para L quanto para R. Para isto, é utilizado algo chamado de caixa de substituição (ou caixa-S), a qual reduz cada pedaço de 6 *bits* para 4 *bits*, retornando assim os 32 *bits* necessários para cada rodada.

Por fim, ao término da última rodada é realizada uma permutação inversa à inicial, resultando assim no texto cifrado. A Figura 5 a seguir demonstra como é a arquitetura da criptografia DES.

Figura 5 – Iteração do algoritmo DES



Fonte: Stallings (2008).

b) 3DES

Conclusões feitas sobre o DES resultaram que este não era totalmente seguro e que demoraria um tempo considerado curto para ser desvendado. A fraqueza conhecida do DES era o seu tamanho de chave de 56 *bits*, de acordo com Erickson (2009). Sendo assim, surgiu o 3DES, criptografando os dados três vezes. Peterson e Davie (2004) relatam que “isso pode ser feito com duas ou três chaves separadas.”.

c) KEA

O Keyphrase Extraction Algorithm (KEA), proposto por Witten (1999), é um algoritmo para extrair automaticamente palavras-chave de textos da Língua Inglesa.

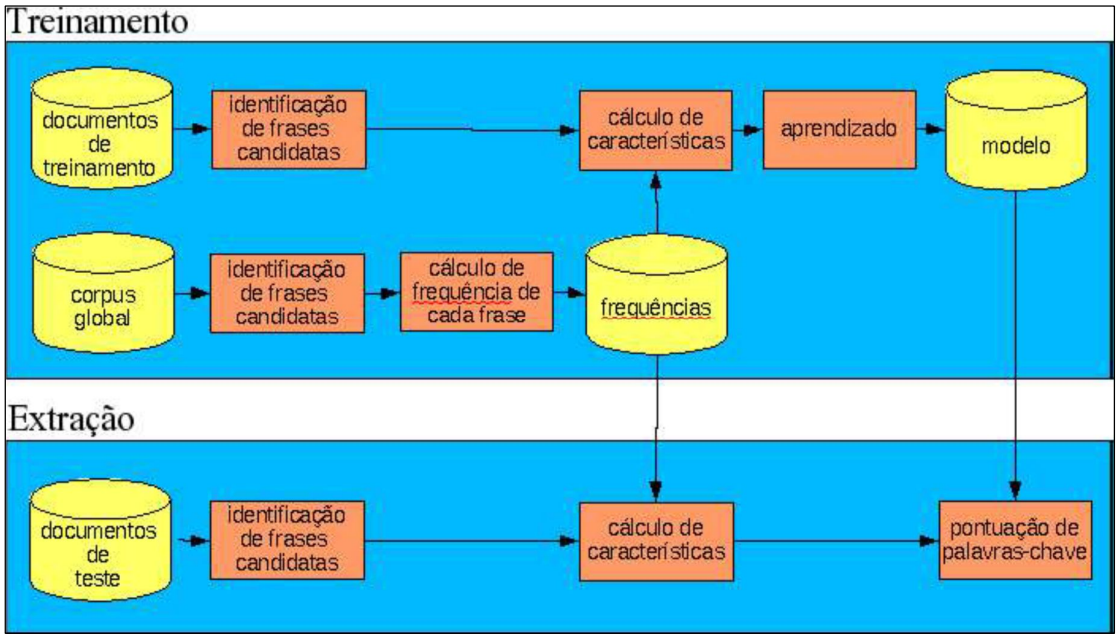
Para isto, ele identifica frases candidatas usando métodos de análise éxica, calculando então características para cada frase candidata e usando a técnica Naive Bayes de Aprendizado de Máquina para treinamento e extração automática de palavras-chave.

A técnica de Aprendizado de Máquina constrói um modelo de predição usando documentos de treinamento com palavras-chave conhecidas e, então, usa o modelo construído para encontrar palavras-chave em novos documentos, ou seja, em documentos cujas palavras-chave não são conhecidas. Uma forma de avaliar a eficácia do KEA é determinar quantas palavras-chave determinadas pelo autor de cada documento são identificadas corretamente.

De acordo com Witten (1999), as palavras-chave determinadas pelo autor e as palavras-chave extraídas automaticamente são bastante similares, mas não é muito difícil adivinhar quais são as do autor. Pode-se verificar que o KEA escolhe diversas palavras chave adequadas, mas também escolhe algumas que são improváveis de um autor utilizar. Apesar destas anomalias, as listas extraídas automaticamente fornecem uma descrição apropriada dos textos. No caso em que nenhuma palavra-chave especificada pelo autor estiver disponível, as escolhas do KEA poderiam ser um recurso valioso para a localização de documentos.

O algoritmo KEA tem dois estágios: treinamento, quando é criado um modelo para identificar palavras-chave usando documentos de treinamento em que as palavras chave do autor são conhecidas; e extração, quando são extraídas palavras-chave de um novo documento usando o modelo de treinamento construído anteriormente.

Figura 6 – Fases do Algoritmo KEA



Fonte: Dias, Malheiros (2005)

O processo todo é ilustrado na Figura 6. Ambos os estágios escolhem um conjunto de frases candidatas a partir da entrada dos documentos, e então calculam os valores de certos atributos, chamados de características, para cada frase candidata.

A fase do KEA que depende da linguagem dos documentos é a denominada *identificação de frases candidatas*, quando são realizados três passos: limpeza do texto de entrada, identificação de frases candidatas e radicalização das frases selecionadas.

A limpeza basicamente remove os símbolos e pontuação das frases candidatas. O algoritmo então considera todas as subsequências em cada linha do texto e determina quais destas são frases candidatas adequadas. Em particular, frases candidatas não podem começar ou terminar com uma *stopword*. Para a radicalização, o KEA usa o método iterativo de Lovins (1968), utilizando este radicalizador para descartar sufixos, repetindo o processo sobre o radical resultante até que não exista mais mudança.

O passo de treinamento utiliza um conjunto de documentos para treinamento em que as palavras-chave são conhecidas. Para cada documento de treinamento, frases candidatas são identificadas e marcadas com sendo das classes “palavra-chave” ou “não palavra-chave”, usando as verdadeiras palavras-chave do

documento. O esquema então gera um modelo que prediz a classe de uma dada palavra.

O passo de extração permite selecionar palavras-chave de um novo documento. Para tanto, o KEA determina frases candidatas deste documento e aplica o modelo construído durante o treinamento. Este modelo determina a probabilidade global de cada frase candidata ser uma palavra-chave, e então uma operação de pós-processamento seleciona o melhor conjunto de palavras-chave.

d) RSA

De acordo com Stallings (2008), foi um método muito interessante descoberto por um grupo de pesquisadores do MIT e é conhecido pelas iniciais dos três estudiosos que o criaram (Rivest, Shamir, Adleman): RSA. Ele sobreviveu a todas as tentativas de rompimento por mais de um quarto de século e é considerado um algoritmo muito forte. Grande parte da segurança prática se baseia nele. Sua principal desvantagem é exigir chaves de pelo menos 1024 bits para manter um bom nível de segurança (em comparação com 128 bits para os algoritmos de chave simétrica), e isso o torna bastante lento.

A segurança do método se baseia na dificuldade de fatorar números extensos. Se pudesse fatorar o valor N (publicamente conhecido), o criptoanalista poderia então encontrar P e Q e, a partir desses, encontrar Z. Com o conhecimento de Z, é possível encontrar D utilizando-se o algoritmo de Euclides Stallings (2008).

Stallings (2008), ainda faz um comentário sobre o algoritmo RSA:

> De acordo com Rivest e seus colegas, a fatoração de um número de 500 dígitos requer 1025 anos, usando-se a força bruta. Nesse caso, eles pressupõem o melhor algoritmo conhecido e um computador com um tempo por instrução de 1 s. Mesmo que os computadores continuem a se tornar cada vez mais rápidos na proporção de uma ordem de magnitude por década, ainda se passarão séculos até que a fatoração de um número de 500 dígitos se torne viável e, nesse tempo, nossos descendentes poderão simplesmente escolher p e q ainda maiores. (STALLINGS, 2008, p. 566).

4.1.3.2 Algoritmos de Autenticação

Esses algoritmos se diferem dos citados previamente, pois são utilizados na autenticação de uma determinada informação e não na criptografia da informação, Stallings (2008)**,** afirma que, “a autenticação de mensagem é um mecanismo ou serviço utilizado para verificar a integridade de uma mensagem”.

As técnicas mais convencionais para a autenticação de mensagem são: código de autenticação de mensagem (MAC – Message Authentication Code), e as funções de *hash,* onde seu produto final chamado de MDC (Modification Detection Code, ou “Código de detecção de modificações”) é capaz de detectar qualquer modificação na mensagem. (FOROUZAN, 2008).

MAC é uma técnica de autenticação de mensagem que envolve o uso de uma chave secreta para gerar um pequeno bloco de dados, conhecido como código de autenticação de mensagem, que é anexado a mensagem e enviada ao destinatário. Este realizará o mesmo procedimento que ao final comparando com resultado MAC do remetente caso ambas forem iguais, segundo Stallings (2008) é possível afirmar que a mensagem se encontra íntegra, e que a mensagem veio do remetente declarado, isso tendo-se em mente que apenas ambos sabiam a chave secreta.

O *hash* é gerado a partir de uma função *hash*, também chamado de resumo de mensagem (*message digest*), o mesmo não possui chaves. Segundo Kurose e Ross (2006), os algoritmos de resumo de mensagem recebem uma mensagem m, de comprimento arbitrário, posteriormente calculando-se uma “impressão digital” dos dados, de comprimento fixo, conhecida como resumo da mensagem H(m).

a) MD5

Message Digest 5 (MD5) possui um processamento em blocos de 512 *bits* similar ao algoritmo SHA-1, e saída de 128 *bits* sendo que sua entrada suporta um comprimento arbitrário. Tanenbaum (2003) explica que o processo se inicia expandindo o tamanho da mensagem com bits 0, com exceção do primeiro bit de expansão até que se chegue a um de 512. Sendo assim, o tamanho do último bloco contendo 448 bits é preenchido com 64 *bits* resultantes do tamanho em *bits* da mensagem original. Caso o tamanho da mensagem não possa ser representado em

64 bits, apenas os *bits* menos significativos serão considerados, segundo (MORENO et al. 2005).

Tanenbaum (2003), explica o processo a cada iteração do algoritmo citado:

> Em cada rodada, um bloco de entrada de 512 bits é extraído e colocado no buffer de 128 bits. Para que os cálculos sejam feitos com maior precisão, também é incluída uma tabela criada a partir da função de seno. O objetivo da utilização de uma função conhecida, como o seno, é evitar qualquer suspeita de que o projetista tenha criado uma armadilha secreta para seu próprio uso, e não pelo fato dessa função ser mais aleatória do que um gerador de números aleatórios. A recusa da IBM em revelar os princípios em que se baseava o projeto das caixas S do DES criou muita especulação sobre esses artifícios secretos. Há quatro rodadas para cada bloco de entrada. Esse processo continua até que todos os blocos de entrada tenham sido consumidos. O conteúdo do buffer de 128 bits forma o sumário de mensagens. (TANENBAUM, 2003, p. 571).

b) SHA-1

Os algoritmos da família SHA foram desenvolvidos pela NSA (National Security Agency), segundo Kurose e Ross (2006), se tornou um padrão federal norte-americano, e é exigido quando aplicações de nível federal precisam de um resumo de mensagem seguro.

O primeiro membro da família foi chamado de SHA, no entanto, frequentemente chamado de SHA-0. Em 1994, foi publicada uma revisão de seu algoritmo, chamado de SHA-1. De acordo com Stallings (2008), existem outras três variantes do mesmo: SHA-256, SHA-384 e SHA-512, sendo que cada um retorna o número em bits correspondente a seus nomes.

A NIST (National Insitute of Standards and Technology), no entanto vem tentando tirar o SHA-1 do mercado, pois o mesmo retorna apenas valores de 160 bits, e por esse motivo, vulnerabilidades estão sendo encontradas desde 2005.

Similar ao algoritmo MD5, é realizado uma expansão de *bits* até que o tamanho seja múltiplo de 512 que posteriormente, segundo Tanembaum (2003, p. 572), “um número de 64 bits contendo o tamanho da mensagem antes do preenchimento é submetido a uma operação OR nos 64 bits de baixa ordem”.

Cinco variáveis de 32 *bits* são criadas (H0 a H4), sendo constantes especificadas no padrão, onde o *hash* é acumulado. Posteriormente os blocos de 512 *bits* são processados, cada bloco de 16 palavras (*words* – unidade que contém 32 *bits*) é copiado para um vetor de 80 posições, os 64 *bits* restantes são

preenchidos utilizando-se uma fórmula específica que faz a rotação circular à esquerda (TANENBAUM, 2003).

Mais cinco variáveis são criadas (A, B, C, D e E), contendo os valores de H0 a H4 respectivamente. Oitenta iterações de cálculos são realizadas utilizando funções diferentes que mudam a cada 20 iterações. Ao término dos cálculos, os valores de A até E são somados aos valores de H0 a H4.

Ao final das operações lógicas neste bloco, o próximo é preparado para repetir o mesmo processo, com vetores de palavras reinicializados, e com o vetor H com os valores obtidos do processo anterior, seguindo esta sequência até que todos os blocos tenham sido processados. Ao final desde processo no último bloco, “as cinco palavras de 32 bits no [vetor] H são transmitidas como saída, formando o *hash* criptográfico de 160 bits.” (TANENBAUM, 2003, p. 573).

5 CERTIFICADO DIGITAL

Atualmente os computadores, juntamente com a Internet estão cada vez mais presentes na troca de informações importantes entre empresas, processamento de dados e transações eletrônicas, justamente por tal importância é necessário a utilização de algum mecanismo de segurança que garanta a autenticidade, confidencialidade e integridade das informações eletrônicas, e o certificado digital é a tecnologia que é cada vez mais utilizada para a satisfação de tais exigências.

Segundo a Imprensa Oficial (2015), o certificado digital é um documento eletrônico que possui dados sobre uma pessoa física ou jurídica, que o utiliza para comprovar sua integridade e autenticidade digitalmente. Funciona como uma carteira de identidade eletrônica, fazendo com que a troca de informações e transações realizadas via internet torne-se perfeitamente segura, sabendo-se que as partes envolvidas terão de apresentar suas credenciais, comprovando, assim, sua identidade.

Em documentos manuscritos oficiais utiliza-se uma assinatura que pode ser reconhecida pelo cartório de forma que uma pessoa possa garantir sua integridade para outro órgão ou pessoa, no caso da certificação digital, o usuário possui uma ferramenta de assinatura digital, permitindo a troca de documentos com autenticação, sigilo e integridade de conteúdo, possuindo, assim, reconhecimento legal como o de uma assinatura em manuscrito.

> Um certificado digital (também chamado do certificado de chave pública) é uma ligação entre a chave pública de uma entidade e um ou mais atributos relacionados a esta entidade, armazenados em um arquivo digital. O usuário neste caso pode ser uma pessoa, dispositivo de hardware ou um processo de software. O certificado digital produz a garantia de que a chave pública pertence à entidade. Além disso, garante também que a entidade (e somente esta entidade) possui de fato a correspondente chave privada. (AQUINO JUNIOR; BATISTA; HOMOLKA; LIMA; CAETANO DA SILVA; CORDEIRO DA SILVA, 2008, p. 26).

5.1 TIPOS DE CERTIFICADOS DIGITAIS

É necessário saber qual tipo de certificado digital é o mais apropriado para a solução de determinado problema e a finalidade de seu uso, pois existem diversos tipos de certificados que podem ser adquiridos atualmente.

A Figura 7 demonstra um comparativo simples entre as duas maiores empresas de prestação de serviços de certificado no Brasil, com suas respectivas descrições e produtos distribuídos pelo mercado.

Figura 7 – Certificados Digitais de Verisign e Serasa Experian

Tipos de Certificados	Verisign Inc. (Certisign)	Serasa Experian
e-CPF	 O e-CPF é a versão eletrônica do CPF, que garante a autenticidade e a integridade nas transações eletrônicas de pessoas físicas.	 O e-CPF é o seu documento de identificação na internet. Com ele, você pode assinar documentos eletrônicos com validade jurídica, autenticar-se em sites, realizar serviços da Receita Federal, como entrega de declarações e acesso ao e-CAC, tanto para a pessoa física quanto para as empresas das quais você for o representante legal.
e-CNPJ	 O e-CNPJ é a versão eletrônica do CNPJ, que garante a autenticidade e a integridade nas transações eletrônicas de pessoas jurídicas.	 O e-CNPJ é o documento de identificação da sua empresa. Com ele, você pode assinar documentos eletrônicos com validade jurídica, autenticar-se em sites, realizar serviços da Receita Federal, como entrega de declarações e acessar o e-CAC.
NF-e	 Criado especialmente para emitir notas fiscais eletrônicas (garantindo sua conformidade na Lei) e atribuir ao funcionário responsável de sua organização a alçada necessária e restrita para emissão e gerenciamento de NF-e.	 A Serasa Experian desenvolveu uma família de certificados digitais para Nota Fiscal Eletrônica (NF-e). Se você emite milhares de notas por dia ou algumas dezenas por mês, temos a solução correta para a sua necessidade.
Servidor Web	 Possui cifragem de 128 bits e conta com uma excelente relação custo-benefício. Este certificado é adotado como prática de segurança por todos os tipos de organizações para proteger suas aplicações web.	 O Certificado de Servidor é o elo de confiança entre sua empresa e seu cliente, garantindo a credibilidade e autenticidade do seu website. Além disso, todas as informações enviadas por meio do site trafegam de forma segura, criptografadas até o servidor da empresa.
Sistema de Pagamento Brasileiro (SPB)	 Conecte-se ao Sistema de Pagamentos Brasileiro com a garantia de segurança dos certificados Certisign.	 Para a participação das instituições no Sistema de Pagamento Brasileiro (SPB), o Banco Central determinou uma série de procedimentos de segurança, entre eles a necessidade de certificado digital específico: o Certificado SPB.
CSS		 O Certificado CCS garante às instituições financeiras total segurança nas operações no Cadastro de Clientes do Sistema Financeiro Nacional. Toda comunicação com o Banco Central (BACEN) trafega criptografada, atendendo aos rigorosos requisitos de segurança da ICP-Brasil.

Fontes: Verisign Inc. (Certisign), 2010 / Serasa Experian, 2010

6 LINUX

Linux é ao mesmo tempo um kernel (ou núcleo) e o sistema operacional que roda sobre ele. Criado em 1991 por Linus Torvalds, hoje é mantido por uma comunidade mundial de desenvolvedores incluindo programadores individuais e empresas como a IBM, a HP e a Hitachi, hoje coordenadas por Linus através da Linux Foundation.

Adotando uma licença de software do tipo livre (*GPL- General Public License*) e aliado a outros softwares livres, pode-se deliberar um ambiente moderno, seguro e estável para estações de trabalho, servidores e sistemas os mais diversos existentes.

Criado inicialmente sem o objetivo de ser uma plataforma portável, o Linux evoluiu nesta direção, sendo considerado um dos mais portáveis existentes, rodando nos mais diversos sistemas disponíveis, desde plataformas baixas, handhelds, centrais de entretenimento, videogames até em mainframes.

É importante notar que este sistema foi concebido para ser portável e ter a habilidade de compilar sistemas de várias outras origens em um só kernel o que contribuiu para ter a sua popularidade ampliada rapidamente.

6.1 UBUNTU SERVER

O site oficial da Ubuntu (2015) cita que, usando o kernel Linux, considerado uma das principais plataformas para a computação em escala, o Ubuntu Server permite tirar o máximo da sua infraestrutura.

Caso seja necessária a implantação de uma nuvem OpenStack (gerenciamento de múltiplas infraestruturas virtualizadas), um cluster Hadoop (plataforma de software em Java de computação distribuída voltada para clusters e processamento de grandes massas de dados) ou um nó 50.000 de Render Farm (grupo de computadores em redes utilizados para a renderização de imagens), o Ubuntu Server oferece o melhor valor de desempenho em escala disponível.

O Ubuntu Server é suportado pela Canonical LTD, seu desenvolvedor, por 5 anos e opera sob licença livre (GPL - General Public License).

6.2 KALI LINUX

Kali Linux é uma avançada distribuição Linux especializada em “Testes de Intrusão” e “Auditoria de Segurança”. É uma reconstrução completa do “BackTrack Linux” totalmente aderente aos padrões de desenvolvimento do Debian.

A Figura 8 mostra as principais características do sistema operacional Kali Linux.

Figura 8 - Principais características do KALI

Testes de intrusão	Dispões de mais de 300 ferramentas de testes de intrusão já incluidas no pacote original
Gratuidade	Como seu predecessor, é sempre gratuita
Repositório Git	Seu repositório Git é distribuído com seus códigos fontes, permitindo que estes sejam adpatados e/ou remontados livremente
Complacente com o padrão FHS	Aderente ao padrão Hierarquico do Sistema de Ariquivos (FHS em inglês), possibilitando que todos os usuários localizem os arquivos binários, de apoio, bibliotecas, etc.
Vasto suporte a dispositivos wireless	Compatibilidade com uma vasta gama de hardware e dispositivos USB
Kernel adaptado para injeção de pacotes	Inclui os principais patches para os testes de intrusão em redes sem fio
Múltiplos idiomas	Suporte a múltiplos idiomas, permitindo aos usuários efetuar seus testes em seus idiomas nativos
Suporte a ARMEL e ARMHF	Instaladores disponíveis para ambientes ARMEL e ARMHF, já ques os sistemas que utilizam processadores ARM possam utilizar a plataforma KALI Linux

Fonte: Kali (2015)

6.2.1 Wireshark

Segundo Sanders (2011) Wireshark é uma ferramenta de análise de pacotes gratuita e de código aberto, desenvolvida nas linguagens C e C++ sendo muito utilizada no diagnóstico de redes. Foi desenvolvida por Gerald Combs, um graduado em ciência da computação na Universidade de Missouri – Cidade do Kansas, onde primeiramente foi lançada com o nome de Ethereal em 1998, porém oito anos depois após problemas com patente da marca, Gerald e seu time renomearam o projeto para Wireshark.

O site oficial do Wireshark (2015), lista todas as funcionalidades do programa, incluindo:

a) Profunda inspeção de centenas de protocolos;
b) Captura ao vivo e análise off-line;
c) Multiplataforma, rodando em Windows, Linux, OS X, Solaris, FreeBSD, NetBSD e muitas outras;
d) Utilização de uma interface GUI para análise dos pacotes;
e) Rica análie VoIP

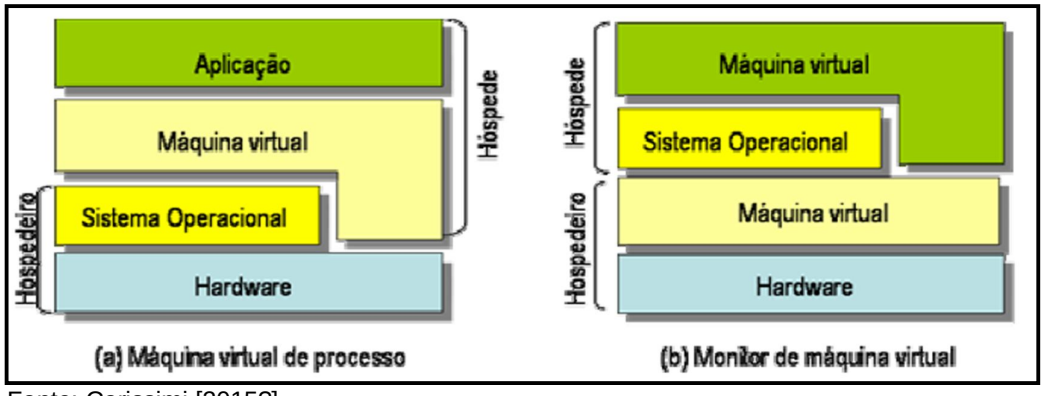
7 MÁQUINAS VIRTUAIS

Uma abordagem para VM (Virtual Machine, em inglês) seria fornecer uma camada de software entre o hardware e o sistema operacional, protegendo o acesso direto deste aos recursos físicos da máquina. Essa camada oferece como interface ao sistema operacional um conjunto de instruções de máquina que pode ser o mesmo do processador físico ou algum outro.

O ponto importante é que essa interface deve estar disponível sempre que o computador estiver ligado e que ela possa ser usada simultaneamente por diferentes programas. O resultado final é a possibilidade de ter diversos sistemas operacionais (programas) executando independentemente na mesma plataforma. Genericamente uma abordagem alternativa é fornecer uma camada de software entre o hardware e o sistema operacional protegendo o acesso direto deste aos recursos físico da máquina. Essa camada oferece como interface ao sistema operacional um conjunto de instruções de máquina que pode ser o mesmo do processador físico, ou um outro. (SMITH; NAIR, 2005).

Essa máquina virtual é referenciada como monitor de máquina virtual (*Virtual Machine Monitor VMM*), também conhecido como *hypervisor*, ou ainda, máquina virtual de sistema. Essa máquina virtual é referenciada como monitor de máquina virtual *(Virtual Machine Monitor* VMM), também conhecido como *hypervisor*, ou ainda, máquina virtual de sistema de acordo com a Figura 9 (SMITH; NAIR, 2005).

Figura 9 - Máquinas virtuais e relações com sistemas hóspede e hospedeiro



Fonte: Carissimi [2015?]

8 SECURE SOCKET LAYER (SSL)

Inicialmente, no final dos anos 80, a internet era utilizada como um sistema de informações baseadas em páginas com hipertexto**.** Em pouco tempo, empresas começaram a utilizar a web para transações financeiras que se utilizavam de cartões de crédito para compra de mercadorias e transações bancárias *online*. Essas aplicações exigiram uma demanda por conexões seguras (TANEMBAUM, 2003).

Nessa época várias empresas apresentaram soluções para garantir que tais vulnerabilidades fossem corrigidas, dentre essas soluções podemos citar o SSL, desenvolvida em 1995 pela Netscape Communications Corp. que até então dominava o mercado de fabricantes de navegadores e que mais tarde tornou-se um padrão de fato, sendo posteriormente incorporada aos principais *browsers* e servidores WEB. (TANEMBAUM, 2003).

> SSL significa Secured Sockets Layer (Camada Segura de Sockets) é um protocolo que permite a transmissão de informações através da Internet de forma criptografada. O SSL garante que a informação seja enviada, sem alterações e exclusivamente para o servidor para o qual pretende enviar. Os sites de compras online com frequência utilizam esta tecnologia para proteger suas informações como dados de cartão de crédito, por exemplo (RABELLO, 2010).

8.1 PROTOCOLOS E MENSAGENS

O SSL é dividido em protocolos e inúmeros tipos de mensagens, cada qual com sua funcionalidade. A seguir os mesmo serão explicados segundo a Internet Engineering Tast Force (2011).

a) Handshake Protocol: o início de toda comunicação é feito por este protocolo, onde o cliente e o servidor negociam uma conexão segura através de uma série de passos onde os mesmos devem decidir os algoritmos de encriptação e autenticação que serão utilizados na conexão e gerar segredos através de funções que geram números aleatórios. Em suma, esse protocolo possui três etapas. Na primeira etapa ocorre a negociação de algoritmos que serão utilizados onde cliente e servidor decidem quais são os algoritmos suportados por ambos, onde o cliente faz o pedido a um servidor que suporte

o protocolo SSL por uma conexão segura e o mesmo já envia uma lista com todos os algoritmos disponíveis para encriptação e autenticação de dados. Na segunda etapa após a escolha do algoritmo mais forte dentre os possíveis, ambos trocam chaves e realizam suas autenticações, lembrando que a autenticação do cliente é opcional. Na terceira e última fase, as mensagens são autenticadas por funções *hash,* e assim garantem a integridade, segurança e autenticação das mesmas após passarem pelo protocolo Record Protocol.

b) Record Protocol: sabendo-se que o SSL se comunica através de pacotes que encapsulam os dados que estão sendo trocados, este protocolo funciona de maneira a tratar esta comunicação, recebendo os dados na entrada e saída da camada. Ele recebe os dados das camadas superiores, os encapsula, encripta e pode também adicionar MACs para garantir ainda mais segurança, integridade e autenticação das mensagens para que o processo inverso seja feito pelo destinatário.

c) Alert Protocol: possui o papel de fazer com que a troca de mensagens que dizem respeito ao funcionamento da transmissão de dados na conexão. Suas mensagens são compostas por dois *bytes*, sendo o primeiro responsável pelo seu tipo, podendo ser do tipo *warning* ou *fatal*, sendo que quando é recebida uma mensagem do tipo *fatal*, o protocolo interrompe a transmissão imediatamente. O segundo byte carrega o código referente a mensagem.

d) Change Cipher Spec: responsável pela mensagem que significará o início de um estágio fixo da comunicação onde a partir dela, todas as mensagens serão criptografadas e autenticadas segundo as negociações contidas no Hello Protocol (contida no Handshake Protocol, referente a escolha das funções de criptografia e autenticação que o cliente possui/suporta que foram acordadas no início da conexáo). Após essas mensagens serem trocadas é função do Record Protocol de fazer a transmissão de dados na sessão.

Toda comunicação dentro do protocolo SSL entre cliente e servidor é feita utilizando-se mensagens. Abaixo serão listadas todas as possíveis mensagens dentro de uma comunicação SSL.

a) Alert – Informa à outra parte de uma possível brecha na segurança ou falha de comunicação.

b) ApplicationData – Informação real que as duas partes irão trocar entre si, que é criptografada, autenticada e/ou verificada por SSL.

c) Certificate – Uma mensagem que carrega o certificado da chave pública do remetente.

d) CertificateRequest – Uma solicitação do servidor para que o cliente forneça seu certificado da chave pública.

e) CertificateVerify – Uma mensagem do cliente que verifica que ele conhece a chave privada correspondente à sua chave pública certificada.

f) ChangeCIpherSpec – Um indicador para começar a utilizar serviços de segurança acordados.

g) ClientHello – Uma mensagem do cliente indicando os serviços de segurança que ele deseja e que ele é suporta.

h) ClientKeyExchange – Uma mensagem do cliente carregando as chaves criptográficas para as comunicações.

i) Finished – Uma indicação de que todas as negociações iniciais estão completas e uma comunicação segura foi estabelecida.

j) HelloRequest – Um pedido do servidor para que o cliente inicie (ou reinicie) o processo de negociação.

k) ServerHello – Uma mensagem do servidor indicando os serviços de segurança que serão utilizados para a comunicação vigente.

l) ServerHelloDone – Uma indicação do servidor de que ele completou todos os seus pedidos para o cliente para estabelecer a comunicação.

m) ServerKeyExchange – Uma mensagem do servidor carregando as chaves criptográficas para a comunicação.

8.2 FUNCIONAMENTO

O protocolo SSL é responsável pela existência de uma conexão segura entre duas entidades, incluindo a negociação de parâmetros entre os mesmos, autenticação mútua, comunicação secreta e proteção na integridade dos dados.

> Efetivamente, trata-se de uma nova camada colocada entre a camada de aplicação e a camada de transporte, aceitando solicitações do navegador e enviando-as ao TCP para transmissão ao servidor. Depois que a conexão

> segura é estabelecida, a principal tarefa da SSL é manipular a compactação e a criptografia. Quando o HTTP é usado sobre a SSL, ele se denomina HTTPS (Secure HTTP), embora seja o protocolo HTTP padrão (TANENBAUM, 2003, p. 609).

A Figura 10 mostra o posicionamento do SSL entre as camadas de aplicação e transporte.

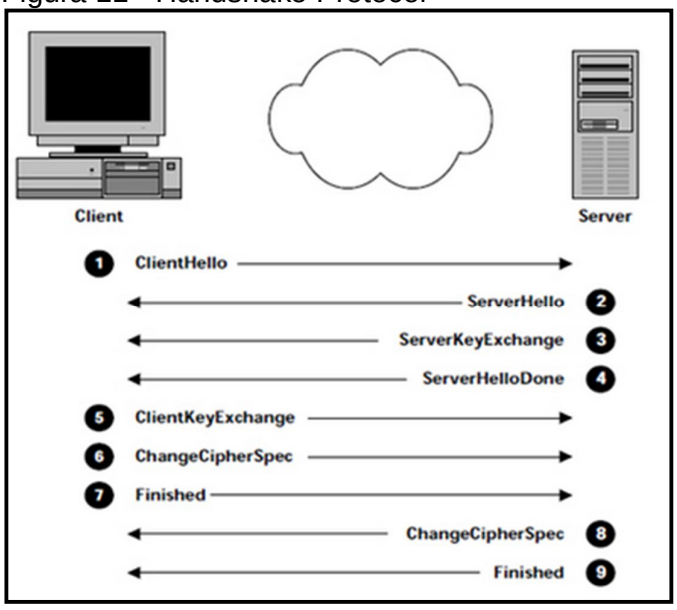
Figura 10 – Pilha de camadas do TCP/IP com SSL

	Camada das Aplicações (http, ftp...)
SSL	SSL Change Cipher, Alert Protocol, Handshake Protocol
SSL	SSL Camada SSL Record
	Camada TCP
	Camada IP

Fonte: Largura (2010)

Tanenbaum (2013) cita que o SSL consiste em dois subprotocolos, um para estabelecer uma conexão segura e outro para usá-la. O primeiro passo é iniciar o protocolo Handshake, demonstrado na Figura 11.

Figura 11 - Handshake Protocol



Fonte: Pinheiro, Vieira, Silva (2011).

Segundo a figura, primeiramente o cliente envia a mensagem *ClientHello* propondo as opções de SSL. O servidor responde com a mensagem *ServerHello* dizendo quais opções serão utilizadas naquela comunicação, envia sua chave pública e conclui sua parte da negociação. Então, o cliente envia as informações da

chave de sessão (Criptografada com a chave pública do servidor), envia a mensagem *ChangeCipherSpec* com o objetivo de ativar as opções negociadas para as mensagens seguintes e por último envia a mensagem *Finished* que após recebida pelo servidor envia a mensagem *ChangeCipherSpec* para ativar as opções negociadas para as mensagens seguintes finalizando com a mensagem *Finished*. A partir desde ponto é reponsabilidade do segundo subprotocolo, *Record protocol*, utilizar esta conexão para fazer a transmissão de dados entre as entidades.

9 TRABALHOS CORRELATOS

A área da computação voltada para a segurança da informação é muito vasta, existindo inúmeros métodos de se proteger informações preciosas a serem dissipadas ou adulteradas por terceiros. Hoje em dia, peritos na quebra dessas barreiras veem descobrindo novas tecnologias de invasões, e em alguns casos acabam utilizando esses conhecimentos para uso e benefício próprio, fazendo com que métodos antigos sejam inválidos hoje.

É possível encontrar vários artigos e trabalhos científicos relacionados à segurança da informação, porém geralmente específicos pelo fato de existirem vários métodos de se quebrar o sigilo de informações, ou então por serem utilizados para se defenderem de determinados ataques. Em relação ao objetivo deste trabalho, após a realização de uma pesquisa, não foram encontradas pesquisas ou artigos científicos com uma análise mais profunda sobre as vantagens do uso do protocolo SSL e seu desempenho diante de ataques que geralmente são utilizados na *web,* somente sobre algumas de suas técnicas de defesa.

A técnica utilizada pelo protocolo SSL para garantir a segurança da informação é a criptografia, onde o mesmo possui funções para estabelecer uma conexão segura entre dois soquetes e posteriormente responsável por fazer a transmissão de dados utilizando-se desta conexão.

Ainda sobre o tópico de criptografia, podemos citar a monografia intitulada “Criptoanálise em Redes sem Fio Utilizando *Lookup Tables*”, de autoria de Deivison Cardoso, em que o mesmo explica detalhadamente alguns algoritmos de criptografia e autenticação de mensagens que são utilizados pelo protocolo SSL como o DES, 3DES, MD5 e SHA-1, porém teve como objetivo verificar a eficiência de quebras de senha utilizando *Lookup Tables* e ao final demonstrando como elevar a segurança de uma rede sem fio contra esse tipo de ataque que se demonstrou eficiente.

A intenção deste trabalho é contribuir com informações e resultados relevantes para a demonstração da importância do SSL nos dias de hoje.

10 METODOLOGIA

10.1 TIPO DE PESQUISA

Foi realizado um levantamento bibliográfico que pudesse demonstrar os princípios do funcionamento do protocolo SSL para a segurança da informação na web, podendo então, realizar o objetivo deste trabalho desenvolvendo um comparativo, que apontou vantagens e recursos de segurança que existem em domínios web que se utilizam do protocolo de criptografia SSL (Secure Socket Layer) e aqueles que não o utilizam.

10.2 RECURSOS

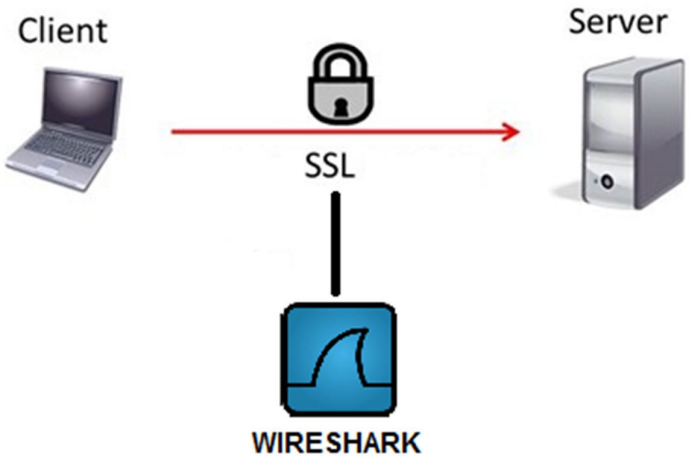
Para realizar os testes foi utilizado um notebook da marca Gateway modelo NE56R13B, com processador Intel Core i5-3230M 3.20GHz, com 4Gb de memória RAM, disco rígido de 500Gb, com Windows 8.

Foi também necessário utilizar o software da Oracle de simulação de máquinas virtuais, VirtualBox – versão 5.0, que foi responsável pela virtualização do ambiente de testes. Como os mesmos foram feitos em um domínio web, foi necessário à utilização de um servidor web, no caso, Apache2 contido na máquina virtual Ubuntu.

10.3 EXECUÇÃO

Após todas as máquinas terem sido virtualizadas, foi criado um servidor web (Apache) que serviu de host local, simulando um ambiente web. Posteriormente foram utilizados dois domínios web, um contendo o protocolo de segurança citado no trabalho, o protocolo SSL (caracterizado por possuir HTTPS:// no URL), que é utilizada na autenticação e integridade entre cliente/servidor e um segundo domínio desprotegido. Ambos foram testados e avaliados com a ferramentas de ataque, exploração de vulnerabilidades e escaneamento Wireshark – versão 1.12.6 localizada na máquina virtual que possui instalado o Kali Linux.

Ao final, após feito um levantamento de dados provenientes dos testes realizados o pesquisador foi capaz de demonstrar através de um comparativo, uma análise crítica que descreve as vantagens que o protocolo SSL proporciona a domínios web e para a segurança das informações que usuários disponibilizam para os sites.



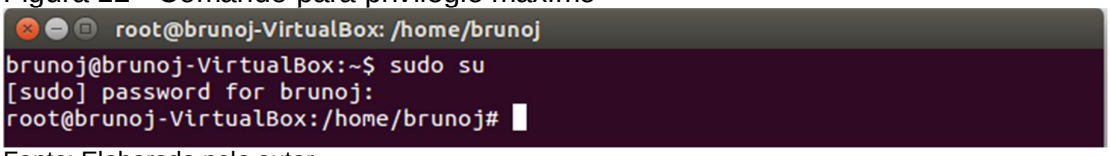
10.4 IMPLEMENTAÇÃO DO AMBIENTE

Primeiramente como um dos objetivos deste trabalho, foi implementado um ambiente que possuísse o protocolo de segurança SSL para que posteriormente se fossem realizados os testes necessários para a construção do comparativo. Todos os passos a seguir foram realizados no terminal da máquina virtual Linux Ubuntu 14.04 LTS.

10.4.1 Pré-requisitos

Para que se possa realizar os comandos necessários para a criação do certificado que utilizará as tecnologias SSL necessita-se de privilégios de usuário root (todos privilégios). Foi utilizado o comando apresentado na Figura 12.

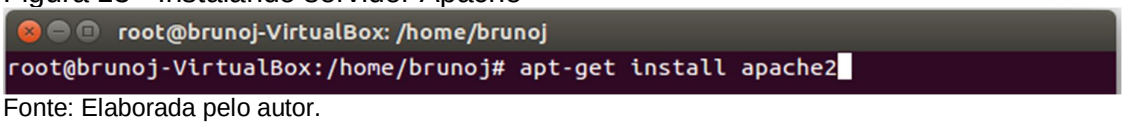
Figura 12 - Comando para privilégio máximo



Fonte: Elaborada pelo autor.

Onde “sudo su” faz o pedido de acesso ao super usuário, em seguida é requisitado a senha do usuário administrador, após inserida a senha correta se possui todos os privilégios para utilizar dos comandos necessários para a implementação e construção do ambiente. Foi necessária a instalação do servidor web, no caso Apache2, conforme ilustra a Figura 13.

Figura 13 - Instalando servidor Apache



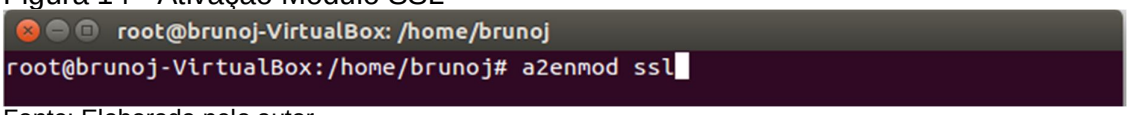
Fonte: Elaborada pelo autor.

Com esses comandos é possível iniciar a criação do ambiente SSL.

10.4.2 Ativação do Módulo SSL

O SSL suporta as configurações padrões contidas no pacote Apache2 que foi previamente instalado, porém necessita-se ativá-lo para que se possa usufruir de suas vantagens no sistema. O comando seguinte, demonstrado na Figura 14, executa essa ativação.

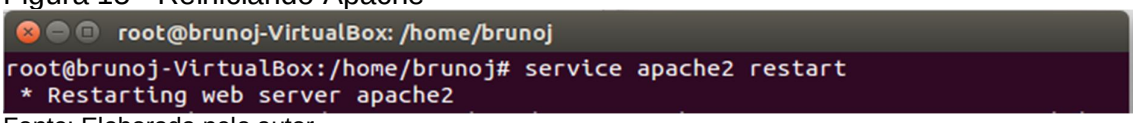
Figura 14 - Ativação Módulo SSL



Fonte: Elaborada pelo autor.

Assim que ativado o modulo SSL, é necessário reiniciar o servidor Apache para que a mudança seja reconhecida pelo servidor web segundo a Figura 15.

Figura 15 - Reiniciando Apache

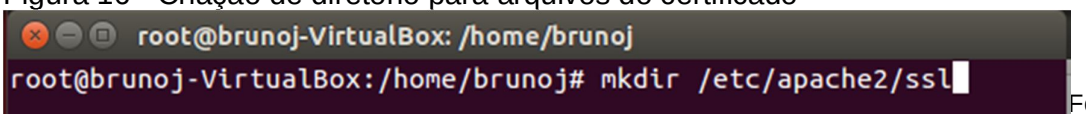


Fonte: Elaborada pelo autor.

10.4.3 Criação do Certificado Auto Assinado SSL

Antes de criar o certificado foi criado um subdiretório nas configurações de hierarquia do Apache como demonstrado na Figura 16, para colocar os arquivos de certificado que serão criados.

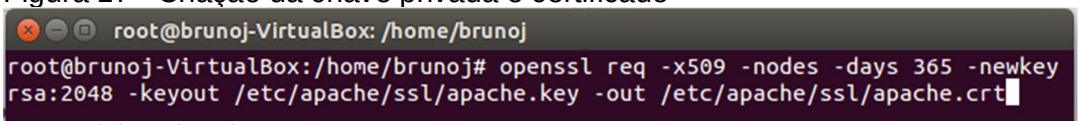
Figura 16 - Criação de diretório para arquivos do certificado



Fonte: Elaborada pelo autor.

Com o comando apresentado na Figura 17 foi possível criar a chave e o certificado necessários ao protocolo SSL e coloca-los no subdiretório que foi criado.

Figura 17 - Criação da chave privada e certificado



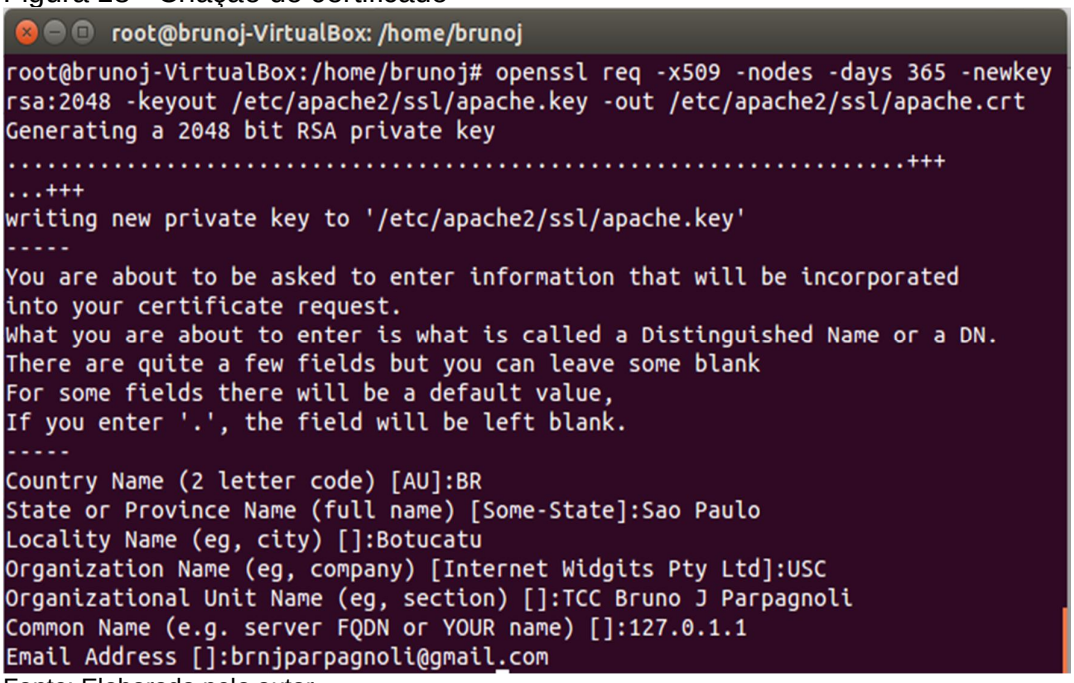
Fonte: Elaborada pelo autor.

Cada um dos parâmetros possui suas propriedades e respectivos valores, onde:

a) Openssl: linha de comando básica proveniente do OpenSSL para criar e gerenciar certificados, chaves, requisição de assinaturas entre outros;
b) Req: especifica um subcomando para X.509 que gerencia requisições de assinatura. X.505 é a infraestrutura padrão que o SSL assume para suas chaves e gerencia de certificados;
c) –x509: Esta opção especifica que será criado um certificado autoassinado ao invés de um que necessita de uma requisição de certificado;
d) –nodes: desabilita o OpenSSL de fazer a segurança do arquivo .key (chave) com uma palavra-chave, caso contrário o mesmo não deixaria que o Apache iniciasse automaticamente sempre que o mesmo fosse reiniciado, requisitando a palavra-chave para que ele inicie;
e) –days 365: especifica que o certificado criado será valido por 365 dias;
f) –newkey rsa:2048: cria o “pedido de certificado” e uma chave privada ao mesmo tempo. Necessário pois não havia nenhuma chave privada atualmente criada. O parâmetro “rsa:2048” faz com que o OpenSSL gere uma chave do tipo RSA de 2048 bits;
g) –keyout: nomeia o arquivo de saída que conterá a chave privada;
h) –out: nomeia o arquivo de saída que conterá o certificado que está sendo gerado.

Assim que este comando foi executado foi necessário responder a uma série de questões que serão implementadas no certificado. A Figura 18 ilustra o processo de resposta.

Figura 18 - Criação do certificado



Fonte: Elaborada pelo autor.

10.4.4 Configurando Apache para Utilizar o Protocolo SSL

Com o certificado e a chave criados, o próximo passo foi configurar o Apache para utilizar esses arquivos em ou host virtual. Foi necessário acessar o arquivo contido em /etc/apache2/sites-availables/default-ssl.conf que contém algumas configurações padrões do SSL e fazer algumas modificações para atender as necessidades do trabalho. Com os comentários removidos o arquivo se apresenta como mostrado na Figura 19:

Figura 19 - Configurações Apache

```
<IfModule mod_ssl.c>
    <VirtualHost _default_:443>
        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html
        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined
        SSLEngine on
        SSLCertificateFile /etc/ssl/certs/ssl-cert-snakeoil.pem
        SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key
        <FilesMatch "\.(cgi|shtml|phtml|php)$">
                        SSLOptions +StdEnvVars
        </FilesMatch>
        <Directory /usr/lib/cgi-bin>
                        SSLOptions +StdEnvVars
        </Directory>
        BrowserMatch "MSIE [2-6]" \
                        nokeepalive ssl-unclean-shutdown \
                        downgrade-1.0 force-response-1.0
        BrowserMatch "MSIE [17-9]" ssl-unclean-shutdown
    </VirtualHost>
</IfModule>
```

Fonte: Elaborada pelo autor.

A Figura 20 apresenta o novo arquivo com as devidas modificações em vermelho.

Figura 20 – Editando configurações do Apache

```
<IfModule mod_ssl.c>
    <VirtualHost _default_:443>
        ServerAdmin admin@example.com
        ServerName your_domain.com
        ServerAlias www.your_domain.com
        DocumentRoot /var/www/html
        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined
        SSLEngine on
        SSLCertificateFile /etc/apache2/ssl/apache.crt
        SSLCertificateKeyFile /etc/apache2/ssl/apache.key
        <FilesMatch "\.(cgi|shtml|phtml|php)$">
                        SSLOptions +StdEnvVars
        </FilesMatch>
        <Directory /usr/lib/cgi-bin>
                        SSLOptions +StdEnvVars
        </Directory>
        BrowserMatch "MSIE [2-6]" \
                        nokeepalive ssl-unclean-shutdown \
                        downgrade-1.0 force-response-1.0
        BrowserMatch "MSIE [17-9]" ssl-unclean-shutdown
    </VirtualHost>
</IfModule>
```

Fonte: Elaborada pelo autor.

O objetivo dessas modificações é apenas para configurar o virtual host que será aplicado o SSL (ServerAdmin, ServerName, ServerAlias e DocumentRoot) e para mostrar ao Apache onde ele devera procurar pelo certificado SSL e a chave privada.

10.4.5 Ativando o Host Virtual SSL

O próximo passo tem como objetivo ativar o virtual host e pode ser feito com a linha de comando na Figura 21.

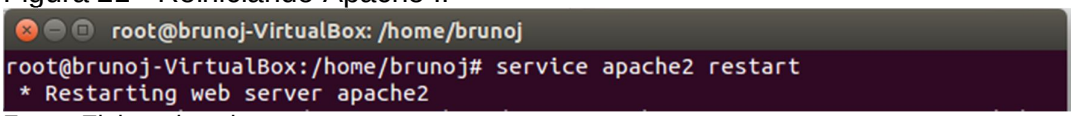
Figura 20 – Ativação do Virtual Host

```
root@brunoj-VirtualBox: /home/brunoj
root@brunoj-VirtualBox:/home/brunoj# a2ensite default-ssl.conf
```

Fonte: Elaborada pelo autor.

Como passo final, foi necessário reiniciar o Apache novamente para carregar o novo Virtual Host contendo SSL com o comando ilustrado na Figura 22.

Figura 21 - Reiniciando Apache II



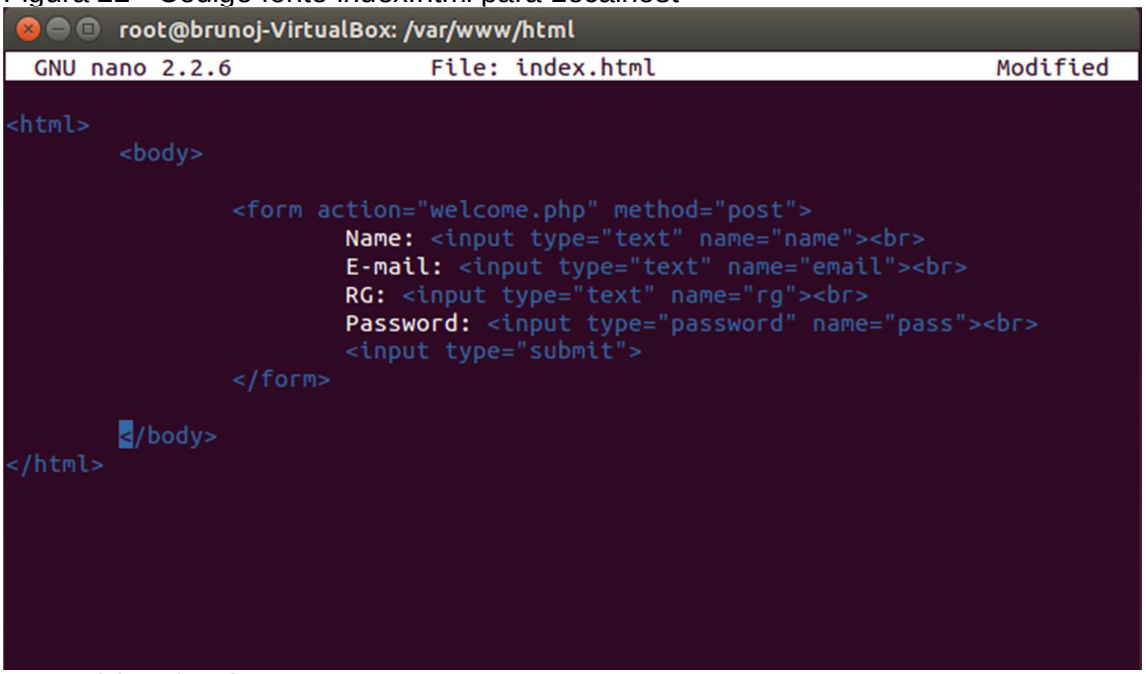
Fonte: Elaborada pelo autor.

Isso faz com que o novo virtual host realize a encriptação dos dados utilizando o certificado SSL que foi criado.

10.4.6 Geração de Tráfego na Rede para Realização dos Testes

Para que os testes fossem realizados foi necessário gerar algum tipo de tráfego de dados na rede para que então esses dados que circulariam pudessem ser analisados enquanto seguros pelo protocolo SSL e de modo não seguro. Para que o mesmo ocorresse foi necessário a edição da página index em localhost (página inicial do Apache). O novo código fonte da página está ilustrado na Figura 23.

Figura 22 - Código fonte index.html para Localhost

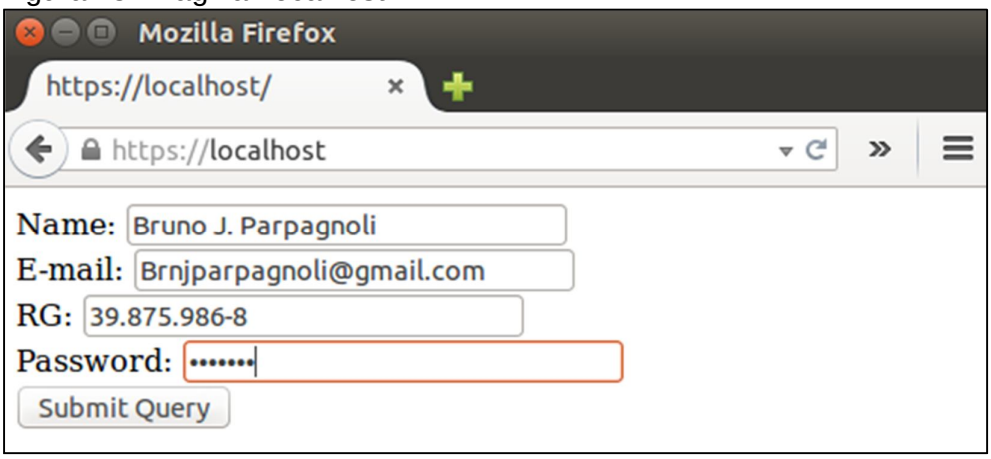


Fonte: Elaborado pelo autor.

O objetivo desta edição era criar uma página com alguns formulários a serem preenchidas bem como um método POST para que as mesmas informações fossem

enviadas para o servidor para processamento, gerando assim o tráfego na rede necessário. A compilação deste código gerou a página demonstrada na Figura 24.

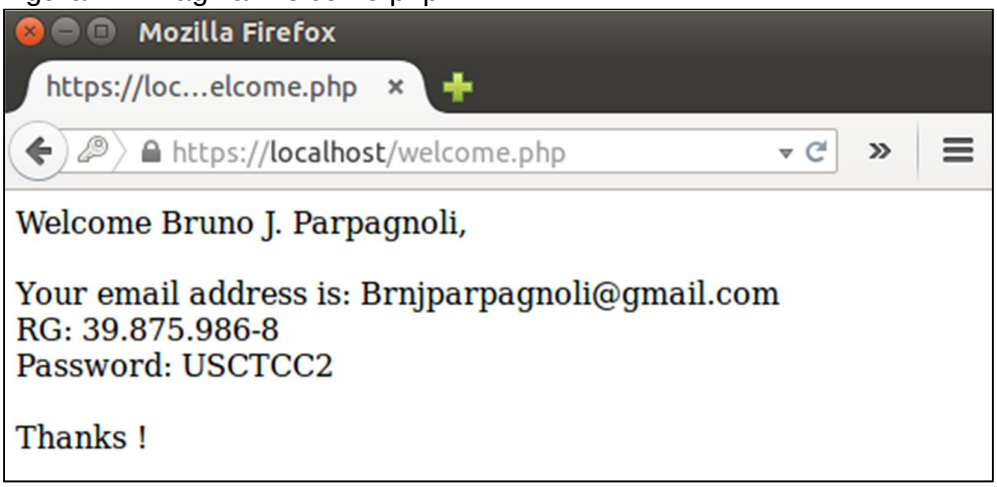
Figura 23 - Página Localhost



Fonte: Elaborada pelo autor.

Assim que o botão “Submit Query” for pressionado as informações nos formulários serão transferidas para a página Welcome.php, que como dito, farão o processamento de dados, retornando para o usuário a seguinte página ilustrada na Figura 25.

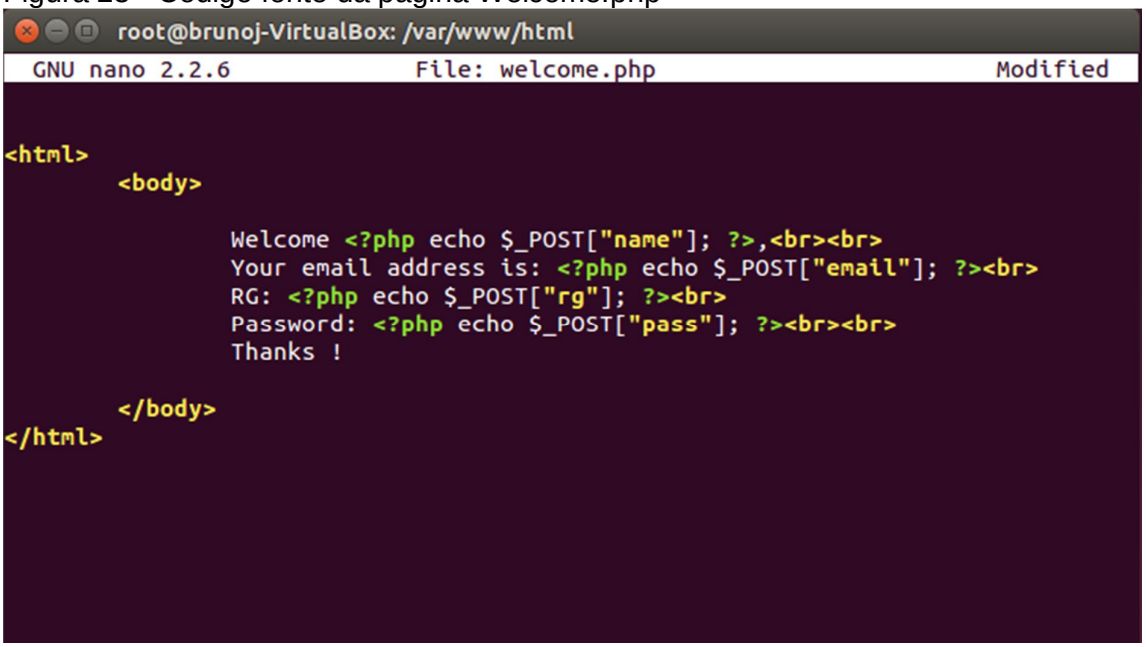
Figura 24 - Página Welcome.php



Fonte: Elaborada pelo autor.

O código-fonte desta página demonstrado na Figura 25 está apresentado na Figura 26.

Figura 25 - Código fonte da página Welcome.php



Fonte: Elaborada pelo autor.

Com isso, foi possível gerar o tráfego de dados na rede, para que os testes necessários fossem realizados com sucesso.

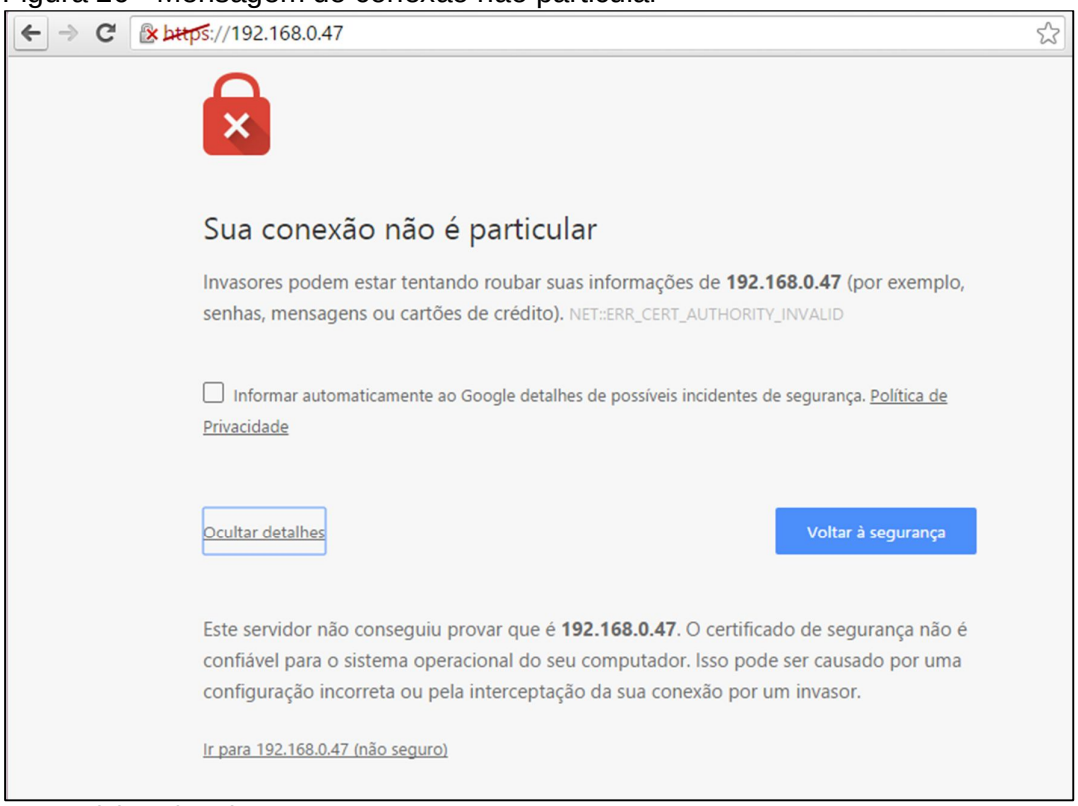
11 RESULTADOS

A realização dos testes aqui citados foi feita sem a utilização de Internet, pois não havia necessidade da utilização da mesma, apenas por uma rede “interna” criada entre todas as máquinas virtuais e a máquina host. É possível notar também que a diferença entre os valores de IP entre ambos os testes se deve pelo fato de que houve uma reinicialização no sistema, e com isso foi atribuído um IP novo ao servidor web no teste com SSL ativado em relação ao mesmo com o protocolo desativado, dado que ele foi feito posteriormente.

11.1 TESTANDO A IMPLEMENTAÇÃO DO AMBIENTE PARA TESTES

Com todas as configurações feitas, certificados e chaves criadas, ao acessar o URL https://192.168.0.47 (IP relativo ao endereço do ambiente onde está localizado o servidor web, no caso, a máquina virtual Ubuntu, que contém o servidor Apache) em qualquer navegador na mesma rede que o servidor nos dá a seguinte mensagem conforme a Figura 27.

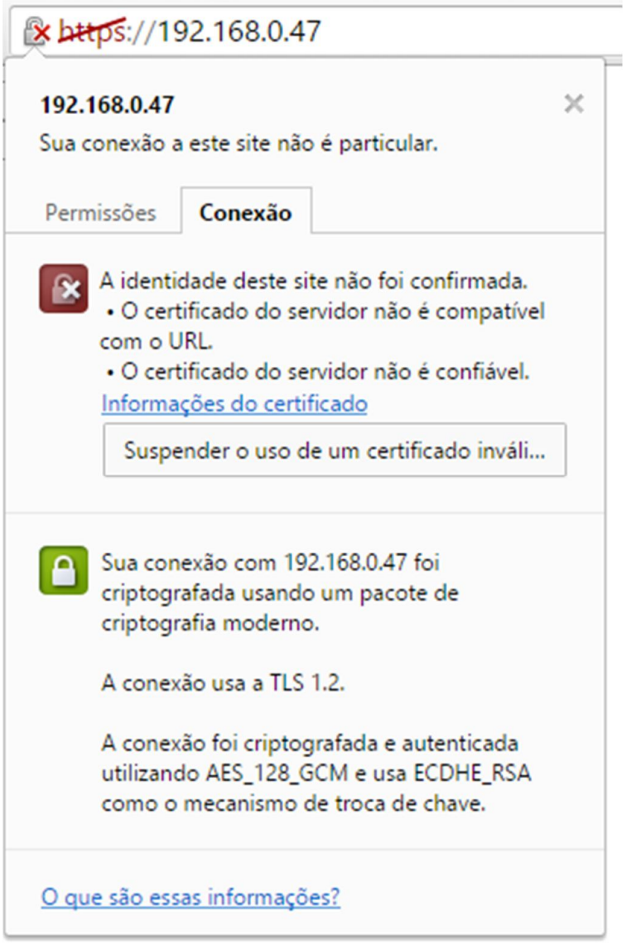
Figura 26 - Mensagem de conexão não particular



Fonte: Elaborada pelo autor.

Essa mensagem diz que o navegador não consegue verificar a identidade do servidor pois ele não foi assinado por uma terceira autoridade certificadora em que ele confia. Isso é esperado, pois foi criado um certificado auto assinado e para o navegador não é seguro. Apesar desta mensagem de aviso o servidor ainda será capaz de criptografar a comunicação. Ao proceder para o link percebe-se acessando as propriedades da página que a comunicação está sendo criptografada como demonstrado na Figura 28.

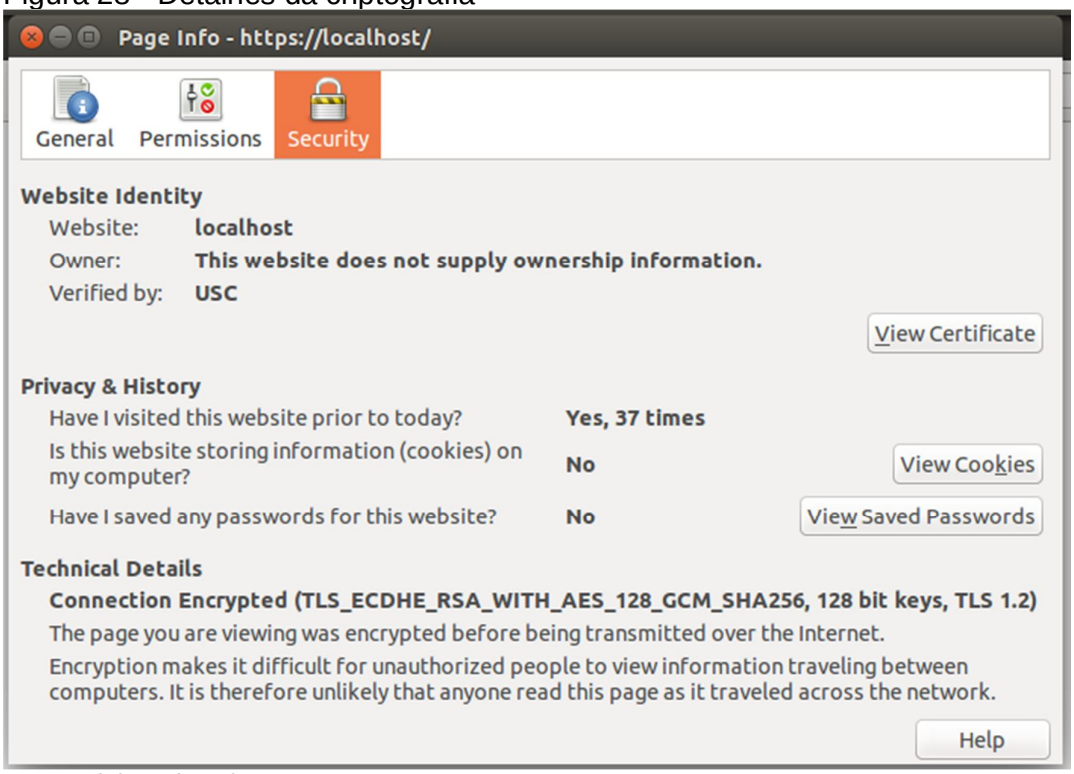
Figura 27 - Conexão criptografada



Fonte: Elaborada pelo autor.

Em seguida, ao acessar o link em azul “Informações do certificado” contido na Figura 28 se obtém mais detalhes sobre essa nova conexão em relação ao certificado criado. A Figura 29 apresenta esses dados.

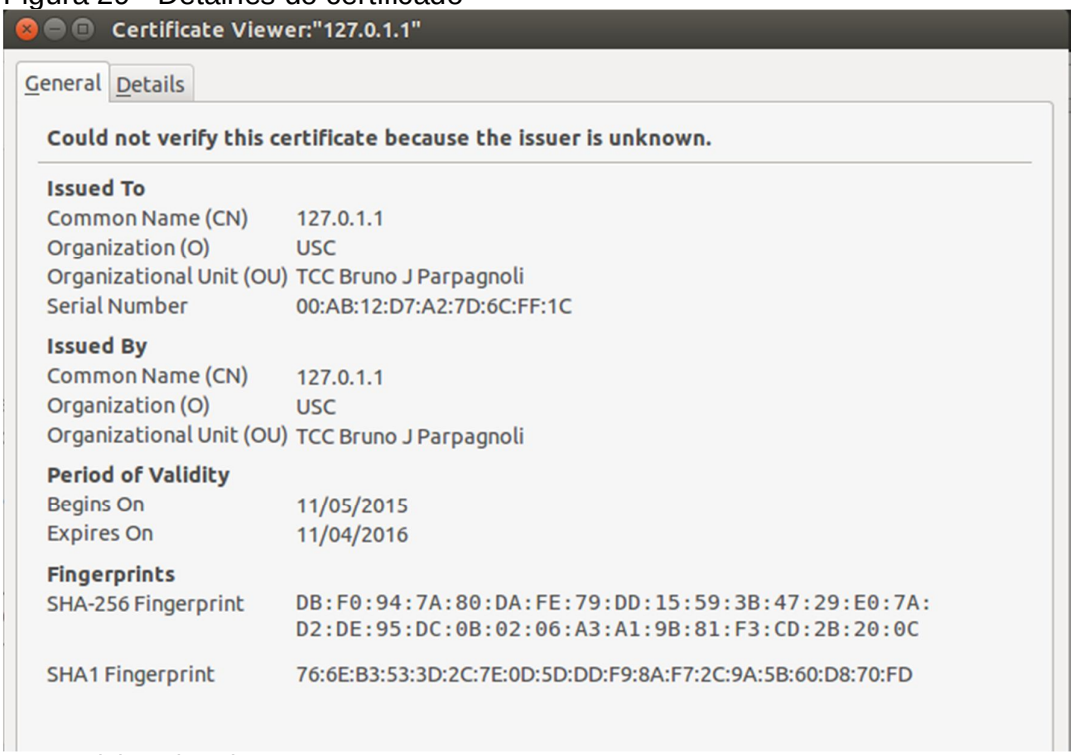
Figura 28 - Detalhes da criptografia



Fonte: Elaborada pelo autor.

Com isso temos todos os detalhes sobre o tipo de criptografia que está sendo utilizada na comunicação desta página com qualquer um que a acesse. Clicando em “View Certificate” foi possível ver as informações contidas no certificado como demonstrado na Figura 30.

Figura 29 - Detalhes do certificado

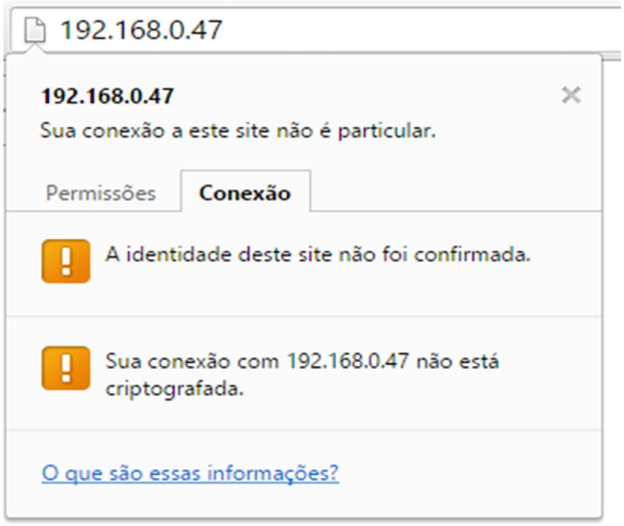


Fonte: Elaborada pelo autor.

Nesta figura estão contidas todas as informações do certificado que vieram das questões que o serviço requisitou quando a linha de comando que criaria o certificado foi executada (demonstrado na Figura 18), que foram citadas na metodologia.

Com ambiente para testes com protocolo SSL foi criado com êxito, foi necessário verificar se o processo reverso poderia ser feito ao editar o link de acesso “https://192.168.0.47” para apenas “192.168.0.47”, fazendo com que o protocolo SSL seja desativado para o ambiente. A Figura 31 ilustra o processo.

Figura 30 - Conexão não criptografada



Fonte: Elaborada pelo autor.

Como visto na Figura 31 foi possível utilizar o mesmo domínio para a realização de ambos os testes, os que foram feitos com o protocolo SSL ativado e com protocolo SSL desativado.

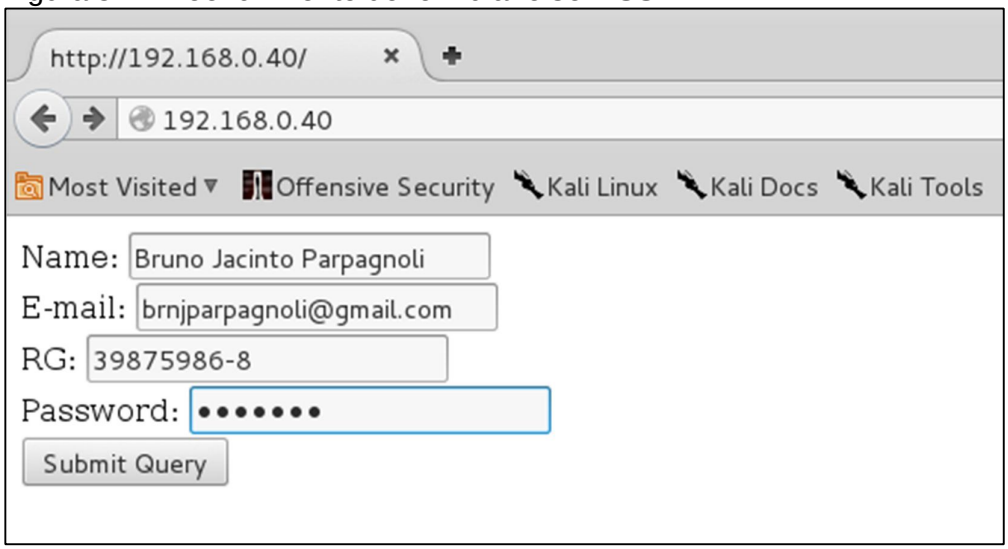
11.2 ANÁLISE NO TRÁFICO UTILIZANDO A FERRAMENTA WIRESHARK

Para que os seguintes testes fossem realizados, foi utilizado a ferramenta de scaneamento de redes Wireshark, o processo para a realização de ambos os testes foi, primeiramente, se certificar que todas as máquinas virtuais estivessem na mesma rede, posteriormente iniciar o escaneamento da rede, acessar os ambientes, gerar o tráfego na rede e, ao final, analisar o que foi coletado pela ferramenta.

11.2.1 Análise sem SSL

Após ativado o escaneamento da rede pelo Wireshark, foi acessado pelo browser (navegador) da máquina Kali Linux o endereço de IP 192.168.0.40 associado ao servidor web que continha o ambiente com o protocolo de rede SSL desativado. Foi então, digitado nos formulários *Name, E-mail, RG, Password,* respectivamente os valores: Bruno J. Parpagnoli, Brnjparpagnoli@gmail.com, 39.875.986-8, USCTCC2 conforme a Figura 32

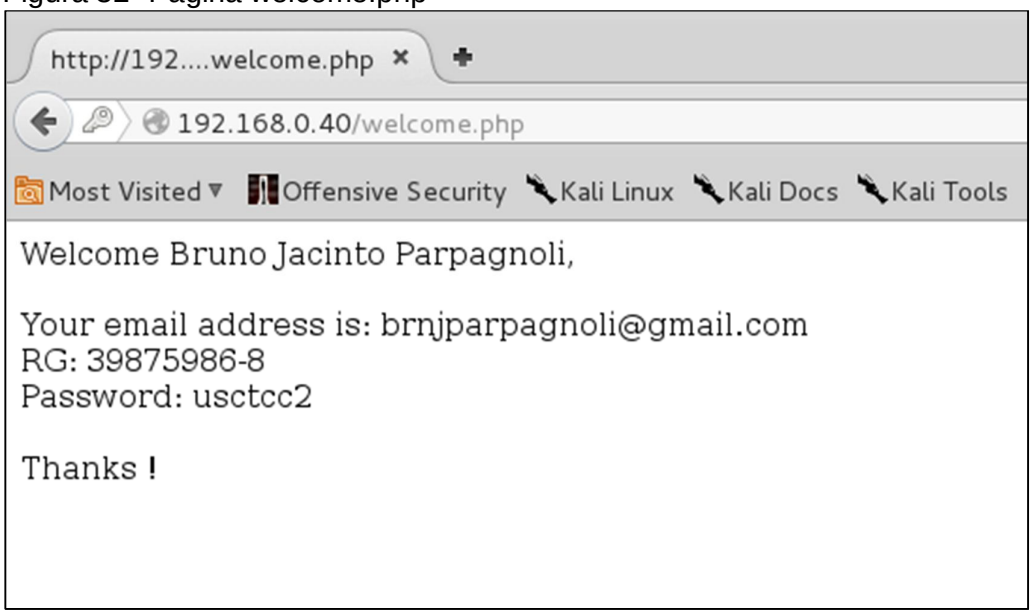
Figura 31 - Preenchimento do formulário sem SSL



Fonte: Elaborada pelo autor.

Feito isso, foi realizado a submissão do método POST clicando no botão “Submit Query”, assim indo para a próxima página web denominada welcome.php como ilustrado na Figura 33.

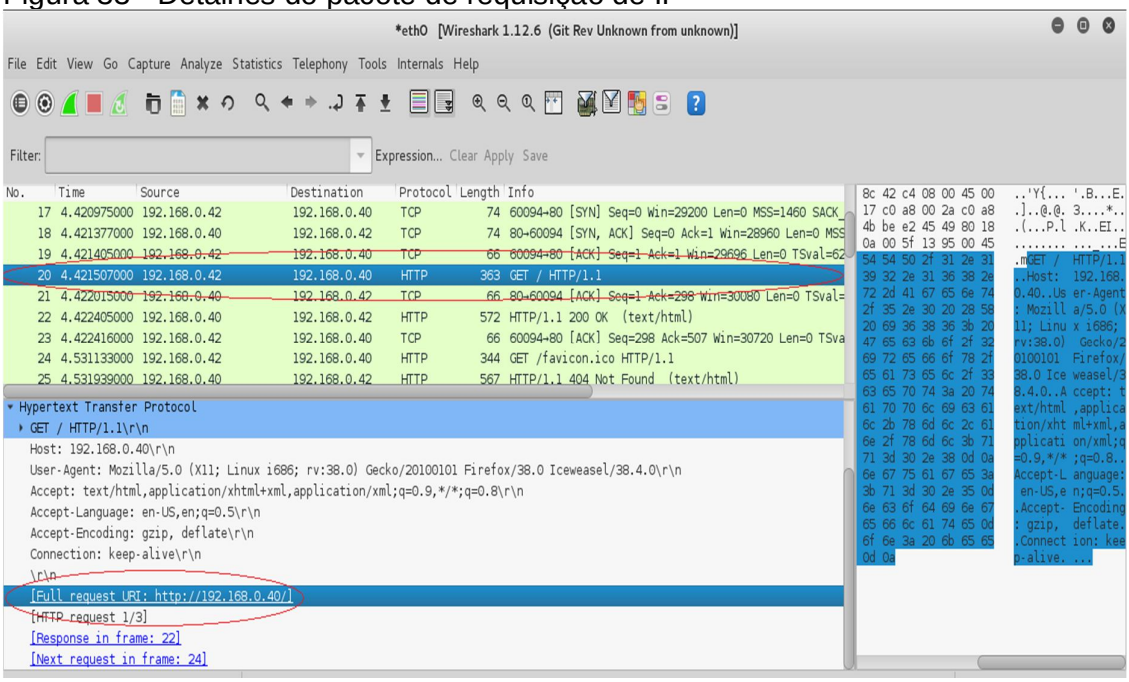
Figura 32- Página welcome.php



Fonte: Elaborada pelo autor

Esta página resumidamente traz todas as informações que foram digitadas nos formulários da página anterior em uma forma de apresentação de texto. Em seguida foi parado o escaneamento de rede e analisado os pacotes que foram coletados durante este pequeno processo. No total, foram analisados quatro pacotes, que já possuiam todas as informações que o teste buscava. Seguindo a ordem de captura dos pacotes foi possível notar a presença do pacote de requisição de IP, de quando foi acessado o ambiente em primeiro lugar, pelo navegador. A Figura 34 apresenta detalhes do pacote coletado.

Figura 33 - Detalhes do pacote de requisição de IP

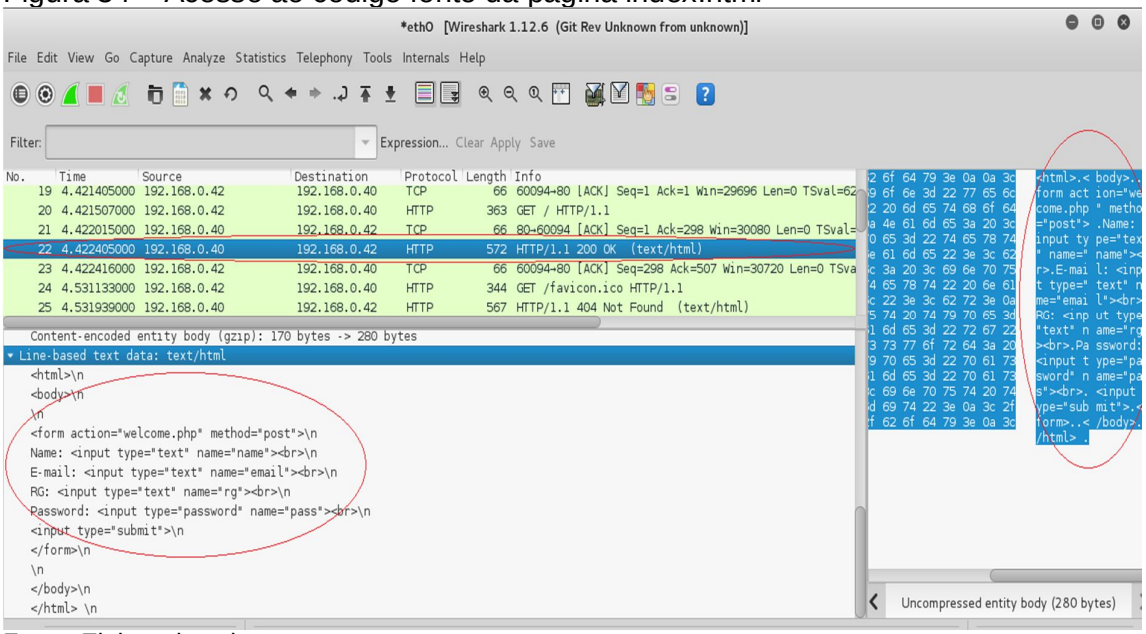


Fonte: Elaborada pelo autor.

É possível notar pela demarcação em vermelho superior da Figura 33 o número do pacote (20), a fonte dessa requisição, no caso, o browser que acessou o ambiente, destino (servidor), protocolo utilizado (HTTP) e a utilização do método GET. A demarcação inferior está associada ao mesmo pacote com a adição de algumas informações relativas ao *browser*.

O próximo pacote analisado foi o pacote de número 22, demarcado na Figura 35.

Figura 34 – Acesso ao código fonte da página index.html

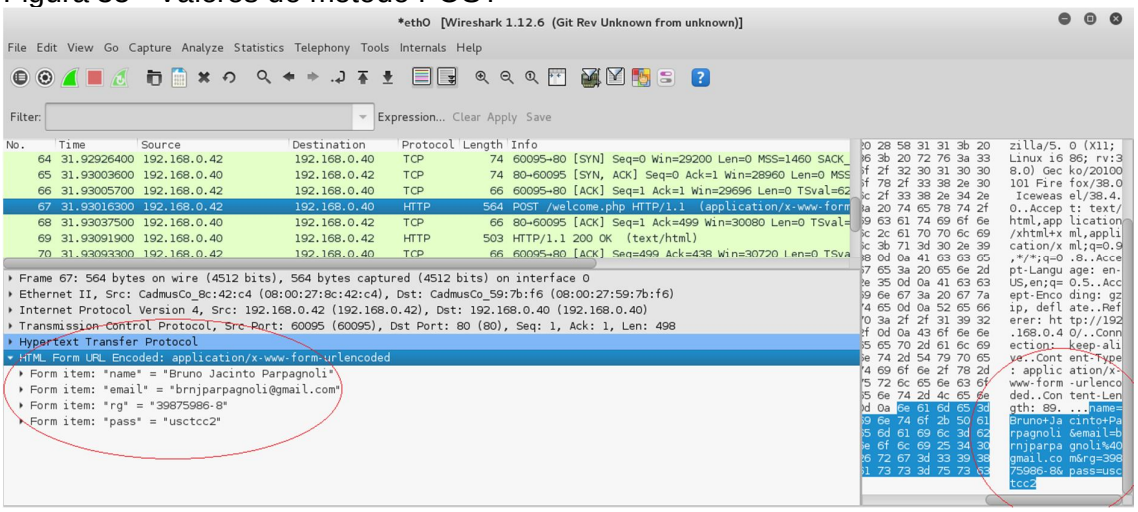


Fonte: Elaborada pelo autor.

Acessando os detalhes deste pacote foi possível ter acesso à todas as informações do código fonte presente na página, bem como todas propriedades dos formulários, o método utilizado, no caso, POST e também o nome da página que receberia as informações que foram inseridas nos formulários, welcome.php. No canto direito da figura está também apresentado o código-fonte da página, porém em código de máquina, porém sua tradução se encontra ao lado.

Na Figura 36 apresentada a seguir, foi analisado o pacote de número 67. Ao acessar seus detalhes foi possível ver nas demarcações em vermelho da figura todas informações que estão sendo transmitidas pelo método POST da página relativa à Figura 35 para a página welcome.php.

Figura 35 - Valores do método POST

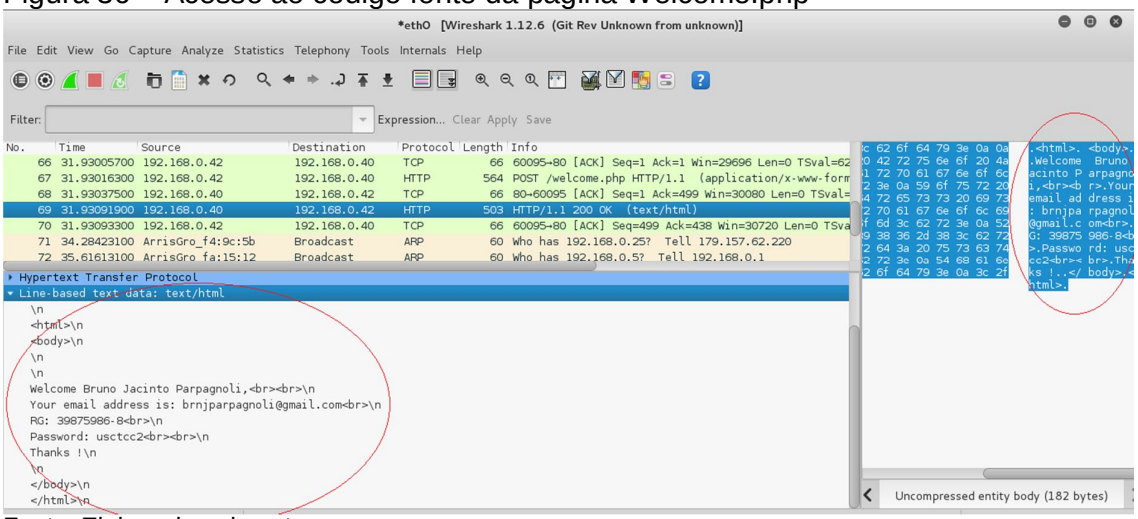


Fonte: Elaborada pelo autor.

Ficou claro nas demarcações os nomes de cada formulário, bem como o valor do que foi inserido no mesmo, no caso, Bruno Jacinto Parpagnoli para *name,* brnparpagnoli@gmail.com para *email*, 39.875.986-8 para *rg* e usctcc2 para *pass*.

O último pacote analisado neste teste foi o de número 69, onde o mesmo é relativo à página welcome.php, que apresenta todos os dados digitados pelo usuário em forma de apresentação, conforme demonstra a Figura 37.

Figura 36 – Acesso ao código fonte da página Welcome.php



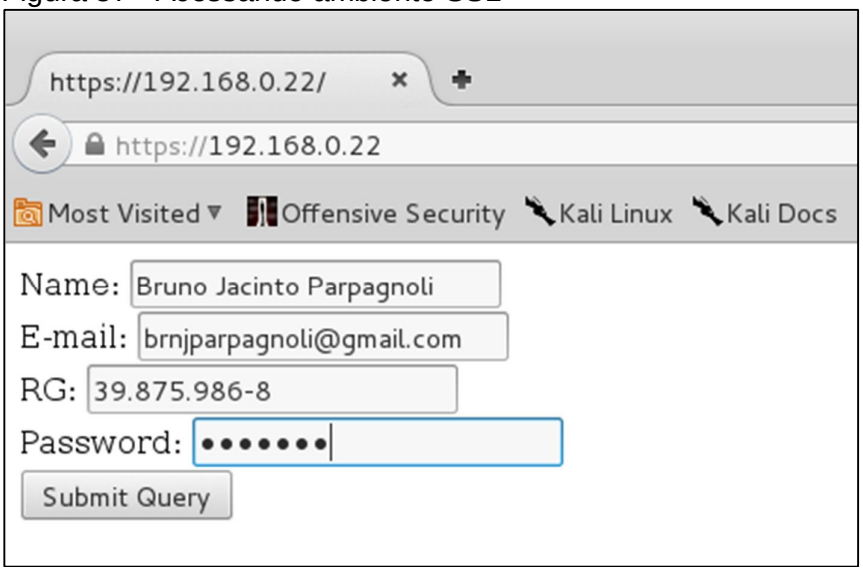
Fonte: Elaborada pelo autor.

Neste pacote também fica evidente após acessado os detalhes do mesmo, todo código fonte da página na área demarcada da figura, contendo todos os dados digitados previamente pelo usuário.

11.2.2 Análise com SSL

Após ativado uma segunda vez o escaneamento da rede pelo Wireshark, foi acessado pelo browser (navegador) da máquina Kali Linux o endereço de IP https://192.168.0.22 associado ao servidor web que continha o ambiente com o protocolo de rede SSL ativado desta vez. Foi então, digitado nos formulários *Name, E-mail, RG, Password,* respectivamente os valores: Bruno J. Parpagnoli, Brnjparpagnoli@gmail.com, 39.875.986-8, USCTCC2 conforme ilustra a Figura 38.

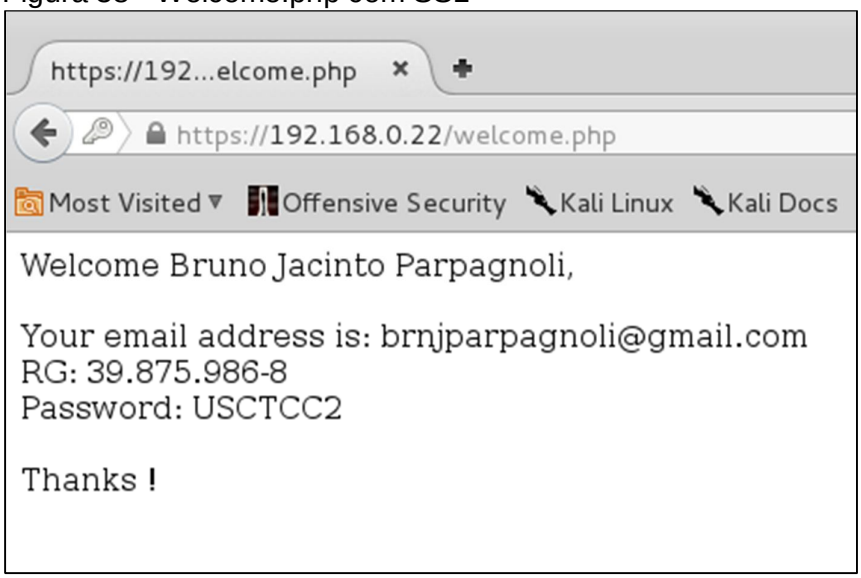
Figura 37 - Acessando ambiente SSL



Fonte: Elaborada pelo autor.

Seguindo os mesmos passos do processo de análise com o protocolo SSL desativado, após inserido os devidos valores em cada formulário, foi acessado o botão “Submit Query”, responsável por gerar o tráfego na rede pelo método POST. A Figura 39 demonstra o resultado seguinte.

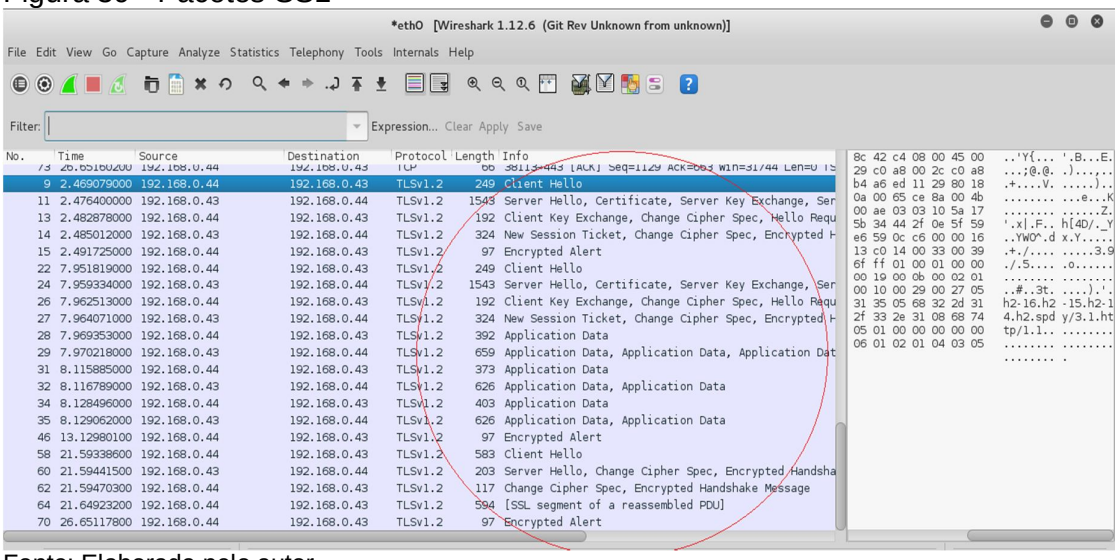
Figura 38 - Welcome.php com SSL



Fonte: Elaborada pelo autor.

Feito isso, foi interrompido o escaneamento de rede realizado pelo Wireshark. Após analisar os resultados obtidos, foi possível perceber ao fazer uma filtragem de pacotes por “protocolo”, que o protocolo TSLv1.2 possuía todos os processos de criptografia, autenticação e protocolos SSL estudados como apresenta a Figura 40.

Figura 39 - Pacotes SSL



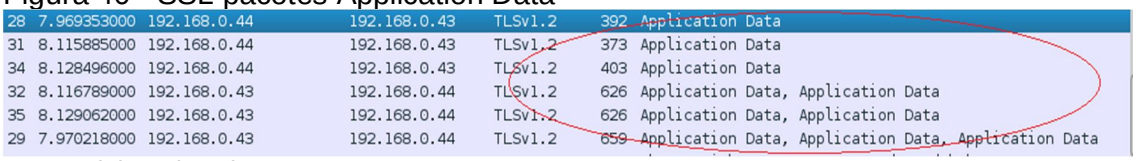
Fonte: Elaborada pelo autor.

Foi notada a presença de todos os protocolos de segurança providos pelo SSL dentro da demarcação em vermelho na figura. O protocolo Client Hello, junto

com Server Hello, Certificate, Server Key Exchange e etc., responsáveis por determinar os parâmetros de criptografia que serão utilizados na comunicação, junto com as trocas de chaves, públicas e privadas que serão utilizadas. É notado também o protocolo Change Cipher Spec, que avisa que os métodos de comunicação já foram estabelecidos e a comunicação será criptografada logo após de “Encrypted Alert” que estabelece o início formal nova comunicação.

Os pacotes mais importantes no teste estão apresentados na Figura 41.

Figura 40 - SSL pacotes Application Data

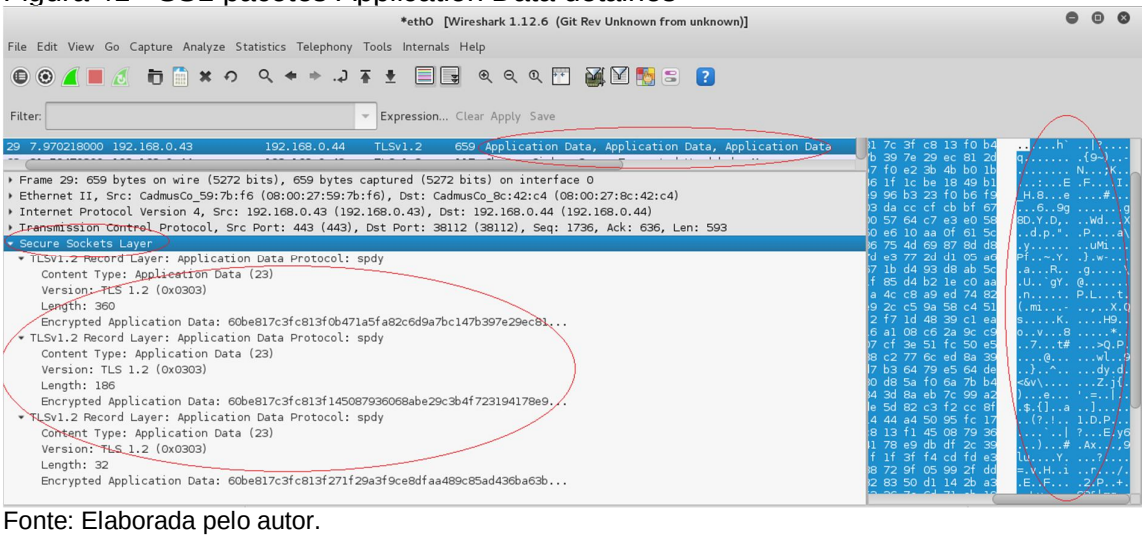


Fonte: Elaborada pelo autor

Os pacotes denominados *Application Data*, carregam as informações reais, trocadas entre as duas entidades, que é criptografada e autenticada pelo SSL. É esperado que as informações que foram previamente digitadas pelo usuário nos formulários, estejam contidas nesses pacotes, junto com outras requisições e informações relacionadas a qualquer troca de informação ou requisição que tenha ocorrido entre cliente e servidor, no caso, o *browser* do Kali Linux e a VM Ubuntu.

A Figura 42 possui os detalhes contidos dentro destes pacotes.

Figura 41 - SSL pacotes Application Data detalhes



Fonte: Elaborada pelo autor.

Ao acessar os detalhes da demarcação em vermelho com conteúdo *Secure Sockets Layer* foi notada a presença de três informações que foram criptografadas em *Encrypted Application Data*, isso explica o porquê no pacote em si na demarcação superior da figura existe o nome *Application Data* repetido três vezes. Mesmo após de vasculhar por todo as informações do pacote não foi possível extrair nada de útil em relação aos valores referentes aos formulários, apenas seu valor criptografado e seu tamanho, também presente na demarcação à direita da figura, isso inclui todos os pacotes ilustrados na Figura 41.

12 CONSIDERAÇÕES FINAIS

Este trabalho teve como objetivo principal analisar o protocolo SSL, encontrado na comunicação existente entre duas entidades dentro da rede, verificando se o mesmo atende as necessidades de segurança que são requeridas hoje em dia, uma época em que uma grande quantidade de informações vitais está trafegando pela rede. Para tal, foi necessário como objetivo paralelo, ampliar os conhecimentos sobre o funcionamento do mesmo para que fosse possível fazer a análise correta das vantagens de sua utilização e implementação do mesmo em um ambiente para realização dos testes.

O primeiro teste realizado no ambiente sem a proteção SSL demonstrou a falta de recursos de segurança que protegem não só as informações que trafegaram entre cliente/servidor, mas também os códigos fonte das páginas acessadas, que se encontravam no servidor web Apache. Por padrão, também não foi encontrando nenhum protocolo “padrão” pela ferramenta Wireshark que se fosse capaz de contribuir com o mínimo de segurança para proteger essas informações. Sem muita dificuldade o pesquisador foi capaz de coletar todas informações trocadas na comunicação que ocorreu na rede.

No segundo teste, realizado com o protocolo SSL ativado no ambiente, após a análise dos pacotes coletados, o primeiro item notado foi a aparição de pacotes contendo o protocolo TSLv1.2, junto desses pacotes estavam listados todos os subprotocolos de segurança SSL estudados durante a pesquisa, bem como o *Handshake Protocol*, responsável por realizar a configuração dos métodos de criptografia que foram utilizados na comunicação, o algoritmo de *hash*, para autenticação e a troca de certificados e chaves, em seguida o pacote contendo *Encryption Alert* que alertava o início da criptografia a partir daquele ponto, e por último os pacotes *Application Data* que continham dados criptografados referentes a qualquer tráfego de dados entre as duas entidades que ocorreu dentro do ambiente.

Após a comparação de resultados coletados entre os ambientes ficou claro como se torna perigoso a navegação web por domínios que não possuem este protocolo de segurança, pois com pouco esforço o pesquisador foi capaz de coletar dados muito importantes realizados as duas entidades sem a utilização do SSL, estando apenas na mesma rede, ou seja, existe o perigo de que informações trocadas com domínios sem essa segurança estejam sendo “ouvidas” por terceiros,

por isso fica evidente a vantagem de navegar e/ou possuir domínios com o protocolo SSL, pois o mesmo deixou explícito sua capacidade de não somente criptografar dados importantes mas como ser o responsável pela certificação de que cada uma das partes seja realmente quem ela diz ser, no caso realizando a autenticação pelo menos do servidor (onde a autenticação do cliente é opcional) através da emissão de certificados válidos por autoridades certificadoras e, por último, a utilização de algoritmos *hash* que possibilita notar a evidência de fraude onde mesmo que informações sejam comprometidas, se a mesma for alterada por terceiros ficará evidente após a comparação dos *hash's* obtidos.

Essas foram todas as vantagens na utilização do protocolo SSL reconhecidas pelo pesquisador através deste trabalho onde a metodologia utilizada foi o suficiente para elaboração deste comparativo.

É importante saber da possibilidade de ocorrerem trabalhos futuros, onde o mesmo ambiente aqui desenvolvido pode ser analisado através de muitas outras ferramentas, não somente as disponíveis no Kali Linux, gerando diferentes dados a serem analisados e adicionados ao comparativo aqui estabelecido.

REFERÊNCIAS

ANDRADE, Eder. **História da Criptografia**. Disponível em: <http://www.dsc.ufcg.edu.br/~pet/jornal/abril2014/materias/historia_da_computacao.html>. Acesso em 27 de maio 2015.

AQUINO J, HOMOLKA, H.; O Ivanildo SOUZA, I. J; LIMA, M F.. **Certificação Digital: Conceitos e Aplicações, Modelos Brasileiro e Australiano**. 1º edição, Rio de Janeiro: Ciência Moderna, 2008.

BERNARDINELLI, M. **Segurança da Informação no Ambiente da Internet com Ênfase em Certificado Digital.** 2010. 62 f. Monografia (Graduação em Processamento de Dados) – Faculdade de Tecnologia do America, 2010. Disponível em: <http://www.mariolb.com.br/mlb/upload/Monografia-SegInfoAmbInternetCertDigital-MarioCesar.pdf>. Acesso em: 28 maio 2015.

CAMPOS, A. **O que é Linux**. BR-Linux. Florianópolis, mar. 2006. Disponível em: <http://br-linux.org/faq-linux>. Acesso em: 28 maio 2015.

CARDOSO D. **Criptoanálise em redes sem fio utilizando lookup tables,** 2013.

CARISSIMI, A. – **Virtualização: Da Teoria a Soluções**. Disponível em: <http://www.jvasconcellos.com.br/unijorge/wp-content/uploads/2012/01/cap4-v2.pdf>. Acesso em: 28 maio. 2015.

CENTRO DE ESTUDO, RESPOSTA E TRATAMENTO DE INCIDENTES DE SEGURANÇA NO BRASIL. **Incidentes Reportados ao CERT.br**: Estatísticas dos Incidentes Reportados ao CERT.br, [2000]. Disponível em: <http://www.cert.br/stats/incidentes/> Acesso em: 30 maio 2015.

CISCO – NETWORKING ACADEMY. **CCNA Exploration 4.0**. Disponível em: <https://135603.netacad.com/courses/24731>. Acesso em: 10 maio 2015.

DIAS, M. A. L.; MALHEIROS, M. G. – **Extração Automática de Palavras-chave de Textos da Língua Portuguesa** 2006

FOROUZAN, B. A. **Comunicação de Dados e Redes de Computadores**. 4. ed. São Paulo: Mcgraw-hill, 2008.

IETF. Internet Engineering Task Force. Disponível em: <https://tools.ietf.org/html/rfc6101#section-5.5> Acesso em: 28 maio 2015.

IMPRENSA OFICIAL – Disponível em: <https://www.imprensaoficial.com.br/PortalIO/Certificacao/Sobre/Apresentacao_7_.aspx>. Acesso em: 12 maio 2015.

KALI. **Kali Linux Official Document** – Disponível em: <http://br.docs.kali.org/introduction-pt-br/o-que-e-o-kali-linux>. Acesso em: 28 maio 2015.

KEA – **Keyphrase Extraction Algorithm** – Disponível em: <http://www.nzdl.org/Kea/>. Acesso em: 25 maio 2015.

KUROSE, J. F.; ROSS, K. W. **Redes de Computadores e a Internet: Uma Abordagem Top-down**. São Paulo: Pearson Addison Wesley, 2006.

LARGURA, L. **Monografia sobre SSL para o Curso de Extensão - Segurança em Redes de Computadores.** Brasília, 2010.

LOVINS, J. B. **Development of a Stemming Algorithm In Mechanical Translation and Computational Linguistics.** Massachusetts, Massachusetts Institute of Technology, 1968.

MORENO, E. D.; PEREIRA, F. D.; CHIARAMONTE, R. B. **Criptografia em Software e Hardware**. São Paulo: Novatec, 2005.

MORENO, T. Faturamento do e-commerce brasileiro. **Revista Exame**, São Paulo, 2015. Tecnologia. Disponível em: <http://exame.abril.com.br/tecnologia/noticias/faturamento-do-e-commerce-brasileiro-cresce-24-em-2014>. Acesso em: 30 maio 2015.

PETERSON, L. L.; DAVIE, B. S. **Redes de Computadores: Uma Abordagem de Sistemas**. Rio de Janeiro: Elsevier, 2004.

PINHEIRO, F. V.; VIEIRA, G. S.; SILVA, L. G. **SSL & TLS**. 2011 – Disponível em: <http://www.gta.ufrj.br/grad/11_1/tls/index.html>. Acesso em: 27 maio 2015.

SAMPAIO, E. **Criptografia: Conceito e Aplicações**. 2013. Disponível em: <http://www.devmedia.com.br/criptografia-conceito-e-aplicacoes-revista-easy-net-magazine-27/26761> Acesso em: 20 maio. 2015

SANDERS, C. **Practical Packet Analysis**. 2ªed. Canada: Editor William Pollock, 2011.

SMITH, J.; NAIR, Ri. **Virtual Machines: Versatile Platforms for Systems and Processes**. Elsevier, 2005.

SOARES, L. F. G.; LEMOS, G.; COLCHER, S. **Redes de Computadores**. 2ª ed. Rio de Janeiro: Editora Campus, 1995.

SSL. **O básico do protocolo SSL** – Disponível em: <http://www.webartigos.com/artigos/o-basico-do-protocolo-ssl/51248/> Acesso em: 27 maio 2015.

STALLINGS, W. **Criptografia e Segurança de Redes**. São Paulo: Pearson Prentice Hall, 2008.

TANEMBAUM, ANDREW S. **Redes de Computadores**. 3ª ed. Rio de Janeiro: Editora Campus, 1997.

TANENBAUM, A. S. **Redes de Computadores**. 4ª ed. Rio de Janeiro: Elsevier, 2003.

UBUNTU – Disponível em: <http://www.ubuntu.com/server/>. Acesso em: 28 maio 2015.

ULBRICH, H. C.; DELLA VALLE, J. **Universidade H4ck3r**. São Paulo: Digerati Books, 2004.

WIRESHARK – Disponível em <https://wireshark.org>. Acesso em: 23 nov. 2015.

WITTEN I. H.: **Practical Automatic Keyphrase Extraction**. In: Proceedings of the Fourth ACM Conference on Digital Libraries, 1999.