UNIVERSIDADE FEEVALE

FELIPE MARCANTH LOPES

ESTUDO COMPARATIVO DA SEGURANÇA DE PLATAFORMAS DE E-COMMERCE DE CÓDIGO ABERTO

Novo Hamburgo
2012

FELIPE MARCANTH LOPES

ESTUDO COMPARATIVO DA SEGURANÇA DE PLATAFORMAS DE E-COMMERCE DE CÓDIGO ABERTO

Trabalho de Conclusão de Curso
apresentado como requisito parcial
à obtenção do grau de Bacharel em
Ciência da Computação pela
Universidade Feevale

Orientador: Marcelo Iserhardt Ritzel

Novo Hamburgo
2012

AGRADECIMENTOS

Gostaria de agradecer a todos que de alguma maneira contribuíram para a realização deste trabalho de conclusão, em especial:

Aos amigos e as pessoas que convivem comigo diariamente, minha gratidão pelo apoio emocional nos períodos mais difíceis do trabalho.

RESUMO

A evolução da prática do comércio juntamente com a popularização da internet resultou no surgimento de um novo modelo para os negócios, o e-commerce, que se encontra nos últimos anos em crescente expansão. Este crescimento, paralelamente ao aumento do número de incidentes de segurança, mostra que, cada vez mais, estes sistemas estão expostos a ameaças. Logo, este trabalho visa realizar uma análise de vulnerabilidade das plataformas de comércio eletrônico de código aberto mais conhecidas, empregando a metodologia da Open Web Application Security Project (OWASP), para verificar seus níveis de segurança, apresentar os principais aspectos deste problema e as estratégias de redução.

Palavras-chave: OWASP, vulnerabilidades, e-commerce, segurança.

ABSTRACT

The evolution of the practice of trade, along with the popularization of Internet, resulted in the appearance of a new model for business, the e-commerce, which in last years is becoming increasingly widespread. This growth, in addition to increasing the number of incidents safety, shows that, more and more, these systems are exposed to threats. Therefore, this study aims at an analysis vulnerabilities of e-commerce platforms code open more known, using the methodology of the Open Web Application Security Project (OWASP), to check their levels of security, present the main aspects of this problem and reduction strategies.

Key words: OWASP vulnerabilities, e-commerce security.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ANATEL	Agência Nacional de Telecomunicações
B2B	*Business-to-Business*
B2C	*Business-to-Consummer*
CERT	Centro de Estudos, Resposta e Tratamento de incidentes de segurança
CSRF	*Cross-site request forgery*
C2C	*Consummer-to-Consummer*
DDL	*Data Definition Language*
DML	*Data Manipulation Language*
DOM	*Document Object*
DoS	*Denial of Service*
GB	*Gygabyte*
GNU	*General Public License*
HTML	*Hypertext Markup Language*
HTTP	*hypertext transfer protocol*
IBGE	Instituto Brasileiro de Geografia e Estatística
KDE	*K Desktop Environment*
OWASP	*Open Web Application Security Project*
PDF	*Portable Document Format*
PHP	*Hypertext Preprocessor*
SGDB	Sistema Gerenciador de Banco de dados
SQL	*Structered Query Language*
URL	*Uniform Resource Locator*
XML	*Extensible Markup Language*
XSS	*Cross-site Scripting*

SUMÁRIO

INTRODUÇÃO

O mercado de comércio eletrônico tem apresentado uma significativa evolução nos últimos anos. Este setor teve seu surgimento no final dos anos de 1970 mudando o modelo de vendas e nos últimos dez anos tem crescido em média, 43,5% (E-BIT, 2012). Alguns dos motivos que levaram a isso foram a popularização da internet e o aumento das velocidades de conexão. Além disso, a grande variedade de produtos, preços baixos, facilidade do uso, a agilidade na comparação de preços e a comodidade foram elementos valiosos para o aumento de 37% no número de consumidores no ano de 2011 em relação a 2010, bem como a disseminação desses serviços (E-BIT, 2012).

O crescimento deste mercado deve muito à tecnologia e à internet, as quais são uma realidade para a maioria da população, pois, conforme pesquisa realizada pelo Instituto Brasileiro de Geografia e Estatística (IBGE) em 2009, 67,9 milhões de pessoas eram usuárias de internet. Outro fato importante é a popularização dos telefones celulares que tinham como finalidade a comunicação rápida com acesso para de qualquer lugar, porém, hoje possuem diversas funcionalidades, com aparelhos cada vez menores, a preços mais acessíveis e, consequentemente, maior poder de processamento (IBGE, 2009). Com a evolução dos aparelhos e desses fatores, tais dispositivos passaram a ter presença constante em todas as classes sociais. A Agência Nacional de Telecomunicações (ANATEL) confirmou tal realidade em dados divulgados em março de 2012, quando mostrou que em fevereiro do mesmo ano, havia, no Brasil 247 milhões de linhas ativas, e a tele densidade chegou a 126,45 por 100 habitantes, estes números mostram um aumento de 16% e 15,5%, respectivamente, se comparado com o mesmo período do ano anterior.

O desenvolvimento tecnológico trouxe consigo novas oportunidades, tanto para operadoras de telefonia móvel, em conjunto com as operadoras de cartão de crédito, quanto para os bancos. Estas parcerias e a *near field comunication* (comunicação por campo de curta distância, traduzido por Neto e Campos (2012), isto é, tecnologia que permite conectividade sem fio, de curto alcance) possibilitam aos consumidores de comércio eletrônico, um novo meio para se realizar um pagamento, e de forma mais cômoda por meio de seus aparelhos celulares (FILHO, 2010).

O crescimento do comércio eletrônico trouxe inúmeras preocupações, uma delas refere-se à segurança da informação transmitida nesses serviços. Paralelas a esta evolução e ao grande número de consumidores, estão as ameaças de crimes virtuais cuja finalidade é

obter informações pessoais dos consumidores (PAGSEGURO, 2010). Além disso, há outra questão que preocupa o usuário: a falta de segurança do software. Segundo Hoglund e McGraw (2006, p. 1), "uma invasão, na maioria das vezes, tem como ponto vulnerável o software que, geralmente, em qualquer computador, é a raiz do problema".

Com a popularização das aplicações web, a sua utilização nos últimos anos apresenta alguma importância para diversos fins, tais como: comércio eletrônico, *homebanking*, gestão empresarial, etc. Tais aplicações necessitam de soluções mais seguras pelo fato de processarem informações sigilosas. Portanto, é indispensável uma atenção maior durante a fase de desenvolvimento, assim como na especificação dos aspectos de segurança quanto no uso de ferramentas apropriadas, a fim de garantir mais segurança. Embora exista a preocupação em evitar que o software possua falhas, e possa causar danos, não é uma tarefa fácil devido a alguns fatores, tais como os citados a seguir (ALBUQUERQUE, 2002).

Uma das origens deste problema é a complexidade dos sistemas computacionais que pode chegar a milhares de linhas de código, aliada ao número de tecnologias para desenvolvimento à disposição e que colaboram para que o software tenha algum defeito significativo. Um dos responsáveis, conforme Hoglund e McGraw, é a conectividade dos computadores por meio da internet, os quais ficam mais suscetíveis aos ataques e aumentam a possibilidade de espalharem falhas que acarretem na suspensão de serviços (2006, p. 19).

O Centro de Estudos, Resposta e Tratamento de incidentes de segurança (CERT.br), grupo de resposta a incidentes para a internet, mantido pelo Comitê Gestor da Internet no Brasil, é o responsável por coletar dados e classificá-los por tipos. Essa atividade tem papel importante para verificar o nível de segurança das redes e aplicações existentes no Brasil, além de tentar colaborar para a diminuição de incidentes. No entanto, os dados estatísticos são contrários aos de 2011, quando se obteve um total de 399.515 notificações obtidas e que comparados com 2010, esse número foi quase três vezes maior (CERT, 2011). Outro fato que evidencia o problema é o aumento de 78% de ataques aos servidores web, quando são exploradas vulnerabilidades em aplicações para hospedar arquivos que possam infectar ou auxiliar em outros ataques (CERT, 2011).

Com o aumento destes incidentes, que apresentam uma quantidade de ameaças as quais os sistemas na internet estão expostos, é compreensível a necessidade de adotar normas e procedimentos de segurança na tentativa de reduzir tais ocorrências, assim como a necessidade de um órgão como o CERT.br.

Este projeto tem como foco realizar uma análise de vulnerabilidades por meio de testes de intrusão, cujo objetivo é encontrar falhas e verificar a segurança de um ambiente. Esta técnica, também conhecida como teste de penetração ou *pentest*, é um meio utilizado para simular métodos que invasores podem usar para burlar a segurança (KENNEDY et al, p.1).

Foram selecionadas algumas das principais plataformas de *e-commerce open-source* que serão objeto de estudo desta proposta (MOIP LABS, 2011). Uma delas foi a *Magento Community Edition,* versão livre deste sistema que possui ainda outras duas versões pagas (MAGENTO, 2012). Ela foi criada utilizando a linguagem *Hypertext Preprocessor* (PHP) e possui uma estrutura dividida em módulos o que facilita a implementação de novas funções. Devido ao grande número de desenvolvedores e a variedade de funcionalidades, ficou popular e vem sendo utilizada por empresas conhecidas como Samsung e Lenovo (MAGENTO, 2012).

Outro sistema escolhido foi o osCommerce, uma popular loja on-line, distribuída sobre licença *General Plublic License* (GNU). Este sistema possui uma comunidade de 260 mil membros que desenvolvem, prestam serviços e contribuem para melhorias. Ela é uma solução de comércio eletrônico utilizada por mais de 12 mil lojas e que possui apoio de grandes parceiros como Paypal e Amazon (OSCOMMERCE, 2012). Complementando os ambientes selecionados, o PrestaShop foi escolhido, uma vez que possui uma documentação bastante abrangente em diversos idiomas, e foi desenvolvida na linguagem PHP. Além disso, a possibilidade de integração com sites de terceiros, juntamente com muitas outras funcionalidades, contribui por ser utilizado em mais de 100 mil lojas em diversos países (PRESTASHOP, 2012).

É importante ressaltar que para validar os resultados que serão obtidos com a técnica de *pentest,* é necessário definir a metodologia a ser aplicada. Logo, as metodologias mais populares avaliadas por Borges foram: NIST, ISSAF, OSSTMM e OWASP (BORGES, 2011). Esta última foi escolhida por ser direcionada a testes em aplicações web, melhor adaptável ao ambiente, o que a diferencia das demais que são mais generalistas.

Para atender ao objetivo final, será aplicado a metodologia OWASP, estudada as técnicas de exploração mais críticas e testada a segurança dos softwares selecionados, para assim apresentar um comparativo das plataformas analisadas e as vulnerabilidades de cada uma, juntamente com as falhas mais comuns encontradas nestes ambientes, de modo a permitir a percepção do grau de impacto que um incidente pode causar nestes ambientes de *e-*

commerce. E, com isso, mostrar a preocupação necessária com a segurança da informação, não só em rede ou físico, mas também em nível de software. Com a análise dos dados apresentados, percebe-se a problemática do tema abordado. Os sistemas de comércio eletrônico estão seguros perante as técnicas de ataque que os invasores tem a sua disposição?

O trabalho está organizado da seguinte forma:

- O capítulo 1 introduz os conceitos de e-*commerce*, seus tipos, características assim como também, cita exemplos de algumas lojas virtuais.
- A metodologia OWASP é tratada no capítulo 2, onde são apresentadas as fases do *pentest* dela, além dos tipos de testes que podem ser realizados e os princípios.
- O capítulo 3 aborda as técnicas mais utilizadas nas explorações de vulnerabilidades, mas também mostra algumas medidas preventivas para tais problemas.
- O capítulo 4 sugere algumas ferramentas para auxiliar na execução dos testes citados na sessão de técnicas e OWASP e também podem automatizar algumas tarefas.
- No capítulo 5 são exibidas algumas informações a respeito da arquitetura computacional e sobre a metodologia utilizada no estudo, além dos ambientes a serem testados.
- Os dados obtidos nos testes e os resultados de cada plataforma testada é apresentada no capítulo 6, juntamente com a forma como foram classificadas as vulnerabilidades quanto ao seus riscos, e também sugerida algumas melhorias para os problemas encontrados.

1 COMÉRCIO ELETRÔNICO

De acordo com Albertin (2004, p. 15), o comércio eletrônico é a realização de toda a cadeia de valor dos processos de negócio num ambiente eletrônico, por meio da aplicação intensa das tecnologias de comunicação e de informação, atendendo aos objetivos de negócio. NOVAES afirma (2007, p. 75) que "o comércio tradicional e o eletrônico se diferenciam pelos seguintes fatores":

a) Comunicação: compradores e vendedores possuem um canal de comunicação para trocar informações, o qual utiliza linguagens computacionais como *Hypertext Markup Language* (HTML) ou linguagens dinâmicas, como também, existe *Extensible Markup Language* (XML) para transferir grandes quantidades de dados e possibilitar a padronização de um meio para descrever, processar e transmitir os dados.

b) Dados: gerenciamento das informações que tem papel importante na manutenção da base de dados como também na descoberta destas sobre seus usuários. Esta descoberta se dá por meio de *cookies*, quando os clientes navegam pelo site. Com esses dados, juntamente com técnicas estatísticas são analisados os hábitos dos consumidores para assim realizar algumas ações de marketing.

c) Segurança: mecanismos de segurança que pretendem garantir a integridade e privacidade das informações, os quais são importantes devido à distância entre o comprador e o vendedor.

O comércio eletrônico pode potencializar as vendas de uma empresa aumentando o alcance da mesma, e também a sua presença no mercado por meio deste novo canal de vendas. Do mesmo modo, possibilita reduzir custos operacionais e aproximar a empresa de seus clientes (O'CONNEL, 2002). Apesar desses benefícios que o *e-commerce* traz, há grande preocupação com a privacidade e a segurança, pois o tráfego de informações pela internet pode trazer inúmeros problemas. Segundo Albertin (2004), essas preocupações podem ser divididas em:

- Segurança em cliente-servidor: métodos de autorização com o objetivo de garantir que somente usuários registrados tenham acesso aos dados do sistema. Mecanismos de segurança como *firewall*, criptografia, biometria e outros;
- Segurança de dados e transmissão: esse segundo tipo tem como finalidade assegurar a privacidade e confidencialidade das informações que trafegam no

sistema. Incluindo a autenticação de usuários e pagamentos *on-line*, e dessa forma proteger o cliente de fraudes.

Essa preocupação se deve em parte ao crescimento e ao faturamento nos últimos anos, como se pode observar na figura 1.1. Outro fator que colaborou para essa evolução foram os sites de compras coletivas que atingiram faturamento de R$ 1,6 bilhão em 2011 (E-BIT EMPRESA, 2012). Estas informações evidenciam a preocupação com a segurança já que este setor pode ser visado por atacantes com o objetivo de obter informações sigilosas como número de cartões de crédito.

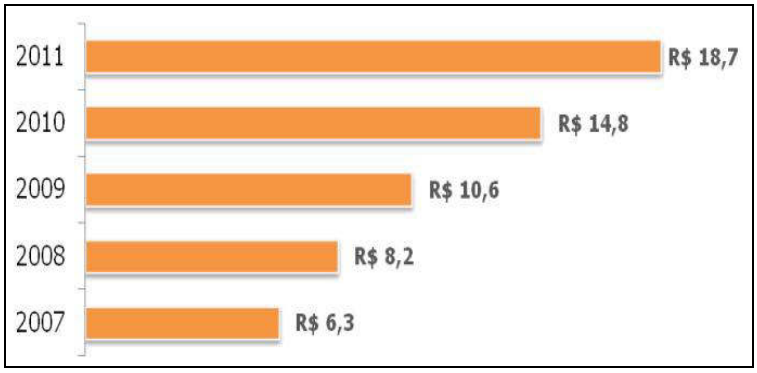

Figura 1.1 – Gráfico do crescimento do comércio eletrônico.
Fonte:E-BIT, 2012.

1.1 Tipos de comércio eletrônico

Um modelo de negócio de acordo com Osterwalder e Pigneur (2010, p. 14) descreve a base sobre a qual uma empresa cria, fornece e capta valor. Sendo assim, no *e-commerce* como em outras áreas existem alguns modelos que são diferenciados pela forma de atuação, os participantes e o público alvo, conforme se pode observar na figura 1.2.

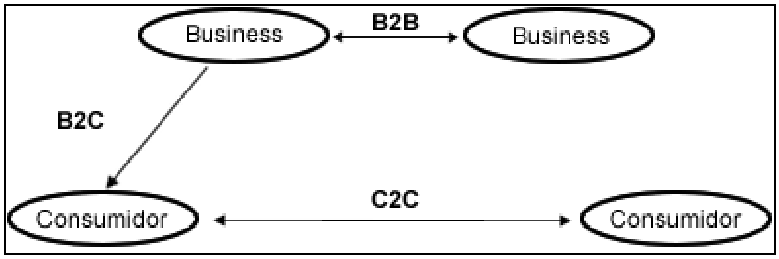

Figura 1.2 – Modalidades de comércio eletrônico.
Fonte: Adaptado de Albertin, 2004.

Tais participantes são as partes envolvidas na negociação, tanto como vendedores, quanto compradores. Eles podem ser consumidores ou empresas, porém para cada combinação dessas pessoas jurídicas ou físicas há um modelo de negócio diferente.

1.1.1 Business-to-Business (B2B)

Business-to-Business (B2B) é a transação de produtos ou serviços realizada entre empresas, caracterizando-se assim o envolvimento de apenas pessoas jurídicas (figura 1.3). Este modelo pode ser considerado como a substituição dos processos físicos envolvidos nas negociações, pois permite uma comunicação e organização melhor destes processos entre clientes, parceiros, fornecedores e distribuidores. No entanto, em compras B2B se tem um risco superior envolvido, devido ao maior investimento que normalmente é feito. No caso de ser adquirido um produto, uma quantidade errada desse ou até mesmo a forma de pagamento não ser muito favorável, pode trazer complicações para o negócio como um todo (LEAL, 2004).

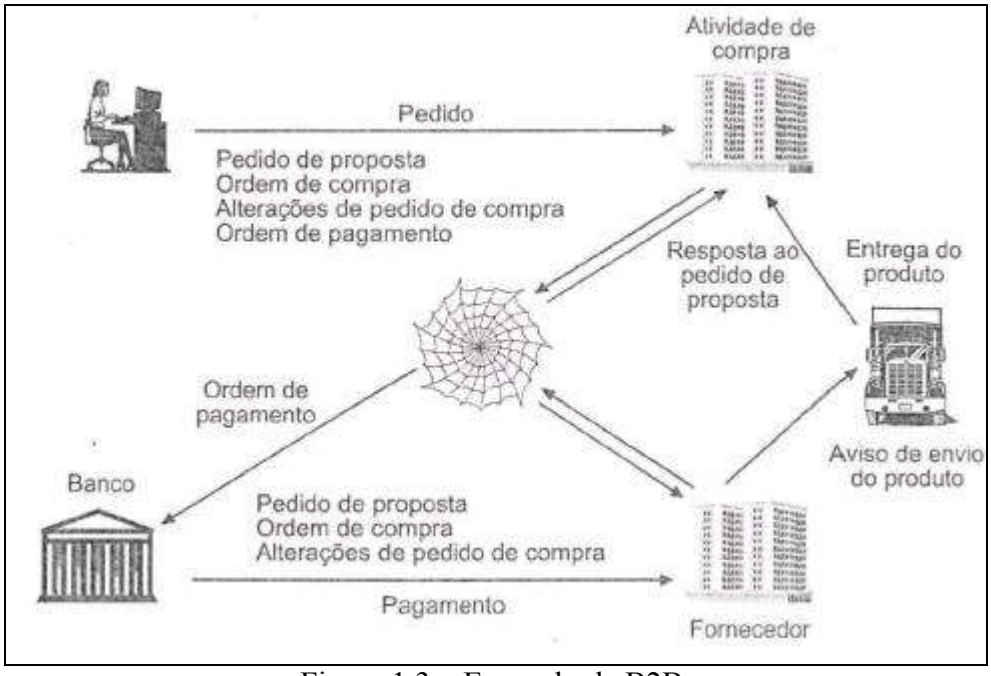

Figura 1.3 – Exemplo de B2B.
Fonte: Nunes, 2010.

Por exemplo, uma empresa disponibiliza para a empresa cliente informações de seus produtos juntamente com os preços. O cliente após a escolha dos produtos envia uma ordem de compra através do sistema. Sendo possível através dele, acompanhar este pedido via internet, receber a fatura para fazer o pagamento e trocar informações com o fornecedor.

1.1.2 Business-to-Consummer (B2C)

Nesse tipo de transação eletrônica, o processo é entre empresas e consumidores finais, representando assim, a virtualização da compra e da venda, se diferenciando pela forma como os produtos são escolhidos e pagos. Normalmente, não estão envolvidos grandes

montantes de dinheiro nessas transações. Dois exemplos dessa modalidade são as lojas virtuais, Submarino e Americanas, como se pode observar na figura 1.4 (ALBERTIN, 2004).

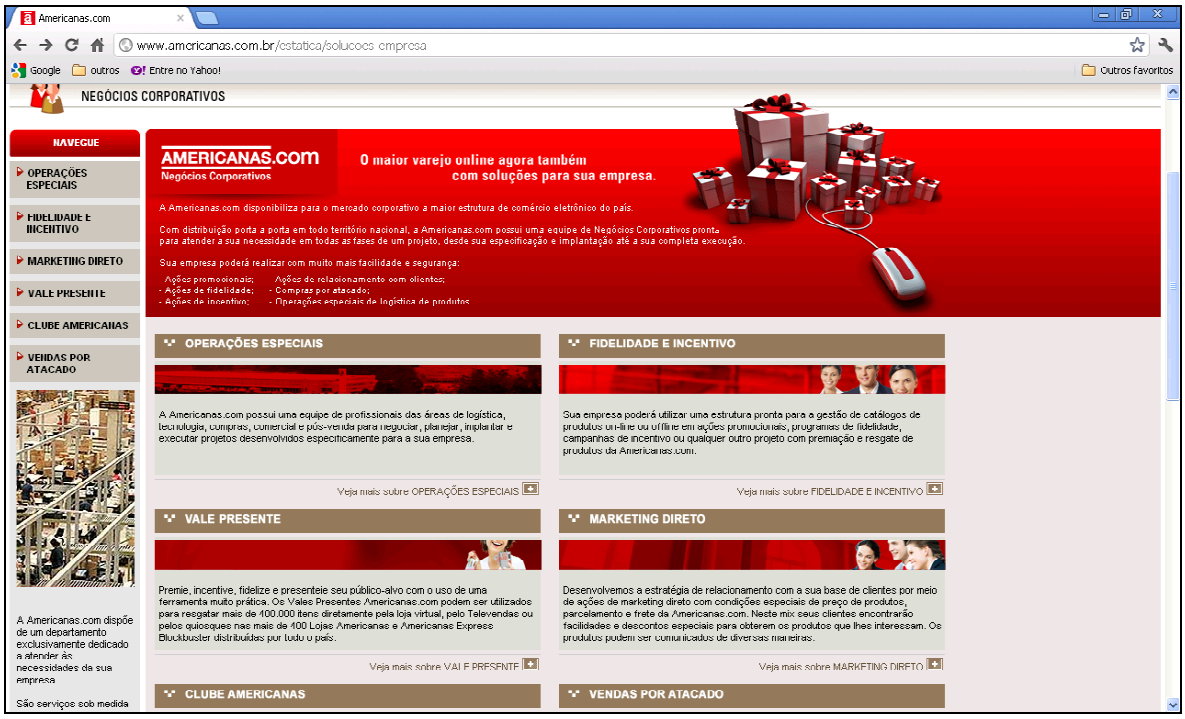

Figura 1.4 – Loja virtual Americanas.

Apesar dessas lojas serem conhecidas como *Business-to-Consummer* (B2C) elas possuem a opção de negócios corporativos, incluindo assim algumas características do modelo B2B como a solicitação de cotação, opções especiais de preço, entre outras. Mesmo sendo possível no tipo B2C, a compra de grandes quantidades nesse modelo de comércio, não são concedidos descontos nesses casos, o que pode inviabilizar para as empresas e o contrário para as pessoas físicas que adquirem quantidades pequenas para uso pessoal.

1.1.3 Consummer-to-Consummer (C2C)

Este modelo de comércio eletrônico é baseado na negociação de consumidores com consumidores e normalmente não envolvem grandes valores. Neste tipo não existem empresas diretamente envolvidas na negociação, mas sim para intermediar a comunicação entre as partes. Este sistema pode parecer inseguro já que as partes envolvidas não se conhecem, porém um exemplo de *Consummer-to-Consummer* (C2C) bastante conhecido no Brasil que mostra o contrário disso é o Mercado Livre (figura 1.5), o qual disponibiliza o serviço de compra e venda para seus usuários que sujeitam-se a respeitar algumas regras (LEAL, 2004).

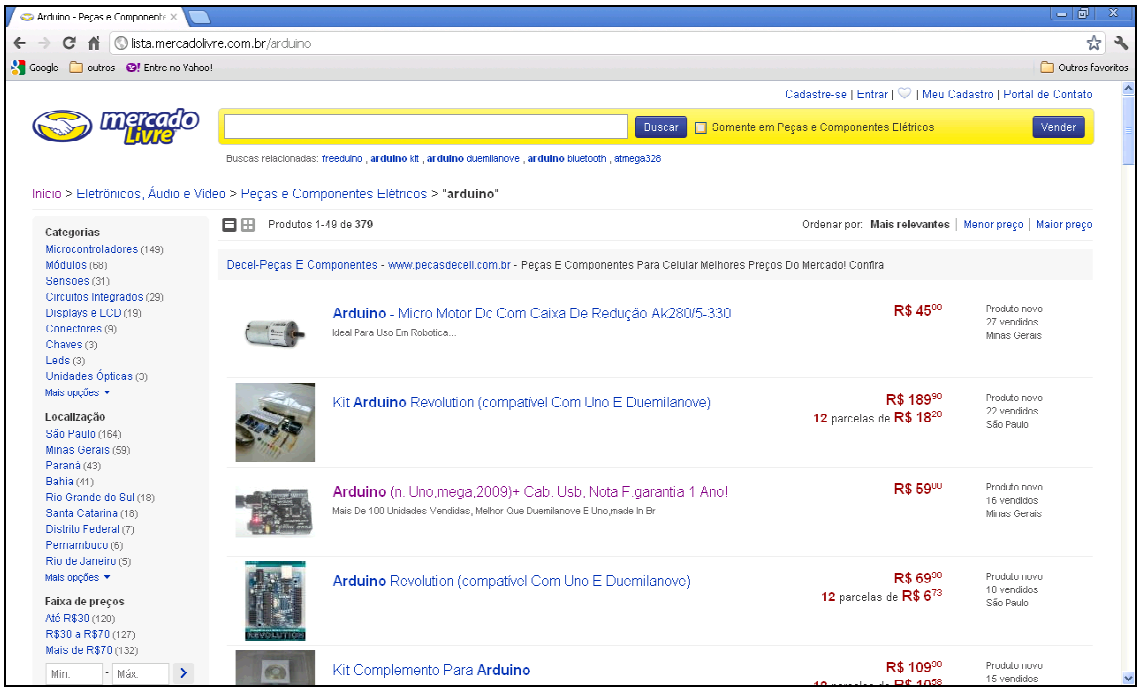

Figura 1.5 – Plataforma Mercado Livre.

Este serviço, possui também uma plataforma de pagamento denominada “Mercado Pago”, a qual tem a finalidade de oferecer maior segurança tanto para quem vende quanto para quem compra. Outro fator que traz maior segurança são as avaliações que compradores e vendedores podem fazer um do outro, possibilitando assim que futuros compradores possam verificar a reputação dentro do sistema e o número de vendas realizadas, juntamente com as avaliações feitas após a compra realizada e concretizada entre ambos.

Neste capítulo foram apresentadas as definições dos tipos de comércio eletrônico, suas características e alguns exemplos de lojas. No próximo capítulo será apresentada a metodologia OWASP que é a base para a realização do estudo com a finalidade de analisar a segurança dos ambientes selecionados.

2 OWASP

OWASP é uma comunidade aberta sem ligação com nenhuma empresa comercial, onde qualquer pessoa interessada pode participar, tendo como objetivo colaborar para a construção de softwares mais seguros (OWASP, 2008). Nesse sentido, existem vários documentos publicados por ela, mas três deles são os principais que abordam a fase de desenvolvimento, a revisão de código e os testes de segurança, com a finalidade de obter mais segurança.

Conforme consta no *OWASP Testing guide*, o qual será a base para a realização deste trabalho, o mesmo pode ser usado por diversos profissionais:

- Desenvolvedores: para garantir que está sendo produzido um software seguro;
- Testador: para aumentar o número de casos de testes;
- Especialistas de segurança: para combinar o conhecimento com novas técnicas.

O guia possibilita assim que as empresas compreendam a maneira que devem ser testados os aplicativos em um *pentest*. Esta técnica é usada com alguma frequência nas empresas para testar o funcionamento dos softwares ou redes. Segundo Danhieux e Wouter (2006, APUD JUNIOR, 2010) o *pentest* é um método para avaliar a segurança dos sistemas de informação simulando um ataque. Seu objetivo é sondar e identificar falhas em sistemas de informação.

A aplicação destes testes é norteada por alguns princípios durante todo o seu processo, a seguir listam-se alguns deles (OWASP, 2008):

- Não existe “bala de prata”: scanner ou *firewall* não irão fornecer todas as defesas ou identificar todas as brechas abertas;
- Entender o escopo da segurança: saber quanto de segurança um projeto vai exigir para que as informações fiquem protegidas. Classificá-las como devem ser tratadas, por exemplo: confidencial, secreto, etc;
- Desenvolver a mentalidade certa: pensar nos testes como se fosse um invasor, não testar apenas o comportamento normal de um usuário;
- Entenda o assunto: para um bom programa de segurança, exigir a documentação da aplicação, como, casos de uso, diagramas de fluxo e todo tipo de documento para ser analisado;

- Use as ferramentas certas: apesar de não identificarem todas as falhas, elas têm função importante, pois aceleram o processo, mas para isso é preciso conhecer a ferramenta para que seja usada de forma correta.

Além desses princípios existem outros que podem ser encontrados no guia de testes da OWASP.

A metodologia de teste de penetração é do tipo *black box*, onde a execução é sem prévio conhecimento do software (WHITAKER; NEWMAN, 2006). Sendo assim, o modelo de testes da OWASP consiste em três fatores:

1. O testador que realiza a execução
2. A metodologia e as ferramentas, que são o núcleo da técnica
3. O software que é a caixa preta a ser testada.

Além disso, este modelo é divido em duas fases (OWASP, 2008):

- Modo passivo: é onde o testador procura entender a lógica da aplicação. Nessa etapa podem ser usadas ferramentas para buscar informações, como por exemplo, sobre os protocolos de comunicação que são usados, bem como, os possíveis pontos de acesso que serão usados na segunda etapa, como o da figura 2.1, que mostra o endereço de login de uma aplicação, juntamente com dois parâmetros;

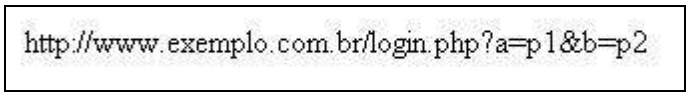

Figura 2.1 – Exemplo de ponto de acesso.
Fonte: Adaptado de OWASP, 2008.

- Modo ativo: nesta fase são executados os testes propriamente ditos, estes são divididos em nove categorias e sessenta e seis possíveis aéreas a serem testadas.

A tabela 2.1, apresenta as categorias mencionadas e as áreas que podem ser analisadas, na busca por alguma vulnerabilidade:

Tabela 2.1 - Categoria de testes.

Categoria	Número de ref.	Nome do teste
Obtenção de informação	OWASP-IG-001	*Spiders, Robots and Crawlers*
	OWASP-IG-002	*Search Engine discovery/Reconnaissance*
	OWASP-IG-003	*Identify application entry points*
	OWASP-IG-004	*Testing for Web Application Fingerprint*
	OWASP-IG-005	*Application Discovery*
	OWASP-IG-006	*Analysis of Error Codes*

Categoria	Número de ref.	Nome do teste
Teste de gestão de configuração	OWASP-CM-001	*SSL/TLS Testing*
	OWASP-CM-002	*DB Listener Testing*
	OWASP-CM-003	*Infrastructure Configuration Management Testing*
	OWASP-CM-004	*Application Configuration Management Testing*
	OWASP-CM-005	*Testing for File Extensions Handling*
	OWASP-CM-006	*Old, backup and unreferenced files*
	OWASP-CM-007	*Infrastructure and Application Admin Interfaces*
	OWASP-CM-008	*Testing for HTTP Methods and XST*
Teste de autenticação	OWASP-AT-001	*Credentials transport over an encrypted channel*
	OWASP-AT-002	*Testing for user Enumeration*
	OWASP-AT-003	*Testing for Guessable (Dictionary) User Account*
	OWASP-AT-004	*Brute Force Testing*
	OWASP-AT-005	*Testing for bypassing authentication schema*
	OWASP-AT-006	*Testing for vulnerable remember password and pwd reset*
	OWASP-AT-007	*Testing for Logout and Browser Cache Management*
	OWASP-AT-008	*Testing for CAPTCHA*
	OWASP-AT-009	*Testing Multiple Factors Authentication*
	OWASP-AT-0010	*Testing for Race Conditions*
Gerenciamento de sessão	OWASP-SM-001	*Testing for Session Management Schema*
	OWASP-SM-002	*Testing for Cookies Attributes*
	OWASP-SM-003	*Testing for Session Fixation*
	OWASP-SM-004	*Testing for Exposed Session Variables*
	OWASP-SM-005	*Testing for CSRF*
Teste de autorização	OWASP-AZ-001	*Testing for Path Traversal*
	OWASP-AZ-002	*Testing for bypassing authorization schema*
	OWASP-AZ-003	*Testing for Privilege Escalation*
Teste de lógica do negócio	OWASP-BL-001	*Testing for business logic*
Teste de validação de dados	OWASP-DV-001	*Testing for Reflected Cross Site Scripting*
	OWASP-DV-002	*Testing for Stored Cross Site Scripting*
	OWASP-DV-003	*Testing for DOM based Cross Site Scripting*
	OWASP-DV-004	*Testing for Cross Site Flashing*
	OWASP-DV-005	*SQL Injection*
	OWASP-DV-006	*LDAP Injection*
	OWASP-DV-007	*ORM Injection*
	OWASP-DV-008	*XML Injection*
	OWASP-DV-009	*SSI Injection*
	OWASP-DV-010	*XPath Injection*

Categoria	Número de ref.	Nome do teste
	OWASP-DV-011	*IMAP/SMTP Injection*
	OWASP-DV-012	*Code Injection*
	OWASP-DV-013	*OS Commanding*
	OWASP-DV-014	*Buffer overflow*
	OWASP-DV-015	*Incubated vulnerability Testing*
	OWASP-DV-016	*Testing for HTTP Splitting / Smuggling*
Teste de negação de serviço	OWASP-DS-001	*Testing for SQL Wildcard Attacks*
	OWASP-DS-002	*Locking Customer Accounts*
	OWASP-DS-003	*Testing for DoS Buffer Overflows*
	OWASP-DS-004	*User Specified Object Allocation*
	OWASP-DS-005	*User Input as a Loop Counter*
	OWASP-DS-006	*Writing User Provided Data to Disk*
	OWASP-DS-007	*Failure to Release Resources*
	OWASP-DS-008	*Storing too Much Data in Session*
Teste Web Services	OWASP-WS-001	*WS Information Gathering*
	OWASP-WS-002	*Testing WSDL*
	OWASP-WS-003	*XML Structural Testing*
	OWASP-WS-004	*XML content-level Testing*
	OWASP-WS-005	*HTTP GET parameters/REST Testing*
	OWASP-WS-006	*Naughty SOAP attachments*
	OWASP-WS-007	*Replay Testing*
Teste Ájax	OWASP-AJ-001	*AJAX Vulnerabilities*
	OWASP-WS-002	*AJAX Testing*

Fonte: OWASP Testing Guide, 2008.

Porém, a natureza desses testes não é uma ciência exata, a documentação cita que deve-se tentar abranger todos os pontos possíveis e que há algo importante com que se preocupar, não passar neles é sinal que há problemas sérios de segurança, mas, como pode-se pensar, a aprovação nos testes não significa que não há problema algum mas que este não foi encontrado (OWASP, 2008).

Este capítulo tratou de apresentar a metodologia OWASP, detalhando seu princípios, fases, categorias de testes e também para quais profissionais ela é destinada. No próximo capítulo será exposta as principais técnicas utilizadas para exploração de vulnerabilidades e a partir desse conhecimento entender melhor como tornar os sistemas mais seguros.

3 TÉCNICAS

Para proteger um sistema é preciso conhecer as técnicas que são utilizadas por invasores e como elas funcionam. Tendo esse conhecimento, pode-se tomar medidas com o objetivo de evitar ou amenizar os danos. Desta forma, este capítulo tem como o objetivo apresentar as principais técnicas de ataque.

3.1 Footprint

Esta técnica é realizada anteriormente ao ataque, ela envolve a obtenção de informações sobre sistema alvo, e onde poderão ser encontradas possíveis falhas. O processo de *footprint* começa com a definição do sistema alvo, para depois, coletar informações específicas sobre a empresa (GRAVES, 2007). Esta coleta pode ser feita por meio de um motor de busca. Esse processo ficou conhecido como *Google hacking*, por meio dele tem-se a possibilidade de tomar conhecimento sobre funcionários, além de outras informações, e assim planejar um ataque de engenharia social (GRAVES, 2007). Seguem alguns comandos que podem ser usados para a sua execução (CARMONA, 2004):

- Filetype: procura apenas dentro de um determinado tipo de arquivo. Deve ser informada a extensão do arquivo sem o ponto;
- Intitle: busca por um termo no título de um arquivo;
- Inurl: pesquisa apenas dentro de uma URL;
- Site: procura somente em um domínio específico;
- Link: busca dentro de um hyperlink.

Um exemplo dessa ferramenta para reunir informações ficaria como a figura 3.1:

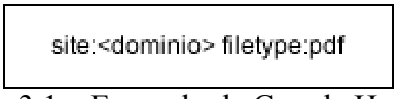

Figura 3.1 – Exemplo de Google Hacking.

O resultado desse comando seriam arquivos *Portable Document Format* (PDF) armazenados no domínio do site informado. Além desse método para obter dados, existem outros, como o uso do *whois*, que é um protocolo para consultar informações sobre um determinado domínio (BRAGA; TAVARES, 2011). Com ele é possível conhecer quem é o

responsável e também alguns dados para contato como pode ser observado na figura 3.2 (JUNIOR, 2010).

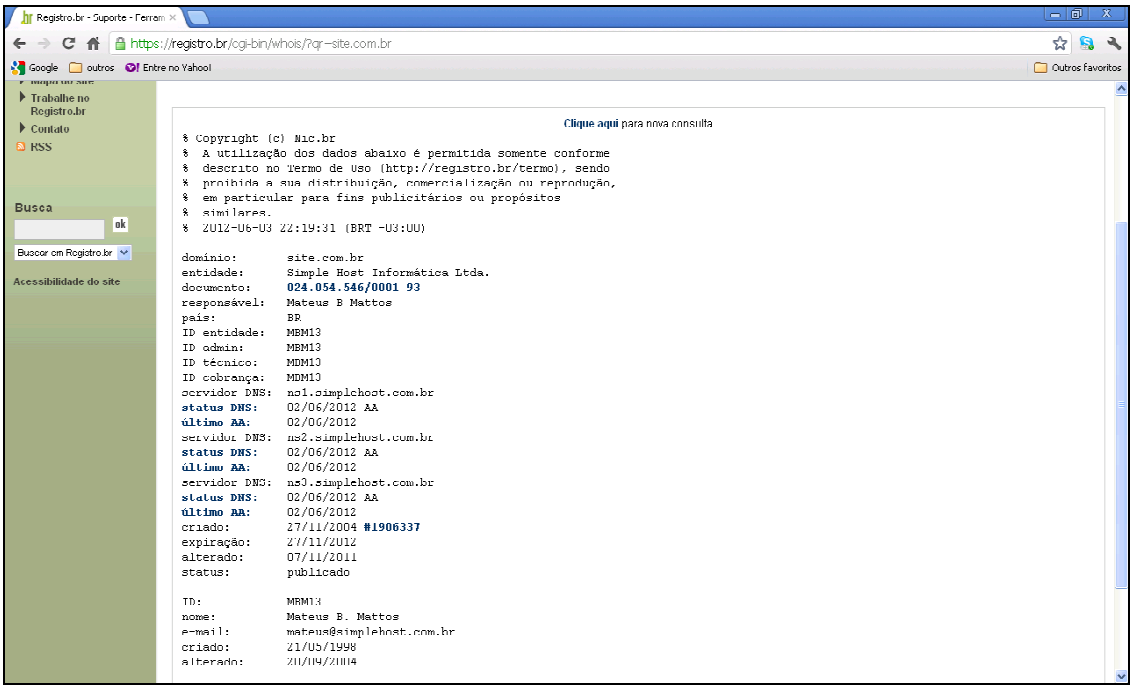

Figura 3.2 – Exemplo de pesquisa whois.

3.2 SQL injection

Sql injection é uma das técnicas mais utilizadas para exploração de vulnerabilidades em aplicações *web*. Ela consiste na inserção de códigos *Structered Query Language* (SQL) em campos ou parâmetros para aqueles que não forem validados, sejam executados de maneira arbitrária. Sua execução pode ser simples se o usuário de acesso ao banco de dados tiver privilégios restritos, ou mais complexa, sendo possível excluir, adicionar ou alterar os registros (UTO; MELO, 2009). Este tipo de ataque acontece em decorrência da construção dos comandos de forma dinâmica, que são concatenados com os dados recebidos do usuário. Esses dados sem o devido tratamento podem adicionar um novo comando ou alterar a estrutura do que foi definido. O teste para verificar se o software possui esse defeito é simples, informando um apóstrofo no campo ou na URL, como mostra a figura 3.3 (UTO; MELO, 2009):

```
http://localhost/site/noticias.php?id=11'
```
Figura 3.3 – Exemplo de teste SQL injection.

Caso for retornada alguma mensagem de erro, como exemplifica a figura 3.4, a resposta é positiva.

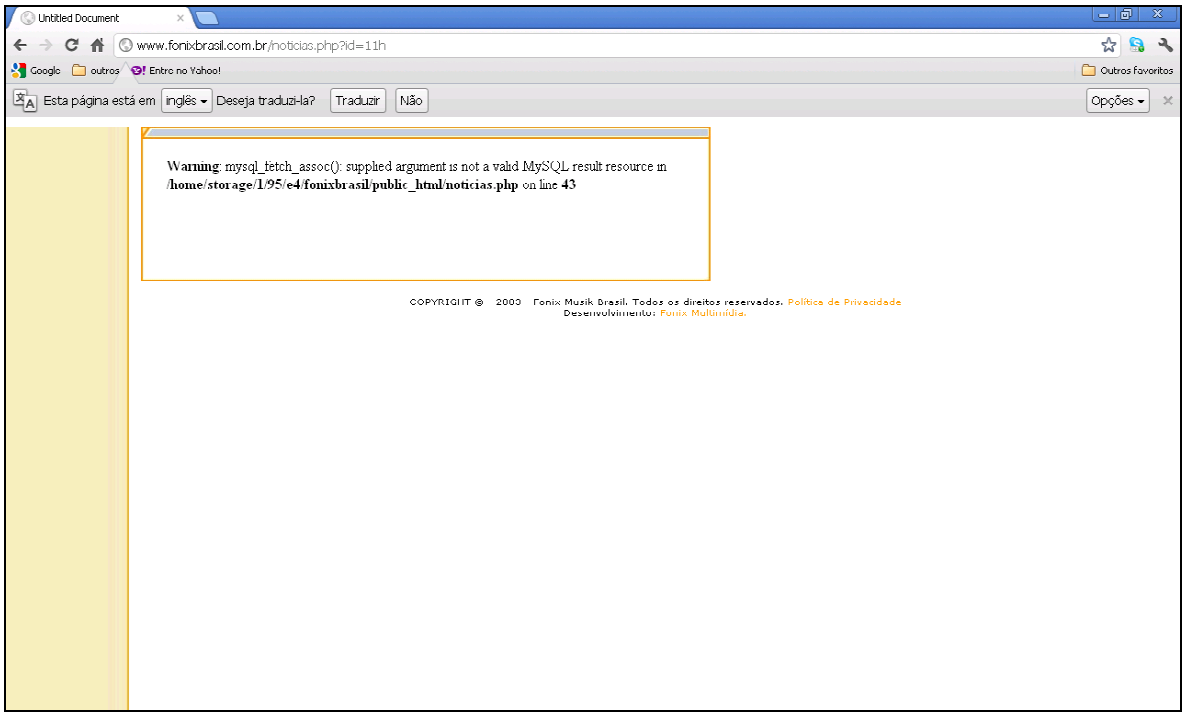

Figura 3.4 – Exemplo de resposta de teste SQL injection.

Para melhor entender considere uma página que utiliza linguagem dinâmica (PHP, JSP ou ASP) e serve para fazer a alteração da senha de um usuário. Ela recebe como entrada usuário e uma nova senha para assim, construir uma instrução SQL que irá fazer a atualização dos dados, como representa o código na figura 3.5 (FARIAS, 2009):

```
UPDATE usuário SET senha='nova_senha' WHERE usuario='usuario'
```

Figura 3.5 – Exemplo de comando SQL.
Fonte: Farias, 2009.

O problema é que se as entradas não forem validadas de maneira correta, a lógica da instrução pode ser alterada e com isso causar alguns transtornos, como a alteração da senha de todos os usuários. Isto acontece se a entrada "nova_senha' *where* 1=1 --" for informada como sendo a nova senha, com isso, a instrução resultante ficaria como a figura 3.6 (FARIAS, 2009):

```
UPDATE usuario SET senha='nova_senha' where 1=1 --' where usuario='joao';
```

Figura 3.6 – Exemplo de SQL injection.
Fonte: Farias, 2009.

Como pode ser observado, o dado informado foi concatenado com o comando, alterando assim a cláusula *where* original. Devido à sequência "--" atuar como um

comentário, este comando SQL alteraria as senhas de todos os usuários do sistema, possibilitando ao invasor usar qualquer conta (FARIAS, 2009).

Neste tipo vulnerabilidade é possível efetuar um ataque mais elaborado não usando apenas comandos do tipo *Data Manipulation Language* (DML) como *SELECT*, *DELETE*, *UPDATE* como também do tipo *Data Definition Language* (DDL). Este outro tipo permite a manipulação de estruturas, como criar ou apagar tabelas. No entanto, para se ter sucesso é necessário que tenha os privilégios mínimos para essas ações.

3.2.1 Medidas preventivas

Para tentar evitar que um sistema possua tal vulnerabilidade e seja alvo de um ataque ou na tentativa de amenizar os danos causados é necessário empregar alguns métodos (GUMERATO, 2009):

- Receber todas as informações do usuário, como se fossem maliciosas. Assim, todos os dados recebidos devem passar por uma validação antes de serem processados ou concatenados.
- Tratar os erros recebidos do sistema gerenciador de banco de dados(SGDB) para que sejam exibidas mensagens genéricas e que não forneçam informações que possam ser importantes para o atacante.
- Utilizar uma conta de acesso ao banco de dados com apenas os privilégios necessários para a execução do sistema. Nunca utilize usuários administrativos, pois seriam potencializados os danos, sendo possível remoção, inclusão e alteração de tabelas, registros ou índices.
- Fortalecer o SGDB também é um passo que pode ajudar. Através da retirada de objetos, usuários e privilégios que não serão utilizados, com essa ação, caso um ataque desse tipo seja bem sucedido, os danos seriam amenizados.

3.3 Cross-site scripting

Cross-site scripting também conhecido como XSS, tira proveito da confiança que o navegador tem no servidor, com isso possibilita o envio de código malicioso, normalmente

JavaScript, para o usuário (SOUZA, 2009). Com esta técnica é possível roubar o histórico de navegação e descobrir consultas realizadas em mecanismos (GROSSMAN et al, 2007). Essa vulnerabilidade acontece quando uma aplicação *web*, não trata de forma adequada as informações recebidas e as insere diretamente na página que é gerada dinamicamente. Isso ocorre, pois o navegador reconhece como sendo um código legítimo da aplicação, com isso, o mesmo é executado, afetando a ele e não ao servidor. O XSS é dividido em três tipos (JUNIOR, 2009):

1. Persistente.
2. Não persistente.
3. *Document Object Model (DOM) Based*.

3.3.1 Cross-site scripting não persistente

Este é o tipo de *cross-site scripting* mais comum na internet (JUNIOR, 2009). Ele depende da ação do usuário, na maioria das vezes com um clique em algum *link*, por isso é um dos responsáveis por ataques de *phishing*, que tem como objetivo adquirir dados da vítima. Essa técnica de XSS gera uma URL com um domínio confiável e com isso tira proveito da confiança que o usuário tem neste domínio (GROSSMAN et al, 2007). Para contextualizar melhor, imagine um site de comércio eletrônico onde os clientes para realizar a autenticação informam nome de usuário e senha, caso corretos, recebem um *cookie* como retorno. Neste site existe um formulário para busca, onde os dados de entrada não recebem o devido cuidado. Com isso o *hacker* realiza a busca com um código JavaScript para confirmar a vulnerabilidade, como mostra a figura 3.7 (GROSSMAN et al, 2007).

Figura 3.7 – Exemplo de XSS.
Fonte: Grossman et al, 2007.

Neste momento, após a confirmação, o atacante elabora um código malicioso para gerar uma URL usando o mesmo método e envia-a por e-mail ou mensagem instantânea. A vítima ao clicar no *link*, entrará no site confiável e terá o código embutido executado pelo navegador. Possibilitando assim, que o criminoso colete informações e as envie para um servidor próprio como no exemplo da figura 3.8 em que o *cookie* de autenticação é roubado.

```
http://server/shopping_cart.cgi?search="><SCRIPT>var+img=new+Image();
img.src="http://hacker/"%20+%20document.cookie;</SCRIPT>
```

Figura 3.8 – Exemplo de URL com XSS.
Fonte: Grossman et al, 2007.

3.3.2 Cross-site scripting persistente

O tipo de XSS persistente diferente do anterior, não necessita do envio de uma URL para a vítima, pois a página que será visitada irá conter o código malicioso, que foi inserido pelo *hacker* (JUNIOR, 2009). O código normalmente fica armazenado em um banco de dados. Esta técnica pode passar a ideia de que não tem poder de causar grandes problemas, mas com a escrita de um código mais elaborado, é possível (UTO; MELO, 2009):

- Instalar *malware*;
- Sequestrar sessão;

- Redirecionar para outras páginas;
- Escravizar navegadores.

Com essas possibilidades exibidas, percebe-se a gravidade e os efeitos que podem ser causados. Uma ilustração (figura 3.9) de como acontece a escravização de *browsers* pode começar com um visitante postando um comentário em um fórum. Caso o site aceite *tags* HTML no campo de comentários e não valide esses dados, o atacante poderá inserir uma instrução maliciosa nesse campo. Quando o usuário tentar realizar a autenticação será redirecionado para a página que contém o código inserido, que ao ser executado conecta-se com o gerenciador, e com isso o navegador acaba sendo escravizado. A partir desse ponto o atacante poderá executar instruções no "zumbi", para realizar diversas tarefas como as já citadas anteriormente.

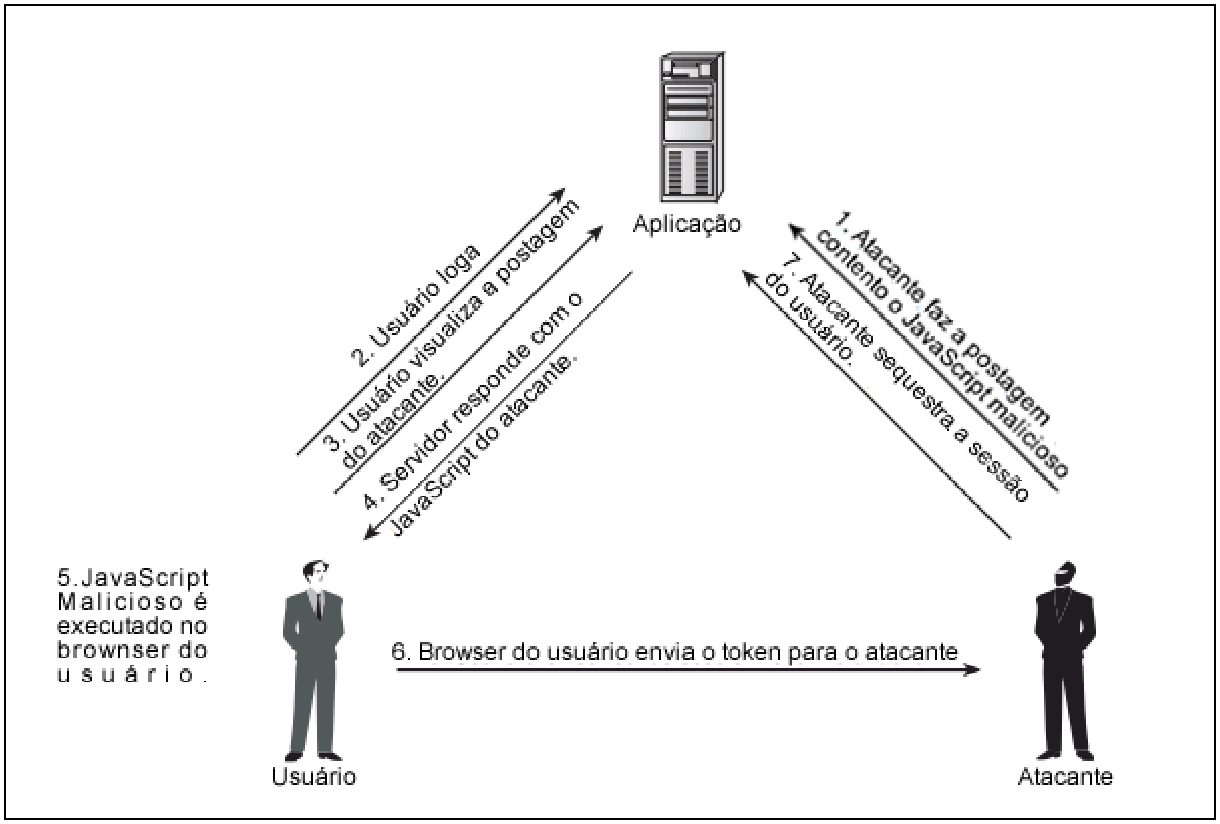

Figura 3.9 – Funcionamento do XSS persistente.
Fonte: Fonte: Adaptado de Stuttard e Pinto, 2008.

3.3.3 Cross-site scripting DOM Based

Cross-site scripting DOM based é outra forma de XSS, diferente das duas apresentadas anteriormente, ela tem um comportamento de manipular os dados e retornar o resultado no código fonte da página, tornando-a insegura. Este terceiro tipo atua conforme o processo a seguir e pode ser visualizado na figura 3.10 (STUTTARD; PINTO, 2008):

- O usuário faz a solicitação de uma página fornecida pelo atacante e que contém um código JavaScript;
- A resposta do servidor não possui nenhum script malicioso em qualquer forma;
- Quando o navegador processa a resposta o script é executado.

Este ataque se torna possível quando os parâmetros são extraídos e processados, gerando um conteúdo que depois será inserido na página de forma dinâmica (STUTTARD; PINTO, 2008).

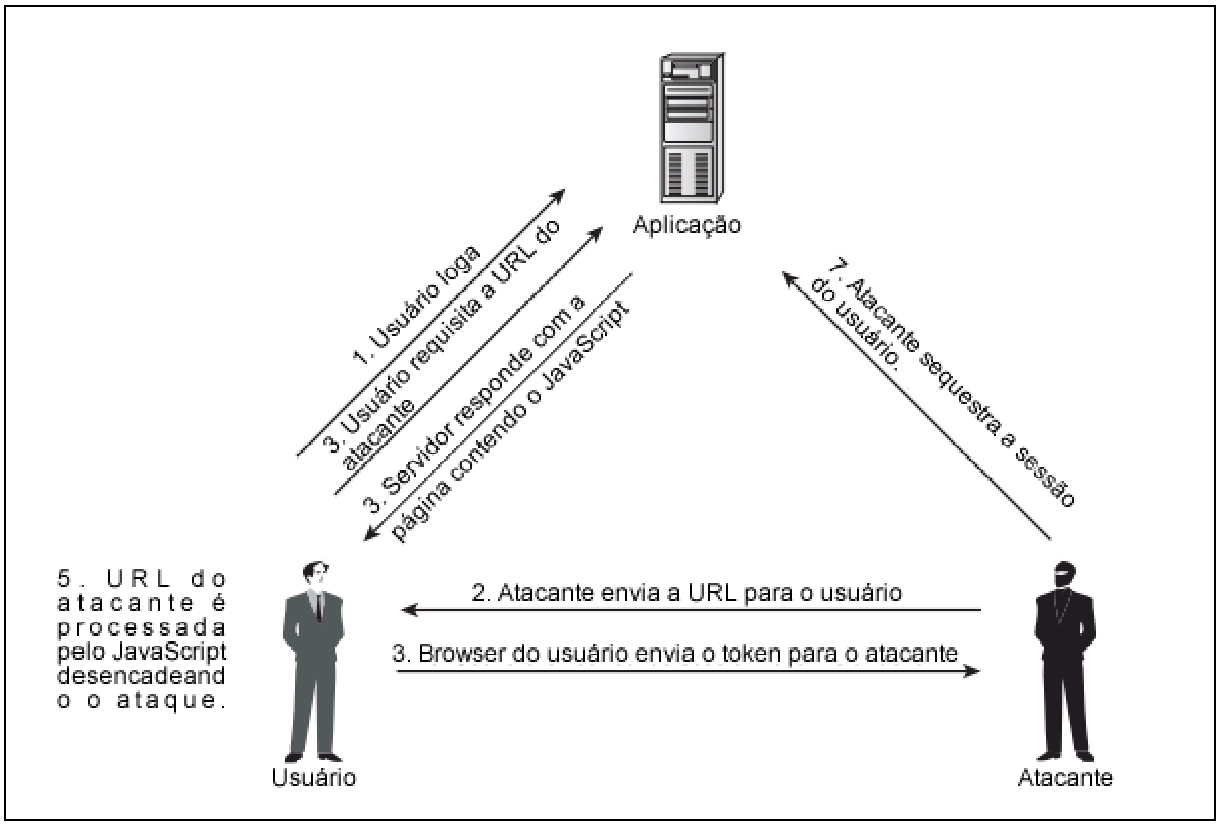

Figura 3.10 – Funcionamento do XSS DOM Based.
Fonte: Adaptado de Stuttard e Pinto, 2008.

O trecho da figura 3.11 extrai o valor do parâmetro mensagem e escreve nele. Desse modo é possível escrever no conteúdo da página de forma dinâmica, parecendo assim como se tivesse sido retornado pelo servidor.

```
<script>
var a = document.URL;
a = unescape(a);
document.write(a.substring(a.indexOf("mensagem=")+8,a.length));
</script>
```

Figura 3.11 – Exemplo de comando JavaScript para exploração XSS DOM Based.
Fonte: Stuttard e Pinto, 2008.

3.3.4 Medidas preventivas

Tendo conhecimento da técnica de XSS, é necessário empregar algumas estratégias para tentar solucionar esse problema, como as seguintes (SCAMBRAY; LIU; SIMA. 2011):

- Receber todas as informações do usuário, como se fossem maliciosas. Assim, todos os dados recebidos devem passar por uma validação antes de serem processados
- Restrinja o tamanho do campo para que contenha apenas o necessário.
- Utilizar a codificação HTML na saída dos dados, substituindo caracteres usados em scripts maliciosos, pois este processo os substitui por um correspondente em HTML. Sendo assim, se for informada uma string como "<script>" no momento de ser exibida para o usuário ela seria codificada, ficando <script>. Na tabela 3.1 estão os caracteres mais problemáticos e as suas codificações.

Tabela 3.1 – Caracteres usados em scripts maliciosos.

Caractere	"	'	&	<	>
Código	"	'	&	<	>

Fonte: Stuttard e Pinto, 2008.

3.4 Cross-site request forgery

Cross-site request forgery (CSRF), também conhecido como *session ridding* e XSRF é uma técnica que utiliza a sessão de um usuário já autorizado pela aplicação vulnerável, para aproveitar-se da confiança que a aplicação tem de que é o usuário que está fazendo as requisições, e com isso realizar ações sem o consentimento da vitima (SOUZA, 2009). A exploração de CSRF acontece, quando um sistema web não garante que as requisições de um usuário estão em um mesmo contexto, possuindo uma lógica de execução, ou seja, que os dados de entrada tenham vindo da tela de captura, e processados da forma planejada pelo programador. A preocupação que se deve ter é devido ao problema de como verificar se uma

solicitação foi intencional ou não. Apesar dos desenvolvedores não se preocuparem muito o CSRF, é aconselhável ter uma atenção a ele de igual forma como para as outras técnicas, pois já causou bastante incômodo a sites populares como: Twitter, Facebook, MySpace e outras redes sociais. Um site que sofreu com esse método foi o Netflix em 2009, quando era possível realizar as ações de (SCAMBRAY; LIU: SIMA, 2011):

- Adicionar filmes para fila de locações;
- Habilitar e desabilitar informações adicionais de um filme;
- Alterar e-mail e senha de uma conta;
- Alterar nome e endereço de uma conta;
- Cancelar contas.

3.4.1 Medidas preventivas

A seguir algumas formas para tentar evitar ataques CSRF (SCAMBRAY;LIU:SIMA, 2011):

- Prefira utilizar o método de envio de formulários *POST* no lugar de *GET*, pois isso irá dificultar esse tipo de técnica;
- Ao realizar ações críticas, como excluir um dado importante, pedir para que o usuário se autentique novamente, para assim, caso a solicitação seja feita pelo atacante, a mesma não seja aceita;
- Em transações sensíveis utilizar um campo do tipo *hidden* para enviar o ID da sessão atual e armazenar esta informação em um *cookie* no lado do cliente. O servidor ao receber o formulário vai verificar se a informação do cabeçalho *hypertext transfer protocol* (HTTP) vai coincidir com o valor do *cookie*. Caso os valores sejam diferentes rejeitar o pedido e registrar em um *log*.

3.5 Denial of Service

Ataques *Denial of Service (DoS)* também conhecidos como, negação de serviço consiste na tentativa de impedir que um usuário legítimo de um sistema o utilize (BERTHOLDO; ANDREOLI; TAROUCO, 2003). Esse impedimento se dá por meio de uma

inundação de requisições falsas para um determinado site, até o ponto de causar uma sobrecarga, com isso, ele acaba recusando as solicitações de usuários reais, impedindo-lhes que acessem alguns dados aos quais queiram. Há uma sub-categoria desse tipo que é o ataque de negação de serviço distribuído. A diferença é basicamente a atuação de mais de um atacante disparando contra um mesmo alvo (MEDEIROS, 2001). Esses atacantes podem ser vários computadores zumbis comandados por um *hacker* ou com a ajuda de outros participantes. Este tipo de técnica é empregada quando se tem como objetivo causar um prejuízo a uma empresa ou desativar seu site temporariamente.

Este tipo de ataque é de fácil uso, mas em contrapartida de difícil defesa, pois podem explorar alguns fatores para alcançar o objetivo de indisponibilizar o serviço, tais como (2000, APUD MEDEIROS, 2001):

- *Flooding:* O envio de várias solicitações em curto período de tempo, sobrecarregando a vítima que acaba não conseguindo processar todas as informações e descartando algumas requisições legitimas;
- Pacotes anormais: O protocolo TCP/IP não trata o recebimento de pacotes incorretos, sendo assim, a máquina que receber pode acabar tendo problemas de travamento;
- *Buffer overflow*: Outro fator que pode ser explorado com a técnica de DoS é caso o software tenha uma falha que envolva a alocação de memória. Com este problema de implementação é possível até mesmo executar algum código remotamente.

Mesmo não causando a perda de informações, nem o roubo, essa técnica ao indisponibilizar o sistema pode resultar em muitos prejuízos para a empresa (LAUFER, 2005).

3.5.1 Medidas preventivas

Mesmo com a dificuldade de evitar ataques do tipo *Denial of Service,* há algumas medidas que podem ser realizadas na tentativa de solucionar (LAUFER, 2005):

- Manutenção dos softwares atualizados;
- Filtragem dos pacotes forjados pelos endereços IP de origem;

Caso estas tentativas não funcionem, há a possibilidade de alguma medida reativa, através do rastreamento dos pacotes para identificar o emissor e com essa informação entrar com ação judicial para que os ataques cessem.

3.6 Buffer overflow

Este tipo de vulnerabilidade é explorada quando ocorre um erro no acesso a partes da memória e assim possibilita realizar ações não permitidas. Essa exploração consiste na inserção de dados em um campo, onde o tamanho desses dados é maior do que ele suporta (STUTTARD; PINTO, 2008). Exemplificando, imagine dois campos de 32 bytes onde devem ser informados o usuário e a senha para obter acesso. Caso o usuário informado seja maior que 32 caracteres, ocorreria o estouro do *buffer*. Sendo assim, o atacante pode inserir mais de 32 caracteres e com isso, os dados excedentes substituiriam instruções nos espaços de memória adjacentes, logo, possibilitando ataques *Denial of Service* e ganhar acesso privilegiado (MEDEIROS, 2001). Apesar de já haver proteções contra esse tipo de ataque, ela pode trazer consequências consideráveis se for explorada, como mencionado anteriormente. A busca dessa falha dá-se através do envio de uma quantidade grande de dados e o monitoramento da resposta do servidor, caso ela esteja presente será possível identificá-la através desse processo (SCAMBRAY; LIU; SIMA, 2011).

3.6.1 Medidas preventivas

A falha de *buffer overflow* dá-se devido a uma limitada ou inexistente checagem de erro. Em virtude disso são apresentadas algumas maneiras para se proteger (ROMERO; KACUTA; OLIVEIRA, 2003):

- Aplicar os *patches* do fornecedor ou instalar a última versão do software;
- Testar aplicações chaves;
- Executar o software com o menor privilégio possível;
- Filtrar o tráfego específico no *firewall*.

3.7 Path transversal

O ataque de path transversal tem como objetivo acessar arquivos ou diretórios que estão armazenados além do diretório da aplicação. Com essa técnica o invasor pode ter acesso a dados sensíveis como senhas, *logs* ou até mesmo arquivos do sistema operacional. Esse tipo de falha acontece quando a aplicação acessa arquivos que estão no servidor para leitura ou escrita. Com isso, se esses parâmetros não forem tratados da forma adequada o atacante pode comprometer o sistema (STUTTARD; PINTO, 2008). Como exemplo, imagine um aplicativo que em determinada página exibe uma imagem, a qual é passada como parâmetro (figura 3.12):

```
http://aplicativo.com/getImage.aspx?arquivo=imagem1.jpg
```

Figura 3.12 – Exemplo de URL vulnerável a path transversal.
Fonte: Stuttard e Pinto, 2008.

Quando o servidor *web* recebe essa requisição, ele pega o parâmetro passado e acrescenta o caminho onde as imagens estão armazenadas (figura 3.13):

```
C:\aplicativo\imagens\
```

Figura 3.13 – Exemplo do caminho a ser explorado com path transversal.
Fonte: Stuttard e Pinto, 2008.

Assim, o servidor lê o arquivo com o nome informado e monta a página dinamicamente, que é exibida no navegador do usuário. Dessa forma, se o invasor utilizar comandos para recuar diretórios, ele poderá acessar outros lugares, como mostra a figura 3.14:

```
http://aplicativo.com/getImage.aspx?arquivo=..\..\windows\
```

Figura 3.14 – Exemplo de exploração com path transversal.
Fonte: Stuttard e Pinto, 2008.

Quando o parâmetro é concatenado com o caminho onde estão as imagens, o atacante estaria tendo acesso à pasta Windows (figura 3.15):

```
C:\aplicativo\imagens\..\..\windows\
```

Figura 3.15 - Exemplo de caminho alcançado com path transversal.
Fonte: Stuttard e Pinto, 2008.

Neste exemplo, a partir desse ponto o atacante poderia realizar outras ações e causar alguns transtornos.

3.7.1 Medidas preventivas

Com a intenção de eliminar e evitar que algum site tenha essa vulnerabilidade é necessário seguir algumas sugestões (STUTTARD; PINTO, 2008):

- Utilizar URL para acesso direto aos arquivos;
- respeitar uma lista dos tipos de arquivos permitidos;
- verificar se o nome do arquivo contém uma sequência de caracteres que realiza a mudança de diretório, caso encontrado, parar o processamento imediatamente;
- confirmar que arquivo passado como parâmetro está dentro do diretório inicial, especificado pelo aplicativo;
- implementar na aplicação a funcionalidade de gerar *logs* e enviar alertas para o administrador sempre que receber um pedido potencialmente malicioso.

No corrente capítulo foram apresentas as principais técnicas utilizadas por invasores, para no próximo mostrar algumas ferramentas que auxiliam na execução destas explorações e podem até automatizar essa tarefa.

4 FERRAMENTAS

Conhecendo as principais ameaças e técnicas empregadas na exploração de falhas, foram selecionadas algumas ferramentas sugeridas pela metodologia OWASP, que serão empregadas para identificar essas vulnerabilidades nos sistemas de comércio eletrônico.

4.1 Nessus

É um scanner de vulnerabilidades disponível para plataformas Windows, Linux e Mac Os. Desenvolvido pela *Tenable Network Security Inc* que aprimora o motor detecção e cria *plugins* para disponibilizar novas funcionalidades. O Nessus é dividido em duas partes, a primeira é o cliente, responsável por fazer as requisições de testes e através da sua interface web, representada pela figura 4.1, permite visualizar relatórios, administrar usuários, gerenciar os *scans* e as políticas (TENABLE NETWORK SECURITY, 2011). Estas políticas são as opções de configuração empregadas nas buscas por vulnerabilidades, as quais podem ser criadas ou modificar as já existentes conforme as necessidades do profissional que a utilize. Esta personalização possibilita a escolha dos métodos de busca por portas abertas e *plugins* que serão utilizados (TENABLE NETWORK SECURITY, 2011). Já a segunda parte do software, é o servidor, que processa as requisições, gera os relatórios e armazena a base de conhecimento e os *plugins* (DERAISON, 2004).

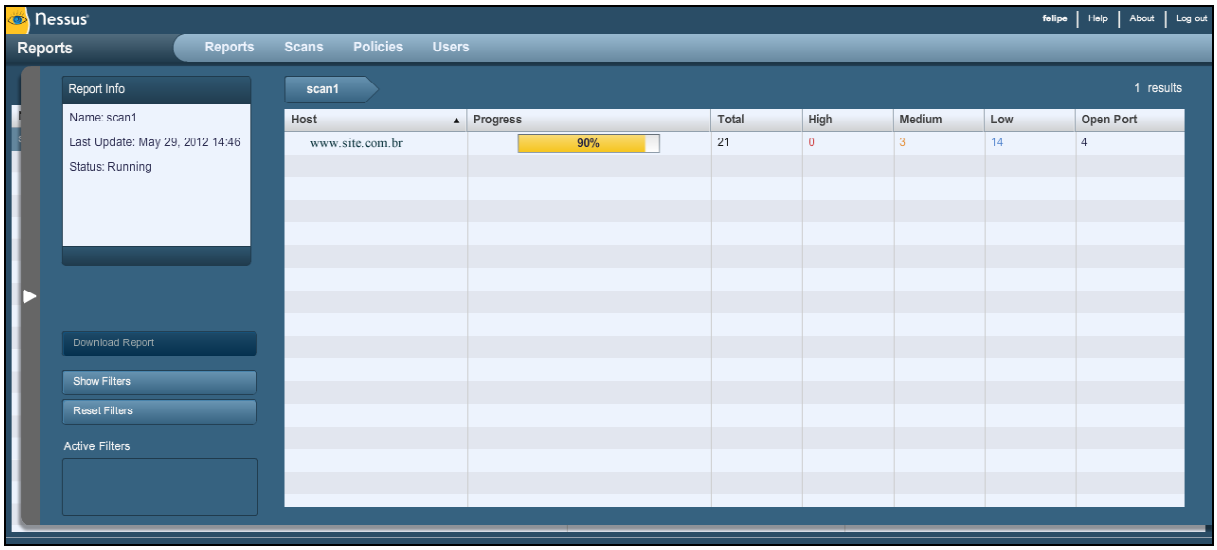

Figura 4.1 – Tela cliente do nessus em execução.

O funcionamento desta ferramenta começa com a autenticação do usuário. Se autorizado ele conecta-se ao servidor por meio do cliente e através deste escolhe a política, de acordo com o objetivo do *scan*. Com isso, é enviada uma solicitação para que o servidor inicie

a varredura de portas no *host* alvo para detectar servidores ativos, após verificar quais estão abertas, são executados códigos de exploração para realizar testes na busca por falhas. Durante a execução, o software permite acompanhar o andamento da atividade, exibindo as falhas encontradas e os níveis de riscos delas, como mostra as figuras 4.1 e 4.2. No segundo exemplo é exibida a descrição dos problemas, caso necessário é possível obter mais informações sobre eles e sugestões para a solução, clicando sobre seus nomes.

Figura 4.2 – Tela nessus vulnerabilidades encontradas.

4.2 WebScarab

É um *framework* multiplataforma desenvolvido em Java que é utilizado para análise de aplicações que utilizam os protocolos de comunicação HTTP e HTTPS. Além de sua portabilidade, outra vantagem é os diversos modos de operação, *fuzzer*, *webservices*, xss, etc. O WebScarab (figura 4.4) intercepta as solicitações e respostas entre o servidor e a aplicação, podendo modificá-las ou não (OWASP, 2012). Ele funciona como um *proxy*, portanto, é preciso configurar o navegador para utilizá-lo como tal, como mostra a figura 4.3, que representa a configuração no navegador Mozilla Firefox.

Figura 4.3 – Exemplo de configuração para o WebScarab.

Para dar início as interceptações é preciso ir à aba *Intercept* da ferramenta, marcar as opções *intercept requests* e *intercept reponses*. O software também permite selecionar o tipo de requisição, *get* ou *post*, que serão capturados. Com o início das capturas ele exibe automaticamente cada requisição realizada e caso desejado alguma alteração pode ser feita. No decorrer de sua atividade, já é indicado possíveis falhas dos tipos *injection* ou *cross-site scripting*, além de outras informações como: origem, método utilizado, parâmetros, etc.

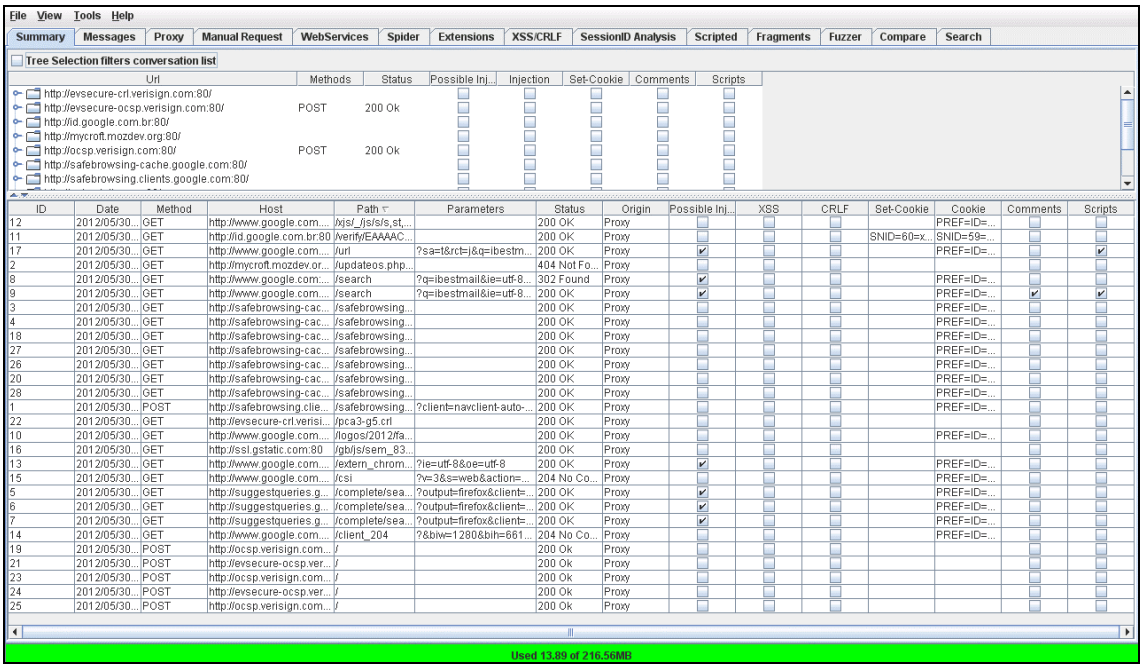

Figura 4.4 – Tela do WebScarab em uso.

Ele é um projeto mantido pela OWASP e seu uso é indicado para desenvolvedores que querem depurar problemas ou especialistas de segurança que buscam identificarem falhas em softwares (OWASP, 2012).

4.3 SQLmap

É uma ferramenta de código aberto executada em linha de comando, usada em testes de intrusão para automatizar a exploração de falhas *sql injection*. Esta ferramenta foi desenvolvida com o uso da linguagem Python e pode ser utilizada tanto em Linux como Windows, sendo preciso apenas um interpretador para Python. Ela possui suporte aos bancos de dados: Oracle, Sql Server e MySql, SQLite e outros (DAMELE; STAMPAR, 2011). Para seu uso é necessário identificar manualmente uma URL vulnerável e a partir de então com o uso da ferramenta, começar a exploração (DAMELE; STAMPAR, 2011).

No exemplo foi informada uma URL juntamente com o parâmetro –b que retornou o *banner* do banco de dados com as informações de nome e a versão (figura 4.5).

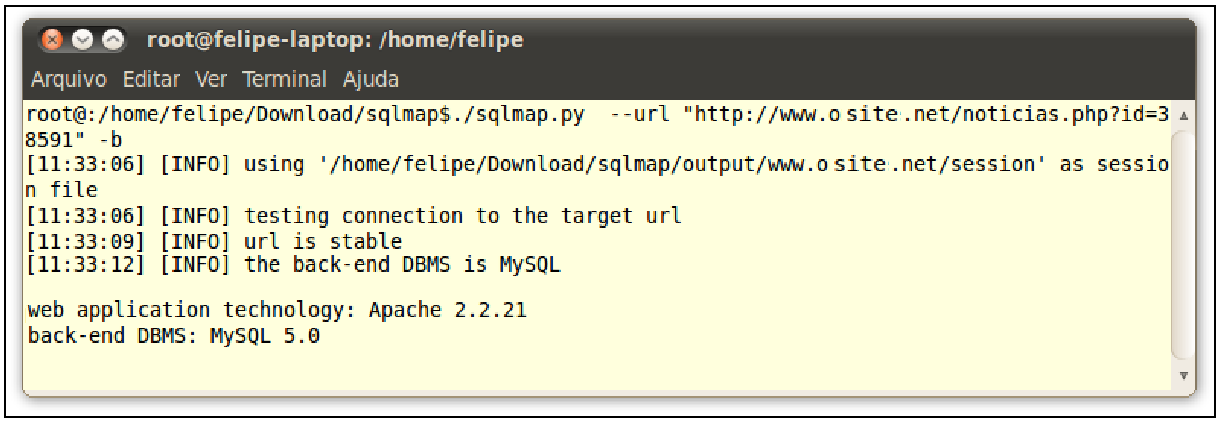

Figura 4.5 – Exemplo 1 de uso do SQLmap.

Logo em seguida foi passado outro parâmetro para a obtenção do nome da base dados e a resposta do software foi ‘site’ (figura 4.6).

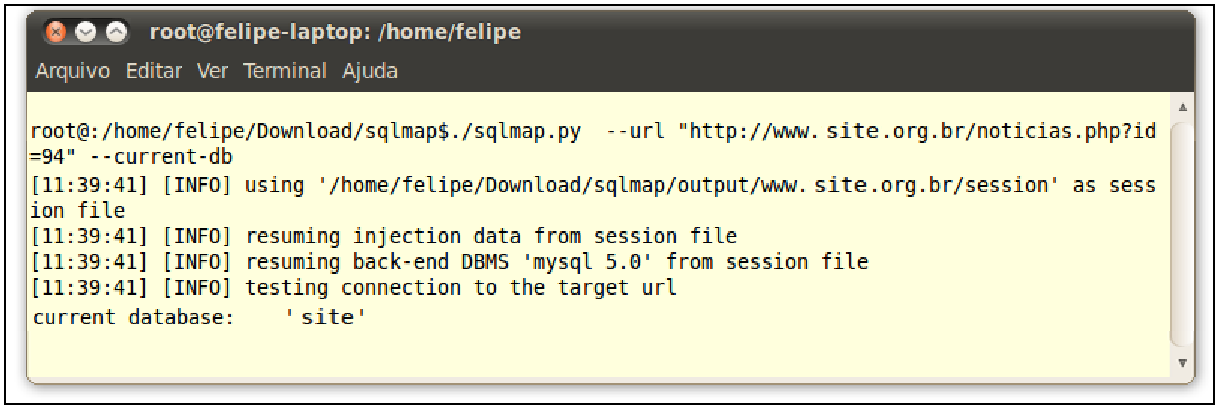

Figura 4.6 – Exemplo 2 de uso do SQLmap.

Com essas informações em mãos é possível obter diversos dados importantes sobre a aplicação alvo. Na figura 4.7 observa-se que foi capturada a lista de todas as tabelas e armazenadas em arquivo, todas as informações exibidas no terminal, ficam em *log*.

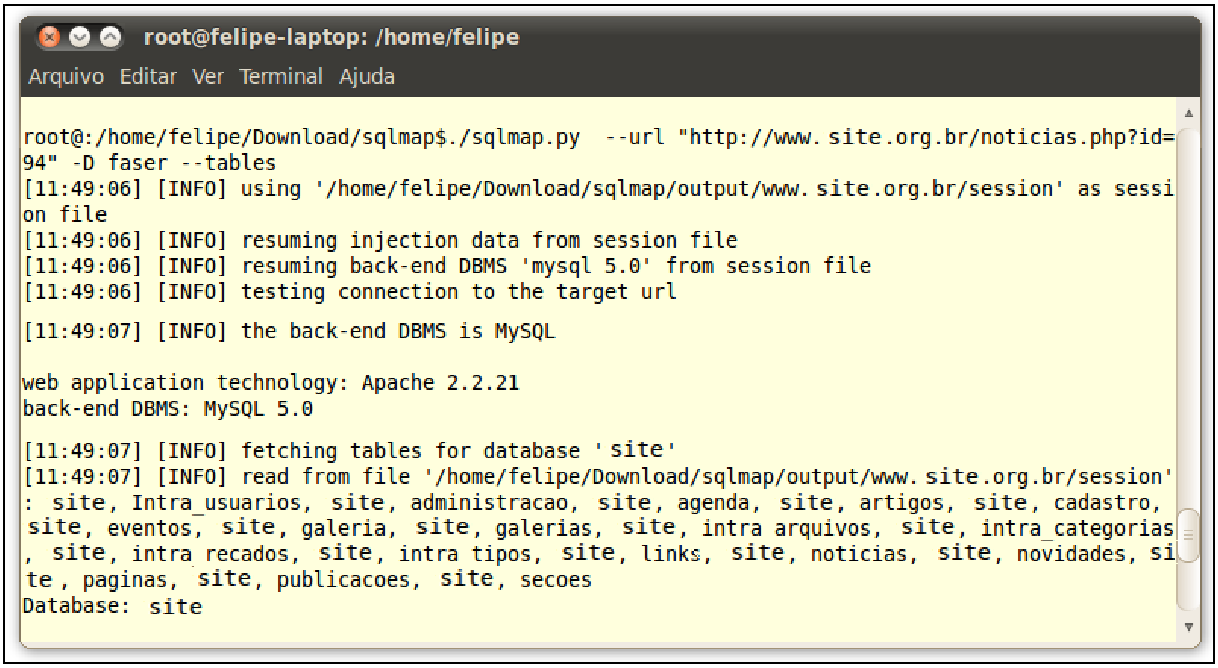

Figura 4.7 - Exemplo 3 de uso do SQLmap.

Além dessas opções, também é possível, copiar as informações, ver os privilégios do usuário do sistema, abrir um terminal com *shell* interativo entre outras que se encontram no manual da ferramenta no link http://sqlmap.sourceforge.net/doc/README.pdf (DAMELE; STAMPAR, 2011).

4.4 Add N Edit Cookies

É um complemento para o navegador Mozilla Firefox, criado pelo desenvolvedor de apelido Goodwill. A ferramenta possibilita editar os *cookies* de sites através da tela da figura 4.8. Com isso, é possível obter acesso não autorizado se a aplicação web não tiver um tratamento adequado na área de *logout* (GOODWILL, 2008).

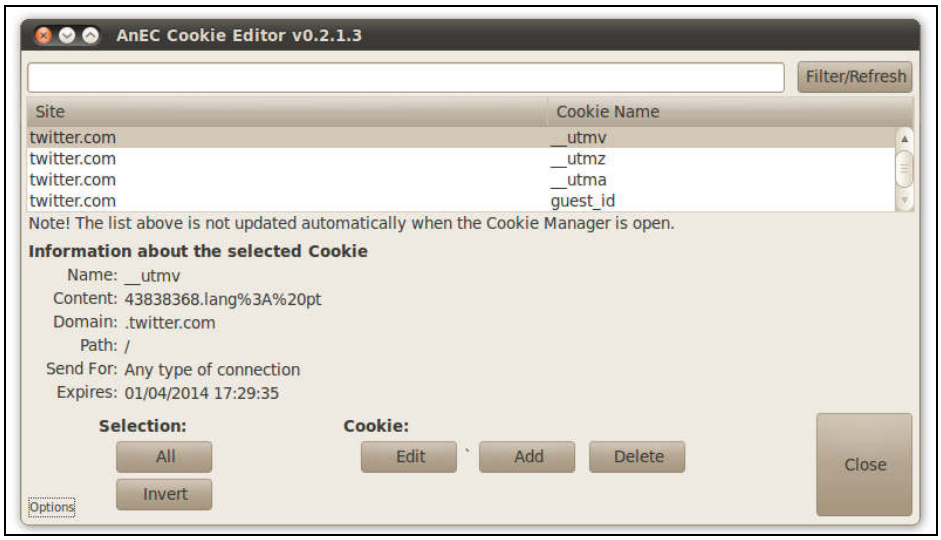

Figura 4.8 – Tela do Add N Edit Cookies.

4.5 OWASP Live CD

É uma distribuição Linux focada em testes de segurança para aplicativos web (figura 4.9). A distribuição possuí interface gráfica *K Desktop Environment* (KDE), o que torna seu uso fácil. Nela está reunida a documentação disponibilizada pela instituição e as ferramentas de que são sugeridas e utilizadas no guia de testes. Além da documentação de testes ela traz consigo livros produzidos pela instituição que abordam a utilização de seus projetos como o WebGoat e outros (OWASP, 2008).

Figura 4.9 – Tela do OWASP Live CD.

Este capítulo foi utilizado para indicar algumas ferramentas para ajudar na execução de testes, enquanto, o próximo será para o detalhamento da metodologia do trabalho, no qual

serão apresentados dados relevantes sobre a arquitetura computacional e os ambientes a serem testados.

5 METODOLOGIA E VALIDAÇÃO

Conforme Prodanov (2009), esse trabalho se enquadra em sua natureza como uma pesquisa aplicada, pois tem como objetivo gerar conhecimento para uma aplicação prática na resolução de problemas. Já quanto os objetivos, são do tipo experimental, pois tem a finalidade de obter mais informações sobre o tema. Por fim, os procedimentos técnicos aplicados neste estudo para a obtenção dos dados, foram categorizados como experimental, já que as variáveis do objeto de estudo são manipuladas, e os efeitos que podem ser produzidos são observados em um ambiente controlado e com o uso dos instrumentos adequados.

Nesse sentido, iniciou-se uma busca por informações as quais pudessem ser importantes nos testes propriamente ditos. Para isso, mapearam-se as aplicações, identificando possíveis pontos de entrada, métodos de envio de formulários utilizados e campos do tipo *hidden*. Identificaram-se também os serviços que estavam sendo executados no servidor e suas respectivas versões. Na etapa seguinte testou-se o gerenciamento de configuração, no qual foram efetuados testes sobre os métodos HTTP, extensões de arquivos aceitas e averiguou-se a localização das interfaces administrativas. A próxima fase abrangeu testes de autenticação, no qual checaram-se a possibilidade de enumeração de usuários, força bruta, timeout, fatores de autenticação entre outras coisas.

Após os testes da categoria de autenticação para continuar a lógica da aplicação verificou-se o gerenciamento de sessão, onde aplicaram-se testes para CSRF, atributos dos *cookies*, fixação de sessão , etc. O passo seguinte foi sobre autorização, no qual foram tratados os seguintes pontos: ultrapassar esquema de autorização, escalada de privilégio e *path* transversal. Com o decorrer das etapas anteriores chegou o momento de testar a validação dos dados, na qual são verificados diversos tipos de injeção de código, entre elas estão xss, SQL *injection*, *buffer overflow* e outros.

Prosseguindo a investigação alcançou-se a categoria negação de serviço, onde foram aplicadas técnicas para bloqueio de contas, *buffer overflow*, alocação de objetos, armazenamento de dados em sessão, etc. No final foram executados testes para a tecnologia AJAX e averiguou-se a lógica do negócio, se era possível ultrapassar alguma etapa do processo de compra.

Sendo assim, como instrumento para validação dos dados obtidos, utilizou-se uma tabela sugerida pelo guia de testes da OWASP que é indicado para relatórios técnicos e pode ser observada na tabela 5.1.

Tabela 5.1 – Tabela para validação dos testes.

Categoria	Número de ref.	Nome do teste	Encontrado	Solução	Risco
Obtenção de informação	OWASP-IG-001	*Spiders, Robots and Crawlers*			
	OWASP-IG-002	*Search Engine discovery/Reconnaissance*			
	OWASP-IG-003	*Identify application entry points*			
	OWASP-IG-004	*Testing for Web Application Fingerprint*			
	OWASP-IG-005	*Application Discovery*			
	OWASP-IG-006	*Analysis of Error Codes*			
Teste de gestão de configuração	OWASP-CM-001	*SSL/TLS Testing*			
	OWASP-CM-002	*DB Listener Testing*			
	OWASP-CM-003	*Infrastructure Configuration Management Testing*			
	OWASP-CM-004	*Application Configuration Management Testing*			
	OWASP-CM-005	*Testing for File Extensions Handling*			
	OWASP-CM-006	*Old, backup and unreferenced files*			
	OWASP-CM-007	*Infrastructure and Application Admin Interfaces*			
	OWASP-CM-008	*Testing for HTTP Methods and XST*			
Teste de autenticação	OWASP-AT-001	*Credentials transport over an encrypted channel*			
	OWASP-AT-002	*Testing for user Enumeration*			
	OWASP-AT-003	*Testing for Guessable (Dictionary) User Account*			
	OWASP-AT-004	*Brute Force Testing*			
	OWASP-AT-005	*Testing for bypassing authentication schema*			
	OWASP-AT-006	*Testing for vulnerable remember password and pwd reset*			
	OWASP-AT-007	*Testing for Logout and Browser Cache Management*			

Categoria	Número de ref.	Nome do teste	Encontrado	Solução	Risco
	OWASP-AT-008	*Testing for CAPTCHA*			
	OWASP-AT-009	*Testing Multiple Factors Authentication*			
	OWASP-AT-0010	*Testing for Race Conditions*			
Gerenciamento de sessão	OWASP-SM-001	*Testing for Session Management Schema*			
	OWASP-SM-002	*Testing for Cookies Attributes*			
	OWASP-SM-003	*Testing for Session Fixation*			
	OWASP-SM-004	*Testing for Exposed Session Variables*			
	OWASP-SM-005	*Testing for CSRF*			
Teste de autorização	OWASP-AZ-001	*Testing for Path Traversal*			
	OWASP-AZ-002	*Testing for bypassing authorization schema*			
	OWASP-AZ-003	*Testing for Privilege Escalation*			
Teste de lógica do negócio	OWASP-BL-001	*Testing for business logic*			
Teste de validação de dados	OWASP-DV-001	*Testing for Reflected Cross Site Scripting*			
	OWASP-DV-002	*Testing for Stored Cross Site Scripting*			
	OWASP-DV-003	*Testing for DOM based Cross Site Scripting*			
	OWASP-DV-004	*Testing for Cross Site Flashing*			
	OWASP-DV-005	*SQL Injection*			
	OWASP-DV-006	*LDAP Injection*			
	OWASP-DV-007	*ORM Injection*			
	OWASP-DV-008	*XML Injection*			
	OWASP-DV-009	*SSI Injection*			
	OWASP-DV-010	*XPath Injection*			
	OWASP-DV-011	*IMAP/SMTP Injection*			

Categoria	Número de ref.	Nome do teste	Encontrado	Solução	Risco
	OWASP-DV-012	*Code Injection*			
	OWASP-DV-013	*OS Commanding*			
	OWASP-DV-014	*Buffer overflow*			
	OWASP-DV-015	*Incubated vulnerability Testing*			
	OWASP-DV-016	*Testing for HTTP Splitting / Smuggling*			
Teste de negação de serviço	OWASP-DS-001	*Testing for SQL Wildcard Attacks*			
	OWASP-DS-002	*Locking Customer Accounts*			
	OWASP-DS-003	*Testing for DoS Buffer Overflows*			
	OWASP-DS-004	*User Specified Object Allocation*			
	OWASP-DS-005	*User Input as a Loop Counter*			
	OWASP-DS-006	*Writing User Provided Data to Disk*			
	OWASP-DS-007	*Failure to Release Resources*			
	OWASP-DS-008	*Storing too Much Data in Session*			
Teste Web Services	OWASP-WS-001	*WS Information Gathering*			
	OWASP-WS-002	*Testing WSDL*			
	OWASP-WS-003	*XML Structural Testing*			
	OWASP-WS-004	*XML content-level Testing*			
	OWASP-WS-005	*HTTP GET parameters/REST Testing*			
	OWASP-WS-006	*Naughty SOAP attachments*			
	OWASP-WS-007	*Replay Testing*			
Teste Ájax	OWASP-AJ-001	*AJAX Vulnerabilities*			
	OWASP-WS-002	*AJAX Testing*			

Fonte: OWASP Testing Guide, 2008.

Com uso das ferramentas apresentadas no capítulo anterior, para a coleta de informações na realização de testes e a tabela 5.1 para a validação deles, se tem como objetivo

encontrar algumas vulnerabilidades de risco baixo e médio, além de possivelmente alguma de risco alto. Mas, esta última não se tem muita expectativa devido às plataformas de comércio eletrônico que serão testadas são populares, possuem diversos desenvolvedores e ainda é utilizada por muitas lojas virtuais.

5.1 Arquitetura computacional

Para validar a pesquisa se teve a necessidade da apresentação da arquitetura computacional, na qual foram instalados os portais de e-commerce, devido à possível existência de problemas de segurança já conhecidos em alguns sistemas operacionais e softwares. Nesse sentido, foi utilizado um serviço de hospedagem como ambiente computacional para tornar o experimento próximo de um ambiente real de testes. O servidor usado utiliza sistema operacional Linux distribuição CentOs 6x e versão de kernel 2.6.18-308.11.1.el5. Ainda, possui 12 gigabytes (Gb) de memória e processador 8 Intel(R) Xeon(R) CPU E5620 @ 2.40GHz. Além dessas informações, ele executa os seguintes serviços:

- Apache versão 2.2.22;
- PHP versão 5.3.14;
- MySQL versão 5.1.63-cl;
- Exim smtpd 4.77;
- OpenSSH 4.3.

Foram identificadas, também, outras portas abertas, que são associadas a alguns serviços:

- FTP – porta 21.
- POP3 – porta 110.
- IMAP – porta 143.

5.2 Ambientes a serem testados

Os ambientes a serem testados, são *open-source* como já citados anteriormente, e são os seguintes:

- Magento: mantido pela empresa Varien, é considerado uma das plataformas de comércio eletrônico mais completa, flexível e personalizável, adaptando-se conforme

as necessidades do usuário e do negócio. A versão utilizada nos testes foi a Magento community edition 1.7.0.2, a qual tem como requisitos para sua instalação: PHP 5.2.0 ou mais recente, MySQL 4.1.20 ou mais recente. Outros detalhes que podem ser encontrados em http://www.magentocommerce.com/system-requirements (MCCOMBS; BANH, 2010);

Figura 5.1 – Logomarca do Magento.

- OsCommerce: iniciado em 2000 e distribuído sob licença *General Public license*, possui mais de cinco mil complementos disponíveis para atender todo o tipo de necessidade do proprietário. OsCommerce versão 2.3.2, foi desenvolvido com a linguagem PHP e utiliza como servidor de banco de dados o MySQL, dessa forma, necessita o PHP 5.0 ou superior e MySQL 5.0 ou superior para que funcione de modo adequado. Demais informações sobre o ambiente são encontrados na documentação http://www.oscommerce.info/confluence/dashboard.action (OSCOMMERCE, 2012);

Figura 5.2 – Logomarca do OsCommerce.

- Prestashop: disponível em 41 idiomas e utilizado por mais de 100 mil lojas, o PrestShop 1.4.8.2 é um software livre que possui 275 funcionalidades e pode ser configurado de acordo com as necessidades da empresa. Sua customização é desde o layout até o método de envio dos produtos. Seus requisitos para instalação são muito parecidos com os anteriores, sendo necessários os seguintes softwares: Apache 1.3, PHP 5.1 ou superior, MySQL 5.0 ou superior. Algumas configurações opcionais podem ser encontradas em http://www.prestashop.com/pt/requisitos-sistema (PRESTASHOP, 2012).

Figura 5.3 – Logomarca do PrestaShop.

Então, neste capítulo foram apresentados os ambientes que serão objeto de estudo neste trabalho, como também informações sobre a arquitetura computacional. No capítulo seguinte será analisado, validado os resultados de cada ambiente e sugerido algumas melhorias.

6 VALIDAÇÃO

Tão importante quanto a descoberta de vulnerabilidades é estimar a gravidade delas para que possam ser priorizadas as correções e, assim, diminuir o risco ao negócio. Como uma vulnerabilidade pode ter seu risco ao negócio avaliado diferentemente entre empresas, será apresentado um modelo simples da metodologia OWASP que posteriormente pode ser customizado de acordo com as necessidades (2008).

Sendo assim, o modelo parte de uma equação, como se pode visualizar na figura 6.1, que utiliza duas variáveis como ponto de partida. Inicialmente é avaliada a probabilidade de acontecer a exploração e o impacto na segurança da aplicação. Com a combinação destes dados se obtém o risco global para que se possam tomar medidas corretivas.

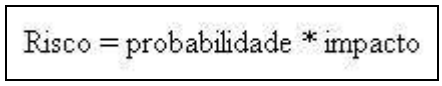

Figura 6.1 – Equação do risco.
Fonte: Adaptado de OWASP Testing Guide, 2008.

O primeiro passo é a identificação de uma falha de segurança. Com tal identificação é preciso reunir informações sobre ela, como: o impacto da exploração sobre o negócio e o tipo de possíveis atacantes.

6.1 Probabilidade

Com as informações reunidas sobre a falha, elas são utilizadas para determinar a probabilidade de uma provável descoberta e exploração por algum atacante. Essa identificação é simples e não necessita de muitos detalhes, sendo suficiente a determinação se a probabilidade é baixa, média ou alta. Para realizar tal avaliação há alguns fatores a serem considerados, sendo o primeiro conjunto relacionado ao agente da ameaça. Essa etapa tem como finalidade determinar a possibilidade de êxito de um grupo de atacantes. Este processo é feito com um questionário, onde cada pergunta possui diversas opções e cada uma delas tem um índice de probabilidade de 0 a 9 associado. Para responder as perguntas deve-se levar em consideração o agente da ameaça do pior caso (OWASP, 2008).

Como é qualificado tecnicamente o grupo de atacantes?

- Sem habilidades técnicas (1);

- Algumas habilidades técnicas (3);
- Usuário de computador avançado (4);
- Habilidades de rede e programação (6);
- Habilidades de segurança e *pentest* (9).

Como é motivado este grupo de atacantes a encontrar e explorar essa vulnerabilidade?

- Recompensa baixa ou nenhuma (1);
- Possível recompensa (4);
- Recompensa alta (9).

Quantas oportunidades esse grupo de atacantes tem para encontrar e explorar essa vulnerabilidade?

- Sem acesso conhecido (0);
- Acesso limitado (4);
- Acesso completo (9).

Qual o tamanho deste grupo de atacantes?

- Desenvolvedores (2);
- Administradores de sistemas (2);
- Usuários da intranet (4);
- Parceiros (5);
- Usuários autenticados (6);
- Usuários de internet anônimos (9).

O outro conjunto de fatores a levar em consideração é a probabilidade de a vulnerabilidade ser encontrada e explorada pelo agente da ameaça avaliado anteriormente.

Como é a facilidade desse grupo de atacantes descobrirem esta vulnerabilidade?

- Praticamente impossível (1);
- Difícil (3);
- Facilmente (7);
- Ferramentas automatizadas disponíveis (9).

Como é a facilidade desse grupo de atacantes explorar essa vulnerabilidade?

- Teórica (1);
- Difícil (3);
- Fácil (5);
- Ferramentas automatizadas disponíveis (9).

Como é o conhecimento sobre esta vulnerabilidade por este grupo de atacantes?

- Desconhecido (1);
- Oculto (4);
- Óbvio (6);
- Conhecimento público (9).

Qual a probabilidade de uma exploração ser detectada?

- Detecção ativa na aplicação (1);
- Registrado e analisado (3);
- Somente registrado (8);
- Não registrado (9).

6.2 Impacto

Após a identificação do nível de probabilidade a etapa seguinte é avaliar o impacto de um ataque bem sucedido. Nesse momento deve ser avaliado o impacto técnico, que é o qual atua sobre aplicação, dados e funções utilizados por ela. Já o impacto no negócio é sobre como será afetada a empresa e as suas operações. Por isso devem ser respondidos os questionários a seguir que possui igualmente, como os anteriores, um conjunto de opções e um índice de impacto associado. Primeiramente segue o questionário sobre o impacto técnico (OWASP, 2008).

Quantos dados podem ser divulgados e qual a sensibilidade deles?

- O mínimo de dados não confidenciais divulgados (2);
- O mínimo de dados críticos divulgados (6);
- Extensos dados não sensíveis divulgados (6);
- Extensos dados críticos divulgados, todos os dados divulgados (9).

Qual a quantidade de dados pode ser corrompida e danificada?

- O mínimo de dados ligeiramente corrompidos (1);
- O mínimo de dados seriamente corrompidos (3);
- Extensos dados e pouco corrompidos (5);
- Extensos dados e seriamente corrompidos (7);
- Todos os dados totalmente corrompidos (9).

Quanto à disponibilidade, qual a quantidade de serviços perdidos e quão vital isso é?

- O mínimo de serviços secundários interrompidos (1);
- O mínimo de serviços primários interrompidos (5);
- Extensos serviços secundários interrompidos (5);
- Extensos serviços primários interrompidos (7);

- Todos os serviços perdidos (9).

As ações dos invasores são rastreáveis?

- Totalmente rastreável (1);
- Possivelmente rastreável (7);
- Completamente anônimo (9).

O impacto do negócio decorre do impacto técnico, mas possibilita maior compreensão do que a empresa precisa para utilizar a aplicação. Sendo assim, o questionário que contém perguntas sobre o impacto no negócio tem o objetivo de fornecer dados para os executivos e diretores tomarem decisões de investimento na resolução dos problemas (OWASP, 2008).

Quanto de dano financeiro resultará uma exploração?

- Menos do que o custo para corrigir a vulnerabilidade (1);
- Pequeno efeito no lucro anual (3);
- Efeito significativo no lucro anual (7);
- Falência (9).

Quanto de danos à reputação e prejudicaria o negócio uma exploração?

- Danos mínimos (1);
- Perda de grandes contas (4);
- Perda de clientela (5);
- Danos à marca (9).

Quanto de não cumprimento de serviços aos clientes pode gerar uma exploração?

- Violação pequena (2);

- Violação clara (5);
- Violação grande (7);

Quanta informação pessoal e identificável pode ser divulgada?

- Uma pessoa (3);
- Centenas de pessoas (5);
- Milhares de pessoas (7);
- Milhões de pessoas (9).

6.3 Gravidade do risco

Depois de avaliar a probabilidade e o impacto, as duas variáveis da equação da figura 6.1, será calculada a gravidade global da vulnerabilidade. Para isso, com os índices das respostas dos questionários será estabelecido o nível de risco da probabilidade e o impacto (OWASP, 2008). A primeira etapa é a seleção dos índices das respostas dos questionários sobre probabilidade e inseri-los em uma tabela como exemplifica a figura 6.2 que calcula a sua média. Com o resultado da média e o auxílio da tabela 6.1 que contém a escala, será estabelecido o nível da probabilidade global.

Tabela 6.1 – Níveis de probabilidade e impacto.

Níveis de probabilidade e impacto	
0 até < 3	Alto
3 até < 6	Médio
6 até 9	Baixo

Fonte: Adaptado de OWASP Testing Guide, 2008.

Fatores do agente				Fatores da vulnerabilidade			
Nível de habilidade	Motivo	Oportunidade	Tamanho	Facilidade de descoberta	Facilidade de explorar	Conhecimento	Detecção de intrusão
5	2	7	1	3	6	9	2
Probabilidade global = 4.375 Médio							

Figura 6.2 – Cálculo da probabilidade global.
Fonte: Adaptado de OWASP Testing Guide, 2008.

O passo seguinte é verificar o impacto, assim, serão distribuídas as respostas em uma tabela de modo parecido com o anterior, como exemplifica a figura 6.3. Mas a escala utilizada é diferente da anterior. Nesse caso, se a média for menor que 3 é baixo, de 3 até 6 é médio e de 6 até 9 é alto.

Impacto técnico				Impacto no negócio			
Perda de confidencialidade	Perda de integridade	Perda de disponibilidade	Rastreabilidade	Danos financeiros	Danos a reputação	Descumprimento	Violação de privacidade
9	7	5	8	1	2	1	5
Impacto técnico global = 7.25 Alto				Impacto no negócio global = 2.25 Baixo			

Figura 6.3 – Cálculo do impacto.
Fonte: Adaptado de OWASP Testing Guide, 2008.

Com as respostas da probabilidade e o impacto, será determinada a gravidade final da falha. Dessa forma, a avaliação será realizada por meio da classificação dessas variáveis. No entanto, deve se levar em consideração o uso de apenas um tipo de impacto, para isso a sua escolha deverá ser comparada com qual das duas se tem mais informações. Após esta escolha e a ajuda da figura 6.4 é executada uma combinação do impacto e a probabilidade para se chegar a uma classificação final.

Gravidade do risco final				
Impacto	Alto	Médio	Alto	Crítico
	Médio	Baixo	Médio	Alto
	Baixo	Observação	Baixo	Médio
	x	Baixo	Médio	Alto
	Probabilidade			

Figura 6.4 – Cálculo do impacto.
Fonte: Adaptado de OWASP Testing Guide, 2008.

6.4 Tabelas de resultados

Foi realizado um levantamento minucioso sobre os resultados dos testes após a execução de todos contidos na documentação da metodologia. Considerou-se todo o processo citado no *Testing Guide* e as ferramentas apresentadas anteriormente. Os resultados dos testes foram conforme a expectativa, já que identificaram-se vulnerabilidades de três categorias diferentes, que serão apresentadas em tabelas separadas por cada ambiente de comércio eletrônico e após uma tabela com os resultados gerais.

6.4.1 Resultados Magento

Depois da realização de todos os testes no ambiente Magento foi obtida apenas uma falha como pode ser observado na tabela 6.2.

Tabela 6.2 – Resultados do Magento.

Categoria	Número de ref.	Nome do teste	Encontrado	Solução	Risco
Teste de autenticação	OWASP-AT-004	*Brute Force Testing*	Sem limite de tentativas de login na interface de clientes.	Estabelecer um limite de tentativas de login e habilitar re*CAPTCHA* após 3 tentativas.	Baixo

O resultado do Magento foi o melhor entre os três. O problema de segurança encontrado é igualmente presente em diversas lojas virtuais, mas, seu risco não preocupa devido a ser baixo e relacionado com a interface do cliente, sendo assim, não muito visado por criminosos, no entanto na sessão de análise será proposta uma melhoria para resolvê-la.

6.4.2 Resultados OsCommerce

Com a realização dos mesmos testes executados nas três plataformas, a OsCommerce apresentou alguns problemas de segurança apresentados na tabela 6.3.

Tabela 6.3 – Resultados do OsCommerce.

Categoria	Número de ref.	Nome do teste	Encontrado	Solução	Risco
Teste de autenticação	OWASP-AT-004	Brute Force Testing	Sem limite de tentativas de login na interface de clientes.	Estabelecer um limite de tentativas de login e habilitar re*CAPTCHA* após 3 tentativas.	Baixo
Gerenciamento de sessão	OWASP-SM-003	Testing for Session Fixation	Não é modificado cookie após a autenticação na interface administrativa	Implementar a modificação de *cookie* após a autenticação	Baixo

	OWASP-SM-005	Testing for CSRF	Na interface administrativa é possível fazer requisições falsas com um cookie roubado	Utilizar campos do tipo *hidden* com um *token* gerado de forma aleatória em formulários e armazenar na sessão do usuário, ao ser enviado é comparado os *tokens* e caso diferentes não executa a requisição.	Médio
Teste de negação de serviço	OWASP-DS-002	Locking Customer Accounts	Na interface administrativa após 3 tentativas de login.	Estabelecer um limite de tentativas de login e habilitar re*CAPTCHA* após 3 tentativas.	Baixo

A plataforma OsCommerce apresentou a vulnerabilidade CSRF de risco médio já encontrado em serviços conhecidos como Netflix, Twitter e outros como citado no capítulo de técnicas. Igualmente como os demais softwares encontrou-se também a presença da falha de força bruta.

6.4.3 Resultados PrestaShop

Igualmente com as duas plataformas de e-commerce que foram apresentados os resultados anteriormente, o Prestashop passou pelo mesmo processo e constatou-se maior vulnerabilidades como se percebe na tabela 6.4.

Tabela 6.4 – Resultados do PrestaShop.

Categoria	Número de ref.	Nome do teste	Encontrado	Solução	Risco
Teste de autenticação	OWASP-AT-004	Brute Force Testing	Sem limite de tentativas de login na interface de clientes.	Estabelecer um limite de tentativas de login e habilitar re*CAPTCHA* após 3 tentativas.	Baixo

	OWASP-AT-004	Brute Force Testing	Sem limite de tentativas de login na interface administrativa.	Estabelecer um limite de tentativas de login e habilitar re*CAPTCHA* após 3 tentativas..	Baixo
	OWASP-AT-007	Testing for Logout and Browser Cache Management	Sem limite de timeout na interface de clientes	Estabelecer limite de *timeout*.	Médio
	OWASP-AT-007	Testing for Logout and Browser Cache Management	Sem limite de timeout na interface administrativa	Estabelecer limite de *timeout*	Médio
Gerenciamento de sessão	OWASP-SM-002	Testing for Cookies attributes	Na interface de clientes cookies válidos por 20 dias.	Diminuir a duração dos *cookies* para 1 dia.	Baixo
	OWASP-SM-002	Testing for Cookies attributes	Na interface administrativa cookies válidos por 20 dias.	Diminuir a duração dos *cookies* para 1 dia.	Baixo
	OWASP-SM-005	Testing for CSRF	Na interface de clientes é possível fazer requisições falsas mesmo após o logout do usuário do qual o cookie foi roubado.	Utilizar campos do tipo *hidden* com um *token* gerado de forma aleatória em formulários e armazenar na sessão do usuário, ao ser enviado é comparado os *tokens* e caso diferentes não executa a requisição.	Médio

	OWASP-SM-005	Testing for CSRF	Na interface administrativa é possível falsificar requisições mesmo após o logout do usuário do qual o cookie foi roubado.	Utilizar campos do tipo *hidden* com um *token* gerado de forma aleatória em formulários e armazenar na sessão do usuário, ao ser enviado é comparado os *tokens* e caso diferentes não executa a requisição.	Médio

Comparando os resultados do PrestaShop com os demais, este apresentou nível de segurança mais baixo devido as oito vulnerabilidades terem sido descobertas nesse ambiente. A preocupação maior se deve a metade desses problemas de segurança ser de nível médio, necessitando assim atenção especial para essas informações para que algumas ações sejam tomadas para solucionar.

6.5 Tabela geral

Os dados gerais foram distribuídos na tabela 6.5 para que seja possível a visualização mais prática das falhas contidas nos ambientes testados.

Tabela 6.5 – Resultados dos gerais.

Nome do teste	PrestaShop	OsCommerce	Magento	Total (%)	Total
Brute force	2	1	1	30,8	4
Session fixation	0	1	0	7,70	1
CSRF	2	1	0	23,1	3
Locking account	0	1	0	7,70	1
Cookie attribute	2	0	0	15,4	2
Logout and browser cache management	2	0	0	15,4	2
Total (%)	61,6	30,8	7,70	100	
Total	8	4	1		13

Observa-se na tabela 6.5 duas informações que se destacam das demais. Sendo que na primeira mais da metade das vulnerabilidades foram encontradas no PrestaShop, enquanto o

outro dado que se verifica é o percentual de falhas encontradas com o teste de força bruta sendo de aproximadamente 30%.

6.6 Análise

Para dar início ao *pentest* foi feita uma avaliação e o recolhimento de informações sobre os aplicativos a serem testados. Essa primeira etapa foi realizada para verificar possíveis pontos de entrada nas plataformas e áreas suscetíveis a falhas, além da identificação de serviços que são executados pelo servidor. Com o conhecimento desses dados iniciais deu-se o início dos testes propriamente dito. Tais testes abordaram gerenciamento de configuração, autenticação, gerenciamento de sessão, validação de dados entre outros.

As figuras 6.5 e 6.6 representam o número de vulnerabilidades encontradas nas três aplicações de e-commerce, na qual é possível verificar maior quantidade de defeitos nas categorias autenticação e gerenciamento de sessão, igualmente com 6 vulnerabilidades cada uma, totalizando 46,15% do total cada uma dessas categorias. Conclui-se, então, esses serem pontos frágeis e mais suscetíveis a explorações.

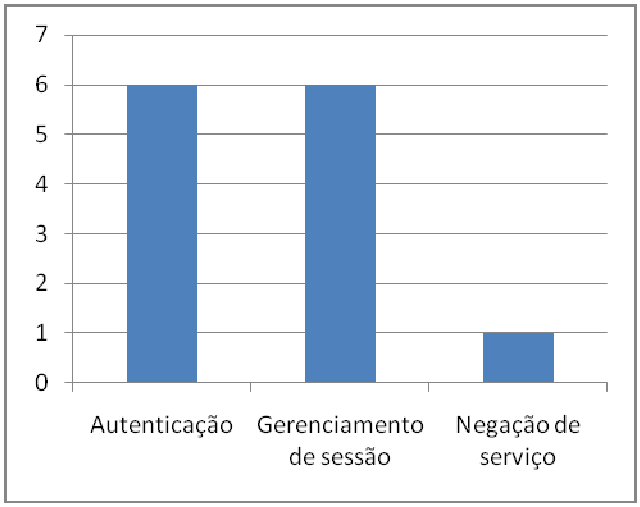

Figura 6.5 – Resultado por categorias.

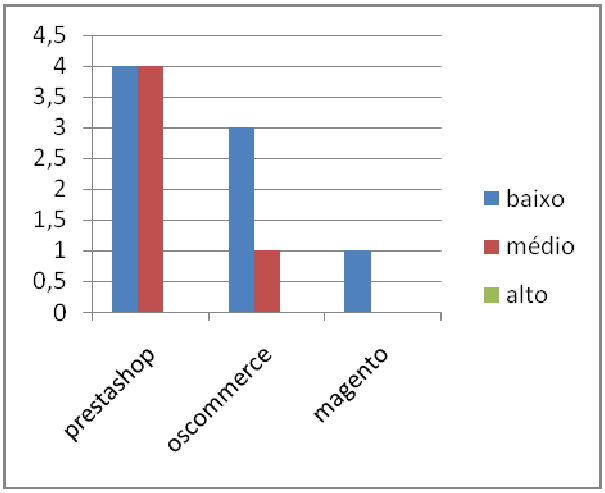

Figura 6.6 – Resultado por ambiente.

Outro ponto importante a se observar é a quantidade de falhas encontradas em cada plataforma que podem ser visualizadas nos gráficos de resultados imagens 6.7 e 6.8. O ambiente Prestashop foi o qual encontrou-se maior número de defeitos, um total de 8 ou 61,54%, sendo quatro desses de risco médio.

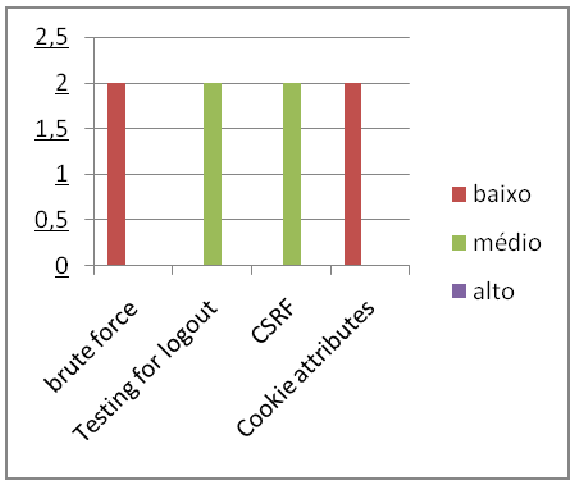

Figura 6.7 – Resultado Prestashop.

Em segundo, no software OsCommerce, foram encontradas 4 vulnerabilidades, com apenas uma de nível médio. Por fim, deve-se ressaltar o ambiente Magento, com apenas uma falha, esta de força bruta e considerada de risco baixo. Por isto, considerado o mais seguro entre os três ambientes nos testes realizados neste estudo.

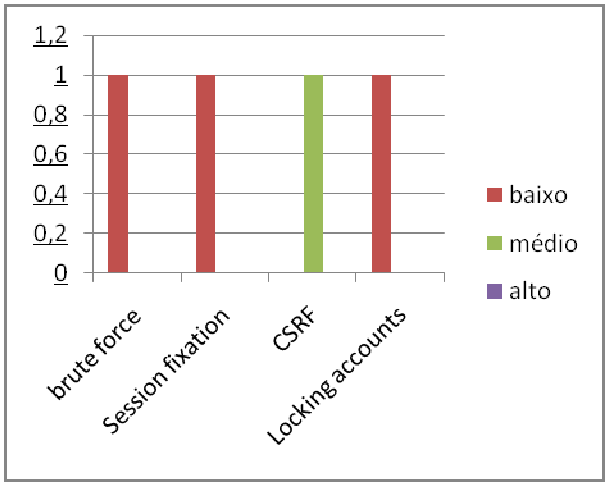

Figura 6.8 – Resultado OsCommerce.

6.7 Sugestões de melhoria

Para o combate de tais falhas, foram propostas algumas soluções. Primeiramente, quanto à vulnerabilidade de força bruta contida nas três plataformas testadas, a sugestão é de dispor uma interface na qual é possível configurar e limitar as tentativas de login e também um módulo re*CAPTCHA* para ser habilitado após atingir tal limite. A escolha dessa tecnologia dá-se devido sua maior segurança já que o *captcha* pode ser mais facilmente quebrado como mostrado por Yan e El (2008, APUD MUNIZ, 2011). O objetivo dessa sugestão é permitir que os usuários continuem tentando efetuar o login, mas ao mesmo tempo impedir que ferramentas automatizadas realizem a tarefa de descobrir um usuário e ou senha existentes. Outro motivo para tal escolha é devido a algumas aplicações apenas bloquearem a conta por determinado tempo, no entanto essa alternativa não é a ideal para um cliente nem para um usuário administrativo. O cliente que realizará uma compra muitas vezes quer agilidade e facilidade, e com esse recurso iria dificultar a aquisição. Já para o administrador o bloqueio não é o mais indicado, pois poderia causar problemas de negação de serviço. Esta mesma solução é sugerida para a vulnerabilidade de *locking accounts* encontrada no ambiente OsCommerce, pois consegue sanar o problema e garantir a segurança dos usuários.

Quanto aos defeitos CSRF encontrados nos ambientes PrestaShop e OsCommerce o mecanismo sugerido para resolver tal problema é de utilizar campos do tipo *hidden* para ser atribuído a ele um *token* gerado de forma aleatória a cada requisição e salvar este *token* na sessão do usuário. Ao ser enviado o formulário, o servidor irá processar a requisição e comparar se o dado do campo é igual ao da sessão, caso positivo é realizada a ação. Esta sugestão dificultaria de forma significativa um ataque deste tipo, já que o invasor teria que

adivinhar o *token* devido ele ser alterado a cada solicitação HTTP. Outra medida para complementar a solução é verificar que o usuário veio da página onde esta o formulário.

A falha de *session fixation* contida no OsCommerce pode favorecer sequestros de sessão quando a aplicação não renova o *cookie* após a autenticação bem sucedida. Dessa forma, é sugerido que se desenvolva um mecanismo para realizar essa renovação.

O problema de *cookie attribute* encontrado no PrestaShop é que ele tem uma duração de 20 dias, um tempo relativamente grande para manter esses dados válidos e permitir que um invasor capture essa informação e acesse a aplicação. Para solucionar essa vulnerabilidade na interface administrativa sugere-se a retirada do atributo *expires*, pois assim o mesmo é válido apenas para a sessão atual e quando o navegador for fechado ele é invalidado. Já para a interface de clientes a sugestão é de reduzir o tempo de duração para 1 dia, porque informações do carrinho de compras são armazenadas nele possibilitando que uma compra seja concluída nesse período.

A vulnerabilidade encontrada com o teste *logout and browser cache management* no ambiente PrestaShop é que a aplicação não tem limite de *timeout*, com o tempo de inatividade do usuário, a solução para isso é a implementação de um limite de 30 minutos. Considera-se que a escolha desse tempo deu-se analisando a segurança e a usabilidade dois fatores importantes para essa decisão como citado no *OWASP Testing Guide* (2008).

CONCLUSÃO

Com o crescimento da utilização de aplicação *web* e dos usuários de internet no Brasil é necessária a existência de maior proteção dos dados nesses sistemas para garantir os princípios de segurança: autenticidade, confiabilidade e disponibilidade.

A realização do trabalho partiu do interesse do autor pela área de segurança da informação, para obter um conhecimento sobre a mesma e alcançar alguns objetivos, tais como: estudar as técnicas de exploração e também a metodologia OWASP, observando os dados estatísticos sobre o setor de comércio eletrônico, pode se notar a preocupação com a segurança nas transações e com a privacidade de seus usuários.

O trabalho buscou com o uso da metodologia OWASP, a avaliação e análise sobre as principais plataformas de comércio eletrônico de código aberto. Essa análise de vulnerabilidades deu-se a partir de vários testes que tinham como alvo a autenticação, o controle de acesso, a validação dos dados de entrada, o gerenciamento da sessão e outros. O projeto, também, proporciona ao leitor, entender o processo de teste de penetração seguindo a metodologia escolhida e aplicada nestes ambientes propostos. Como também sugestão de algumas medidas preventivas ou para a solução dos problemas encontrados.

Com as técnicas apresentadas e com o *pentest*, pode-se garantir aos ambientes disponíveis na *web* maior segurança e suportem aos ataques identificados no decorrer do projeto, não proporcionando assim que um usuário mal intencionado ameace as informações e a privacidade dos clientes.

Além dessas conclusões, fica como sugestão de trabalhos futuros, a possibilidade de realização de testes em ambientes reais, com determinadas cargas de dados e usuários acessando simultaneamente alguns recursos das aplicações apresentadas, com o objetivo de encontrar algumas vulnerabilidades diferentes.

REFERÊNCIAS BIBLIOGRÁFICAS

AGÊNCIA NACIONAL DE TELECOMUNICAÇÕES. **Em fevereiro, telefonia móvel alcança mais de 247 milhões de linhas ativas.** Disponível em:<http://www.anatel.gov.br/Portal/exibirPortalNoticias.do?acao=carregaNoticia&codigo=24913>. Acesso em: 03/04/2012.

AGÊNCIA NACIONAL DE TELECOMUNICAÇÕES. **Brasil fecha fevereiro com 207,5 milhões de acessos móveis.** Disponível em:<http://www.anatel.gov.br/Portal/exibirPortalNoticias.do?acao=carregaNoticia&codigo=22356>. Acesso em: 03/04/2012.

ALBERTIN, Alberto Luiz. **Comércio eletrônico: modelo, aspectos e contribuições de sua aplicação.** São Paulo: Atlas, 2004.

ALBUQUERQUE, Ricardo; RIBEIRO, Bruno. **Segurança no desenvolvimento de software.** Rio de Janeiro: Campus, 2002.

BERTHOLDO, Leandro Márcio; ANDREOLI, Andrey Vedana; TAROUCO, Liane. **Compreendendo Ataques Denial of Services.** Porto Alegre: CERT-RS, 2003.

BORGES, Cristiano Goulart. Estudo comparativo de metodologias de pentests. Canoas: Ulbra, 2011. Disponível em:<http://pt.scribd.com/doc/65629231/Plugin-Artigo-Pentest-Cristiano>. Acesso em: 11/04/2012.

BRAGA, Gabriel Velasco; TAVARES, Isabella Pinheiro. **PhiNo - Phishing Notifier:** Ferramenta para notificar e-mails com códigos maliciosos que utilizam técnicas de Phishing e Spam. Brasília: Universidade de Brasília, 2011. Disponível em:< http://monografias.cic.unb.br/dspace/bitstream/123456789/341/1/monografia.pdf>. Acesso em: 18/06/2012.

CARMONA, Tadeu. **Segredos do Google:** desvende os recursos não revelados do poderoso sistema de busca. São Paulo: Digerati, 2004.

CERT. **Estatísticas do CERT.br:** incidentes. 2011. Disponível em:< http://www.cert.br/stats/incidentes/2011.html>. Acesso em 20/06/2012.

DAMELE, Bernardo; STAMPAR, Miroslav. **Sqlmap user's manual.** 2011. Disponível em:<http://sqlmap.sourceforge.net/doc/README.pdf >. Acesso em: 18/06/2012.

DANHIEUX, P; WOUTER, C. **Penetration Testing:** The Third Party Hacker. 2006. In: JUNIOR, Odilo Schwade. **Roteiro para a realização de testes de penetração em cenários turn-keys.** Itajaí, SC: Universidade do Vale do Itajaí, 2010.

DERAISON, Renaud et al. **Nessus:** network auditing. [S.l.]:Syngress, 2004.

E-BIT EMPRESA. **WebShoppers 25º edição.** São Paulo, [2011 ou 2012]. Disponível em:<http://www.webshoppers.com.br/webshoppers/WebShoppers25.pdf>. Acesso em: 03/04/2012.

FARIAS, Marcelo Bukowski. **Injeção de SQL em aplicações web:** causas e prevenção. Porto Alegre: UFRGS, 2009.

FILHO, Onildo Luciano de Souza Ferraz. **Comunicação NFC (Near Field Comunication) entre dispositivos ativos.** Recife: UFPE, 2010.

GOODWILL. **Complementos para Firefox:** Add N Edit Cookies. 2008. Disponível em: <https://addons.mozilla.org/pt-br/firefox/addon/add-n-edit-cookies/>. Acesso em: 18/06/2012.

GRAVES, Kimberly. **Official Certified Ethical Hacker:** review guide. Indianapolis: Wiley, 2007.

GROSSMAN, Jeremiah et al. **XSS Attacks:** cross site scripting exploiting and defense. Burlington: Syngress, 2007.

GUMERATO, Ronaldo. **SQL injection em aplicações web**. Uberlândia: Uniminas, 2009.

HOGLUND, Greg; MCGRAW, Gary. **Como quebrar códigos: a arte de explorar (e proteger) software.** São Paulo: Pearson Makron Books, 2006.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATISTICA. **PNAD 2009.** Disponível em:<http://www.ibge.gov.br/home/presidencia/noticias/noticia_visualiza.php?id_noticia=1708>. Acesso em: 03/04/2012.

JUNIOR, Armando Gonçalves da Silva. **Cross-Site Scripting:** Uma Análise Prática. Recife: Universidade Federal de Pernambuco, 2009.

JUNIOR, Odilo Schwade. **Roteiro para a realização de testes de penetração em cenários turn-keys.** Itajaí, SC: Universidade do Vale do Itajaí, 2010.

KENNEDY, David; O'GORMAN, Jim; KEARNS, Devon; AHARONI, Mati. **Metasploit:**The penetration tester's guide. San Francisco: no starch press, 2011.

LAUFER, Rafael P. et al. **Negação de Serviço:** Ataques e Contramedidas. Rio de Janeiro: Universidade Federal do Rio de Janeiro, [2005?]. Disponível em:< http://www.gta.ufrj.br/ftp/gta/TechReports/LMVB05a.pdf>. Acesso em: 18/06/2012.

LEAL, Frederico Franklin Albuquerque. **Especificação de requisitos de um modelo de agronegócio.** Montes Claros: Unimontes, 2004. Disponível em:< http://www.ccet.unimontes.br/arquivos/monografias/15.pdf >. Acesso em: 18/06/2012.

LIMA, Marcelo B. **Firewalls:** Uma Introdução à segurança. Curitiba, fev. 2000. In MEDEIROS, Carlos Diego Russo. **Implantação de Medidas e Ferramentas de Segurança da informação.** Joinville: Univille, 2001.

MAGENTO**. [Comparação das versões].** 2012. Disponível em: <http://www.magentocommerce.com/product/compare>. Acesso em: 12/04/2012.

MAGENTO. **[Casos de sucesso].** 2012. Disponível em: <http://www.magentocommerce.com/showcase>. Acesso em: 12/04/2012.

MCCOMBS, Adam; BANH, Robert. **Magento:** guia definitivo. New York: Novatec, 2010.

MEDEIROS, Carlos Diego Russo. Implantação de Medidas e Ferramentas de Segurança da informação. Joinville: Univille, 2001.

MOIP LABS. **E-commerce e suas plataformas open-source.** 2011. Disponível em: <http://labs.moip.com.br/2011/02/10/ecommerce-e-suas-plataformas-open-source-2/>. Acesso em: 12/04/2012.

NFC BRASIL. **Mas o que é nfc?** 2009. Disponível em:<http://www.nfcbrasil.com.br/?p=49>. Acesso em: 11/04/2012.

NOVAES, Antonio Galvão. **Logística e gerenciamento da cadeia de distribuição:** estratégia, operação e avaliação. Rio de Janeiro: Campus, 2007.

NUNES, Rodolfo Modrigais Strauss. **Análise da contribuição do comércio eletrônico Business to Business na gestão de cadeias de suprimentos.** São Paulo:Unip, 2010.

O'CONNEL, Brian. **B2B.COM:** ganhando dinheiro no e-Commerce Business-to-Business. São Paulo: Makron Books, 2002.

OSCOMMERCE. **osCommerce, Open Source Online Shop E-commerce Solutions.** 2012.

Disponível em: <http://www.oscommerce.com/>. Acesso em: 12/04/2012.

OWASP. **OWASP Testing guide**. [S.l.]: 2008.

OWASP. **OWASP Live CD Project.** [2008]. Disponível em:< https://www.owasp.org/index.php/Category:OWASP_Live_CD_Project>. Acesso em: 18/06/2012.

OWASP. **OWASP WebScarab Project.** 2012. Disponível em:< https://www.owasp.org/index.php/Category:OWASP_WebScarab_Project>. Acesso em: 18/06/2012.

OSTERWALDER, Alexander; PIGNEUR, Yves. **Business Model Generation.** New Jersey: Wiley, 2010.

PAGSEGURO. O comércio eletrônico na América Latina: pesquisa comScore. 2010.

Disponível em:<http://blogpagseguro.com.br/2010/12/o-comercio-eletronico-na-america-latina-pesquisa-comscore/>. Acesso em: 10/05/2012.

PRESTASHOP. **[Funcionalidades].** 2012. Disponível em:<http://www.prestashop.com/pt/funcionalidades>. Acesso em: 12/04/2012.

ROMERO, Luiz Carlos; KACUTA, Luiz Yukishigue; OLIVEIRA, Viviane Luciana de. **Segurança da informação: tipos de ataques.** Unicamp, 2003. Disponível em:< http://www.dcc.unicamp.br/~rdahab/cursos/mp202/Welcome_files/trabalhos/Ataques/texto/Tipos_Ataque.pdf>. Acesso em: 20/06/2012.

SCAMBRAY, Joel; LIU, Vincent; SIMA, Caleb. **Hacking Exposed web applications 3:** Web applications security secrets & solutions. Nova York: McGraw Hill, 2011.

SOUZA, Johnny. **Cross-Site Scripting & Cross-Site Request Forgery.** Brasília: Universidade de Brasília, 2009.

STUTTARD, Dafydd; PINTO, Marcus. **The Web Application Hacker's Handbook:** Discovering and exploiting security flaws. Indianapolis: Wiley, 2008.

TENABLE NETWORK SECURITY. **Guia do usuário.** Columbia, 2011.

UTO, Nelson; MELO, Sandro Pereira de. **Vulnerabilidades em Aplicações Web e Mecanismos de Proteção.** SP, 2009. Disponível em:< http://pt.scribd.com/doc/91767961/15/Contramedidas>. Acesso em: 18/06/2012.

WHITAKER, Andrew; NEWMAN, Daniel P. **Penetration testing and network defense.** Indianapolis: Cisco press, 2006.

YAN, Jeff; EL, Ahmad Salah. **A Low-cost Attack on a Microsoft CAPTCHA:** School of Computing Science Technical Report, Newcastle University, England 2008. In: MUNIZ, Marcos de Paula. **Protegendo formulários web de robôs.** Uberlândia, MG: Faculdade Pitágoras, 2011.

ANEXO A – RESPOSTAS DO QUESTIONÁRIO DE PROBABILIDADE E IMPACTO DA VULNERABILIDADE DE FORÇA BRUTA DO MAGENTO

Questões sobre probabilidade:

Como é qualificado tecnicamente o grupo de atacantes?

- Algumas habilidades técnicas (3).

Como é motivado este grupo de atacantes a encontrar e explorar essa vulnerabilidade?

- Recompensa baixa ou nenhuma (1).

Quantas oportunidades esse grupo de atacantes tem para encontrar e explorar essa vulnerabilidade?

- Acesso completo (9).

Qual o tamanho deste grupo de atacantes?

- Usuários de internet anônimos (9).

Como é a facilidade desse grupo de atacantes descobrirem esta vulnerabilidade?

- Facilmente (7).

Como é a facilidade desse grupo de atacantes explorar essa vulnerabilidade?

- Difícil (3).

Como é o conhecimento sobre esta vulnerabilidade por este grupo de atacantes?

- Óbvio (6).

Qual a probabilidade de uma exploração ser detectada?

- Somente registrado (8).

Questões sobre impacto técnico:

Quantos dados podem ser divulgados e qual a sensibilidade deles?

- O mínimo de dados não confidenciais divulgados (2).

Qual a quantidade de dados pode ser corrompida e danificada?

- O mínimo de dados ligeiramente corrompidos (1).

Quanto à disponibilidade, qual a quantidade de serviços perdidos e quão vital isso é?

- O mínimo de serviços secundários interrompidos (1).

As ações dos invasores são rastreáveis?

- Possivelmente rastreável (7).

Questões sobre impacto no negócio:

Quanto de dano financeiro resultará uma exploração?

- Menos do que o custo para corrigir a vulnerabilidade (1).

Quanto de danos à reputação e prejudicaria o negócio uma exploração?

- Danos mínimos (1).

Quanto de não cumprimento de serviços aos clientes pode gerar uma exploração?

- Violação pequena (2).

Quanta informação pessoal e identificável pode ser divulgada?

- Uma pessoa (3).

ANEXO B – RESPOSTAS DO QUESTIONÁRIO DE PROBABILIDADE E IMPACTO DA VULNERABILIDADE DE FORÇA BRUTA DO OSCOMMERCE

Questões sobre probabilidade:

Como é qualificado tecnicamente o grupo de atacantes?

- Algumas habilidades técnicas (3).

Como é motivado este grupo de atacantes a encontrar e explorar essa vulnerabilidade?

- Recompensa baixa ou nenhuma (1).

Quantas oportunidades esse grupo de atacantes tem para encontrar e explorar essa vulnerabilidade?

- Acesso completo (9).

Qual o tamanho deste grupo de atacantes?

- Usuários de internet anônimos (9).

Como é a facilidade desse grupo de atacantes descobrirem esta vulnerabilidade?

- Facilmente (7).

Como é a facilidade desse grupo de atacantes explorar essa vulnerabilidade?

- Difícil (3).

Como é o conhecimento sobre esta vulnerabilidade por este grupo de atacantes?

- Óbvio (6).

Qual a probabilidade de uma exploração ser detectada?

- Somente registrado (8).

Questões sobre impacto técnico:

Quantos dados podem ser divulgados e qual a sensibilidade deles?

- O mínimo de dados não confidenciais divulgados (2).

Qual a quantidade de dados pode ser corrompida e danificada?

- O mínimo de dados ligeiramente corrompidos (1).

Quanto à disponibilidade, qual a quantidade de serviços perdidos e quão vital isso é?

- O mínimo de serviços secundários interrompidos (1).

As ações dos invasores são rastreáveis?

- Possivelmente rastreável (7).

Questões sobre impacto no negócio:

Quanto de dano financeiro resultará uma exploração?

- Menos do que o custo para corrigir a vulnerabilidade (1).

Quanto de danos à reputação e prejudicaria o negócio uma exploração?

- Danos mínimos (1).

Quanto de não cumprimento de serviços aos clientes pode gerar uma exploração?

- Violação pequena (2).

Quanta informação pessoal e identificável pode ser divulgada?

- Uma pessoa (3).

ANEXO C – RESPOSTAS DO QUESTIONÁRIO DE PROBABILIDADE E IMPACTO DA VULNERABILIDADE DE SESSION FIXATION DO OSCOMMERCE

Questões sobre probabilidade:

Como é qualificado tecnicamente o grupo de atacantes?

- Habilidades de segurança e *pentest* (9).

Como é motivado este grupo de atacantes a encontrar e explorar essa vulnerabilidade?

- Possível recompensa (4).

Quantas oportunidades esse grupo de atacantes tem para encontrar e explorar essa vulnerabilidade?

- Acesso completo (9).

Qual o tamanho deste grupo de atacantes?

- Usuários de internet anônimos (9).

Como é a facilidade desse grupo de atacantes descobrirem esta vulnerabilidade?

- Difícil (3).

Como é a facilidade desse grupo de atacantes explorar essa vulnerabilidade?

- Difícil (3).

Como é o conhecimento sobre esta vulnerabilidade por este grupo de atacantes?

- Oculto (4).

Qual a probabilidade de uma exploração ser detectada?

- Não registrado (9).

Questões sobre impacto técnico:

Quantos dados podem ser divulgados e qual a sensibilidade deles?

- O mínimo de dados não confidenciais divulgados (2).

Qual a quantidade de dados pode ser corrompida e danificada?

- O mínimo de dados ligeiramente corrompidos (1).

Quanto à disponibilidade, qual a quantidade de serviços perdidos e quão vital isso é?

- O mínimo de serviços secundários interrompidos (1).

As ações dos invasores são rastreáveis?

- Possivelmente rastreável (7).

Questões sobre impacto no negócio:

Quanto de dano financeiro resultará uma exploração?

- Menos do que o custo para corrigir a vulnerabilidade (1).

Quanto de danos à reputação e prejudicaria o negócio uma exploração?

- Danos mínimos (1).

Quanto de não cumprimento de serviços aos clientes pode gerar uma exploração?

- Violação pequena (2).

Quanta informação pessoal e identificável pode ser divulgada?

- Uma pessoa (3).

ANEXO D – RESPOSTAS DO QUESTIONÁRIO DE PROBABILIDADE E IMPACTO DA VULNERABILIDADE DE CSRF DO OSCOMMERCE

Questões sobre probabilidade:

Como é qualificado tecnicamente o grupo de atacantes?

- Habilidades de segurança e *pentest* (9).

Como é motivado este grupo de atacantes a encontrar e explorar essa vulnerabilidade?

- Possível recompensa (4).

Quantas oportunidades esse grupo de atacantes tem para encontrar e explorar essa vulnerabilidade?

- Acesso limitado (4).

Qual o tamanho deste grupo de atacantes?

- Administradores de sistemas (2).

Como é a facilidade desse grupo de atacantes descobrirem esta vulnerabilidade?

- Difícil (3).

Como é a facilidade desse grupo de atacantes explorar essa vulnerabilidade?

- Difícil (3).

Como é o conhecimento sobre esta vulnerabilidade por este grupo de atacantes?

- Oculto (4).

Qual a probabilidade de uma exploração ser detectada?

- Não registrado (9).

Questões sobre impacto técnico:

Quantos dados podem ser divulgados e qual a sensibilidade deles?

- Extensos dados não sensíveis divulgados (6).

Qual a quantidade de dados pode ser corrompida e danificada?

- Extensos dados e pouco corrompidos (5).

Quanto à disponibilidade, qual a quantidade de serviços perdidos e quão vital isso é?

- O mínimo de serviços primários interrompidos (5).

As ações dos invasores são rastreáveis?

- Possivelmente rastreável (7).

Questões sobre impacto no negócio:

Quanto de dano financeiro resultará uma exploração?

- Pequeno efeito no lucro anual (3).

Quanto de danos à reputação e prejudicaria o negócio uma exploração?

- Danos mínimos (1).

Quanto de não cumprimento de serviços aos clientes pode gerar uma exploração?

- Violação pequena (2).

Quanta informação pessoal e identificável pode ser divulgada?

- Centenas de pessoas (5).

ANEXO E – RESPOSTAS DO QUESTIONÁRIO DE PROBABILIDADE E IMPACTO DA VULNERABILIDADE DE LOCKING ACCOUNTS DO OSCOMMERCE

Questões sobre probabilidade:

Como é qualificado tecnicamente o grupo de atacantes?

- Algumas habilidades técnicas (3).

Como é motivado este grupo de atacantes a encontrar e explorar essa vulnerabilidade?

- Recompensa baixa ou nenhuma (1).

Quantas oportunidades esse grupo de atacantes tem para encontrar e explorar essa vulnerabilidade?

- Sem acesso conhecido (0).

Qual o tamanho deste grupo de atacantes?

- Usuários de internet anônimos (9).

Como é a facilidade desse grupo de atacantes descobrirem esta vulnerabilidade?

- Facilmente (7).

Como é a facilidade desse grupo de atacantes explorar essa vulnerabilidade?

- Ferramentas automatizadas disponíveis (9).

Como é o conhecimento sobre esta vulnerabilidade por este grupo de atacantes?

- Óbvio (6).

Qual a probabilidade de uma exploração ser detectada?

- Não registrado (9).

Questões sobre probabilidade impacto técnico:

Quantos dados podem ser divulgados e qual a sensibilidade deles?

- O mínimo de dados não confidenciais divulgados (2).

Qual a quantidade de dados pode ser corrompida e danificada?

- O mínimo de dados ligeiramente corrompidos (1).

Quanto à disponibilidade, qual a quantidade de serviços perdidos e quão vital isso é?

- O mínimo de serviços secundários interrompidos (1).

As ações dos invasores são rastreáveis?

- Possivelmente rastreável (7).

Questões sobre probabilidade impacto no negócio:

Quanto de dano financeiro resultará uma exploração?

- Menos do que o custo para corrigir a vulnerabilidade (1).

Quanto de danos à reputação e prejudicaria o negócio uma exploração?

- Danos mínimos (1).

Quanto de não cumprimento de serviços aos clientes pode gerar uma exploração?

- Violação pequena (2).

Quanta informação pessoal e identificável pode ser divulgada?

- Uma pessoa (3).

ANEXO F – RESPOSTAS DO QUESTIONÁRIO DE PROBABILIDADE E IMPACTO DA VULNERABILIDADE DE FORÇA BRUTA DO PRESTASHOP

Questões sobre probabilidade:

Como é qualificado tecnicamente o grupo de atacantes?

- Algumas habilidades técnicas (3).

Como é motivado este grupo de atacantes a encontrar e explorar essa vulnerabilidade?

- Recompensa baixa ou nenhuma (1).

Quantas oportunidades esse grupo de atacantes tem para encontrar e explorar essa vulnerabilidade?

- Acesso completo (9).

Qual o tamanho deste grupo de atacantes?

- Usuários de internet anônimos (9).

Como é a facilidade desse grupo de atacantes descobrirem esta vulnerabilidade?

- Facilmente (7).

Como é a facilidade desse grupo de atacantes explorar essa vulnerabilidade?

- Difícil (3).

Como é o conhecimento sobre esta vulnerabilidade por este grupo de atacantes?

- Óbvio (6).

Qual a probabilidade de uma exploração ser detectada?

- Somente registrado (8).

Questões sobre impacto técnico:

Quantos dados podem ser divulgados e qual a sensibilidade deles?

- O mínimo de dados não confidenciais divulgados (2).

Qual a quantidade de dados pode ser corrompida e danificada?

- O mínimo de dados ligeiramente corrompidos (1).

Quanto à disponibilidade, qual a quantidade de serviços perdidos e quão vital isso é?

- O mínimo de serviços secundários interrompidos (1).

As ações dos invasores são rastreáveis?

- Possivelmente rastreável (7).

Questões sobre impacto no negócio:

Quanto de dano financeiro resultará uma exploração?

- Menos do que o custo para corrigir a vulnerabilidade (1).

Quanto de danos à reputação e prejudicaria o negócio uma exploração?

- Danos mínimos (1).

Quanto de não cumprimento de serviços aos clientes pode gerar uma exploração?

- Violação pequena (2).

Quanta informação pessoal e identificável pode ser divulgada?

- Uma pessoa (3).

ANEXO G – RESPOSTAS DO QUESTIONÁRIO DE PROBABILIDADE E IMPACTO DA VULNERABILIDADE DE FORÇA BRUTA DO PRESTASHOP

Questões sobre probabilidade:

Como é qualificado tecnicamente o grupo de atacantes?

- Algumas habilidades técnicas (3).

Como é motivado este grupo de atacantes a encontrar e explorar essa vulnerabilidade?

- Possível recompensa (4).

Quantas oportunidades esse grupo de atacantes tem para encontrar e explorar essa vulnerabilidade?

- Acesso completo (9).

Qual o tamanho deste grupo de atacantes?

- Usuários de internet anônimos (9).

Como é a facilidade desse grupo de atacantes descobrirem esta vulnerabilidade?

- Facilmente (7).

Como é a facilidade desse grupo de atacantes explorar essa vulnerabilidade?

- Difícil (3).

Como é o conhecimento sobre esta vulnerabilidade por este grupo de atacantes?

- Óbvio (6).

Qual a probabilidade de uma exploração ser detectada?

- Somente registrado (8).

Questões sobre impacto técnico:

Quantos dados podem ser divulgados e qual a sensibilidade deles?

- O mínimo de dados não confidenciais divulgados (2).

Qual a quantidade de dados pode ser corrompida e danificada?

- O mínimo de dados ligeiramente corrompidos (1).

Quanto à disponibilidade, qual a quantidade de serviços perdidos e quão vital isso é?

- O mínimo de serviços secundários interrompidos (1).

As ações dos invasores são rastreáveis?

- Possivelmente rastreável (7).

Questões sobre impacto no negócio:

Quanto de dano financeiro resultará uma exploração?

- Pequeno efeito no lucro anual (3).

Quanto de danos à reputação e prejudicaria o negócio uma exploração?

- Danos mínimos (1).

Quanto de não cumprimento de serviços aos clientes pode gerar uma exploração?

- Violação pequena (2).

Quanta informação pessoal e identificável pode ser divulgada?

- Centenas de pessoas (5).

ANEXO H – RESPOSTAS DO QUESTIONÁRIO DE PROBABILIDADE E IMPACTO DA VULNERABILIDADE DE TIMEOUT PRESTASHOP

Questões sobre probabilidade:

Como é qualificado tecnicamente o grupo de atacantes?

- Habilidades de segurança e *pentest* (9).

Como é motivado este grupo de atacantes a encontrar e explorar essa vulnerabilidade?

- Recompensa baixa ou nenhuma (1).

Quantas oportunidades esse grupo de atacantes tem para encontrar e explorar essa vulnerabilidade?

- Acesso completo (9).

Qual o tamanho deste grupo de atacantes?

- Parceiros (5).

Como é a facilidade desse grupo de atacantes descobrirem esta vulnerabilidade?

- Facilmente (7).

Como é a facilidade desse grupo de atacantes explorar essa vulnerabilidade?

- Difícil (3).

Como é o conhecimento sobre esta vulnerabilidade por este grupo de atacantes?

- Oculto (4).

Qual a probabilidade de uma exploração ser detectada?

- Não registrado (9).

Questões sobre impacto técnico:

Quantos dados podem ser divulgados e qual a sensibilidade deles?

- O mínimo de dados não confidenciais divulgados (2).

Qual a quantidade de dados pode ser corrompida e danificada?

- O mínimo de dados ligeiramente corrompidos (1).

Quanto à disponibilidade, qual a quantidade de serviços perdidos e quão vital isso é?

- O mínimo de serviços secundários interrompidos (1).

As ações dos invasores são rastreáveis?

- Possivelmente rastreável (7).

Questões sobre impacto no negócio:

Quanto de dano financeiro resultará uma exploração?

- Menos do que o custo para corrigir a vulnerabilidade (1).

Quanto de danos à reputação e prejudicaria o negócio uma exploração?

- Danos mínimos (1).

Quanto de não cumprimento de serviços aos clientes pode gerar uma exploração?

- Violação pequena (2).

Quanta informação pessoal e identificável pode ser divulgada?

- Uma pessoa (3).

ANEXO I – RESPOSTAS DO QUESTIONÁRIO DE PROBABILIDADE E IMPACTO DA VULNERABILIDADE DE TIMEOUT PRESTASHOP

Questões sobre probabilidade:

Como é qualificado tecnicamente o grupo de atacantes?

- Habilidades de segurança e *pentest* (9).

Como é motivado este grupo de atacantes a encontrar e explorar essa vulnerabilidade?

- Recompensa baixa ou nenhuma (1).

Quantas oportunidades esse grupo de atacantes tem para encontrar e explorar essa vulnerabilidade?

- Acesso limitado (4).

Qual o tamanho deste grupo de atacantes?

- Parceiros (5).

Como é a facilidade desse grupo de atacantes descobrirem esta vulnerabilidade?

- Facilmente (7).

Como é a facilidade desse grupo de atacantes explorar essa vulnerabilidade?

- Difícil (3).

Como é o conhecimento sobre esta vulnerabilidade por este grupo de atacantes?

- Oculto (4).

Qual a probabilidade de uma exploração ser detectada?

- Não registrado (9).

Questões sobre impacto técnico:

Quantos dados podem ser divulgados e qual a sensibilidade deles?

- O mínimo de dados críticos divulgados (6).

Qual a quantidade de dados pode ser corrompida e danificada?

- O mínimo de dados seriamente corrompidos (3).

Quanto à disponibilidade, qual a quantidade de serviços perdidos e quão vital isso é?

- O mínimo de serviços secundários interrompidos (1).

As ações dos invasores são rastreáveis?

- Possivelmente rastreável (7).

Questões sobre impacto no negócio:

Quanto de dano financeiro resultará uma exploração?

- Pequeno efeito no lucro anual (3).

Quanto de danos à reputação e prejudicaria o negócio uma exploração?

- Danos mínimos (1).

Quanto de não cumprimento de serviços aos clientes pode gerar uma exploração?

- Violação pequena (2).

Quanta informação pessoal e identificável pode ser divulgada?

- Centenas de pessoas (5).

ANEXO J – RESPOSTAS DO QUESTIONÁRIO DE PROBABILIDADE E IMPACTO DA VULNERABILIDADE DE COOKIES ATTRIBUTES PRESTASHOP

Questões sobre probabilidade:

Como é qualificado tecnicamente o grupo de atacantes?

- Algumas habilidades técnicas (3).

Como é motivado este grupo de atacantes a encontrar e explorar essa vulnerabilidade?

- Possível recompensa (4).

Quantas oportunidades esse grupo de atacantes tem para encontrar e explorar essa vulnerabilidade?

- Acesso completo (9).

Qual o tamanho deste grupo de atacantes?

- Usuários autenticados (6).

Como é a facilidade desse grupo de atacantes descobrirem esta vulnerabilidade?

- Facilmente (7).

Como é a facilidade desse grupo de atacantes explorar essa vulnerabilidade?

- Difícil (3).

Como é o conhecimento sobre esta vulnerabilidade por este grupo de atacantes?

- Óbvio (6).

Qual a probabilidade de uma exploração ser detectada?

- Não registrado (9).

Questões sobre impacto técnico:

Quantos dados podem ser divulgados e qual a sensibilidade deles?

- O mínimo de dados não confidenciais divulgados (2).

Qual a quantidade de dados pode ser corrompida e danificada?

- O mínimo de dados ligeiramente corrompidos (1).

Quanto à disponibilidade, qual a quantidade de serviços perdidos e quão vital isso é?

- O mínimo de serviços secundários interrompidos (1).

As ações dos invasores são rastreáveis?

- Possivelmente rastreável (7).

Questões sobre impacto no negócio:

Quanto de dano financeiro resultará uma exploração?

- Menos do que o custo para corrigir a vulnerabilidade (1).

Quanto de danos à reputação e prejudicaria o negócio uma exploração?

- Danos mínimos (1).

Quanto de não cumprimento de serviços aos clientes pode gerar uma exploração?

- Violação pequena (2).

Quanta informação pessoal e identificável pode ser divulgada?

- Uma pessoa (3).

ANEXO K – RESPOSTAS DO QUESTIONÁRIO DE PROBABILIDADE E IMPACTO DA VULNERABILIDADE DE COOKIES ATTRIBUTES PRESTASHOP

Questões sobre probabilidade:

Como é qualificado tecnicamente o grupo de atacantes?

- Algumas habilidades técnicas (3).

Como é motivado este grupo de atacantes a encontrar e explorar essa vulnerabilidade?

- Possível recompensa (4).

Quantas oportunidades esse grupo de atacantes tem para encontrar e explorar essa vulnerabilidade?

- Acesso completo (9).

Qual o tamanho deste grupo de atacantes?

- Usuários autenticados (6).

Como é a facilidade desse grupo de atacantes descobrirem esta vulnerabilidade?

- Facilmente (7).

Como é a facilidade desse grupo de atacantes explorar essa vulnerabilidade?

- Difícil (3).

Como é o conhecimento sobre esta vulnerabilidade por este grupo de atacantes?

- Óbvio (6).

Qual a probabilidade de uma exploração ser detectada?

- Não registrado (9).

Questões sobre impacto técnico:

Quantos dados podem ser divulgados e qual a sensibilidade deles?

- O mínimo de dados críticos divulgados (6).

Qual a quantidade de dados pode ser corrompida e danificada?

- Extensos dados e pouco corrompidos (5).

Quanto à disponibilidade, qual a quantidade de serviços perdidos e quão vital isso é?

- O mínimo de serviços secundários interrompidos (1).

As ações dos invasores são rastreáveis?

- Possivelmente rastreável (7).

Questões sobre impacto no negócio:

Quanto de dano financeiro resultará uma exploração?

- Menos do que o custo para corrigir a vulnerabilidade (1).

Quanto de danos à reputação e prejudicaria o negócio uma exploração?

- Danos mínimos (1).

Quanto de não cumprimento de serviços aos clientes pode gerar uma exploração?

- Violação pequena (2).

Quanta informação pessoal e identificável pode ser divulgada?

- Centenas de pessoas (5).

ANEXO L – RESPOSTAS DO QUESTIONÁRIO DE PROBABILIDADE E IMPACTO DA VULNERABILIDADE DE CSRF PRESTASHOP

Questões sobre probabilidade:

Como é qualificado tecnicamente o grupo de atacantes?

- Habilidades de segurança e *pentest* (9).

Como é motivado este grupo de atacantes a encontrar e explorar essa vulnerabilidade?

- Possível recompensa (4).

Quantas oportunidades esse grupo de atacantes tem para encontrar e explorar essa vulnerabilidade?

- Acesso limitado (4).

Qual o tamanho deste grupo de atacantes?

- Desenvolvedores (2).

Como é a facilidade desse grupo de atacantes descobrirem esta vulnerabilidade?

- Difícil (3).

Como é a facilidade desse grupo de atacantes explorar essa vulnerabilidade?

- Difícil (3).

Como é o conhecimento sobre esta vulnerabilidade por este grupo de atacantes?

- Oculto (4).

Qual a probabilidade de uma exploração ser detectada?

- Não registrado (9).

Questões sobre impacto técnico:

Quantos dados podem ser divulgados e qual a sensibilidade deles?

- O mínimo de dados não confidenciais divulgados (2).

Qual a quantidade de dados pode ser corrompida e danificada?

- O mínimo de dados ligeiramente corrompidos (1).

Quanto à disponibilidade, qual a quantidade de serviços perdidos e quão vital isso é?

- O mínimo de serviços secundários interrompidos (1).

As ações dos invasores são rastreáveis?

- Possivelmente rastreável (7).

Questões sobre impacto no negócio:

Quanto de dano financeiro resultará uma exploração?

- Menos do que o custo para corrigir a vulnerabilidade (1).

Quanto de danos à reputação e prejudicaria o negócio uma exploração?

- Danos mínimos (1).

Quanto de não cumprimento de serviços aos clientes pode gerar uma exploração?

- Violação pequena (2).

Quanta informação pessoal e identificável pode ser divulgada?

- Uma pessoa (3).

ANEXO M – RESPOSTAS DO QUESTIONÁRIO DE PROBABILIDADE E IMPACTO DA VULNERABILIDADE DE CSRF PRESTASHOP

Questões sobre probabilidade:

Como é qualificado tecnicamente o grupo de atacantes?

- Habilidades de segurança e *pentest* (9).

Como é motivado este grupo de atacantes a encontrar e explorar essa vulnerabilidade?

- Possível recompensa (4).

Quantas oportunidades esse grupo de atacantes tem para encontrar e explorar essa vulnerabilidade?

- Acesso limitado (4).

Qual o tamanho deste grupo de atacantes?

- Administradores de sistemas (2).

Como é a facilidade desse grupo de atacantes descobrirem esta vulnerabilidade?

- Difícil (3).

Como é a facilidade desse grupo de atacantes explorar essa vulnerabilidade?

- Difícil (3).

Como é o conhecimento sobre esta vulnerabilidade por este grupo de atacantes?

- Oculto (4).

Qual a probabilidade de uma exploração ser detectada?

- Não registrado (9).

Questões sobre impacto técnico:

Quantos dados podem ser divulgados e qual a sensibilidade deles?

- Extensos dados não sensíveis divulgados (6).

Qual a quantidade de dados pode ser corrompida e danificada?

- Extensos dados e pouco corrompidos (5).

Quanto à disponibilidade, qual a quantidade de serviços perdidos e quão vital isso é?

- O mínimo de serviços primários interrompidos (5).

As ações dos invasores são rastreáveis?

- Possivelmente rastreável (7).

Questões sobre impacto no negócio:

Quanto de dano financeiro resultará uma exploração?

- Pequeno efeito no lucro anual (3).

Quanto de danos à reputação e prejudicaria o negócio uma exploração?

- Danos mínimos (1).

Quanto de não cumprimento de serviços aos clientes pode gerar uma exploração?

- Violação pequena (2).

Quanta informação pessoal e identificável pode ser divulgada?

- Centenas de pessoas (5).