REDE DE ENSINO DOCTUM

FACULDADES INTEGRADAS DE CARATINGA

HENRIQUE DE JESUS DIAS

METODOLOGIA PARA TESTES DE INTRUSÃO EM APLICAÇÕES WEB ATRAVÉS DE UMA PERSPECTIVA DE SEGURANÇA DA INFORMAÇÃO

CARATINGA

2017

HENRIQUE DE JESUS DIAS

METODOLOGIA PARA TESTES DE INTRUSÃO EM APLICAÇÕES WEB ATRAVÉS DE UMA PERSPECTIVA DE SEGURANÇA DA INFORMAÇÃO

Trabalho de Conclusão de Curso apresentado a Faculdades Integradas de Caratinga, como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Msc. Elias de Souza Gonçalves

CARATINGA

2017

rede de ensino DOCTUM

FACULDADES INTEGRADAS DE CARATINGA

TERMO DE APROVAÇÃO

O trabalho de Conclusão de Curso intitulado: METODOLOGIA PARA TESTES DE INSTRUSÃO EM APLICAÇÕES WEB ATRAVÉS DE UMA PERSPECTIVA DE SEGURANÇA DA INFORMAÇÃO, elaborado pelo aluno HENRIQUE DE JESUS DIAS, foi aprovado por todos os membros da Banca Examinadora e aceita pelo curso de Ciência da Computação das Faculdades Integradas de Caratinga, como requisito parcial da obtenção do título de

BACHAREL EM CIÊNCIA DA COMPUTAÇÃO.

Caratinga, 13 de Dezembro 2017

Elias de Souza Gonçalves

Prof. Orientador

Fabrícia Pires Souza

Prof. Examinador 1

Wanderson Miranda Nascimento

Prof. Examinador 2

AGRADECIMENTOS

Agradeço a todos os professores pelo conhecimento compartilhado, aos meus pais, que com certeza sem eles não seria possível alcançar este objetivo, à minha namorada Laís por me incentivar sempre, ao Gustavo que me ajudou quando mais precisei, e principalmente a Deus.

DEDICATORIA

Dedico este trabalho ao meu pai Geraldo, minha mãe Rosangela, minha irmã Stella e a minha amada namorada Laís.

RESUMO

Com o aumento exponencial da utilização dos meios computacionais para realização de atividades que envolvem os pilares da segurança da informação (autenticidade, confidencialidade, integridade, disponibilidade) na internet, os criminosos virtuais se aproveitaram da nova tendência para empregar seus conhecimentos técnicos e tirar proveito de diversas vulnerabilidades nesses sistemas. Muitas vulnerabilidades podem ser corrigidas com atualização do sistema e instalação de pacotes de correção. Buscou-se realizar uma revisão bibliográfica com o intuito de expor os conceitos de segurança da informação, ameaças, vulnerabilidades, além da aplicação de metodologias existentes e de sua importância para manter um sistema de informação seguro. Através da aplicação de tais testes é possível identificar e corrigir vulnerabilidades antes que um atacante mal intencionado tire proveito das mesmas. Como foi demonstrado em aplicação prática do *pentest,* o mesmo pode auxiliar a encontrar vulnerabilidades em aplicações *Web* e tem sua efetividade comprovada, uma vez que através dos testes foi possível analisar a segurança da aplicação no que tange a proteção das informações e dos usuários. Espera-se que a pesquisa seja utilizada em caráter de aprendizado para todos os interessados no assunto e não para motivos torpes.

Palavras-chave: Segurança da Informação; Teste de intrusão; Teste de invasão em sistema Web; Pentest Web; Penetration-Test.

ABSTRACT

With the exponential increase in the use of computing resources to carry out activities that involve the pillars of information security (authenticity, confidentiality, integrity, availability) on the Internet, virtual criminals have taken advantage of the new tendency to employ their technical knowledge and take advantage of vulnerabilities in these systems. Many vulnerabilities can be fixed by updating the system and installing patch packs. A bibliographic review was sought to expose the concepts of information security, threats, vulnerabilities, as well as the application of existing methodologies and their importance in maintaining a secure information system. By applying such tests it is possible to identify and fix vulnerabilities before a malicious attacker takes advantage of them. As demonstrated in the practical application of pentest, it can help to find vulnerabilities in Web applications and has its proven effectiveness, once through the tests it was possible to analyze the security of the application regarding the protection of information and users. It is hoped that the research will be used as a learning tool for all those interested in the subject and not for stupid reasons.

Keywords: Information Security; Intrusion test; Invasion test in Web system; Pentest Web; Penetration-Test.

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE ABREVIATURAS

CERT	Centro de Estudos, Resposta e Tratamento de Incidentes no Brasil
ESG	ESG Corp – Gestão, Tecnologia e conhecimento ao seu alcance
IDC	*International Data Corporation*
IDS	*Intrusion Detection System*
ISO	*Institute Standard Organization*
NDA	*Non Disclosure Agreement*
TCU	Tribunal de Contas da União
WEB	*World Wide Web*

SUMÁRIO

INTRODUÇÃO

A era da tecnologia trouxe consigo uma grande dependência dos sistemas computadorizados e da informação digital. O valor de uma informação é pertinente aos seus respectivos donos, e por vezes é imensurável, portanto, a proteção da mesma é de extrema importância, uma vez que se divulgada, pode perder parte do seu valor.

De acordo Mahidhar, Schatsky e Bissell (2013), os ataques cibernéticos estão se tornando cada dia mais sofisticados devido aos novos métodos de distribuir, ocultar e evitar a detecção dos *malwares*. Segundo os investigadores de segurança da CISCO (2017) ciberatacantes estão utilizando técnicas de engenharia social para aproveitar da inocência de usuários de aplicativos e redes sociais, incentivando-os a clicar em *links* ou abrir anexos infectados por algum tipo de vírus de computador.

O grande problema é que com esse grande avanço tecnológico e fácil acesso à internet, a quantidade registrada de ataques cibernéticos cresceram gradativamente e esses números mostram a necessidade de estar se protegendo e tomando certos cuidados ao utilizar um computador na internet.

Segundo o estudo "*Networking Skills*", encomendado pela Cisco à IDC cerca de 49% do déficit de profissionais de rede, em 2015 na América Latina, foi pela falta de profissionais capacitados na área de segurança da informação. Estima-se também que em 2019 estes profissionais ainda representarão um déficit de 46%. O estudo aponta ainda, que 86% das empresas na América Latina já possuem plano de contingência para segurança. Porém, apenas 42% desses indicaram que gerenciam as vulnerabilidades dos sistemas. O que evidencia a importância de um profissional em segurança da informação especializado.

Problema de pesquisa

Então como saber se um *Web* sistema/site é confiável/seguro e não colocará em risco todos os recursos computacionais que compõe seu ambiente?

Os testes de intrusão são um conjunto de ataques do administrador da rede e dos sistemas computacionais com o intuito de identificar vulnerabilidades, falhas, problemas em sistemas operacionais e configurações mal feitas em softwares e hardwares sob sua gestão.

Objetivos

A realização deste trabalho depende de objetivos gerais e específicos que ao serem alcançados, permitirão concluir sobre os benefícios e importância dos testes de intrusão.

Geral

Aplicar uma metodologia de *pentest* em um estudo de caso e através do mesmo demonstrar os conceitos e como tais testes auxiliam na verificação da segurança de um site/sistema *Web*, além de conscientizar os desenvolvedores sobre a importância de realizar testes específicos sobre o site/sistema.

Específicos

- Pesquisar e realizar uma revisão bibliográfica sobre as áreas de redes de computadores, segurança da informação e *pentest* principalmente;
- Verificar as metodologias de teste de intrusão;
- Analisar os resultados obtidos.

Organização do trabalho

Este trabalho foi subdividido em três capítulos: O capítulo um é o referencial teórico, onde serão citados e definidos os conceitos necessários para compreensão e leitura do trabalho. São definidos os conceitos de: Segurança da informação; Ameaças, vulnerabilidades e ferramentas de controle; Técnicas e tecnologias de defesa; Políticas de segurança da informação. Em segurança da informação, serão abordados os conceitos, características e sua importância.

No capítulo dois, é definida a metodologia proposta no trabalho, assim como o ambiente de testes, a definição das vulnerabilidades testadas, além dos tipos de testes realizados buscando a constatação de tais vulnerabilidades. É descrito ainda a configuração utilizada nas máquinas virtuais.

No capítulo três, denominado Experimentos e resultados serão detalhadas as aplicações dos testes propostos no capítulo dois (Metodologia), bem como seus

objetivos, pré-requisitos e os resultados obtidos através da aplicação dos testes, no ambiente realizado.

1 REFERÊNCIAL TEÓRICO

A Revisão ou Levantamento de Literatura é a localização e obtenção de documentos para avaliar a disponibilidade de material que subsidiará o tema do trabalho de pesquisa. Este levantamento é realizado junto às bibliotecas ou serviços de informações existentes e visa demonstrar conceitos introdutórios, que servirão como base para a leitura e compreensão de todo o trabalho.

1.1 Segurança da Informação e seus parâmetros

Segundo a *International Organization for Standardization* (ISO, 2013) a informação e os processos relacionados, sistemas, redes e pessoas envolvidas nas suas operações, são informações que, como outros ativos importantes, têm valor para o negócio da organização e, consequentemente, requer proteção contra vários riscos. Com o avanço tecnológico e interconexão propiciada pelo uso da internet, a informação está sujeita a diversos tipos de ameaças.

Um sistema denominado seguro é aquele que provê acesso à informação somente ao indivíduo autorizado. Se o sistema permite acesso ao usuário que deveria ser negado, então o mesmo é denominado inseguro.

Um sistema de segurança da informação deve-se basear em quatro princípios básicos: Disponibilidade, Integridade Confidencialidade e Autenticidade.

1.1.1 Características da informação

A segurança da informação é baseada em quatro pilares essências que quando violados ocorre-se uma quebra da segurança da informação, esses são: integridade, disponibilidade, confidencialidade. Esses atributos são fundamentais para a definição e instauração de um sistema de informação seguro e livre das ameaças, mas não se aplicam somente as informações digitais, engloba também a proteção das mesmas em forma física, além dos ativos informáticos que provém acesso a todas essas informações.

Segundo o TCU (2012), a integridade consiste na fidedignidade de informações. Sinaliza a conformidade de dados armazenados com relação às inserções, alterações e processamentos autorizados efetuados. Para Gonçalves (2014), integridade é a garantia de que se o dado está lá, então se encontra íntegro.

Isto quer dizer que o dado é autêntico e nada foi acrescentado, retirado ou modificado.

De acordo com o TCU (2012), a disponibilidade é caracterizada pela garantia de que as informações estejam acessíveis a todos os que são autorizados, em qualquer momento requerido. Para Alencar (2011), a disponibilidade se define na garantia de que as informações estejam disponíveis para acesso das pessoas e processos autorizados, a qualquer momento que seja solicitado, e sua perda ocorre quando uma pessoa autorizada tenta fazer um acesso à determinada informação e não consegue.

O TCU (2012) define que a confidencialidade consiste na garantia de que somente pessoas autorizadas tenham acesso às informações armazenadas ou transmitidas por meio de redes de comunicação. Manter a confidencialidade pressupõe assegurar que as pessoas não tomem conhecimento de informações, de forma acidental ou proposital, sem que possuam autorização para tal procedimento. Reforçando essa ideia, Maia (2013) defende que este conceito é diferente de um segredo ou algo inacessível, é um conceito que restringe o acesso à informação a quem tem direito, ou seja, somente as pessoas autorizadas pelo proprietário da informação.

Como mostra o TCU (2012), autenticidade consiste na garantia da veracidade da fonte das informações. Por meio da autenticação é possível confirmar a identidade da pessoa ou entidade que presta as informações. No mesmo sentido, Oliveira (2016) destaca que este conceito assegura a identidade do emissor da informação, ou seja, gera o não-repúdio que se dá quando há garantia de que o emissor não poderá negar autoria da mensagem.

1.1.2 O processo de zelo da segurança da informação

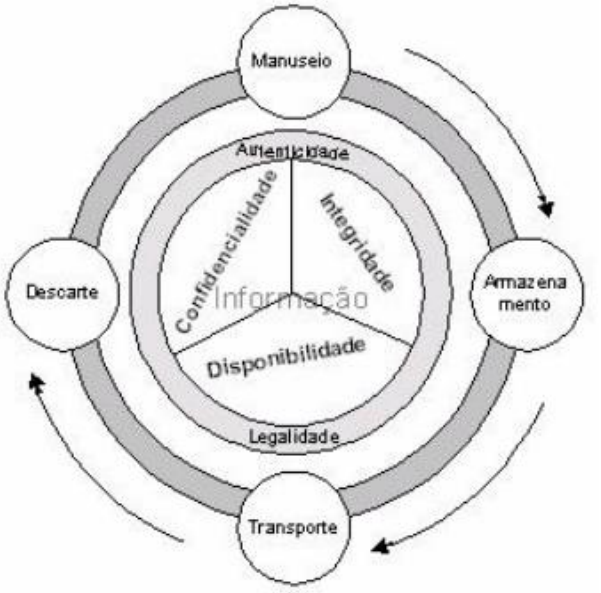
A ISO/IEC 13335-1/2004 caracteriza como ativo qualquer coisa que tenha valor para a organização. Segundo o TCU (2012), a informação é um ativo de extrema importância para qualquer organização, é considerado um patrimônio crítico que quando em posse de pessoas não autorizadas pode ser alterada, divulgada e podendo assim denegrir a imagem da empresa perante os *stakeholders* e até inviabilizar um segmento de negócio. De acordo com a afirmação de Deresky (2004,

p. 25) “a segurança passa a ser crítica na gestão da informação organizacional”. Então assim, é importante enaltecer a utilização de sistemas de segurança como abordagem estratégica para uma gestão proativa dos dados organizacionais e informações.

1.1.3 Ciclo de vida da informação

Fazem parte do ciclo de vida da informação, todos os momentos e etapas vividos pela informação e que a colocam em risco constante. Segundo Sêmola (2003), os ativos físicos, tecnológicos e humanos vivenciam esses momentos quando fazem uso da informação, desenvolvendo os processos que são inerentes aos negócios da organização. A Figura 1 demonstra os momentos e ciclo de vida da informação.

Figura 1 – Momentos do ciclo de vida da informação



Fonte: Juliano (2014)

Dantas (2011), diz que a informação possui um ciclo de vida que começa com sua produção, tem um tempo de vida útil, onde é manuseada, utilizada internamente e externamente, transportada por diversos meios, armazenada, e morre com a sua destruição.

Marcondes (2016) define o ciclo de vida da informação em quatro etapas. Manuseio, nesta etapa a informação é criada e manipulada, seja ao manusear uma

pilha de papéis, utilizar senha de autenticação para acesso ou ainda ao digitar informações recém-geradas em uma aplicação *Web*, por exemplo. Armazenamento, nesse momento do ciclo de vida a informação é armazenada em banco de dados compartilhado, em folha de papel e posteriormente arquivada ou ainda em algum dispositivo de mídia.

Transporte, momento em que a informação é transportada. Pode ser um encaminhamento de e-*mail*, ao falar pelo telefone uma informação ou postar um documento por fax, por exemplo. Descarte, momento em que a informação perde seu valor e é descartada, seja ao jogar um material impresso na lixeira, ao eliminar um documento eletrônico do computador ou ao descartar uma mídia que apresentou determina falha, por exemplo.

1.1.4 Vulnerabilidade da informação

Vulnerabilidades são pontos fracos, onde há uma ou demais falhas presentes e pode acarretar em prejuízo de algum ativo da organização. A definição da NBR ISO/IEC 27002:2005 para vulnerabilidade é como uma fragilidade de um determinado ou grupo de ativos que pode ser explorado.

Sêmola (2003) define as vulnerabilidades como fragilidades presentes ou associadas a ativos de informação, que, ao serem exploradas, permitem a ocorrência de incidente na segurança da informação.

Dantas (2011) afirma que vulnerabilidades são fragilidades que podem provocar danos decorrentes da utilização de dados em qualquer fase do ciclo de vida da informação.

Conforme pode ser observado, nota-se que as vulnerabilidades estão relacionadas intrinsecamente com as fragilidades. As fragilidades podem estar presentes em softwares, hardwares, usuários e processos. Então para manter os pilares da segurança da informação intactos, a mesma deve ser encarada como um projeto contínuo que deve estar em constante avaliação, monitoramento e evolução.

1.2 Ameaças, Vulnerabilidade e Ferramentas de Controle

Esta seção aborda conteúdos bibliográficos que demonstram os conceitos publicados sobre os tipos de ameaças a sistemas *Web*, vulnerabilidades e as técnicas de proteção.

1.2.1 Ameaças

Segundo a ISO uma ameaça é uma potencial causa de acidente, que resulta na perda ou dano de um sistema computacional ou organização, e uma vulnerabilidade é uma fraqueza de um bem ou conjunto de bens, que podem ser explorados por uma ou mais ameaças.

Segundo Bishop (2002), uma ameaça é um ponto que leva a uma possível exploração. Não é necessariamente uma exploração para que se tenha uma ameaça. Somente o fato de que a exploração possa ocorrer, significa que as ações que levam a esta possível exploração devem ser tratadas.

Lehtinen (2006), diz que a segurança de computadores consiste em identificar as vulnerabilidades em sistemas e assim protegê-las contra as possíveis ameaças. Assim sendo, é primordial para a segurança de um computador na rede compreender como o mesmo pode estar vulnerável a sucessivos ataques e contornar essas ameaças com medidas proativas de segurança.

1.2.2 Vírus

Os vírus de computador são trechos de código desenvolvidos para tirar proveito e/ou realizar alguma ação indevida no computador da vítima. Embora existam hoje em dia diversos tipos de vírus conhecidos, há algumas divergências quanto ao primeiro vírus de computador desenvolvido. O que se sabe é que desde os primeiros programas maliciosos desenvolvidos, seu modo de criação e disseminação evoluiu muito e pode contaminar uma larga escala de computadores em um tempo relativamente curto através da internet. A infecção do computador da vítima normalmente acontece após a execução do arquivo contaminado com o vírus em anexo, após sua execução tenta se camuflar em meio aos arquivos do sistema operacional, dificultando sua localização e exclusão.

Bishop (2002), diz que um vírus de computador é um programa que se instala em um ou demais arquivos e executa uma ação maliciosa. Segundo Lehtinen (2006), o vírus de computador realiza alguma ação quando o programa hospedeiro é executado na máquina. Lehtinen (2006), diz ainda que o vírus pode degradar o desempenho do sistema, além de causar outros danos. Os vírus de computador se espalham pelo sistema criando cópias de si mesmo indo da memória principal para o

disco rígido e mídias removíveis, espalhando assim a infecção, ocasionando lentidão do sistema, desabilitando o antivírus, bloqueando o acesso a internet, entre outros problemas causados.

1.2.3 Rootkit

Os *rootkits* são ferramentas que possibilitam o atacante a ter total controle do sistema operacional e consequentemente do computador. Eles camuflam certos processos e serviços, com isso torna-se mais difícil sua detecção.

Segundo Kuhnhauser (2004), com sua utilização é possível obter acesso privilegiado, instalação de diversas ferramentas, de *backdoor* até um sistema para proteção contra outros atacantes.

Segundo a Avast (2017), o *rootkit* é um programa desenvolvido para fornecer aos *crackers* acesso administrativo a determinado computador, sem que a vítima saiba.

1.2.4 Trojans

Lemonnier (2015), diz que os trojans não se replicam ao infectarem o computador, muitas vezes eles tentam se manter ocultos, assim podem coletar informações, configurar alguma brecha de segurança (por exemplo, *backdoor*), mas podem também controlar o computador.

Os cavalos de Tróia ou trojans são programas de computador que executam algumas tarefas no sistema alvo, utiliza o meio de propagação semelhante ao do vírus de computador, ele se instala em um programa hospedeiro, com o intuito de ludibriar a vítima e fazê-la executar o mesmo no computador.

1.2.5 Worms

Segundo Bishop (2002), o *worm* é uma variação do vírus e infecta diversos computadores, criando cópias de si nos mesmos.

Sua propagação segundo o CERT (2006) se dá pela exploração de vulnerabilidades conhecidas ou falhas de configuração em produtos de softwares. Diferentemente do vírus o *worm* não necessita ser executado para se proliferar e

não necessita também de um programa hospedeiro e também contém trechos de códigos maliciosos que executam ações no sistema.

1.2.6 Keyloggers e Screenlogger

Os *keyloggers* são um tipo de *malware* de computador que é desenvolvido com a pura intenção de roubar informações que são digitadas pelo usuário. Esses programas embora não afetem diretamente o sistema, são de grande periculosidade, pois podem capturar senhas e *logins* do usuário e enviar para o atacante. Já um *screenlogger* tem sua ação parecida com o *keylogger*, mas ao invés de salvar as teclas pressionadas, o mesmo irá fazer uma captura de tela de acordo com os cliques realizados pelo mouse e enviar para o atacante.

Segundo o CERT (2017), o *screenlogger* armazena a posição do cursor e a tela do monitor, quando o mouse é acionado pelo usuário. São normalmente utilizados em sites que obrigam o usuário a digitar a senha por um teclado virtual.

A Avast (2017) define que um *keylogger* é um *software* malicioso que fica em constante monitoramento de todas as teclas que são digitadas no computador. O programa posteriormente salva as informações monitoradas em um arquivo de log e envia para um servidor de escolha do atacante, onde o mesmo pode ler todo conteúdo anexado.

1.2.7 Backdoors

São trechos de códigos maliciosos instalados no computador alvo, que fazem alusão a um serviço para enganar o sistema e abrir uma porta, que pode ser utilizada posteriormente pelo atacante para acessar o computador remotamente e executar comandos no mesmo.

Segundo Tiller (2005), a maior parte dos ataques bem sucedidos a computadores começa explorando alguma vulnerabilidade conhecida para conseguir acesso a máquina e instalar um *backdoor* (“porta dos fundos”, em português) permitindo o atacante retornar facilmente sem ser detectado.

1.2.8 Bots e Botnets

Segundo Tacio (2017), uma *botnet* é composta por diversos computadores infectados por um *malware* de acesso remoto formando uma rede, e fica aguardando por instruções do dono para realizar diversos tipos de ataques.

Bantim (2012) define *bot* como um tipo de *malware* utilizado pelos *crackers* para obter acesso total ao computador infectado remotamente, e realizar ataques na internet sem o conhecimento do proprietário.

1.2.9 Hijackers

Duarte (2015), diz que o *hijacker* (“sequestrador”, em português) é um tipo de *spyware* que se instala no computador da vítima e geralmente altera a página inicial do navegador, instala barra de ferramentas esquisitas, além de alterar o mecanismo de busca padrão, e comumente depois de algum tempo começa a abrir páginas aleatórias no computador da vítima.

1.2.10 Adwares e Spwares

Segundo a Avast (2017), o *spyware* é um tipo de *malware* utilizado por crackers para fazer espionagem e conseguir acesso a informações confidenciais. A Avast (2017) define também que o *adware* é um tipo de *malware* mantido por propagandas que abrem como *pop-up* ou barra de ferramenta no navegador, e podem fazer espionagem assim como os spywares e até agir como *keylogger*.

1.3 Spans

Segundo o Cert (2017), *spam* refere-se aos *e-mails* não solicitados, estes estão diretamente ligados aos ataques à segurança do usuário, pois estão comumente ligados a propagação de códigos maliciosos, disseminação de golpes e venda ilegal de produtos.

Atualmente a prática de enviar *e-mail* não solicitado é muito utilizada, e pode trazer alguns problemas para o usuário, entre os seus efeitos colaterais pode-se citar: perdas de mensagens importantes; conteúdos impróprios; gasto de tempo

desnecessário; não recebimento de e-mails. O surgimento dos *spans* se deu no ano de 1994 e deste então o mesmo tem evoluído acompanhando as tendências tecnológicas. Os tipos de *spam* identificados até agora serão descritos a seguir.

1.3.1 Correntes

São muito populares esses tipos de *spam* e também muito irritantes. Geralmente pede para o usuário repassar alguma mensagem um determinado número de vezes e que será recompensado de alguma forma após isso.

Segundo o site Antispam (2017), muitas correntes usam a engenharia social como meio de tentar ludibriar o usuário para que o mesmo repasse a corrente.

1.3.2 Boatos e Lendas urbanas

Os boatos ou *hoaxes* são uma variação das correntes. Utilizam da engenharia social como meio de atrair a atenção do usuário, geralmente usam histórias de cunho apelativo para atrair a atenção e assim redirecionar os usuários para a página desejada.

Segundo o Antispam (2017), a diferença entre os boatos e as correntes está na mensagem de seu conteúdo, pois geralmente estes contam histórias alarmantes e falsas, sensibilizando o usuário a continuar a disseminação. Ainda segundo o site Antispam (2017), as lendas urbanas podem se confundir com os boatos, mas estes se diferenciam um pouco, pois geralmente possuem alguma justificativa que tenta atribuir alguma veracidade ao *spam*.

1.3.3 Propagandas

Talvez seja o tipo mais conhecido e encontrado na internet, tem por objetivo oferecer diversos tipos de produtos aos usuários.

Segundo o Antispam (2017), são conhecidos como *Unsolicited Comercial e-mail*, e pode envolver produtos, serviços, pessoas, sites, entre outros. São controversos, pois é possível fazer *marketing* sem fazer *spam*.

1.3.4 Pornografia

Esses são um dos *spans* mais antigos e populares na internet. Muitas vezes seu conteúdo é de pedofilia, abuso de animais, nudez, entre outros. É de muita importância uma política de segurança e controles de acessos para evitar que crianças se deparem com esse tipo de conteúdo na internet.

Segundo o Antispam (2017), é uma das modalidades mais antigas quando se trata de *spam*. E este tópico levanta duas questões importantes: o recebimento deste tipo de conteúdo por crianças e a propagação de material contendo pedofilia. O primeiro caso deve ser tratado com técnicas que regularão o acesso das crianças, enquanto o segundo deve ser notificado o mais rápido possível aos órgãos competentes.

1.3.5 Programas maliciosos

Há os *spans* que são enviados com o intuito de instalar algum código malicioso no computador da vítima para realizar alguma ação indevida. Nesse tipo de *spam* o *cracker* combina diversas técnicas, como: engenharia social, *phishing*, entre outras técnicas e tenta fazer com que o usuário execute o que se pede.

Segundo o Antispam (2017), diversos tipos de códigos maliciosos (*backdoor, spyware, keylogger, screenlogger*) são anexados em e-mails, utilizando da engenharia social pretendendo manipular o usuário para que o mesmo execute o código malicioso. Muitos desses códigos são utilizados também pelos fraudadores.

1.3.6 Fraudes

As fraudes acontecem das mais variadas formas que se pode imaginar, mas ainda assim, depende muito da engenharia social para fazer com que a vítima caia no golpe e execute a ação do fraudador.

Segundo Antispam (2017), para que se consiga obter alguma vantagem, os fraudadores têm utilizado a engenharia social. Empregando discursos apelativos, que tentam persuadir o usuário a digitar dados pessoais e financeiros. Pode ser

ainda que o usuário seja orientado a instalar algum programa malicioso em seu computador ou então seja redirecionado para uma página falsa.

1.4 Vulnerabilidades

As vulnerabilidades são fatores cruciais, que quando não são identificadas e tratadas, fica-se exposto ao risco de que em algum momento alguma ameaça possa tirar proveito da fraqueza constatada.

Segundo Campos (2014, p. 23),

> Os ativos informatizados, que suportam os processos de negócio, possuem vulnerabilidades. Vale destacar que essas vulnerabilidades são inerentes aos ativos e não são geralmente de origem externa. (CAMPOS, 2014, p.23).

Segundo Oliveira (2016), uma vulnerabilidade é um ponto fraco que possibilita um atacante reduzir a garantia da informação do sistema. A vulnerabilidade é a união de três aspectos: uma suscetibilidade ou falha do sistema, acesso do atacante a falha e a capacidade do atacante de explorar esta falha.

As vulnerabilidades possibilitam dois principais tipos de ataques aos ativos da informação: ataques passivos e ataques ativos. Ataques ativos são aqueles em que há interação direta entre o atacante e o sistema alvo, e por consequência ocorre a quebra de um dos pilares da segurança da informação (autenticidade, confidencialidade, disponibilidade, integridade).

Segundo Oliveira (2016), ataques ativos intervêm no fluxo normal da informação, alterando seu conteúdo e assim quebrando a autenticidade e integridade da mesma.

Ataques passivos são aqueles em que não há interação direta entre o atacante e o sistema alvo, o atacante fica na escuta do canal, observando o tráfego das informações. Segundo Oliveira (2016), ataques passivos não alteram o fluxo normal da informação, apenas ficam na escuta do canal de informações.

1.4.1Tecnologias

Atualmente o termo tecnologia é comumente empregado para se referir aos novos aparelhos desenvolvidos para a transmissão de informações, essa definição não está errada, mas o conceito de tecnologia é bem mais abrangente. Segundo

Kruglianskas (1996), a tecnologia é considerada uma rede de conhecimentos necessária para idealizar, produzir e compartilhar bens e serviços de forma competitiva.

A tecnologia traz consigo comodidade para o usuário, mas também pode expor o mesmo através de vulnerabilidades em sua arquitetura, implementação ou configuração. Por exemplo, as aplicações *Web*. (PINTO; STUTTARD, 2007).

Para Pinto e Stuttard (2007), o principal risco de uma aplicação Web está no fato de que o usuário pode enviar entradas totalmente arbitrárias, com o intuito de interferir na lógica e comportamento da aplicação, ocasionando algum tipo de erro que pode ser explorado posteriormente. Além das possíveis falhas de arquitetura, implementação e configuração, as tecnologias podem estar vulneráveis também, por aspectos físicos: fogo, água, explosões, poeira.

1.4.2 Pessoas

Segundo Mitnick e Simon (2003), a engenharia social é considerada a arte de explorar as falhas humanas em vez de explorar as falhas tecnológicas. Seu objetivo é enganar as pessoas, muitas vezes assumindo identidade de outrem para conseguir uma informação privilegiada, por exemplo.

Considerando os usuários, muitas vezes estes são o ponto chave para o sucesso ou fracasso de uma tentativa de invasão. Através de técnicas de engenharia social, por exemplo, os *crackers* se aproveitam de uma vulnerabilidade humana e assim conseguem tirar proveito dessa falha, obtendo determinado acesso e comprometendo os pilares da segurança da informação: ACID, portanto o investimento em capacitação e conscientização dos colaboradores é de vital importância, assumindo que estes podem envolver-se em algum incidente de segurança da informação.

1.4.3 Processos

Os processos englobam os gerenciamentos de riscos, no que diz respeito a tecnologia da informação, visando manter operativo e em segurança os negócios da organização.

Segundo Stoneburner, Goguen e Feringa (2002), o gerenciamento de riscos é o processo de identificar, avaliar e tomar medidas para reduzir o risco de uma ameaça. Seu objetivo é manter em pleno funcionamento as tarefas organizacionais, através da aplicação de segurança para os sistemas de TI.[1]

1.4.4 Ambientes

Os ambientes tecnológicos são os ativos muito importantes nos dias atuais, pois propiciam a organização maior competitividade, eficácia e eficiência na elaboração de produtos e prestação de serviços.

Segundo Rafael (2014), não importa o tamanho do ambiente tecnológico disposto pela organização, o mesmo deve ter parâmetros e critérios de segurança implementados, diminuindo então os riscos de ameaças iminentes, que poderiam causar a paralização dos negócios da organização, ocasionando assim, prejuízo à mesma.

1.4.5 Engenharia Social

Compreende por engenharia social, o método de ataque que visa explorar a vulnerabilidade dos usuários. O *cracker* aposta na ingenuidade da pessoa, persuadindo a mesma, podendo se fazer passar por outra pessoa, com o objetivo de conseguir acessos indevidos.

Segundo definição de Lafrance (2004), a engenharia social explora o fator humano, é o “ataque contra pessoas”, com base na confiabilidade.

1.5 Ferramentas de controle

Controlar os acessos aos recursos computacionais dispostos na rede é de vital importância para manter as ameaças afastadas e os negócios em funcionamento. No atual cenário globalizado, onde tudo é muito dinâmico e as mudanças ocorrem a todo instante, novas ameaças surgem e evoluem constantemente. Isso define que todo cuidado é pouco, quando o assunto é

[1] Tecnologia da informação - A Tecnologia da Informação ou **TI**, é o conjunto de atividades e soluções envolvendo *hardware*, *software*, banco de dados, e redes que atuam para facilitar o acesso, análise e gerenciamento de informações.

[2] O Metasploitable é uma máquina virtual rodando Ubuntu Server 8.04, com diversos softwares em

segurança na internet. Engana-se quem acredita que um bom antivírus é a solução para todos os problemas, como os relatados neste capítulo.

1.5.1 Firewalls

O advento da internet embora tenha revolucionado todo o cotidiano, trouxe consigo novos meios para realização de crimes. Estar conectado a internet pode representar um risco se não for tomado nenhum cuidado para se proteger de ameaças virtuais. Existem diversos mecanismos para se proteger na rede, um deles é o *firewall*.

O *firewall* (“parede de fogo”, em português) é um mecanismo de defesa que controla todo o tráfego de uma rede computadores. Segundo Stamp (2006), o *firewall* analisa as requisições que trafegam pela rede, e com base nas regras definidas na sua implementação, decide se a requisição pode ou não ser aceita.

O *firewall* é um componente de segurança, que funciona como um porteiro de um prédio, por exemplo, sua função é controlar o tráfego, todas as pessoas que desejam entrar no prédio, devem comunicar sua chegada ao porteiro, e o mesmo com base em alguns critérios e regras libera ou não, a passagem para aquela pessoa. Assim é o *firewall*, ele é um gerenciador entre a rede interna e à internet, todo o tráfego que entra e sai da rede interna, passa pelo *firewall*, este então com base nas suas regras de configuração, verifica se a requisição pode ou não ser autorizada.

Segundo Zwicky, Cooper, Chapman (2000), logicamente um *firewall* é um limitador, separador, analisador. E sua implementação depende muito da demanda e dos componentes da rede. Fisicamente, são um conjunto de componentes interligados, trocando informações constantemente, embora algumas empresas tentem fabricar o *firewall* em um componente de *hardware* apenas.

Como toda tecnologia que vai se tornando obsoleta, os *firewalls* necessitaram de evolução, perante as novas vulnerabilidades descobertas em sistemas computacionais e artimanhas criadas pelos *crackers*. Atualmente existem basicamente três principais tipos de *firewall*.

1.5.2 Antivírus

Assim como os *firewalls* os Antivírus são softwares que têm como objetivo manter os computadores livres de vírus, com isso evitar danos ao sistema operacional e arquivos do usuário. Segundo o Canaltech (2017), antivírus é um *software* de computador que detecta, impede e atua no combate e remoção de programas maliciosos e assim prover segurança ao usuário.

Os antivírus embora tenham diversos meios de verificar se um arquivo ou programa está infectado por algum trecho de código malicioso, ainda podem acontecer falsos positivos, ou seja, um arquivo ser confundido com uma ameaça, mesmo sendo confiável. A seguir são descritos os principais mecanismos de identificação de vírus utilizados pelos antivírus.

1.5.2.1 Escaneamento de vírus conhecido

Segundo o Canaltech (2017), quando um novo vírus é identificado, o antivírus pega seu código e o desmembra em vários pedaços menores denominados caracteres de *string*. Essas *strings* não são encontradas em outros programas do computador que estão limpos de vírus. Com posso dessa *string* o antivírus realiza a varredura do sistema e caso encontre a *string* em alguma parte do computador o usuário recebe um notificação e a *string* é removida para quarentena.

1.5.2.2 Sensoriamento Heurístico

Segundo o Dbios (2017), esse é o segundo passo a ser executado pelo antivírus quando o usuário agenda um escaneamento na máquina. Essa etapa é um método complexo e sujeito a erros, como os falsos positivos. Ela busca por instruções que não são normais e usuais nos programas.

1.5.2.3 Busca Algorítmica

Segundo o Canaltech (2017), essa fase dispõe de diversos algoritmos que irão realizar a varredura do sistema com o intuito de selecionar os resultados em busca de ameaças.

1.5.2.4 Checagem de Integridade

O site Dbios (2017) relata que nessa fase o antivírus cria um banco de dados com informações sobre os programas armazenados no disco do computador que irão garantir sua integridade. Sendo assim, quando for realizar uma varredura do sistema, o banco de dados é consultado e qualquer alteração nas informações que garantem a integridade do arquivo seja constatada, o antivírus alerta o usuário e envia o arquivo para quarentena.

1.6 Realização correta de cópias de segurança

Segundo Macêdo (2012), as cópias de segurança asseguram a possibilidade de restauração dos dados em caso de perda acidental ou proposital. São realizados cópias dos dados presentes no disco rígido, de maneira completa ou parcial. Macêdo (2012), ainda relata que é importante realizar os *backups* periodicamente e realizar os testes para garantir que os dados possam ser lidos e restaurados em caso de perda dos arquivos originais.

1.6.1 Cuidados com os backups

Os cuidados com as cópias de segurança dependem muito da necessidade do usuário e do valor das informações para o mesmo. Informações confidenciais irão demandar maior investimento na proteção e armazenamento.

Segundo um estudo da Veeam (2017) encomendado a ESG com mais de 1.000 entrevistados no mundo todo, existem muitos desafios na proteção, recuperação e disponibilidades das informações. Como resultado, foi constatado que existe uma enorme quantidade de dados que não são protegidos corretamente e os processos de restauração dos mesmos não são eficazes. Os números mostram que o prejuízo causado por incidentes chega à média de US$ 21,8 milhões em tempo de inatividade anual.

Para tentar sanar estes problemas diversos cuidados devem ser tomados em relação aos *backups*, por exemplo, controle físico do local onde serão armazenadas as informações, para evitar roubo ou destruição do mesmo, é recomendável que seja protegida contra eventos da natureza, tais como, fogo, água, poeira, entre outros. Além dos cuidados lógicos que devem ser observados, por exemplo,

verificação das cópias de segurança após sua geração, possibilitando a descoberta de defeitos nas mesmas.

1.6.2 Educação dos usuários finais

Os usuários finais devem ser informados e instruídos acerca dos perigos presentes na internet, uma vez que estes são os alvos prediletos de *crackers*, porque os mesmos apresentam poucos conhecimentos técnicos e muitas vezes são imprudentes com os tipos de conteúdos acessados através da rede e acabam acessando algum arquivo contaminado por código malicioso, causando a infecção do sistema.

Segundo Jorge (2017), as análises apontam que o rápido avanço tecnológico e o aumento do uso de dispositivos informatizados, elevaram a dificuldade e necessidade de segurança da informação. Assim sendo, é de extrema importância que os usuários se conscientizem dos riscos que estão sujeitos.

Por mais que se invista em tecnologias de segurança, os seres humanos são ainda o ponto mais vulnerável de uma rede, pois através de uma simples engenharia social pode-se fazer com que toda uma infraestrutura computacional seja comprometida através deste. Torna-se então necessário investimento em capacitação e conscientização.

1.7 Criptografia

A criptografia é uma proteção que consistem em codificar as informações através de fórmulas matemáticas e armazená-las, surgiu da necessidade dos povos antigos de proteger informações para que seus inimigos não conseguissem ler o conteúdo da mesma, caso caíssem em mãos erradas. Essa técnica é muito usada para manter a integridade e confidencialidade dos dados em um processo de autenticação de usuários, transações bancárias, protegerem conversas sigilosas, por exemplo. Segundo Beal (2008), sem a existência da criptografia não seria possível realizar comércio eletrônico. Com base no tipo de chave usada, as criptografias podem ser: criptografia de chave simétrica ou de chave assimétrica.

Segundo o Cert (2017), a criptografia de chave simétrica, também conhecido por criptografia de chave única ou secreta, faz uso da mesma chave para codificar e

decodificar a informação. É largamente utilizada para garantir a confidencialidade das informações. Alguns exemplos de métodos criptográficos de chave simétrica são: AES, Blowfish, RC4, IDEA E 3DES.

O Cert (2017) define também que, a criptografia de chave assimétrica que também é conhecida como criptografia de chave pública, faz uso de duas chaves diferentes, uma que é pública, para codificar a informação e outra que é privada, para decodificar a informação e deve ser mantida em segredo pelo seu proprietário. Alguns exemplos de métodos criptográficos de chave assimétrica são: RSA, DSA, ECC E Diffie-Hellman.

1.8 Assinatura digital

É um método de criptografia que surgiu para substituir as assinaturas feitas a punho em papel, e é legalmente válida. A QualySign (2017) define que, é uma tecnologia que utiliza de criptografia e a associa ao certificado digital. Com ela a empresa elimina diversos processos manuais que dependem de assinaturas em documentos físicos e arquivamento de papéis, economizando assim tempo e dinheiro.

A Certsign (2013) diz que, a assinatura digital é um dos meios de fazer uso do Certificado Digital ICP-Brasil, documento eletrônico, que irá garantir, por meio da criptografia autenticidade, integridade e não-repúdio às transações digitais.

Uma assinatura digital é realizada com a utilização de uma chave privada e assim é possível emitir informações de forma segura, sem que ninguém se passe por outro. Para transmitir essas então é preciso gerar um *hash*, este é o resultado dos dados criptografados propriamente dito. Assim o certificado digital é gerado e então pode ser repassado para quem tiver a chave assimétrica para decodificação das informações.

1.9 Certificado digital

É um arquivo eletrônico análogo a assinatura digital, sua função é parecida com um documento de identidade. O ITI (2017) define que, o certificado digital é considerado uma identidade virtual que permite realizar transações e operações na internet de forma segura e legítima.

Segundo o Cert (2017), o certificado digital é um registro eletrônico contendo as diversas informações que irão identificar alguém ou algo e associar o mesmo a uma chave pública. Deverá ser emitido por uma entidade certificadora, que será a responsável pela autenticidade das informações do proprietário do certificado digital.

1.10 Técnicas e Tecnologias de Defesa

Esta seção abordará os conceitos de algumas das principais técnicas de proteção disponíveis no mercado.

1.10.1 Filtro de pacote

Esse é o primeiro tipo de *firewall* desenvolvido, e faz o controle de acesso na rede com base no cabeçalho do pacote. Utiliza como parâmetros principais para fazer um bloqueio o número de IP (“*Internet Protocol*”) e o número da porta. Essas informações indicam exatamente a qual rede e qual computador são destinados determinados pacotes. Após saber essas informações verifica-se nas configurações de regras do *firewall* se o pacote tem ou não permissão para adentrar a rede.

1.10.2 Filtro de pacotes com controle de estado

Este tipo tem as mesmas premissas do tipo de *firewall* anterior, mas agrega funções de segurança que o torna mais completo. Ele se baseia no cabeçalho do pacote, buscando nas regras de configuração o IP e a porta, mas ainda monitora toda a conexão, levando em conta também a origem desta, ou seja, se um *host* de dentro da rede faz uma requisição a um servidor de determinada página *Web*, o *firewall* monitora a conexão esperando uma resposta do servidor para o *host* e a uma porta em questão.

Se as informações estiverem certas o pacote tem permissão de adentrar a rede, mesmo que não haja uma regra especificando autorização para aquele IP. Mas se um pacote de fora da rede é enviado, ao tentar estabelecer a conexão com um *host* interno o mesmo é negado e não pode passar pelo *firewall*, mesmo que haja uma regra dando autorização aquele IP.

1.10.3 Proxy firewall

Este tipo é também conhecido por "*application firewall*" e faz a filtragem das mensagens em nível de aplicação. Para entender seu funcionamento, primeiro é necessário entender o que é e o que faz um *proxy*. Um *proxy* é um *host* ou aplicação na rede que age como intermediário entre um cliente e um servidor, fazendo requisições em nome do cliente e devolvendo as respostas em nome do servidor. Desse modo o *proxy firewall* monitora todo tráfego, pois toda requisição será feita por ele.

1.11 Filtros

Essa funcionalidade permite que sejam selecionados quais pacotes podem ou não trafegar pela rede. Os filtros fazem a seleção de um pacote válido para transitar na rede com base em seu cabeçalho de transporte, observando principalmente o IP e portas TCP/UDP de origem e destino.

São mecanismos de controles simples e de baixo custo, mas que podem se tornar difíceis de implementar em casos mais complexos, portanto é recomendado seu uso em casos que não demandem grandes quantidades de regras, e não devem ser também o único meio de segurança disponível, mas deve ser usado como complemento das ferramentas de segurança implementadas.

Zwicky, Cooper, Chapman (2000) definem que, os filtros são mecanismo que permitem ou bloqueiam pacotes na rede. Para realizar a filtragem, são configurados um conjunto de regras especificando os tipos de pacotes que devem ser permitidos e quais devem ser bloqueados.

1.12 Proxies

Um *proxy* é um computador na rede que irá atuar como um intermediário em uma troca de mensagens entre outros dois computadores. Segundo Macêdo (2012), é um servidor que responde às requisições redirecionando o tráfego do cliente à frente: um cliente conecta-se a um servidor *proxy*, solicitando algum serviço, página *Web*, um arquivo, conexão ou algum outro recurso do servidor. Segundo Neto (2009), *proxy* é um termo em inglês que significa algo como: realizar algo em nome de outrem.

Zanoni (2007) diz que, um dos objetivos do *proxy* é fornecer acesso a internet para computadores de uma rede, sem que estes tenham a necessidade de realizar a requisição diretamente. O *proxy* é geralmente instalado em uma máquina de acesso a internet e os demais computadores irão efetuar solicitações de serviços da internet através deste. Por isso é chamado de *proxy* (“procurador”, em português), ou seja, é um sistema que realiza solicitações em nome de outros.

1.12.1 Proxy Cache

O *proxy cache* é responsável por armazenar localmente páginas *Web*, com o objetivo de otimizar o acesso a esses recursos, quando solicitados. Segundo Macêdo (2012), o *cache* é como se fosse um depósito dos sites acessados na rede.

Seu funcionamento consiste em, quando solicitada uma página *Web*, por um *host* qualquer na rede, o *proxy* irá procurar em seu *cache* primeiro se a página se encontra armazenada. Caso encontre, a resposta é devolvida ao *host* solicitante, caso contrário, ele faz a requisição ao servidor, realiza o *download* da página e devolve a resposta ao solicitante.

1.12.2 Proxy Reverso

O *proxy* reverso, realiza a função contrário de um servidor *proxy* comum, ou seja, ele é responsável por pegar o tráfego oriundo da internet e repassar ao servidor. Filho (2008) diz que, todo servidor deve ter um sistema de *proxy* reverso instalado a sua frente, uma vez que sua função é evitar que os clientes realizem requisições direto ao servidor, e assim oferece um mecanismo de segurança a mais para o mesmo.

Segundo Lopes (2006, p. 31):

> Para que o *proxy* consiga interceptar todas requisições oriundas da internet, o mesmo deve se portar como um servidor *web*, desse modo ele consegue centralizar as requisições vindas da *web*, além de ocultar informações ligadas a rede local, para quem está acessando da internet. (LOPES, 2006, p 31).

1.12.3 Bastions hosts

Um *bastion host* é um computador na rede por onde irá transitar todo tráfego de entrada e saída. Nesse sistema, são instalados somente os serviços estritamente necessários para atender a demanda de seus usuários. Normalmente dispõe de duas interfaces de rede, uma conectando a internet e outra conectando a intranet.

Segundo Alecrim (2013), o *bastion* se localiza entre o roteador e a intranet, não permitindo a comunicação direta entre as duas. Trata-se então de um mecanismo de segurança extra, uma vez que toda informação vinda da internet passa primeiro pelo roteador, que filtra as mesmas, encaminhando-as para o *bastion*, que em seguida determina quais pacotes podem ou não ser redirecionados para a rede interna.

1.13 Network Address Translation – NAT

É uma técnica criada para resolver problemas de endereçamento IP em redes de grande porte. Com NAT (“Tradução do endereço da rede”, em português) é possível que uma rede interna dispondo de IP’s privados consiga acessar a internet. Quando uma máquina da rede, com IP privado, solicita uma página *Web*, por exemplo, o NAT é responsável por converter o IP privado em um IP público.

Segundo Lento (2012), o NAT consiste na tradução de endereços de uma rede para outra. A utilização dessa técnica permite esconder os endereços IP da rede interna e a conexão de uma grande quantidade de máquinas à internet, fazendo uso de menor quantidade de IP’s válidos, ou utilizando endereços administrativos.

1.14 Rede privada virtual – VPN

A VPN surgiu a partir da necessidade de trafegar na internet com informações sensíveis de forma segura. Seus pilares são a criptografia e o tunelamento. A criptografia garante a segurança das informações (autenticidade, confidencialidade, integridade e disponibilidade), além de prover a segurança dos túneis, que permitem as informações trafegarem na internet de forma segura, através de um túnel criptografado.

Segundo Chin (1998), a VPN provê um túnel de criptografia entre dois pontos, criados através de redes, sejam elas públicas, ou não, para a transferência de informações de maneira segura, entre as redes corporativas ou usuários remotos.

1.15 Autenticação

Uma a ação que garante a identidade de um usuário na rede, limitando assim o acesso somente às pessoas que têm credenciais para utilização dos recursos disponíveis no *host*.

Segundo Filho (2009), a autenticação depende geralmente de um ou mais fatores dentre os métodos existentes. Como por exemplo: biometria, padrão de retina ou de voz*, smart cards, tokens*, senhas.

1.16 Políticas de Segurança da Informação

A política de segurança é um conjunto de regras normativas que irão impor alguns critérios para estabelecer a segurança da informação e deve ser de conhecimento geral dentro da organização. O conteúdo de uma PSI (Política de Segurança da Informação) depende muito das necessidades e critérios utilizados, mas sempre sofrerá alterações, de acordo com a demanda.

Dantas (2011) define, política de segurança da informação como um documento que impõe os critérios, procedimentos e orientações que devem ser realizados para maior gestão da proteção da informação.

Política de segurança da informação ou somente PSI são um conjunto de diretrizes impostas pela direção da organização de maneira formal, e deve ser apresentado a todos os colaboradores, a fim de que se cumpra o que foi estabelecido em suma.

Para Dantas (2011), a política de segurança da informação é documento formal estabelecendo os princípios, compromissos, valores, orientações, responsabilidades e requisitos sobre os processos para alcançar um padrão desejável de proteção das informações.

1.16.1 A importância de zelar pela SI

Por ser um ativo de vital importância para os segmentos de negócio da organização, pode acontecer dessas serem alvo de espionagem, roubo ou terem alguma forma de violação que comprometa os pilares da segurança da informação: ACID (autenticidade, confidencialidade, integridade, disponibilidade), por dadas circunstâncias é importante que se estabeleçam todos os critérios e meios de segurança possíveis, a fim de minimizar as possibilidades de ocorrência e diminuir os impactos de uma possível violação desses pilares. Além dentre outras coisas, também é importante para a conscientização de todos os colaboradores.

Campos (2001) diz que, a informação é a base para que a evolução e o desenvolvimento humano ocorram progressivamente.

Segundo a ISO/IEC 27002 (2005), a PSI busca fornecer orientações da direção empresarial e apoio para a segurança da informação, com base nos requisitos de negócio e com as leis e regulamentações relevantes.

1.16.2 A responsabilidade por elaborar a política de SI

Para maior consistência e relevância da política de segurança da informação, busca-se criar um grupo de profissionais especialistas em segurança, juntamente com os diretores e profissionais da área jurídica estabelecer o conjunto de regras que irão compor a PSI, além de implantar, divulgar e gerenciar a mesma.

Para Ferreira e Araújo (2008), o ideal é fundar um comitê de segurança da informação, constituído pelos especialistas necessários. Esse comitê será responsável por divulgar e estabelecer os procedimentos de segurança e fazer reuniões periódicas com o objetivo de avaliar e manter a segurança na organização.

1.16.3 Abordagens da PSI

O conteúdo da política de segurança deve abordar além dos aspectos de sistema de informação. Devem ser considerados os objetivos, área de atuação, requisitos de segurança física e operacional, cultura dentre outros fatores importantes. Contudo, não existe uma política de segurança da informação pronta, existe um padrão a ser seguido que auxilia no desenvolvido de uma PSI eficiente,

mas para cada instituição existirá uma política da informação específica, que irá atender as peculiaridades do negócio.

Adachi (2004) realizou um estudo de gestão da segurança em *Internet Banking*, e apontou três aspectos impactantes envolvidos na segurança da informação: físico, lógico e humano. Com isso é essencial que a PSI blinde o máximo esses pontos de vulnerabilidades, tornando a política de segurança mais completa quanto possível.

1.16.4 Implementação de PSI

É um processo normalmente demorado, deve ser realizado formalmente com a presença de todos os colaboradores e com o suporte da alta direção, é recomendável ainda que sejam espalhados avisos sobre a PSI em murais por toda a empresa e que sejam assinados pelos colaboradores. Durante sua implantação é necessário que a mesma esteja aberta a alterações para que todos se adaptem e respeitem a política imposta pela organização.

Segundo Ferreira (2017), é de extrema importância que todos os colaboradores da organização tomem nota das ameaças que podem colocar em risco os ativos e quais medidas devem ser tomadas para a proteção dos mesmos. Durante a implantação, é de extrema importância contar com uma equipe qualificada para gerenciar toda a fase de mudanças e orientar todos os envolvidos.

De acordo com a ISO/IEC 27002 (2005), os colaboradores devem estar cientes das ameaças e vulnerabilidades que envolvem a segurança da informação e que estejam preparados para apoiar a política de segurança da informação na organização durante a execução de suas competências.

1.16.5 Divulgação da PSI

A divulgação deve ser realizada de forma generalizada para todos os colaboradores da organização. Todos devem tomar nota da PSI, porque uma vez implantada, qualquer tipo de violação da mesma, será penalizada por meio verbal, escrito ou até judicialmente. Diversos métodos de divulgação podem ser utilizados, por exemplo: campanhas feitas internamente, palestras de conscientização, folhetos em murais, dentre outros métodos.

Ferreira e Araújo (2008, p. 47) dizem que:

> Para que a cultura da organização seja modificada para atender os parâmetros definidos na política de segurança da informação, é fundamental que todos os colaboradores estejam cientes das mudanças, por meio de avisos, palestras de conscientização, elaboração de guias rápidos de consulta e treinamento direcionado. (FERREIRA E ARAÚJO, 2008. p.47).

1.16.6 Controladoria de acessos

A parte da política de segurança que se refere ao controle de acesso tem como objetivo proteger os ativos da organização contra eventuais ameaças. Podem ser divididos em acesso lógico e físico, sendo que os ativos informatizados não podem ser protegidos somente com proteção física.

Segundo Lento (2012), o controle de acesso limita as ações que os usuários de um sistema computacional podem executar, limitando o que ele pode fazer logicamente como fisicamente.

1.16.6.1 Controle de acesso lógico

São as diretrizes estabelecidas com o intuito de limitar o acesso aos ativos de informação em forma digital. Segundo Lento (2012), os controles lógicos limitam o acesso à informação que estejam representadas eletronicamente e que ficariam expostas a uma possível vulnerabilidade sem a devida realização de controles lógicos.

1.16.6.2 Controle de acesso físico

São as diretrizes estabelecidas com o intuito de limitar o acesso aos ativos em forma física. Segundo Lento (2012), os controles de acesso físico limitam o acesso dos usuários a infraestrutura tecnológica da organização, tornando assim reduzido o número de pessoas que acessarão os recursos computacionais.

1.17 OWASP – Open Web Application Security Project

Segundo Meucci (2008) a *OWASP* (*Open Web Application Security Project*) é uma comunidade aberta, sem fins lucrativos, que visa encontrar e combater falhas de segurança em aplicações. Todos os documentos, fóruns e ferramentas estão disponíveis, de forma gratuita para todos os interessados, que desejam aumentar a segurança das aplicações.

1.17.1 Projetos Top Ten e Testing Guide

***Top Ten*:** Williams e Wichers (2010) dizem que o *Top Ten* é um documento que foca nas vulnerabilidades mais críticas de aplicações *Web*. Objetiva-se a destacar as consequências das vulnerabilidades mais comuns em aplicações *Web*, fornece ainda, técnicas básicas para proteção contra tais vulnerabilidades.

Testing Guide: Meucci (2008) define que, o *Testing Guide* é um guia onde são abordadas as seguintes questões: o que testar, por que testar, quando testar, onde testar e como testar aplicações *Web*. Resumindo, o objetivo é encontrar vulnerabilidades em tais aplicações.

Com isso, a finalização deste capítulo permitirá ao leitor ter maior compreensão acerca das definições técnicas presentes neste trabalho, além da interpretação metodológica descrita no Capítulo 2, bem como a aplicação dos testes propostos no Capítulo 3 com maior clareza.

2 METODOLOGIA

Esta pesquisa se caracteriza por um estudo de caso, que conforme Fonseca (2002) se define como:

> Um estudo de uma entidade bem definida como um programa, uma instituição, um sistema educativo, uma pessoa, ou uma unidade social. Visa conhecer em profundidade o como e o porquê de uma determinada situação que se supõe ser única em muitos aspectos, procurando descobrir o que há nela de mais essencial e característico. O pesquisador não pretende intervir sobre o objeto a ser estudado, mas revelá-lo tal como ele o percebe. O estudo de caso pode decorrer de acordo com uma perspectiva interpretativa, que procura compreender como é o mundo do ponto de vista dos participantes, ou uma perspectiva pragmática, que visa simplesmente apresentar uma perspectiva global, tanto quanto possível completa e coerente, do objeto de estudo do ponto de vista do investigador (FONSECA, 2002, p. 33).

Neste sentido, o autor buscou através de levantamento bibliográfico acerca dos conceitos de segurança da informação e *pentest* (que é parte essencial do trabalho), pois demonstra os tipos de vulnerabilidades, ameaças, metodologias de *pentest* e estabelecer um roteiro a ser explorado em teste de intrusão em ambiente controlado tal como um intruso mal-intencionado faria em um cenário real. Então, após a aplicação dos testes, expor os resultados com o intuito de demonstrar sua efetividade, no que se refere à identificação de vulnerabilidades.

Foram realizadas diversas consultas a livros, dissertações e por artigos científicos selecionados através de busca em diversas bases de dados (livros, repositórios de dados, etc...), em autores como: “Mahidhar (2013)”, “Bortoluzzi (2004)”, “Gonçalves (2014)”. O período dos artigos pesquisados foram os trabalhos publicados principalmente nos últimos “10” anos. As palavras-chave utilizadas nas buscas foram: *pentest*, teste de intrusão, segurança da informação.

Ainda foi utilizado como guia para definir as vulnerabilidades a serem abordadas aquelas que foram encontradas inicialmente por varredura em um sistema e que estão contidas no documento *Owasp Top 10.* Este é um relatório realizado pela organização especialista em segurança *Web* (*Owasp*), que desenvolve diversos projetos e metodologias visando conscientizar os desenvolvedores sobre desenvolvimento seguro e ajudando a identificar falhas em sites e sistemas Web através de ferramentas desenvolvidas pela própria equipe.

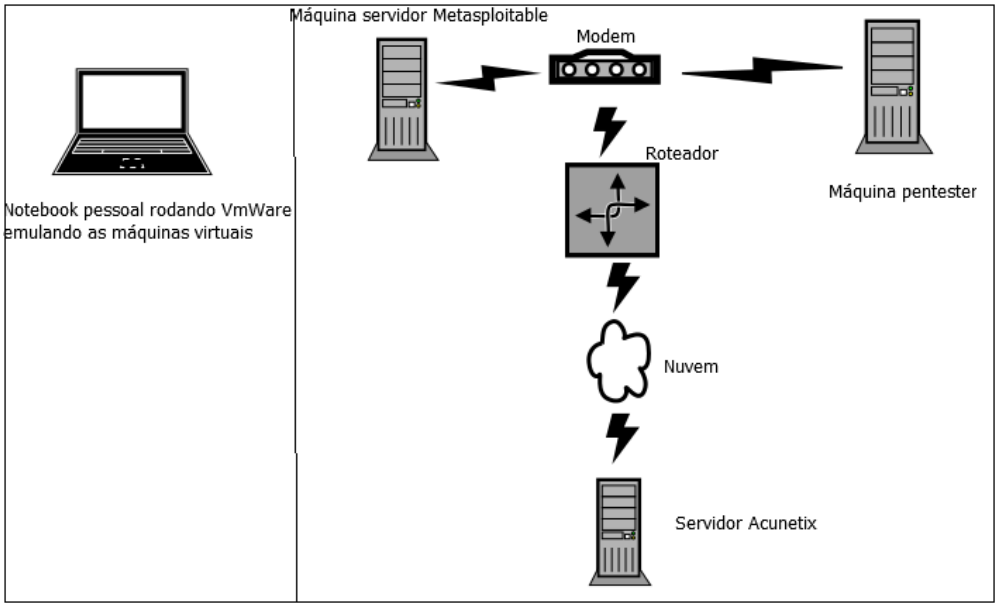
Foram definidas no total sete vulnerabilidades (*Identify application entry points*, *Web Application Fingerprint*, *Reflected Cross Site Scripting*, *Stored Cross Site*

Scripting, *Session Management Schema*, *Cookies Attributes*, *Testing for Bypassing Authentication Schema*) que serão testadas através dos métodos propostos. Foram utilizados ainda o *Owasp Testing Guide V.2*, tal como metodologias independentes para que se tenha um melhor aproveitamento e profundidade nos testes abordados.

Para execução deste trabalho, foram necessários utilização de recursos do próprio autor, uma vez que a metodologia foi executada em ambiente real, porém os responsáveis pelo sistema não autorizaram a divulgação dos resultados neste trabalho. Então foram instalados em máquinas virtuais VMware Workstation (2017), um ambiente cliente/servidor. Desse modo, o ambiente de testes é composto por uma máquina virtual Kali Linux (2017), utilizada pelo *pentester*. Além de outra com Kali Linux (2017), que será testada, foi utilizado também o site "*testphp.vulnweb.com*", da empresa Acunetix, além do metasploitable [2].

Uma visão geral do laboratório de testes configurado pode ser visualizada na Figura 2:

Figura 2 – Laboratório de testes



Fonte: O próprio autor

Como pode ser observado na Figura 2, o laboratório de testes foi emulado em um notebook, utilizando duas máquinas virtuais e fez uso do site disponibilizado pela empresa Acunetix como exemplo em alguns testes e em outros foi utilizado o

[2] O Metasploitable é uma máquina virtual rodando Ubuntu Server 8.04, com diversos softwares em versões com vulnerabilidades conhecidas, tais como Tomcat, TikiWiki, entre outros.

metasploitable para demonstração, no Quadro 01 é detalhado a configuração das máquinas virtuais e referencia o site que a Acunetix autoriza utilizar em caráter de demonstração e aprendizado:

Quadro 01- Ambiente de Testes

Máquina *pentester*	Utiliza Kali linux na versão mais atualizada até o momento (Versão, 2017.2). A máquina virtual foi criada com o VMware, utilizando 20 GB de disco rígido alocado dinamicamente, memória RAM de 1024 MB, interface de rede configurada para rede interna.
Máquina servidor	Utiliza *Metasploitable 2*. A máquina virtual foi criada com o VMware, utilizando 8 GB de disco rígido alocado dinamicamente, memória RAM de 512 MB, interface de rede configurada para rede interna.
Acunetix Site:	Site "*testphp.vulnweb.com*" disponibilizado pela empresa Acunetix para realização de testes de invasão.

Fonte: O próprio autor (2017)

Vale ressaltar que este é um trabalho realizado com recursos próprios e que qualquer tentativa de reprodução do mesmo em cenário real é necessária autorização do proprietário do sistema ou rede para aplicação dos testes.

O objetivo é identificar as principais vulnerabilidades encontradas em *Web* sites/sistemas, a fragilidade e facilidade de exploração, além de evidenciar a importância dos *pentests* no que tange a segurança da informação e ferramenta auxiliar para constatação de vulnerabilidades *Web*, além de servir como material didático para todos os profissionais e curiosos da área.

A abrangência da metodologia proposta não contempla todos os testes para as vulnerabilidades catalogadas no *Owasp Top 10*, mas com base em tal foram selecionadas as vulnerabilidades consideradas críticas, que foram constatadas em cenário real, que contém maior impacto no caso de ocorrência, da facilidade de exploração e risco de ocorrência, então posteriormente utiliza-se adaptativamente o *Owasp Testing Guide V.2* juntamente com as metodologias existentes para realizar os testes em ambiente controlado.

Com base nas características elementares de um *Web* site/sistema, adaptou-se um roteiro de testes que segue alguns métodos do *Owasp Testing Guide V.2*, juntamente com alguns métodos existentes, com o intuito de complementar e assim elencar os testes.

Os testes de intrusão que foram realizados estão listados abaixo e seguem a seguinte convenção de nomenclatura: o padrão *Owasp* - categoria de teste - numeração dentro da categoria e podem ser visualizados no Quadro 02.

Quadro 02 – Metodologias usadas

Metodologia 1	(OWASP-IG-003) *Testing Identify application entry points;*
Metodologia 2	(OWASP-IG-004) *Testing for Web Application Fingerprint;*
Metodologia 3	(OWASP-DV-001) *Testing for Reflected Cross Site Scripting;*
Metodologia 4	(OWASP-DV-002) *Testing for Stored Cross Site Scripting;*
Metodologia 5	(OWASP-SM-001) *Testing for Session Management Schema;*
Metodologia 6	(OWASP-SM-002) *Testing for Cookies Attributes.*
Metodologia 7	(OWASP-AT-005)Testing for Bypassing Authentication Schema

Fonte: O próprio autor (2017)

O presente capítulo destinou-se a apresentar a metodologia utilizada para o desenvolvimento do trabalho, o laboratório de testes e quais testes serão aplicados. A metodologia se caracteriza pelo estudo de caso da aplicação de um *pentest* White-box em um site, sendo executado em ambiente controlado, mas diante da perspectiva Black-box que é aplicado em ambiente real. Assim sendo, o capítulo 3 abordará de fato a execução do teste de intrusão.

3 EXPERIMENTOS E RESULTADOS

O presente capítulo abordará a aplicação dos testes de intrusão para verificação de vulnerabilidades, seguindo a metodologia proposta no capítulo 2.

3.1 Experimentos realizados

Nesta seção estão descritos os testes realizados, para tal a seção foi divida em subseções, onde cada subseção representa um teste e para cada teste específico, são apresentados os objetivos, pré-requisitos e resultados do teste em questão.

3.1.1 Testing Identify application entry points (OWASP-OTG-INFO-006)

Objetivos do teste

- Mapear os pontos de entrada e de possíveis vulnerabilidades.

Pré-requisitos do teste

- Ferramenta *proxy* para interceptação do tráfego da rede. Ex: (*BurpSuite*, *WebScarab*, entre outros).

Para a realização dos testes com maior eficiência e aproveitamento, é primordial que o *pentester* compreenda a interação entre o cliente/servidor que são o usuário/navegador e a aplicação, respectivamente. Essas interações geralmente acontecem por meio do protocolo HTTP, frequentemente pelos métodos *POST* ou *GET*. A análise das requisições e respostas HTTP permite a identificação de pontos de entrada referentes ao processo analisado.

O mapeamento sistemático e detalhado da aplicação permite a identificação de cabeçalhos, parâmetros, campos ocultos presentes nas requisições e respostas do protocolo HTTP. Todas essas informações devem ser armazenadas para análises posteriormente.

No entanto, a quantidade de informações pode ser extensa, dependendo do tamanho da aplicação, tornando o processo redundante. Para isso, a experiência do *pentester* é essencial, pois somente um profissional qualificado irá identificar quais são os pontos mais críticos a serem analisados minuciosamente.

Para a análise das requisições HTTP que são *POST*, é necessário a utilização de uma ferramenta *proxy*, que irá capturar o tráfego da rede entre o usuário e a aplicação.

Os seguintes pontos devem ser observados, quando se está analisando as requisições do HTTP:

- Onde são utilizados os métodos GET e POST;
- Quais parâmetros compõem as requisições POST;
- Campos ocultos em requisições POST;
- Identificação dos parâmetros utilizados na consulta do método GET;
- Cabeçalhos adicionais ou modificados (por exemplo, *debug=false*).

No que se refere às respostas HTTP, o que deve ser observado são:

- Definição, adição e alteração dos *cookies* (*Set-Cookie header*);
- Verificar onde existe redirecionamento de páginas pelo HTTP (verificação dos códigos de estado HTTP);
- Verificar o uso de cabeçalhos específicos (por exemplo, *Server: BIG-IP*).

3.1.1.1 Resultados do teste Identify application entry points

Para a execução foi utilizado o *Owasp ZAP* e então foi possível obter a estrutura de diretórios do site em questão, além dos parâmetros utilizados em requisições GET e POST e as URLs válidas. Foi possível identificar diretórios ocultos para o navegador, além de capturar cada requisição e resposta feita pelo protocolo HTTP do navegador. A Figura 3 mostra a captura de tela da execução do *ZAP*:

Figura 3 – Lista de URLS encontradas

```
http://testphp.vulnweb.com/
http://testphp.vulnweb.com/AJAX/index.php
http://testphp.vulnweb.com/AJAX/styles.css
http://testphp.vulnweb.com/Mod_Rewrite_Shop/
http://testphp.vulnweb.com/Mod_Rewrite_Shop/Details/color-printer/3/
http://testphp.vulnweb.com/Mod_Rewrite_Shop/Details/network-attached-storage-dlink/1/
http://testphp.vulnweb.com/Mod_Rewrite_Shop/Details/web-camera-a4tech/2/
http://testphp.vulnweb.com/artists.php
http://testphp.vulnweb.com/artists.php?artist=3
http://testphp.vulnweb.com/cart.php
http://testphp.vulnweb.com/categories.php
http://testphp.vulnweb.com/disclaimer.php
http://testphp.vulnweb.com/guestbook.php
http://testphp.vulnweb.com/hpp/
http://testphp.vulnweb.com/hpp/?pp=12
http://testphp.vulnweb.com/hpp/params.php?p=valid&pp=12
http://testphp.vulnweb.com/index.php
http://testphp.vulnweb.com/listproducts.php?artist=3
http://testphp.vulnweb.com/listproducts.php?cat=4
http://testphp.vulnweb.com/login.php
http://testphp.vulnweb.com/privacy.php
http://testphp.vulnweb.com/product.php?pic=6
http://testphp.vulnweb.com/robots.txt
http://testphp.vulnweb.com/search.php?test=query
http://testphp.vulnweb.com/secured/newuser.php
http://testphp.vulnweb.com/secured/style.css
http://testphp.vulnweb.com/signup.php
http://testphp.vulnweb.com/sitemap.xml
http://testphp.vulnweb.com/style.css
http://testphp.vulnweb.com/userinfo.php
```

Fonte: O próprio autor

Após a identificação dos caminhos, páginas e parâmetros da aplicação e realização de um relatório com as informações coletadas, tem-se uma base constatada para, a partir daí elaborar testes mais específicos e abrangentes.

Essa etapa do teste pode ser automatizada, como foi o caso, mas dependendo do tamanho da aplicação, é mais eficaz realizar testes manuais, pois as ferramentas possuem limitações que um teste individual feito manualmente pode contornar determinadas situações.

3.1.2 Testing for Web Application Fingerprint (OWASP-IG-004)

Objetivos

- Captura do tipo e versão do servidor *Web*.

Pré-requisitos

- Ferramenta para realização de requisições HTTP (por exemplo, *NetCat*).

As páginas de aplicações *Web* são armazenadas e executadas a partir de um servidor *Web*, com isso as suas peculiaridades são pertinentes aos testes de invasão. Assim sendo, a descoberta do tipo e a versão do servidor *Web* são informações de extrema importância, que em posse destas é possível fazer uma busca na *internet* a procura de vulnerabilidades que possam ser exploradas.

Normalmente, na execução do teste para captura do tipo e versão do servidor são enviadas diversas requisições HTTP para o mesmo, que após o processamento das requisições irá enviar a respostas ao cliente, e com base nas características destas, é possível identificar o tipo e versão do servidor. Apesar disso, alguns servidores distintos podem gerar respostas HTTP semelhantes para requisições diferentes, então quanto mais requisições forem feitas maior a probabilidade de se confirmar o tipo e a versão do servidor.

Para realizar o teste, a utilização da ferramenta *NetCat* foi essencial, pois a partir dela foram efetuadas as requisições HTTP. Com o *NetCat* o *pentester* se conecta ao servidor, então são enviadas diversas requisições e com base nas características do cabeçalho de resposta, especialmente o campo *Server*, pode-se obter o tipo e a versão do servidor. Porém, essas mensagens de resposta podem ser alteradas pelo administrador do servidor. Então, uma análise minuciosa observando a ordem dos cabeçalhos de resposta pode dizer se o cabeçalho foi adulterado ou não, uma vez que cada fabricante de servidor *Web* emite as respostas em uma determinada ordem.

Outra técnica utilizada para a detecção do servidor consiste em enviar requisições HTTP malformadas e com base nas respostas e nos aspectos inerentes da resposta, podem-se identificar padrões em sua estrutura e conteúdo.

3.1.2.1 Resultados do teste Web Application Fingerprint

Utilizando o *NetCat*, conectando-se na porta 80 do servidor, obtém-se algumas respostas do protocolo HTTP, e através das mesmas pode-se analisar diversas informações, inclusive informações sobre versões das tecnologias utilizadas.

Conforme pode ser observado na Figura 4, o cabeçalho de resposta, apontado pelo campo *Server* está indicando o nome do servidor e a versão do mesmo.

Figura 4 – Resposta HTTP

```
HTTP/1.1 200 OK
Server: nginx/1.4.1
Date: Fri, 17 Nov 2017 14:57:46 GMT
Content-Type: text/html
Connection: keep-alive
X-Powered-By: PHP/5.3.10-1~lucid+2uwsgi2
```

Fonte: O próprio autor

Através de uma requisição com a versão do protocolo HTTP inexistente, o servidor retorna a seguinte resposta, representada na Figura 5.

Figura 5 – Versão inexistente do protocolo HTTP

```
root@kali:~# nc testphp.vulnweb.com 80
GET / HTTP/3.0

HTTP/1.1 400 Bad Request
Server: nginx/1.4.1
Date: Thu, 23 Nov 2017 11:22:42 GMT
Content-Type: text/html
Content-Length: 172
Connection: close
```

Fonte: O próprio autor

Em caso de dúvidas sobre a versão identificada do servidor *Web*, como complemento para o *fingerprint*, pode-se realizar uma requisição por um protocolo inexistente ou indisponível através do *NetCat* e assim observar a resposta, que dá indícios da confirmação do servidor e sua versão.

3.1.3 Testing for Reflected Cross Site Scripting (OWASP-DV-001)

Objetivos

- Enumerar os pontos de entrada não filtrados para inserção de dados do usuário;
- Execução de código malicioso no lado cliente em âmbito da enumeração.

Pré-requisitos

- Ferramenta para varredura de páginas *Web*, *Spider/Crawler*;
- Ferramenta *proxy* (*BurpSuite*, *WebScarab*, entre outros);

- Codificador/decodificador de caracteres.

O tipo de ataque *Reflected Cross Site Scripting* explora os campos de entrada que não são filtrados corretamente no lado do servidor. Um atacante mal intencionado pode então com um código do lado cliente, inserir o mesmo em um dos parâmetros da aplicação, com isso uma requisição HTTP maliciosa será feita e assim o servidor pode retornar uma resposta de caráter duvidoso.

Esse tipo de *Cross-Site Scripting*, também chamado de XSS pode ser constatado através dos testes com parâmetros da URL e de formulários da aplicação, desse modo insere-se um *script* nestes parâmetros e se o código for executado pelo navegador, significa que o servidor refletiu o trecho de código e então o mesmo encontra-se vulnerável a XSS. As modificações dos parâmetros podem ser feitas utilizando um interceptador *proxy.*

Existem ferramentas que automatizam a verificação desse tipo de XSS refletido. Essas ferramentas utilizam métodos heurísticos analisando as respostas HTTP equivalentes às requisições feitas com dados não confiáveis. Pode-se observar então que a análise automatizada verifica a resposta HTTP, e não sua execução de fato.

3.1.3.1 Resultados do teste Reflected Cross Site Scripting

Após a realização da identificação das URLs válidas dentro do site, que pode ser visto na Figura 3 da seção 3.1.1.1, têm-se a noção dos possíveis pontos de entradas que estão presentes. Então agora o pentester com base nessas URLs pode realizar os testes para verificação do XSS Refletido.

Pode-se observar na Figura 6, a seguir, um caminho de URL que foi descoberto na varredura da seção 3.1.1.1 com um trecho de código *JavaScript* que visa testar se o parâmetro passado pelo método *GET* da aplicação é filtrado corretamente:

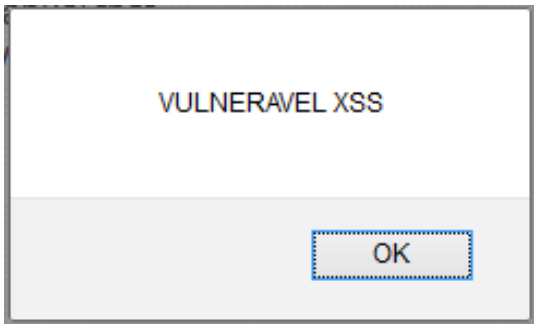
Figura 6 – URL Testada XSS Refletido

testphp.vulnweb.com/listproducts.php?cat=<script>alert("VULNERAVEL XSS")</script>

Fonte: O próprio autor

Então a partir da execução dessa URL pode-se observar que o resultado foi renderizado pelo navegador, conforme observado na Figura 7 e assim constata-se a vulnerabilidade de XSS refletido.

Figura 7 – Vulnerabilidade XSS Refletido constatada



Fonte: O próprio autor

3.1.4 Testing for Stored Cross Site Scripting (OWASP-DV-002)

Objetivos

- Enumerar os pontos de entrada não filtrados para inserção de dados do usuário;
- Execução e armazenamento de código malicioso no lado cliente em âmbito da enumeração.

Pré-requisitos

- Ferramenta para varredura de páginas *Web*, *Spider/Crawler*;
- Ferramenta *proxy* (*BurpSuite*, *WebScarab*, entre outros);
- Codificador/decodificador de caracteres.

O tipo de ataque *Stored XSS*, ou *Cross Site Scripting* persistente, é definido pelo armazenamento do trecho de código malicioso no servidor da aplicação. É um tipo de ataque mais abrangente que o XSS refletido e mais perigoso, pois todos que acessarem a página da aplicação que irá referenciar ao trecho de código malicioso serão afetados. Quando o servidor responde a uma solicitação HTTP ao cliente o

mesmo interpreta a página de resposta como um conteúdo seguro, assim sendo, o navegador executa o código malicioso no âmbito do cliente normalmente.

A constatação desse tipo de vulnerabilidade para injeção de trechos de código maliciosos é bastante similar ao do XSS refletido. Através de testes com os parâmetros passados para as requisições do protocolo HTTP, pode-se armazenar o trecho de código malicioso no banco de dados da aplicação no servidor. Sua confirmação se dá quando os usuários acessam a página que faz referência ao trecho de código malicioso e então o navegador do cliente executa o código, então, tem-se um *Stored Cross Site Scripting.*

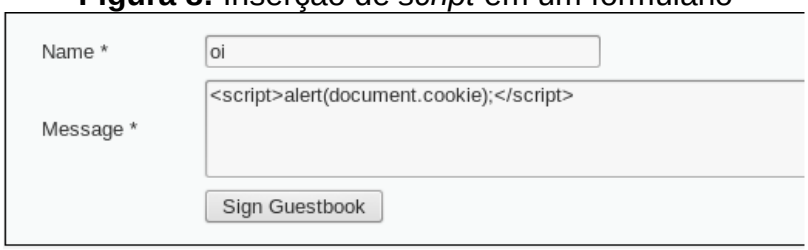
Um detalhe pertinente ao XSS que é muito importante ser verificado, consiste em testar a vulnerabilidade em diferentes navegadores Web. O que acontece é que, muitas vezes, algumas entradas podem ser renderizadas por um determinado navegador, mas por outro não. Isso ocorre devido às particularidades do navegador, o modo como ele realiza a codificação dos caracteres. Então devido a isso, além do emprego de filtros nos campos de entrada de dados (tanto no lado cliente, como no servidor), compete ao *pentester* usar da criatividade e ousadia para manipular as entradas de dados.

3.1.4.1Resultados do Testing for Stored Cross Site Scripting

Sua execução é similar ao XSS Refletido, a diferença é que o XSS Armazenado aproveita da vulnerabilidade de não verificação dos caracteres inseridos nos campos de formulário que serão armazenados no banco de dados. Então, assim o atacante pode inserir um *script* em determinado formulário que será armazenado para uso posterior, desse modo, toda vez que a página com o *script* malicioso for acessada, o *script* será renderizado juntamente com a página *Web.*

Pode-se observar na Figura 8 uma execução de *Cross-Site Scripting Stored*:

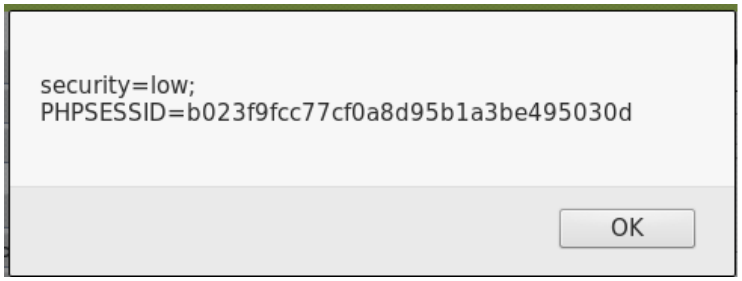
Figura 8: Inserção de *script* em um formulário



Fonte: O próprio autor

Após a inserção do *script* exibido na Figura 8, o mesmo persistirá na página e toda vez que o conteúdo da página for acessado, será exibido uma caixa de alerta exibindo o *cookie* da sessão, Figura 9.

Figura 9 – XSS Armazenado constatado



Fonte: O próprio autor

A execução desta modalidade de XSS tem a peculiaridade de que, não necessariamente a página que executa o código malicioso tenha sido a página que permite a injeção do mesmo. Em algumas situações, o trecho de código malicioso é visto por outro perfil de utilizador, diferente do que injetou o código.

3.1.5 Testing for Session Management Schema (OWASP-SM-001)

Objetivos

- Analisar as características inerentes aos *cookies*;
- Analisar os mecanismos de geração de informação de sessão;
- Investigar e explorar tempo de expiração dos *cookies*;
- Analisar o tipo de transporte utilizado pelos *cookies*.

Pré-requisitos

- Ferramenta de interceptação *proxy* (*BurpSuite*, *WebScarab*, entre outros);
- Ferramenta para análise de sequência de *cookies*.

O protocolo utilizado pela aplicação, que é o HTTP, não tem em suas características inerentes nenhum mecanismo para gerenciamento de sessão. Por conseguinte, para alcançar este objetivo, criaram-se mecanismos de administração de estados e sessão que são combinados com o protocolo HTTP e assim oferece maior segurança.

O *cookie* é uma estrutura de dados que foi desenvolvida para ser responsável por conter as informações pertinentes a identificação da sessão. Essas informações de identificação podem ser transmitidas de algumas formas - por URL ou por campos ocultos.

Cabe ao *pentester* averiguar as seguintes diretivas em relação ao gerenciamento de sessão:

- Os atributos dos *cookies* estão definidos corretamente?
- Através da engenharia reversa é possível descobrir qual a metodologia utilizada para geração dos *cookies*?
- Existe coerência no tempo de expiração dos *cookies*?
- Os *cookies* são transmitidos por meios seguros? Se sim, é obrigatório?

3.1.5.1 Resultado do Testing for Session Management Schema

A execução deste teste partiu da análise da resposta HTTP ilustrada na Figura 10. Ela corresponde à resposta HTTP que define um valor de *cookie*, para a sessão do usuário, com o uso da diretiva *Set-Cookie*.

Alguns campos presentes no cabeçalho de resposta devem ser analisados com atenção, por exemplo, as diretivas *Set-Cookie*, *Expires, Cache-Control*. Com intenção de detalhar um pouco mais análise dos mecanismos de gerenciamento de sessão, serão tratadas as respostas apresentadas na metodologia do atual teste e que ainda não foram respondidas, são elas:

(I) Os atributos de sessão dos *cookies* estão definidos corretamente?

Sempre que um valor pré-definido não é encontrado no *cache* do *browser* um novo *cookie* é atribuído a aplicação. Ou então, quando uma situação de *logout* for realizada. Desse modo, se um *cookie* definido estiver com o tempo de validade não expirado, um atacante pode utilizar a técnica conhecida como *cookie replay*.

Na utilização da técnica de replicação de *cookies*, o atacante utiliza valores válidos em requisições HTTP para a aplicação. Por consequência, o atacante consegue utilizar o perfil do usuário que está simultaneamente e legitimamente, associado a este *cookie*.

O ideal é determinar um tempo para a expiração dos *cookies* da sessão e que estejam adequados às funcionalidades da aplicação e além das necessidades do usuário.

(II) Os *cookies* são transmitidos por meios seguros?

Na Figura 10, pode-se notar a ausência da *flag Secure*, isso não obriga a transmissão dos *cookies* por canais seguros de comunicação. Portanto um inimigo através de um ataque de *man-in-the-middle*[3] pode capturar essas informações.

Figura 10 – Atributos dos *cookies*

```
GET /dvwa/vulnerabilities/csrf/ HTTP/1.1
Host: 192.168.17.133
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0)
Gecko/20100101 Firefox/45.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q
=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.17.133/dvwa/vulnerabilities/xss_s/
Cookie: security=high;
PHPSESSID=1ed67da4d630ae291498736fb6c845f9
Connection: close
Cache-Control: max-age=0
```

Fonte: O próprio autor

Além da *flag secure*, deve ser avaliado também as questões de imprevisibilidade do *cookie*, ou seja, se é possível descobrir o mecanismo de geração dos *cookies.* Avaliar se o *cookie* tem um tempo de expiração coerente. Se esses atributos mínimos de segurança, não estiverem definidos, é muito possível que um atacante consiga realizar um ataque MITM (*man-in-the-middle*) e assim se passar por um usuário legítimo.

3.1.6 Testing for Cookies Attributes (OWASP-SM-002)

Objetivos

- Analisar características inerentes dos *cookies.*

Pré-requisitos

- Ferramenta de interceptação *proxy*.

Os *cookies* são estruturas de dados desenvolvidas para controlar os estados do usuário, mas que também podem ser usados para outros fins. São utilizados em

[3] O man-in-the-middle (pt: Homem no meio, em referência ao atacante que intercepta os dados) é uma forma de ataque em que os dados trocados entre duas partes (por exemplo, você e o seu banco), são de alguma forma interceptados, registrados e possivelmente alterados pelo atacante sem que as vitimas se apercebam.

aplicações *Web* dinâmicas, e os critérios pelos quais estes são manipulados pela aplicação, dependem dos seus atributos. Os atributos relacionam-se com os aspectos de segurança na manipulação dos *cookies*.

A seguir apresenta-se uma lista com os principais atributos para o gerenciamento seguro da sessão do usuário:

Quadro 03 - Atributos dos *Cookies*

Atributo 1	*HttpOnly*: Quando ativado, proíbe que o código cliente aceda ao *cookie;*
Atributo 2	*Secure*: Quando esta *flag* é ativada, garante a transmissão segura dos *cookies* (geralmente por SSL/TLS);
Atributo 3	*Domain*: Limita a manipulação dos *cookies* somente aos domínios e subdomínios;
Atributo 4	*Path*: Geralmente é usado em conjunto com o atributo *Domain*, este atributo define para qual caminho de URL o *cookie* é válido;
Atributo 5	*Expires*: Estabelece um tempo para que os *cookies* persistentes expirem. Se não for definido tempo algum passa a expirar diante do término da sessão do navegador.

Fonte: O próprio autor (2017)

Através da análise de respostas HTTP com o cabeçalho *Cookie* presente, pode-se testar os atributos dos *cookies*. Então nessas respostas analisa-se se estão presentes os atributos listados acima.

Quando são constatadas as ocorrências desses atributos nas respostas analisadas, então estes têm seus valores analisados pela perspectiva de segurança; quando não são constatadas, são avaliados os impactos de suas ausências.

3.1.6.1 Resultado do Testing for Cookies Attributes

Em muitos sites, quando um usuário realiza um processo de *login*, o servidor do site em questão envia um *cookie* para o cliente solicitante, que será armazenado no seu disco rígido e será utilizado durante o período em que o usuário estiver autenticado e utilizando o site.

Este teste visa analisar se os *cookies* foram criados com os atributos necessários para prover maior segurança na utilização do site. É possível analisar através da resposta do servidor apresentado na Figura 10, a ausência de todas as *flags* de segurança descritas anteriormente no Quadro 03, com isso um atacante conseguiria realizar ataques de sequestro de sessão, *man-in-the-middle*, entre outros métodos de exploração.

3.1.7 Testing for Bypassing Authentication Schema (OWASP-AT-005)

Objetivos

- Burlar os mecanismos de autenticação
  + *Direct page request* (*forced browsing*);
  + *Parameter Modification;*
  + *Session ID Prediction;*
  + *Form authentication SQL Injection.*

Pré-requisitos

- Ferramenta de interceptação *proxy* (BurpSuite, WebScarab, entre outros).

Conforme o projeto *OWASP*, existe um conjunto de metodologias para testar a segurança do esquema de autenticação. Os seguintes testes são realizados para avaliar essa questão:

SQL Injection for authentication forms: A condição primordial para um *SQL Injection* é o não tratamento dos caracteres inseridos pelo teclado, no lado cliente. O *pentester* pode testar se a aplicação está vulnerável através de campos para consultas ou através de formulários presentes na aplicação, então realizar inserção de caracteres nos mesmos a fim de manipular as consultas *SQL* (*Structured Query Language*). Assim com base nas respostas de retorno pode-se determinar se é vulnerável a *SQL*, ou podem-se automatizar os testes, para casos mais específicos de *SQL,* como o *Blind SQL Injection*, que é mais exaustivo para testar.

Direct page request (*forced browsing*): Consiste em tentar acessar páginas que devem ser restringidas e exibidas somente mediante as credenciais de acesso legítimas.

Parameter modification: Este teste é específico para casos em que a autenticação é baseada em um parâmetro que é passado para a aplicação. Existem dois métodos para este teste. No primeiro o parâmetro é passado pelo método GET e enviado a URL, então o *pentester* altera o mesmo com o intuito de autenticar-se. Pelo segundo meio, o parâmetro é enviado por POST e então é necessário uma ferramenta *proxy* para interceptar a requisição HTTP e modificar a mesma com o intuito de autenticar-se.

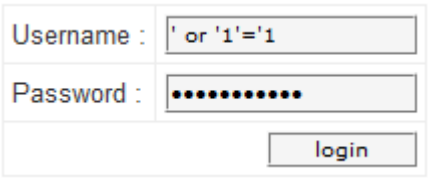
Session ID Prediction: Este caso de teste aplica-se em mecanismo de autenticação baseada na *Session ID*. O *pentester* deve tentar descobrir o padrão de geração dos identificadores de sessão, a fim de autenticar-se. Primariamente avalia-se o uso da *Session ID* na autenticação e posteriormente avalia-se o mecanismo de geração das mesmas.

3.1.7.1 Resultados do Testing for Bypassing Authentication Schema

SQL Injection for authentication forms

A abordagem manual de teste para *SQL Injection* no esquema de autenticação consistiu em inserir em um formulário de login algumas sequências de caracteres ou caractere, com o intuito de autenticar-se sem a inserção do usuário e senha corretos. Na Figura 11, pode-se observar o formulário de *login*.

Figura 11 – Teste em formulário de *login*



Fonte: O próprio autor

Pode-se perceber que foram inseridos nos campos de *Username* e *Password* a expressão “‘ or ‘1’=‘1”, após clicar no botão de *login* é realizada a requisição POST para o servidor e forma-se a seguinte consulta SQL no banco de dados, Figura 12.

Figura 12 – Consulta SQL maliciosa no banco de dados

```
SELECT * FROM users WHERE  username= ‘ ’ or ‘1’=‘1’ AND password=‘ ’ or ‘1’=‘1’
```

Fonte: O próprio autor

E então o processo de autenticação é feito sem necessariamente ter conhecimento do usuário e senha válidos que constem no banco de dados da aplicação, pois esta consulta da Figura 10 invalida a lógica da mesma.

Muitos casos podem não ser tão simples os testes, podem demandar maior criatividade do testador e pode ainda requerer que seja feito por alguma ferramenta automatizada, como o *SQLmap* ou *SQLninja* que estão presentes na distribuição utilizada Kali Linux e são muito eficazes na detecção dessas vulnerabilidades.

***Direct page request* (*forced browsing*)**

A intenção desse teste é verificar se a aplicação faz o controle acesso das páginas restritas. Caso o desenvolvedor tenha implantado o mecanismo de verificação somente na página de *login* podem-se realizar as requisições das páginas restritas aos usuários autenticados e acessar as mesmas de forma direta, sem a necessidade de estar logado com as credenciais de usuário e senha cadastrados no banco de dados.

Algumas das URLs ilustradas na Figura 3 necessitam de autenticação para acesso, então foram digitadas diretamente no navegador, mas não foi possível acessar seu conteúdo de modo direto. Podem-se ver as URLs que foram testadas na Figura 13.

Figura 13 – Lista de URLS acessadas diretamente

```
http://testphp.vulnweb.com/userinfo.php
http://testphp.vulnweb.com/cart.php
```

Fonte: O próprio autor

Parameter modification

Alguns desenvolvedores criam suas páginas de autenticação com base nos conteúdo dos *cookies*. Dessa maneira os controles de acesso às páginas exclusivas aos usuários cadastrados ficam sobre esses *cookies*. Com base nas URLs da Figura 3, foram realizados os testes, porém os mesmos não foram bem sucedidos.

Session ID Prediction

Algumas aplicações *Web* realizam o gerenciamento das autenticações por meio dos identificadores de sessões. Existem diversos meios para gerar esses identificadores e quanto mais complexo for esse mecanismo de geração, mais complicado será para o testador realizar este teste, pois o mesmo obtém grande taxa de sucesso em mecanismos que são previsíveis.

Novamente foram utilizadas as URLs da Figura 3 e não foi possível constatar um mecanismo de geração de identificador para sessão.

3.2 Resultados obtidos

Após a aplicação dos testes, chegaram-se as seguintes conclusões. O primeiro ponto é considerando o ambiente em que foram executados tais testes, este foi emulado pelo autor do presente trabalho. Jorge (2012) diz na conclusão de seu trabalho que, através da aplicação do *pentest* em um cenário real ou controlado é possível identificar diversos problemas referentes à segurança e é possível então aplicar as soluções adequadas. A aplicação em um cenário real não se difere em nada da metodologia exposta neste capítulo, servindo como parâmetro para análise da segurança da informação. Embora os resultados possam ser diferentes dos obtidos neste trabalho, devido à arquitetura em que se encontra a aplicação *Web* testada (se possui *patches* de correção, programas atualizados, mecanismos de proteção, entre outros), os resultados obtidos através da metodologia servirão como base para análise do nível de segurança.

O segundo ponto, se refere aos resultados de cada metodologia aplicada, estes se mostraram satisfatórios, pois demonstraram com clareza o processo pelo

qual um *pentester* começaria a análise de um site/sistema em questão através do escaneamento da aplicação, identificação dos possíveis pontos de entrada e identificação do sistema operacional. Após isso, como realizar os testes em formulários de *logins* e outros formulários que utilizem método POST de requisição HTTP. Demonstrando ainda, como realizar os testes para vulnerabilidades específicas, como o *Cross-Site Scripting* armazenado e refletido, além do *SQL Injection.* Assim sendo, nota-se a importância da aplicação de um *pentest* para avaliação das potenciais falhas e riscos inerentes as aplicações *Web* no que tange a segurança da informação.

CONSIDERAÇÕES FINAIS

A popularização da internet permitiu que muitas empresas e pessoas físicas, realizassem comércio eletrônico, compartilhamento de informações e outras atividades na rede. É inegável que é um grande meio de negócio e comunicação, mas da mesma forma que o lado positivo seja evidente, é também inegável que existem os contras. Pessoas com conhecimentos baixos ou extensos e que estão a espreita, planejando ou esperando uma brecha para comprometer os pilares da segurança da informação.

A partir desse enunciado, o trabalho buscou demonstrar os conceitos que englobam a segurança da informação, tecnologias, ameaças, vulnerabilidades e outros. Então, através destes chegar ao roteiro estabelecido para demonstrar a importância de testar as páginas voltadas para *Web* e como aplicar um teste de intrusão.

O teste de intrusão aplicado no trabalho foi estabelecido com base no guia de teste definido pela grande e consolidada comunidade *OWASP*. Não foram exaustados todos os testes do guia, mas contudo foram explorados os seguintes: *Testing Identify application entry points, Testing for Web application fingerprint, Testing for reflected cross-site scripting, Testing for stored cross-site scripting, Testing for session management schema* e *Testing for cookies attributes.*

Após a realização dos testes, constata-se que os pentests são de suma importância para a avaliação do nível de segurança que um site/sistema *Web* possui. Através dos mesmos, ainda foi possível verificar que algumas vulnerabilidades encontradas, igual *SQL Injection* e *Cross-Site Scripting*, embora não seja uma falha de *software* e nem *hardware*, mas uma falha do programador na hora da codificação das páginas, que não filtra adequadamente as informações trocadas entre o cliente e o servidor. Essas falhas são de risco extremamente alto, pois comprometem as informações do banco de dados do sistema.

Por último, para concluir, é importante lembrar que os testes de intrusão necessitaram de uma autorização pelo administrador do *Web* site/sistema para sua realização. Este trabalho contou com uma execução em cenário real, na qual não podem ser divulgados nomes e dados institucionais em virtude da não autorização do proprietário, bem como para proteção, segurança e sigilo da organização que permitiu a realização do teste.

TRABALHOS FUTUROS

Este trabalho foi realizado no âmbito acadêmico, visando auxiliar e subsidiar informações relevantes aos acadêmicos dos cursos de tecnologia da informação, os desenvolvedores de softwares *Web*, bem como organizações e demais interessados sobre as ameaças que circundam na internet, vulnerabilidades existentes em sistemas computacionais e seus utilizadores, e então com base na aplicação da metodologia de teste de invasão proposta pelo autor foi possível constatar as vulnerabilidades existentes no sistema *Web* emulado e no disponibilizado pela empresa *Acunetix*.

Nesta perspectiva, espera-se que este trabalho sirva de subsidio para trabalhos futuros e novas pesquisas em segurança informacional, onde novos testes poderão ser feitos, novos métodos utilizados para validar os já realizados pelo autor, bem como utilizar novas ferramentas e novos ambientes de testes, objetivando encontrar novas vulnerabilidade e ameaças e assim melhorando a informação construída nesta pesquisa.

Como sugestão pode-se ainda ser realizada a aplicação dos métodos de proteção contra tais vulnerabilidades demonstradas no presente trabalho, além da aplicação de outros testes propostos no *Owasp Testing Guide* e que não foram abordados neste.

REFERÊNCIAS BIBLIOGRÁFICAS

ADACHI, Tomi. **Gestão de Segurança em Internet Banking** - São Paulo: FGV, 2004. 121p. Mestrado. Fundação Getúlio Vargas - Administração. Orientador: Eduardo Henrique Diniz.

ALECRIM, Emerson. **O que é firewall? Conceitos, tipos e arquiteturas**. Disponível em: < https://www.infowester.com/firewall.php >. Acesso em: 29 set. 2017.

ALENCAR, André. **Princípios Básicos da Segurança da Informação**. Vestcon. Disponível em: < http://blog.vestcon.com.br/principios-basicos-da-seguranca-da-informacao-mnemonico-dica/ >. Acesso em: 17 out. 2017.

ANTISPAM. **Tipos de Spam: Correntes**. Disponível em: < http://www.antispam.br/tipos/#1 >. Acesso em: 26 set. 2017.

ANTISPAM. **Tipos de Spam: Boatos**. Disponível em: < http://www.antispam.br/tipos/boatos/ >. Acesso em: 26 set. 2017.

ANTISPAM. **Tipos de Spam: Propagandas**. Disponível em: < http://www.antispam.br/tipos/#2 >. Acesso em: 26 set. 2017.

ANTISPAM. **Tipos de Spam: Pornografia.** Disponível em: < http://www.antispam.br/tipos/#4 >. Acesso em: 26 set. 2017.

ANTISPAM. **Tipos de Spam: Códigos maliciosos.** Disponível em: http://www.antispam.br/tipos/malware/ >. Acesso em: 26 set. 2017.

ANTISPAM. **Tipos de Spam: Fraudes**. Disponível em: < http://www.antispam.br/tipos/fraudes/ >. Acesso em: 26 set. 2017.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISSO/IEC 27002:2005 tecnologia da informação - técnicas de segurança - código de pratica para gestão da informação**. Rio de janeiro: 2005, Disponível em: <https://www.profissionaisti.com.br/2010/03/conheca-a-nbr-isoiec-27002-parte-1/>. Acesso em: 29 out. 2017.

AVAST. **Adware**. Disponível em: < https://www.avast.com/pt-br/c-adware >. Acesso em: 25 set. 2017.

AVAST. **Keylogger**. Disponível em: < https://www.avast.com/pt-br/c-keylogger >. Acesso em: 25 set. 2017.

AVAST. **Rootkit**. Disponível em: < https://www.avast.com/pt-br/c-rootkit >. Acesso em: 25 set. 2017.

AVAST. **Spyware**. Disponível em: <https://www.avast.com/pt-br/c-spyware >. Acesso em: 25 set. 2017.

BANTIM, Rudolfh. **O que é botnet?.** Disponível em: < http://www.techtudo.com.br/artigos/noticia/2012/03/o-que-e-botnet.html >. Acesso em: 25 set. 2017.

BEAL, Adriana. **Segurança da Informação: Princípios e Melhores Práticas para a Proteção dos Ativos da Informação nas Orgnizações**. São Paulo: Editora Atlas, 2008.

BISHOP, M. **Computer security: art and Science**. Boston: Pearson Addison Wesley, 2002. 1136 p. ISBN 0-201-44099-7.

BISHOP, M. **What is computer security? IEEE Security & Privacy**, Davis, Jan. 2003. 67-69.

CAMPOS, Ricardo. **Informação é poder**. Disponível em: < https://ricardocampos.wordpress.com/2008/03/03/informacao-e-poder/ >. Acessado em: 30 set. 2017.

CANALTECH. **O que é Antivírus ?**. Disponível em: < https://canaltech.com.br/antivirus/o-que-e-antivirus/ >. Acesso em: 28 set. 2017.

CERT. **Códigos maliciosos**. Disponível em: < https://cartilha.cert.br/malware/ >. Acesso em: 25 set. 2017.

CERT. **Criptografia**. Disponível em: < https://cartilha.cert.br/criptografia/ >. Acesso em: 25 set. 2017.

CERTSIGN. **Assinatura Digital: O que é e seus benefícios**. Disponível em: < https://www.certisign.com.br/documents/10163/321165/certinews_Assinatura_20131015_baixa_2.pdf >. Acesso em: 29 set. 2017.

CHIN, Liou Kuo. **Rede privada virtual – VPN**. Disponível em: < https://memoria.rnp.br/newsgen/9811/vpn.html >. Acesso em: 30 set. 2017.

CISCO. **Cisco prevê mais Destruição de serviço devido a aumento da escala e impacto das ciberameaças**. Cisco. Disponível em: < https://www.cisco.com/c/pt_pt/about/press/news-archive-2017/20170721.html >. Acesso em: 23 set. 2017.

CISCO. **Networking Skills in Latin America**. Cisco. Disponível em: < https://www.cisco.com/assets/csr/pdf/IDC_Skills_Gap_-_LatAm.pdf >. Acesso em: 27 set. 2017.

COSTA, Aécio. **Riscos, ameaças e vulnerabilidades**. Disponível em: < http://www.aeciocosta.com.br/wp-content/uploads/FG/Introducao%20a%20Seguranca%20da%20Informacao%202014-1/5-ISI-Riscos,%20Ameacas%20e%20Vulnerabilidades.pdf >. Acesso em: 27 set. 2017.

COURY, RICARDO. **Informação é poder**. Disponível em < http://www.timester.om.br/entrevista/artigos/main_artigo.asp?Codigo=424 >, Acessado em 28 de outubro de 2012

DANTAS, Marcus Leal. **SEGURANÇA DA INFORMAÇÃO: UMA ABORDAGEM FOCADA EM GESTÃO DE RISCOS**. 1 ed. Olinda: Livro rápido, 2011.

DERESKY, Helen. 2004. **Administração Global: estratégica e interpessoal. Porto Alegre: Bookman.**

DUARTE, Henrique. **O que são hijackers e como eles podem colocar o seu PC em risco**. Disponível em < http://www.techtudo.com.br/noticias/noticia/2014/02/o-que-sao-hijackers-e-como-eles-podem-colocar-o-seu-pc-em-risco.html >. Acesso em: 25 set. 2017.

FARNSWORTH, ROGER. **CISCO Systems: Introduction to Information Security**. Apresentação de Palestra 302 da CISCO, 1998.

FERREIRA, André L. R. **Como implantar uma política de segurança da informação na sua empresa**. Disponível em: < http://www.netdeep.com.br/blog/geral/como-implantar-uma-politica-de-seguranca-da-informacao-na-sua-empresa.html >. Acesso em: 10 out. 2017.

FERREIRA, Fernando N.F. ; ARAÚJO, Márcio T. **Política de Segurança da Informação: Guia Prático para Elaboração e Implementação**. 2 ed. Rio de Janeiro: Editora Ciência Moderna Ltda, 2008.

FILHO, João Eriberto Mota. **Proxy reverso HTTP com Squid (versão 2.6 ou superior)**. Disponível em: < http://eriberto.pro.br/wiki/index.php?title=Proxy_reverso_HTTP_com_Squid_%28vers%C3%A3o_2.6_ou_superior%29 >. Acesso em: 29 set. 2017.

FILHO, Sócrates. **Segurança da Informação: Autenticação**. Disponível em: < http://waltercunha.com/blog/index.php/2009/08/19/seguranca-da-informacao-autenticacao/ >. Acesso em: 30 set. 2017.

GONÇALVES, Luciano. **Aspecto de segurança para uma arquitetura web. Viva o linux**. Disponível em: < http://www.vivaolinux.com.br/artigo/Aspecto-de-seguranca-parauma-arquitetura-web?pagina=2 >. Acesso em: 17 out. 2017.

ISO, *International Organization for Standardization*. **Diretrizes para a Gestão de Segurança da Informação no âmbito do Poder Judiciário**. Disponível em: < http://www.fieb.org.br/download/senai/NBR_ISO_27002.pdf >. Acesso em: 25 set. 2017.

ITI, *Instituto Nacional de Tecnologia da Informação*. **Certificado Digital**. Disponível em: < http://www.iti.gov.br/certificado-digital >. Acesso em: 26 out. 2017.

JORGE, B.T.C. **Segurança e privacidade numa infraestrutura VoIP.** Disponível em : < http://www.inatel.br/biblioteca/pos-seminarios/seminario-de-redes-e-sistemas-de-telecomunicacoes/v-srst/9506-seguranca-e-privacidade-em-redes-voip/file&usg=AOvVaw3Pya7s3KDufPTtctQv7jDL >. Acesso em: 18 dez. 2017.

JORGE, C. D. **Principais tendências em segurança da informação, segundo ESET.** Disponível em: < http://www.cbsi.net.br/2017/01/principais-tendencias-em-seguranca-da-informacao.html >. Acesso em 29 set. 2017.

JULIANO, Ezequiel. **O Ciclo de Vida da Informação**. Disponível em: < http://www.ezequieljuliano.com.br/?p=27 > Acesso em: 24 out. 2017.

KRUGLIANSKAS, I. **Tornando a pequena e média empresa competitiva**. São Paulo, Instituto de Estudos Gerenciais e Editora, 1996.

KUHNHAUSER, W. (2004, January). Root kits: an operating systems viewpoint. ACM SIGOPS Operating Systems Review, 38(1), 12-23, doi: 10.1145/974104.974105.

LAFRANCE, Y. **Psychology: A Precious Security Tool. SANS InfoSec Reading Room**, 9 jun. 2004. Disponível em: < https://www.sans.org/reading-room/whitepapers/engineering/psychology-precious-security-tool-1409 >. Acesso em: 28 set. 2017.

LEHTINEN, R.; RUSSELL, D.; GANGEMI, G. T**. Computer Security Basics. [S.l.]: O'Reilly Media**, Inc., 2006. ISBN 0596006691.

LEMONNIER, Jonathan**. O que é o malware de cavalo de Tróia ?**. Disponível em: < https://www.avg.com/pt/signal/what-is-a-trojan >. Acesso em: 25 set. 2017.

LENTO, Luiz Otávio Botelho. **A importância de uma NAT e de uma VPN para a segurança da informação**. Disponível em: < http://www.diegomacedo.com.br/a-importancia-de-uma-nat-e-de-uma-vpn-para-a-seguranca-da-informacao/#more-3367 >. Acesso em: 30 set. 2017.

LENTO, Luiz Otávio Botelho. **Mecanismos de controle de acesso**. Disponível em: < http://www.diegomacedo.com.br/mecanismos-de-controle-de-acesso/ >. Acesso em: 10 out. 2017.

LOPES, Leandro de Souza. **Segurança em Servidores Web Utilizando Proxy Reverso**. Uberlândia:, 2006.

LOPES, Lidiomar. **Quais são os métodos de identificações e bloqueios de vírus**. Dbios. Disponível em: < http://www.dbios.com.br/novo/index.php/antivirus-quais-sao-os-metodos-de-identificacoes-e-bloqueios-de-virus/ >. Acesso em: 28 set .2017.

MACÊDO, Diego. **Backup: conceito e tipos.** Disponível em: < http://www.diegomacedo.com.br/backup-conceito-e-tipos/ >. Acesso em: 29 set. 2017.

MACÊDO, Diego. **Proxy Cache e Reverso**. Disponível em: < http://www.diegomacedo.com.br/proxy-cache-e-reverso/ >. Acesso em: 29 set. 2017.

MAHIDHAR, Vikram; SCHATSKY, David; BISSELL, Kelly. **Cyber crime fighting**. Disponível em: < http://dupress.com/articles/cyber-crime-fighting/ >. Acesso em: 23 set. 2017.

MAIA, Marco Aurélio. **O que é segurança da informação. Blog de segurança da informação | Módulo Security**. Disponível em: < http://segurancadainformacao.modulo.com.br/seguranca-da-informacao >. Acesso em: 17 out. 2017

MARCONDES, José Sérgio. **Conceito de Segurança da Informação Organizacion**al. Disponível em: < https://www.gestaodesegurancaprivada.com.br/conceito-de-seguranca-da-informacao-organizacional/ >. Acesso em: 25 out. 2017.

MITNICK, K. D.; SIMON, W. L. **The Art of Deception: Controlling the Human Element of Security**. New York, NY, USA: John Wiley & Sons, Inc., 2001. ISBN 1401463223.

NETO, Alfredo Sabocinski. **Proxy Reverso? O que danado é isso?**. Disponível em < http://alfredosabo.blogspot.com.br/2009/05/proxy-reverso-o-que-danado-e-isso.html >. Acesso em: 29 set. 2017

OLIVEIRA, Paulo Cesar. **Princípios básicos da segurança da informação**. Techtem Tecnologia Fácil. Disponível em: < https://www.profissionaisti.com.br/2013/06/politica-de-seguranca-da-informacao-definicao-importancia-elaboracao-e-implementacao/ >. Acesso em: 24 set. 2017.

OLIVEIRA, Waldes. **Riscos, vulnerabilidade e ameaça em Segurança da Informação**. Disponível em : < http://www.techtem.com.br/seguranca-da-informacao-riscos-vulnerabilidade-e-ameaca/ >. Acesso em : 27 set. 2017.

OWASP, The Open Web Application Security Project. **Owasp Top 10 2017**. Disponível em: < https://www.owasp.org/images/b/b0/OWASP_Top_10_2017_RC2_Final.pdf >. Acesso em: 26 ago. 2017.

OWASP, The Open Web Application Security Project. **Owasp Testing Guide 4.0 Release**. Disponível em: < https://www.owasp.org/images/1/19/OTGv4.pdf >. Acesso em: 26 ago. 2017.

QUALISIGN. **Assinatura Digital – Conceito de Assinatura Digital**. Disponível em: < https://www.documentoeletronico.com.br/assinatura-digital.asp >. Acesso em 29 set. 2017.

RAFAEL, G. C. **A realidade de ambientes de TI em Micro e Pequenas Empresas (MPE)**. Disponível em: < https://www.profissionaisti.com.br/2014/02/a-realidade-de-ambientes-de-ti-em-micro-e-pequenas-empresas-mpe/ >. Acesso em: 28 set. 2017.

SÊMOLA, Marcos. 2003. **Gestão da Segurança da Informação – Uma visão Executiva**. Editora Campus. Rio de Janeiro.

SILVA, E. S; JUNIOR, P. M. **Trabalho de Internet – Firewall**. Diponível em < http://homepages.dcc.ufmg.br/~mlbc/cursos/internet/firewall/fire_def.html >. Acesso em: 28 set. 2017.

STAMP, M. **Information security: principles and practice**. Hoboken: John Wiley & Sons, 2006. 390 p. ISBN 978-0-471-73848-0.

STONEBURNER, Gary; GOGUEN, Alice e FERINGA, Alexis. NIST SP 800-30. **Risk management guide for information technology systems**. Local: ? Ed.: NIST, 2002.

STUTTARD, D.; PINTO, M. **The web application hacker's handbook: discovering and exploiting security flaws**. New York, NY, USA: John Wiley & Sons, Inc., 2007. ISBN 9780470170779.

TACIO, Paulo. **O que é e para que serve uma botnet**. Disponível em : < http://www.mundodoshackers.com.br/o-que-e-e-para-que-serve-uma-botnet >. Acesso em: 25 set. 2017.

TCU. **Boas Práticas em Segurança da Informação**. TCU. Disponível em: < http://www4.planalto.gov.br/cgd/assuntos/publicacoes/2511466.pdf>. Acesso em 23 set. 2017.

TILLER, J. S. **The ethical hack: a framework for business value penetration testing**. Boca Raton: Auerbach Publications, 2005. ISBN 0-8493-1609-X.

VEEAM. **2017 Veeam Availability Report – 3 medidas para evitar US$ 21,8 milhões em custos com tempo de inatividade**. Disponível em: < https://www.veeam.com/br/wp-availability-report-2017-brief.html >. Acesso em: 29 set. 2017.

ZANONI, Guilherme Souza. **Servidor Proxy (Squid)**. Disponível em: < https://www.vivaolinux.com.br/artigo/Servidor-proxy-(Squid) > . Acesso em: 29 set. 2017.

ZWICKY, E. D; COOPER, Simon; CHAPMAN, D.B. **Building Internet Firewalls**. 2. Ed. Sebastopol: O'Reilly & Associates, Inc., 2000. 869 p. ISBN 1-56592-871-7.