

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Adelson Pacheco dos Reis

**Análise de vulnerabilidades de segurança em
sistemas de Internet Banking utilizando
ferramentas de código aberto**

Uberlândia, Brasil

2018

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Adelson Pacheco dos Reis

Análise de vulnerabilidades de segurança em sistemas de Internet Banking utilizando ferramentas de código aberto

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como requisito exigido parcial à obtenção do grau de Bacharel em Sistemas de Informação.

Orientador: Rodrigo Sanches Miani

Universidade Federal de Uberlândia – UFU

Faculdade de Ciência da Computação

Bacharelado em Sistemas de Informação

Uberlândia, Brasil

2018

Adelson Pacheco dos Reis

Análise de vulnerabilidades de segurança em sistemas de Internet Banking utilizando ferramentas de código aberto

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como requisito exigido parcial à obtenção do grau de Bacharel em Sistemas de Informação.

Trabalho aprovado. Uberlândia, Brasil, 21 de dezembro de 2018:

Rodrigo Sanches Miani
Orientador

Professor

Professor

Uberlândia, Brasil
2018

Dedico esse trabalho a meus pais, Adélio e Geralda, meus irmãos, Lucas e Daiane, a meus amigos e professores. Sem eles, o sonho de concluir a graduação seria impossível.

Agradecimentos

Agradeço a Deus, autor e sustentador da vida, por ser minha força, refúgio e fortaleza. Sem Ele, nada disso seria possível.

Agradeço a minha família e amigos, por terem estado ao meu lado durante toda essa caminhada, sempre me apoiando, trazendo palavras de ânimo, mostrando que toda essa luta vale a pena.

A todos os professores e servidores do curso de Sistemas de Informação da Universidade Federal de Uberlândia Campus Monte Carmelo, em especial, ao professor Rodrigo Sanches Miani, meu orientador, pela paciência, apoio e tempo dedicado para que esse trabalho fosse realizado.

Ademais, agradeço a todos aqueles que de alguma forma estiveram e estão próximos de mim, fazendo esta vida valer cada vez mais a pena.

Resumo

Devido a popularização do acesso à Internet, as instituições bancárias encontraram uma boa oportunidade de disseminar melhor seus serviços através dos ditos Internet *Banking*. Contudo, a criação de serviços por meio da Internet trouxe como consequência altos riscos de invasões. Para evitar invasões e outros ataques às suas próprias informações e de terceiros, bancos vêm investindo massivamente em segurança de dados. Este trabalho tem como objetivo avaliar os mecanismos de segurança em sistemas de Internet *Banking* usando ferramentas *black-box*, de modo que seja possível apresentar um resumo sobre o nível de segurança de cada sistema analisado, bem como buscar relacionar a quantidade e o grau das vulnerabilidades encontradas com o número de acessos aos sistemas e o patrimônio líquido dos bancos analisados. Para este trabalho, serão utilizadas as ferramentas de análise de vulnerabilidades *Vega* e *Skipfish*, ambas de código aberto e as vulnerabilidades serão classificadas de acordo com a lista OWASP Top 10 de 2017 – que abrange as vulnerabilidades mais encontradas em sistemas *web* do ano em questão. O desenvolvimento do projeto é feito observando o cenário tecnológico nacional bem como suas estruturas de segurança. Os resultados obtidos demonstram que as vulnerabilidades mais encontradas são: Configuração Incorreta de Segurança, Entidades Externas XML (XXE), Injeção e *Cross-Site Scripting* (XSS). Com exceção da Instituição L, as instituições de maior valor financeiro de um modo geral não apresentam piores desempenhos na análise.

Palavras-chave: Internet *Banking*, Segurança de informação, Vulnerabilidades de segurança

Lista de ilustrações

Figura 1 – Exemplo SQL Injection. Adaptado de (MOCELIN; GONCALVES, 2008)	17
Figura 2 – Exemplo Cross-Site Scripting. Adaptado de Fortify Taxonomy (2016)	17
Figura 3 – Exemplo Quebra de Autenticação e Controle de Sessão. Adaptado de Guide to Cryptography - OWASP (2016)	18
Figura 4 – Exemplo Vega. Imagem produzida pelo autor.	20
Figura 5 – Mapa site Skipfish. Imagem produzida pelo autor.	21
Figura 6 – Esquematização da metodologia utilizada.	25
Figura 7 – Distribuição do total de vulnerabilidades encontradas por ferramenta.	30
Figura 8 – Vulnerabilidades encontradas por ferramenta.	36

Lista de tabelas

Tabela 1 – Instituições Financeiras por Categoria de Patrimônio Líquido e Grupo de Número de Acesso. Extraído de Banco Central do Brasil (2017) e Jornal Estado de Minas (2017)	26
Tabela 2 – Ocorrências de vulnerabilidades detectadas por ferramenta.	28
Tabela 3 – Top 10 vulnerabilidades de segurança de 2017 segundo OWASP.	28
Tabela 4 – Instituições Financeiras ordenadas pelo número de vulnerabilidades do tipo Configuração Incorreta de Segurança.	31
Tabela 5 – Instituições Financeiras ordenadas pelo número de vulnerabilidades do tipo Entidades Externas XML.	32
Tabela 6 – Instituições Financeiras ordenadas pelo número de vulnerabilidades do tipo Injeção.	32
Tabela 7 – Instituições Financeiras ordenadas pelo número de vulnerabilidades do tipo <i>Cross-Site Scripting</i>	33
Tabela 8 – Instituições Financeiras ordenadas pelo número de vulnerabilidades do tipo Exposição de Dados Sensíveis.	34
Tabela 9 – Instituições Financeiras ordenadas pelo número de vulnerabilidades do tipo Quebra de Autenticação e Controle de Sessão.	35
Tabela 10 – Instituições Financeiras ordenadas pelo número de vulnerabilidades do tipo Quebra de Controle de Acesso.	36
Tabela 11 – Instituições Financeiras ordenadas pelo número de vulnerabilidades do tipo Utilização de Componentes Vulneráveis Conhecidos.	36
Tabela 12 – Instituições Financeiras ordenadas pelo nível de criticidade de seus portais de Internet <i>Banking</i>	37
Tabela 13 – Instituições Financeiras pelo Número de Vulnerabilidades e Grupo de Número de Acessos. Extraído de Alexa (2017).	38
Tabela 14 – Instituições Financeiras pelo Número de Vulnerabilidades e Categoria de Patrimônio Líquido. Extraído de Banco Central do Brasil (2017).	39
Tabela 15 – Categoria por Média de Vulnerabilidades Totais e Vulnerabilidades Severas.	40

Lista de abreviaturas e siglas

ATM	<i>Automated Teller Machine</i>
BACEN	Banco Central do Brasil
DDL	<i>Data Definition Language</i>
DDoS	<i>Distributed Denial of Service</i>
DML	<i>Data Manipulation Language</i>
DoS	<i>Denial of Service</i>
FEBRABAN	Federação Brasileira de Bancos
HTTP	<i>Hypertext Transfer Protocol</i>
HTTPS	<i>Hypertext Transfer Protocol Secure</i>
ID	Identidade
IF	Instituição Financeira
JSP	<i>Java Server Pages</i>
LDAP	<i>Lightweight Directory Access Protocol</i>
OS	<i>Operating System</i>
OWASP	<i>Open Web Application Security Project</i>
SQL	<i>Structured Query Language</i>
SQLi	<i>Structured Query Language Injection</i>
XML	<i>Extensible Markup Language</i>
XSS	<i>Cross-Site Scripting</i>
XXE	<i>XML External Entity</i>

Sumário

1	INTRODUÇÃO	10
2	FUNDAMENTAÇÃO TEÓRICA	13
2.1	Segurança da Informação	13
2.2	Vulnerabilidades	14
2.2.1	Injeção	16
2.2.2	Cross Site Scripting	17
2.2.3	Quebra de Autenticação e Controle de Sessão	18
2.3	Ferramentas	19
2.4	Trabalhos correlatos	20
3	DESENVOLVIMENTO	24
3.1	Metodologia	24
3.2	Resultados	27
3.3	Análise dos resultados	29
3.3.1	Configuração Incorreta de Segurança	30
3.3.2	Entidades Externas XML	30
3.3.3	Injeção	31
3.3.4	<i>Cross-Site Scripting</i>	33
3.3.5	Exposição de Dados Sensíveis	33
3.3.6	Quebra de Autenticação e Controle de Sessão	34
3.3.7	Quebra de Controle de Acesso e Utilização de Componentes Vulneráveis	
	Conhecidos	35
3.3.8	Vulnerabilidades por ferramenta	36
3.3.9	Discussão	36
4	CONCLUSÃO	41
	REFERÊNCIAS	42

1 Introdução

Segundo a [FEBRABAN \(2011\)](#), um banco é uma instituição pertencente ao Sistema Financeiro Nacional, regulado pelo Banco Central do Brasil que rentabiliza as economias e poupanças das pessoas e empresas através do pagamento de juros, realiza serviços de pagamentos e recebimentos e financia o consumo e o investimento das pessoas e empresas cobrando para isso juros.

Ainda segundo ela, existem três tipos de bancos: Bancos Comerciais, Bancos de Investimento e Bancos Múltiplos, que são definidos da seguinte forma:

- Banco Comercial: Banco que pode ser nacional ou não, de caráter público ou privado, que as pessoas e empresas utilizam no cotidiano, para abrir uma conta corrente e realizar movimentações bancárias, pagar contas e boletos, fazer empréstimos ou solicitar cartões de crédito, por exemplo. Estes bancos têm a característica de prestar serviços à sociedade e para tal, possuem redes de agências e postos de atendimento bancário, oferecendo ainda canais eletrônicos para seu público, tais como Internet Banking, Central Transacional, Central de Atendimento ao Cliente (SAC), Central de Reclamações (Ouvidoria) e Caixas Eletrônicos (ATMs);
- Banco de Investimento: De caráter privado, é uma instituição financeira especializada em financiamento da atividade produtiva, para suprimento de capital de giro e fixo à longo prazo, operações de participação societária com caráter temporário, bem como administração de recursos de terceiros. Estes bancos não disponibilizam aos clientes abertura de conta corrente, e captam recursos via depósito a prazo, repasse de recursos externos e internos, e venda de cotas de fundos de investimento por eles administrados;
- Banco Múltiplo: Instituição financeira que possui as funções supracitadas, referente à Banco Comercial e Banco de Investimento.

A indústria bancária tem sido significativamente influenciada pela evolução da tecnologia. As crescentes aplicações computadorizadas, destinadas ao sistema bancário, reduziram o custo de transações e aumentaram a velocidade do serviço substancialmente.

De acordo com [Hole, Moen e Tjostheim \(2006\)](#), a natureza dos intermediários financeiros fizeram com que os bancos melhorassem suas tecnologias de produção, focando na distribuição de produtos. Em outras palavras, a evolução da tecnologia empregada nos sistemas bancários tem sido impulsionada, principalmente pela mudança nos canais

de distribuição, como se tem observado nos caixas eletrônicos, e mais recentemente, no Internet *Banking* e *Mobile Banking*.

Tjøstheim e Moen (2005) caracterizam Internet *Banking* como um processo inovador por proporcionar aos consumidores, a possibilidade de controle de suas contas bancárias, sem a necessidade de ir à algum posto de atendimento. Isso também permite que consumidores visitem ambientes virtuais dos bancos via *web*, o que possibilita aos mesmos, mais fácil e rápido acesso as suas informações bancárias, bem como agilidade e comodidade nas transações financeiras.

Conforme Brei (2001), em 1998, metade dos bancos brasileiros já ofereciam serviços via Internet, mas apenas 0,5% das transações bancárias eram conduzidas pelos clientes via *web*.

Segundo Pires e Filho (2008), o emprego da tecnologia em serviços bancários possibilitou um aumento de aproximadamente 257% na utilização do Internet *Banking* pelos clientes de 2001 a 2005, passando à 3,5% do total de transações em 2001 e atingindo 9,02% em 2005. Nesse mesmo ano, do total de 70,5 milhões de contas bancárias no Brasil, 37,3% já utilizavam serviços bancários *online*.

Uma década depois, de acordo com Deloitte (2016), 17 bancos representavam 93% dos ativos da indústria bancária do país. Ainda em 2015, foram registradas 11,2 bilhões de transações bancárias via *web* – 54% do total –, representando um crescimento de 138% em relação a 2014, que atingiu o número de 4,7 bilhões de transações realizadas utilizando o mesmo meio. O número de contas com acesso à Internet *Banking* também sofreu um aumento significativo entre esses dois anos, saltando de 56 milhões em 2014, para 62 milhões em 2015.

De fato, a disseminação do Internet *Banking* mudou significativamente a interação das pessoas com o sistema bancário. Com todo este crescimento, vem surgindo cada vez mais a necessidade de estudos e aperfeiçoamento de mecanismos para garantir a segurança das informações, tanto do cliente quanto das instituições financeiras.

Neste intuito Correia et al. (2008) apresenta um estudo detalhado sobre a implementação de alguns mecanismos de segurança, utilizados em sistemas de Internet *Banking*, já os estudos de Antunes e Vieira (2015) e Vieira, Antunes e Madeira (2009) comparam de formas distintas, ferramentas de verificação de vulnerabilidades.

Por outro lado, criminosos vêm utilizando cada vez mais esta tecnologia para cometer crimes virtuais, se aproveitando das vulnerabilidades que essa nova maneira de interação possui. Somente em 2015, ataques a Internet *Banking* no Brasil levaram bancos a terem perdas de cerca de R\$ 1,8 bilhão (ESTADÃO, 2015). O portal Digital (2017) também mostra que somente em 2017, bancos perderam mais de US\$ 1,7 milhão somente por falhas de segurança em Internet *Banking*.

Conforme informado pelo [Tecnundo \(2015\)](#) em 2015, neste ano o serviço de Internet *Banking* foi o principal alvo de ataques de criminosos no Brasil, e neste sentido, verifica-se que [Gallao e Junior \(2016\)](#) demonstraram, através de análises estatísticas, a criticidade dos impactos causados tanto ao cliente e aos Bancos referentes a ataques virtuais, além das principais características do Internet *Banking* que se tornam alvos fáceis de extravio de informações e deixam a desejar na segurança desta ferramenta.

Todas estas ocorrências e inúmeras outras, vem gerando nos clientes a desconfiança da segurança e confiabilidade deste serviço, conforme dados de [Avellareduarte \(2017\)](#) onde 97% dos usuários de Internet *Banking* se mostraram preocupados com a segurança do serviço, e 77% estavam muito ou extremamente apreensivos com estas vulnerabilidades.

Por isso, o objetivo desse trabalho é avaliar a segurança dos sistemas bancários sob o ponto de vista de uma das ferramentas mais modernas e disponíveis para o público atualmente, o Internet *Banking*, oferecendo um ampla análise da segurança em ambientes bancários. O que nos leva para a seguinte hipótese: existe uma relação entre segurança dos sistemas e o patrimônio líquido de cada IF aos quais eles pertençam.

O método empregado na concretização desse trabalho é subdividido em quatro etapas principais:

1. Seleção das ferramentas: Escolha dentre as diversas ferramentas black-box gratuitas de verificação de vulnerabilidades;
2. Seleção das instituições financeiras (IFs): Escolha de 20 sistemas de Internet *Banking* de IFs atuantes no Brasil, tendo como critério de avaliação seus respectivos patrimônios líquidos;
3. Coleta das vulnerabilidades de segurança dos sistemas das IF: varredura utilizando todas as ferramentas;
4. Análise dos resultados obtidos: análise feita buscando validar a hipótese supracitada.

Este estudo encontra-se subdividido em capítulos, organizados da seguinte forma: o capítulo 2 trata-se da fundamentação teórica e os trabalhos correlatos, responsáveis pelo o entendimento e estruturação de todo o estudo; já o capítulo 3 abrange o desenvolvimento da pesquisa, sendo composto pela metodologia, resultados e sua análise. Finalmente, no capítulo 4 é desenvolvida a conclusão, que reúne as considerações finais obtidas na execução deste estudo.

2 Fundamentação Teórica

Desde o início das redes de computadores, *hacking* vem sendo uma atividade comumente praticada através de infraestruturas de tecnologia da informação, e estão cada vez mais sendo motivadas pelo avanço de tecnologias de sistemas e a popularização do acesso à Internet. De acordo com [Al-Sanea e Al-Daraiseh \(2015\)](#), ataques podem ser praticados por diferentes motivos, como políticos ou para obter vantagens através de roubo de informações de clientes. A disponibilidade de novas ferramentas cada vez mais poderosas, simples e de mais fácil manuseio também despertam motivação para seu uso.

[Al-Sanea e Al-Daraiseh \(2015\)](#) mostram que atacantes utilizam dois métodos principais para atacar sistemas. Um deles é atacar uma vulnerabilidade até então desconhecida ou sem um pacote de segurança desenvolvido para tal, e o outro método é descobrir as vulnerabilidades do sistema e atacá-las com o auxílio de códigos maliciosos e ferramentas especialmente desenvolvidas para esta ação. Por realizar ataques à uma aplicação sabendo o que se deve atacar, o segundo método é menos complexo e mais eficiente que o primeiro.

Na subseção 2.1 será abordado alguns conceitos básicos de Segurança de Informação, onde, de modo breve, serão definidos aspectos como integridade, confidencialidade e disponibilidade. Na subseção 2.2, será abordada diferentes maneiras de análise de vulnerabilidades, bem como as principais vulnerabilidades de segurança segundo a OWASP, organização sem fins lucrativos que tem como objetivo melhorar da segurança de *software*, permitindo que empresas tomem decisões baseadas em seus estudos. A subseção 2.3 define das ferramentas de *black-boxing test* – explicadas na subseção 2.2 – utilizadas nesse trabalho. Por fim, a subseção 2.4 aborda os trabalhos relacionados a este.

2.1 Segurança da Informação

De acordo com [Almeida \(2007\)](#), a missão da Segurança de Informação é proteger sistemas de informação contra o uso, divulgação, modificação ou destruição de informações confidenciais por meio de acessos não autorizados. Está relacionada a três aspectos: integridade, confidencialidade e disponibilidade. De forma breve, [Netto e Silveira \(2007\)](#) caracteriza estes aspectos como:

- Integridade: Busca manter a exatidão das informações, de modo a garantir que apenas pessoas autorizadas possam adicioná-las, modificá-las ou excluí-las. *Shell Injection* é um exemplo ataque à integridade.
- Confidencialidade: Garante que apenas pessoas autorizadas possam acessar deter-

minadas informações. Busca proteger cada informação de acordo com seu grau de sigilo. Como exemplos de ataques à confidencialidade temos os ataques passivos – escuta, interceptação e análise de tráfego de rede, afim de obter acesso a informações confidenciais.

- Disponibilidade: Garante que informações possam ser acessadas a qualquer momento pelas pessoas autorizadas a fazê-lo. Ataques como DoS – ataque de negação de serviço – e DDoS – ataque distribuído de negação de serviço – são exemplos de ataques à disponibilidade.

2.2 Vulnerabilidades

Vulnerabilidade, no universo da segurança de informação, refere-se às falhas de um sistema que podem deixá-lo sujeito à ataques. Uma vulnerabilidade pode também se referir a qualquer tipo de deficiência de um sistema de informação, seja um conjunto de procedimentos falhos pelos quais um intruso pode executar códigos maliciosos ou informações confidenciais expostas que possam ser exploradas (SOLMS; NIEKERK, 2013).

Para sua detecção, dentre outras abordagens, está a identificação em fóruns e redes sociais do surgimento de novas vulnerabilidades. Como exemplo de tais detecções podemos listar a comparação das datas divulgação de vulnerabilidades de bases oficiais com as primeiras menções em redes sociais (RODRIGUEZ et al., 2018), extração de alertas de segurança como o lançamento de atualizações importantes ou novas ameaças (QUEIROZ; KEEGAN; MTENZI, 2017) e criação de classificadores que estimam a probabilidade de uma vulnerabilidade ser explorada no mundo real com base em informações coletadas em fóruns Deep/DarkWeb (ALMUKAYNIZI et al., 2017).

Para Makino e Klyuev (2015), quase todas as ferramentas de detecção de vulnerabilidades em aplicações *web* consistem em três principais componentes: *crawling*, ataque, e análise de componentes. De forma rápida, há duas principais maneiras de analisar as vulnerabilidades:

- *White boxing test*: Consiste na análise manual do código das aplicações ou usando alguma ferramenta. Porém, por conta da dificuldade do código, utilizar ferramentas para análise pode dificultar o encontro de todas as falhas.
- *Black boxing test*: Consiste na análise da aplicação usando ferramentas de varredura que não têm o conhecimento interno da aplicação. Normalmente utilizam-se técnicas *fuzzy* nas requisições HTTP.

Além disso, também é necessário considerar quais tipos de vulnerabilidades cada ferramenta de varredura deve encontrar, e para tal, foi considerada a lista das Top 10

vulnerabilidades da [OWASP \(2017\)](#). Esta lista é um documento composto pelas vulnerabilidades mais críticas no ano de sua criação e representa um amplo consenso sobre os riscos de segurança mais críticos para aplicações *web*, cujo objetivo é conscientizar organizações e indivíduos afim de minimizar esses riscos e produzir códigos mais seguros.

As vulnerabilidades da lista Top 10 da [OWASP \(2017\)](#) estão descritas abaixo:

1. Injeção: ocorre quando o banco de dados é alvo de ataques SQL, onde o atacante envia comandos de definição de dados – DDL (*create*, *alter*, *drop*), comandos de manipulação de dados – DML (*select*, *update*, *insert*, *delete*) e/ou consultas não confiáveis. Isso faz com que o invasor engane o interpretador do banco de dados e o faça executar comandos maliciosos, retornando dados sem a devida autorização. Tais comandos maliciosos podem retornar dados confidenciais de clientes por atacantes, resultando em acesso às suas contas bancárias, por exemplo.
2. Quebra de Autenticação e Controle de Sessão: ocorre quando há deficiências na implementação de funções de gerenciamento de autenticação e sessão da aplicação, abrindo brechas para que invasores explorem outras falhas da aplicação, comprometam senhas e *tokens* de sessão ou se apropriem da identidades de outros usuários. Atacantes podem obter acesso às contas bancárias de clientes que utilizem senhas frágeis em sistemas de Internet *Banking* que não as validem.
3. Exposição de Dados Sensíveis: ocorre quando não é dada a devida proteção à dados sensíveis, abrindo margem para que invasores consultem ou alterem esses dados. Esta vulnerabilidade quebra a segurança de dados nos sistemas das instituições financeiras, facilitando acesso de atacantes aos seus dados confidenciais e de seus clientes.
4. Entidades Externas XML (XXE): ocorre quando *softwares* ultrapassados e codificação desatualizada são utilizadas. Entidades externas podem ser usadas para acessar, divulgar e compartilhar arquivos internos, executar códigos remotamente, varrer portas internas e realizar ataques DoS. Ataques DoS podem deixar indisponíveis serviços bancários por meio da Internet, prejudicando clientes que precisem realizar operações financeiras com certa urgência.
5. Quebra de Controle de Acesso: ocorre quando controle de acesso não é aplicado corretamente, permitindo que invasores explorem essa falha para acessar funcionalidades e/ou dados disponíveis em níveis mais altos, como visualizar arquivos confidenciais, acessar e modificar dados de outros usuários, alterar direitos de acesso, etc. Informações confidenciais ou sensíveis disponíveis em uma URL, por exemplo, podem ser utilizadas por atacantes para obter acesso à contas bancárias de clientes de IFs cujos sistemas de Internet *Banking* possuem tal vulnerabilidade.

6. Configuração Incorreta de Segurança: ocorre quando a configuração de segurança de uma aplicação é incorreta, ocasionando uma configuração incompleta ou insegura. Essa falha de segurança pode resultar de mensagens de erro contendo dados confidenciais, cabeçalhos HTTP configurados incorretamente, etc. Mensagens de erro contendo dados confidenciais podem ocasionar vulnerabilidades como Exposição de Dados Sensíveis e Quebra de Autenticação e Controle de Sessão.
7. *Cross-Site Scripting* (XSS): XSS consiste em vulnerabilidades causadas pela falta de validações de parâmetros de entradas de usuários ou resposta do servidor da aplicação *web*. Este ataque permite que códigos HTML e *scripts* maliciosos sejam inseridos no navegador do usuário. Como o navegador do usuário não consegue identificar a real intenção de tais *scripts*, eles podem acessar cookies, *tokens* de sessão e outras informações sensíveis de clientes retidas pelo navegador, as tornando disponíveis para atacantes.
8. Desserialização Insegura: ocorre quando aplicações desserializam objetos implantados ou alterados por atacantes, modificando a lógica da aplicação ou realizando a execução remota de códigos, ataques de repetição, ataques de injeção e ataques de escalonamento de privilégios. Esta vulnerabilidade pode resultar em ataques DoS em sistemas de Internet *Banking*, tornando indisponível o acesso de clientes às suas respectivas contas bancárias.
9. Usando Componentes com Vulnerabilidades Conhecidas: ocorre quando são usados componentes (como bibliotecas, por exemplo) com vulnerabilidades conhecidas, tornando a aplicação também vulnerável. Tal vulnerabilidade pode resultar em outras como Quebra de Autenticação e Controle de Sessão, Exposição de Dados Sensíveis e Quebra de Controle de Acesso.
10. Registro e Monitoramento Insuficientes: esta vulnerabilidade permite que atacantes continuem violando a aplicação, extraindo, alterando ou destruindo dados. Devido à ineficiência desta vulnerabilidade, outras podem persistir por um considerável período.

As subseções 2.2.1, 2.2.2 e 2.2.3 exemplificam os ataques de Injeção (*Injection*), XSS *Cross Site Scripting* e Quebra de Autenticação e Controle de Sessão, respectivamente. Tais vulnerabilidades são as 3 mais frequentes do ano de 2017 (OWASP, 2017).

2.2.1 Injeção

A Injeção (*Injection*), mais especificamente o Injeção de SQL é um tipo de ataque simples que se baseia na execução de código SQL, tanto comandos de definição de dados – DDL (*create*, *alter*, *drop*) quanto comandos de manipulação de dados – DML (*select*,

update, *insert*, *delete*). Esses comandos são executados através de formulários *web*, que por falhas nas aplicações, resultam no acesso indevido ou em alterações no banco de dados.

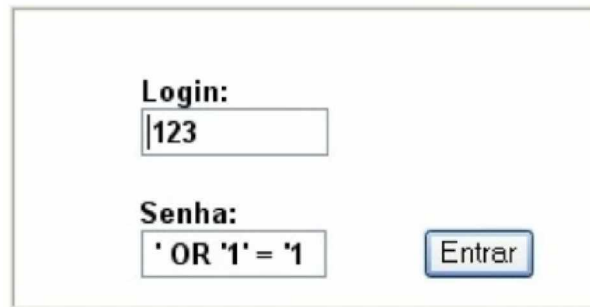


Figura 1 – Exemplo SQL Injection. Adaptado de (MOCELIN; GONCALVES, 2008)

O problema ocorre quando a validação do usuário acontece internamente pela aplicação, que verifica se existe um usuário com o respectivo *login* e senha:

```
SELECT * FROM user_table WHERE login = 'login' AND password = 'password'
```

Geralmente seria digitado o *login* e senha de modo correto, e a aplicação verificaria junto ao banco de dados se os dados digitados estariam corretos. Porém a validação interna abre margem para que sejam aceitos termos que podem confundir a aplicação. Como se pode observar na Figura 1, o código passado no campo senha resulta na seguinte consulta:

```
SELECT * FROM user_table WHERE login = '123' AND password = ' or '1' = '1'
```

Observe que independentemente do *login* e senha informados, o comando passado no campo senha garante que as condições de *login* sejam sempre verdadeiras, o que acarreta o acesso à informações restritas.

2.2.2 Cross Site Scripting

O ataque *Cross-site scripting* (XSS) acontece quando há uma falha na validação de parâmetros de entrada do usuário e resposta do servidor *web*, permitindo que código HTML seja inserido de modo abrupto no navegador do usuário alvo.

```
<% String employeeID = request.getParameter("employeeID"); %>
...
Employee ID: <%= employeeID %>
```

Figura 2 – Exemplo Cross-Site Scripting. Adaptado de Fortify Taxonomy (2016)

Existem vários tipos de ataque XSS, como o ataque XSS refletido, tratado nesse exemplo. Esse ataque é caracterizado por Vogt et al. (2007) pelo uso de engenharia social,

como por exemplo o *phishing*, para ludibriar vítimas. Essa prática faz com que as mesmas enviem requisições ao servidor, que transmite dados XSS e executa o *script* refletido em seus navegadores.

Uma vez que o XSS refletido não é um ataque persistente, o invasor precisa entregar o código malicioso para cada uma das vítimas, e as redes sociais vêm sendo frequentemente usadas para a disseminação desse tipo de ataque.

O segmento de código JSP da Figura 2 lê o ID de um funcionário, “employeeID”, por meio de uma requisição HTTP e o mostra para o usuário.

Este código da Figura 2 funciona corretamente se a variável “employeeID” conter apenas dados alfanuméricos. Se “employeeID” conter valores que incluem metacaracteres ou código-fonte, o código vai ser executado pelo navegador como uma resposta HTTP.

Inicialmente isto pode não parecer uma vulnerabilidade, pois não há motivos aparentes para alguém executar um código malicioso em seu próprio computador. O real perigo é que o atacante pode criar um código malicioso, armazená-lo num endereço *web* e, por meio de engenharia social ou mídias sociais, enviá-lo para suas vítimas. Quando as vítimas clicam no endereço *web*, o conteúdo malicioso do código criado é automaticamente refletido para suas máquinas através da aplicação *web* vulnerável a XSS. Esse mecanismo de exploração de aplicações *web* é conhecido como XSS refletido.

2.2.3 Quebra de Autenticação e Controle de Sessão

Autenticação e controle de sessão inclui todos os aspectos de gerenciamento de autenticação de usuários e gerenciamento de sessões ativas. A autenticação é um aspecto crítico desse processo, mas mesmo os mecanismos sólidos de autenticação podem ser prejudicados por funções de gerenciamento de credenciais defeituosas, incluindo “alteração de senha”, “esqueci minha senha”, “lembre-se da minha senha” entre outras funções relacionadas.

`http://example.com/sale/saleitems?sessionId=586179370&dest=London`

Figura 3 – Exemplo Quebra de Autenticação e Controle de Sessão. Adaptado de [Guide to Cryptography - OWASP \(2016\)](#)

Como exemplo de gerenciamento defeituoso de credenciais, pode-se pensar em um usuário que acabou de realizar a compra de uma viagem de férias. No momento da compra, este usuário está autenticado – com e-mail e senha validados – no site da agência de viagens. Por estar fervoroso com a compra ele decide informar seu amigo, enviando-o por e-mail o endereço *web* observado na Figura 3. Sem saber, ao enviar este e-mail, ele

também está dando sua identificação de sessão. Quando seu amigo acessar o endereço em questão, ele ganhará acesso a sua sessão e cartão de crédito.

Pensando no âmbito de criminosos *web*, por meio de engenharias sociais, conseguir esse tipo de informação da vítima pode ser uma tarefa fácil, como Huluka e Popov (2012) dizem em seu estudo.

2.3 Ferramentas

Ferramentas *black-box* de detecção de vulnerabilidades são ferramentas automáticas de testes de segurança de aplicações *web* sem a necessidade de acessar seu código-fonte. Apesar de limitações como falta de solidez em seus resultados, essas ferramentas possuem uma série de vantagens quando comparadas com os métodos de análise estático de código (BAU et al., 2010). Por exemplo: elas imitam ataques promovidos de hackers, colocando à prova os métodos de detecção de um leque importante de vulnerabilidades. Geralmente o uso de ferramentas de varredura *black-box* está diretamente relacionado a sua habilidade de detectar vulnerabilidades, o que é de extrema utilidade para desenvolvedores, *testers* e pesquisadores.

Considerando as principais vulnerabilidades apresentadas na sessão anterior, foram escolhidas duas ferramentas de varreduras para este trabalho, a saber, *Vega* e *Skipfish* disponíveis em <https://subgraph.com/vega> e <https://tools.kali.org/web-applications/skipfish>, respectivamente. Essas duas ferramentas de varredura *black-box* possuem código aberto e são recomendadas pela OWASP, o que significa que é provido suportes para testadores, analisadores e pesquisadores. Porém, alguns problemas podem ser encontrados em ambas, como a incapacidade de detectar falsos-positivos e ataques diferentes daqueles existentes em suas bases de dados.

A seguir, um breve sumário sobre as duas ferramentas escolhidas.

- *Vega*: plataforma de código aberto escrita em Java que pode ser executada em Linux, Windows e OS X, sua interface pode ser vista na Figura 4. Desenvolvida para realizar busca de vulnerabilidades em testes de segurança de aplicações *web* e sendo capaz de detectar, dentre outras vulnerabilidades, informações confidenciais, *Vega* é capaz de encontrar deficiências bem conhecidas, como *SQL Injection* e *Cross-Site Scripting* (XSS) (BROWN et al., 2005) e (ĐURIĆ, 2014).
- *Skipfish*: ferramenta de verificação de vulnerabilidades de aplicações *web* de código aberto. Ao criar um mapa do site através de buscas recursivas e consultas em testes baseados em dicionários, o profissional *web* pode avaliar a segurança de uma aplicação analisando o resultado final gerado por esta ferramenta. Considerado mais eficiente tanto em velocidade quanto em consumo de recursos comparado às fer-

ramentas com um certo grau de similaridade, como a *w3af*, *Skipfish* vem sendo utilizado para verificações de segurança em instituições financeiras. Sua interface pode ser observada na Figura 5 (VIEIRA et al., 2016).

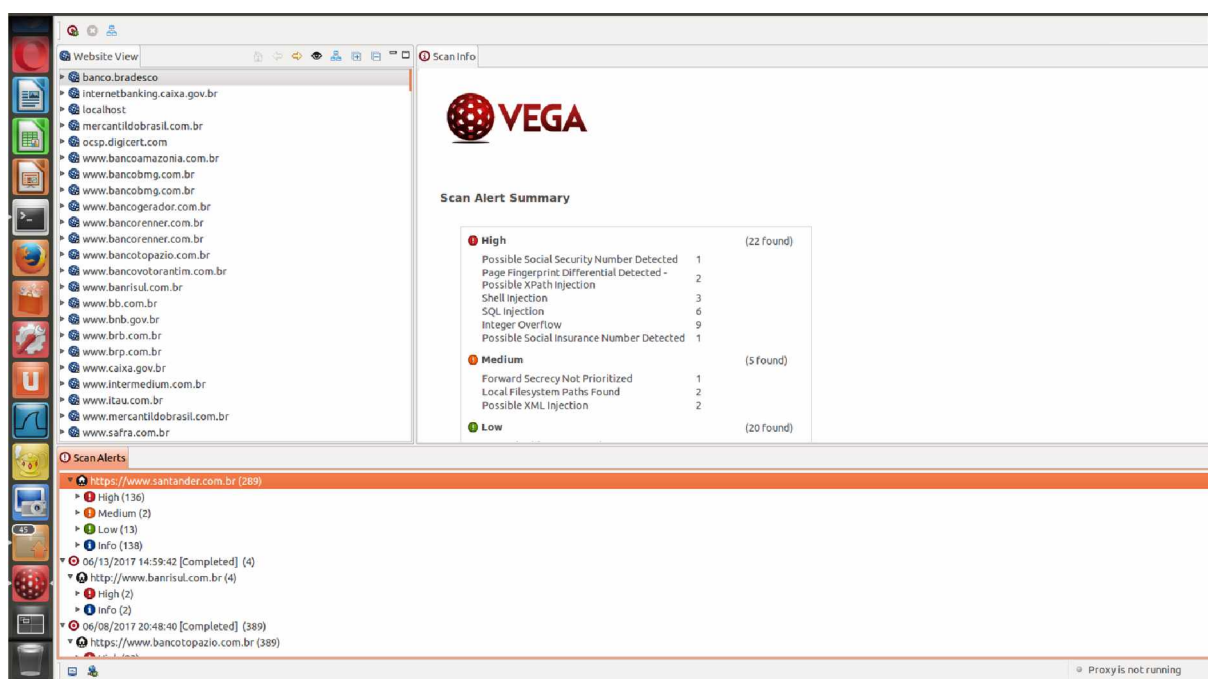


Figura 4 – Exemplo Vega. Imagem produzida pelo autor.

2.4 Trabalhos correlatos

De acordo com Correia et al. (2008), a popularização das redes de computadores, em especial a Internet, gerou um aumento da utilização da mesma para acesso a serviços de Internet *Banking* no Brasil e no mundo. Por aliar comodidade e conveniência para o usuário, além de baixo custo para as instituições, as transações bancárias via Internet já superaram as transações realizadas pelos meios tradicionais.

Porém com sua popularização, o uso de Internet *Banking* trouxe muitas preocupações no que se refere à segurança em diferentes perspectivas, tais como: governamentais, de negócios, tecnológicas e dos próprios bancos (BRAR; SHARMA; KHURMI, 2012).

Com a consolidação das práticas acima mencionadas, notou-se que a Internet ampliou o leque de possibilidades de fraudes no sistema financeiro, levando assim preocupação aos usuários desse serviço *online* de transações bancárias. Lau (2006) baseia-se num estudo de caso para identificar tendências de ataques à Internet *Banking* sob o contexto social e tecnológico. Além disso, traz como resultados recomendações para prevenir e conter ataques de acordo com as perspectivas dos usuários finais, do provedor de serviço e dos atacantes.

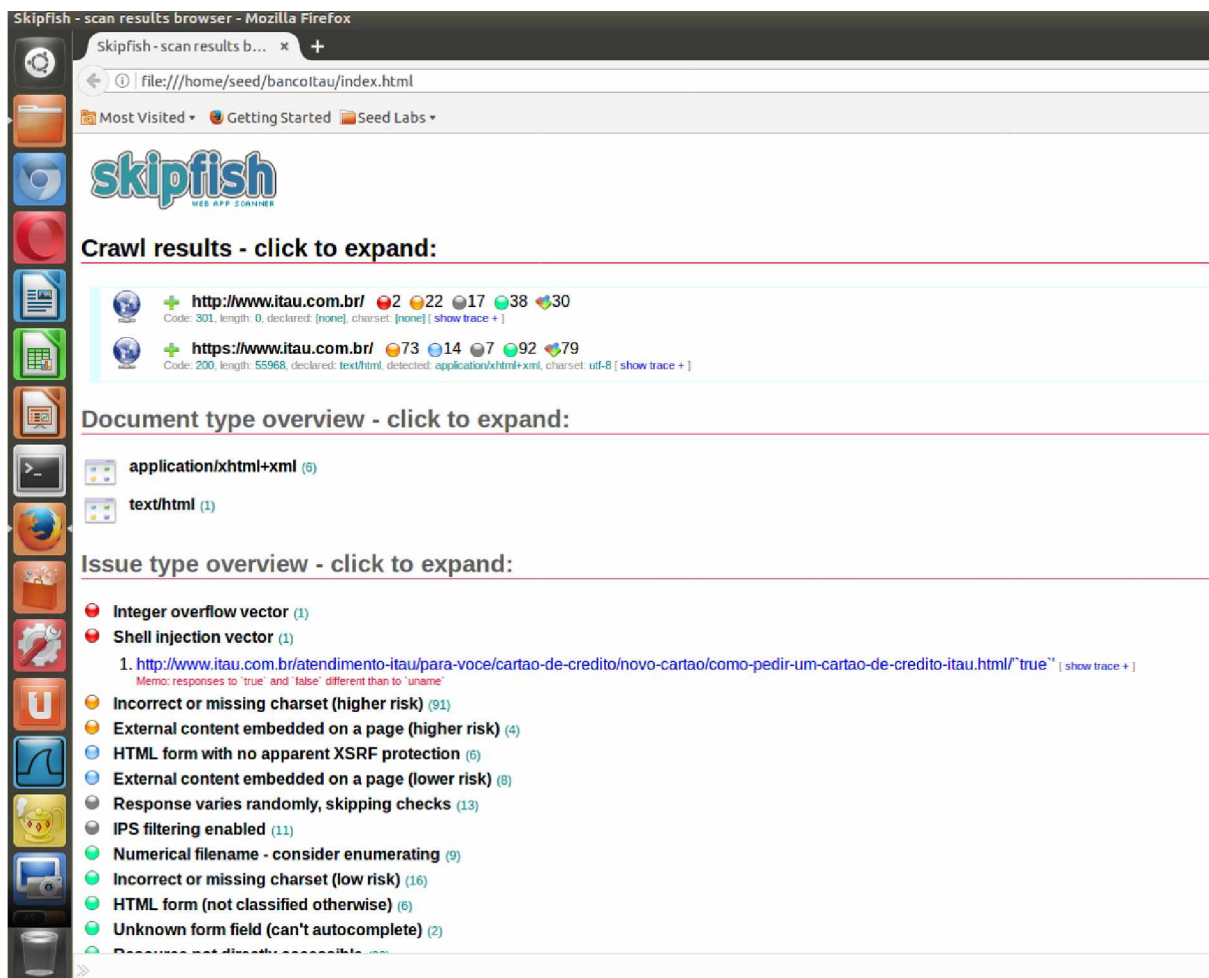


Figura 5 – Mapa site Skipfish. Imagem produzida pelo autor.

Através de análise estatística de dados sobre de ataque, dos impactos causados e seus principais alvos, o estudo de [Gallao e Junior \(2016\)](#) foca na avanço da segurança para utilização do serviço de Internet *Banking*. Aborda ataques a servidores de nomes (DNS), que respondem erroneamente requisições de um determinado cliente e como sites falsos podem ser utilizados em ataques.

Por outro lado, o trabalho de [Correia et al. \(2008\)](#) tem como objetivo levantar mecanismos de segurança utilizados em sistemas de Internet *Banking* das maiores instituições brasileiras. O trabalho também apresenta um estudo detalhado sobre a implementação de alguns desses mecanismos, constatando vulnerabilidades provenientes da utilização de práticas não recomendadas nas suas elaborações.

Em outra vertente, [Bau et al. \(2010\)](#) analisam ferramentas *black-box* de detecção de vulnerabilidades em aplicações *web* e recomendam as principais ferramentas e as respectivas vulnerabilidades cobertas. Questões como: “Quais vulnerabilidades são testadas pela ferramenta?”, “Quão representativos são os testes de vulnerabilidades realizados pelas ferramentas?” e “Quão efetivo é a ferramenta?” são respondidas em seu trabalho. O

autor mensura o tempo de execução e o tráfego de rede de cada ferramenta, além de testar suas respectivas detecções de vulnerabilidades e performance com falso-positivos.

O trabalho de [Antunes e Vieira \(2015\)](#) propõe uma abordagem para avaliação comparativa de ferramentas de detecção de vulnerabilidades para *web services*. Por meio de duas métricas, a abordagem proposta especifica todos os componentes e passos necessários para realizar avaliações comparativas. Estas métricas são:

- Proporção de vulnerabilidades detectadas corretamente pelo número de vulnerabilidades detectadas;
- Proporção de vulnerabilidades corretamente identificadas para o número de todas as vulnerabilidades conhecidas.

Um aspecto-chave é que a abordagem proposta pelo autor é genérica e pode ser usada para especificar comparações diferentes para diferentes domínios de aplicativos e tipos de vulnerabilidades.

Com o objetivo de estudar a efetividade das ferramentas e tentar identificar similaridades entre vulnerabilidades em *web services*, o trabalho de [Vieira, Antunes e Madeira \(2009\)](#) comparou quatro ferramentas de varredura. Os resultados obtidos mostraram que ferramentas diferentes tendem a detectar vulnerabilidades diferentes. Também foi observado que há um alto grau de ocorrência de falso-positivos (vulnerabilidades detectadas que na realidade não existem) e uma baixa cobertura (vulnerabilidades existentes, mas não detectadas), mostrando a limitação das ferramentas de verificação na detecção de vulnerabilidades.

Já o trabalho de [Fonseca et al. \(2014\)](#) estuda as deficiências de códigos-fonte que geram as principais vulnerabilidades de segurança em aplicações web – *Cross-Site Scripting* (XSS) e Injeção de SQL (SQLi), por exemplo. Ao detalhar a distribuição e o processo de análise de falhas de *software*, bem como a classificação de cada vulnerabilidade, o objetivo de tal pesquisa é entender falhas recorrentes que estão por trás da maioria das vulnerabilidades encontradas em aplicações *web*, levando em consideração suas relações com diferentes linguagens de programação de tipagem forte, como Java, C# e VB. Para entender a relevância dessas vulnerabilidades para criminosos, seu trabalho também analisa alguns códigos usados para explorá-las.

A comparação dos trabalhos anteriores mostra que [Bau et al. \(2010\)](#), [Antunes e Vieira \(2015\)](#) e [Vieira, Antunes e Madeira \(2009\)](#) compararam ferramentas de verificação de vulnerabilidades, o que, de maneira diferente, se assemelha com este estudo em relação aos objetivos buscados. Os estudos de [Lau \(2006\)](#), responsável por realizar uma análise dos fatos que propiciam fraudes a *Internet Banking*, e de [Gallao e Junior \(2016\)](#) e [Correia et al. \(2008\)](#) que abordam mecanismos de segurança utilizados em *Internet Banking*, se

relacionam a este trabalho no sentido buscar abordagens mais seguras para o uso de sistemas bancários *online*. Por fim, o estudo de [Fonseca et al. \(2014\)](#) sobre falhas por trás de vulnerabilidades encontradas, classifica tais vulnerabilidades, assim como o presente trabalho se propõe a fazer.

3 Desenvolvimento

Neste capítulo será apresentado e discutido o método de desenvolvimento do estudo considerando cada fase do processo. Inicialmente será detalhada a metodologia utilizada e, então, uma análise dos resultados obtidos.

3.1 Metodologia

A metodologia utilizada neste trabalho se baseou nos métodos mais utilizados na literatura correlata e segue os passos, abaixo, conforme esquematizado na Figura 6:

1. Escolha das ferramentas;
2. Seleção das instituições financeiras (IF);
3. Coleta das vulnerabilidades de segurança dos sistemas das IF;
4. Análise dos resultados obtidos.

O critério central para escolha das ferramentas é a disponibilidade, ou seja, elas devem ser de fácil acesso e gratuitas. Seguindo esse critério, duas foram escolhidas: *Vega* e *Skipfish*.

Tais ferramentas, além de seguirem o critério central, também possuem alto grau de popularidade, sendo frequentemente utilizadas para detectar vulnerabilidades presentes na lista da OWASP (SAGAR et al., 2018), (IDRISSI et al., 2017), (CARVALHO et al., 2013) e (GUAMÁN et al., 2017).

Para conhecer e testar a eficiência de cada uma das ferramentas em detectar vulnerabilidades de segurança, foi utilizado o ambiente controlado OWASP Mutillidae (https://www.owasp.org/index.php/OWASP_Mutillidae_2_Project), uma aplicação *web* de código aberto e deliberadamente vulnerável, que simula um ambiente composto por um conjunto de alvos ideais para teste artificial de segurança da informação.

As instituições foram selecionadas a partir da lista das 1454 IFs atuantes no Brasil e classificadas primeiramente por seu patrimônio líquido. Ambas as informações estão disponíveis no *website* do BACEN (BANCO CENTRAL DO BRASIL, 2017).

Após a classificação dos bancos por patrimônio, uma segunda métrica foi utilizada: o número de acessos mensais aos sistemas de Internet *Banking* de cada instituição. Tal informação foi coletada em Outubro de 2017, utilizando o *website* <https://alexa.com>, capaz de medir o tráfego mensal e realizar inferências estatísticas à outros *websites*.

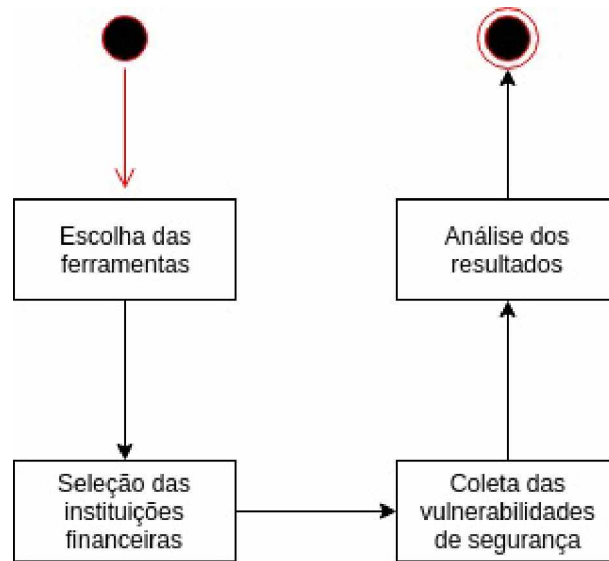


Figura 6 – Esquematização da metodologia utilizada.

Após a aplicação da segunda métrica, as instituições foram separadas nas seguintes categorias, utilizando como base a segmentação do [Jornal Estado de Minas \(2017\)](#):

- B1: com o patrimônio líquido igual ou superior a 10% do PIB brasileiro do ano de 2016;
- B2: com o patrimônio líquido entre 1% e 10% do PIB brasileiro do ano de 2016;
- B3: com o patrimônio líquido abaixo de 1% do PIB brasileiro do ano de 2016.

Em seguida, cada grupo supracitado teve suas instituições financeiras embaralhadas entre si e então foram sorteadas 5 instituições da categoria B1, 6 instituições da categoria B2 e 9 instituições da categoria B3, o que totaliza uma amostra de 20 instituições.

Tendo como base o número de acesso de cada IF no mês supracitado, tal amostra foi dividida em 4 grupos:

- C1: com mais de 10 milhões de acessos;
- C2: entre 1 milhão e 10 milhões de acessos;
- C3: entre 100 mil e 1 milhão de acessos;
- C4: com menos de 100 mil acessos.

Dessa forma, o grupo C1 foi composto por 5 IFs, o grupo C2 por 4 IFs, o grupo C3 por 8 IFs e o grupo C4 por 3 IFs, como observado na Tabela 1.

<i>Instituição Financeira</i>	<i>Categoria Patrimônio Líquido</i>	<i>Grupo Número de Acesso</i>
Instituição H	B1	C1
Instituição B	B1	C1
Instituição E	B1	C1
Instituição L	B1	C1
Instituição S	B1	C1
Instituição K	B2	C3
Instituição T	B2	C2
Instituição P	B2	C4
Instituição R	B2	C2
Instituição M	B2	C2
Instituição F	B2	C3
Instituição A	B3	C3
Instituição C	B3	C3
Instituição Q	B3	C3
Instituição D	B3	C3
Instituição I	B3	C3
Instituição N	B3	C3
Instituição G	B3	C2
Instituição J	B3	C4
Instituição O	B3	C4

Tabela 1 – Instituições Financeiras por Categoria de Patrimônio Líquido e Grupo de Número de Acesso. Extraído de [Banco Central do Brasil \(2017\)](#) e [Jornal Estado de Minas \(2017\)](#)

A motivação para a escolha de um número relativamente pequeno de IFs foi basicamente a demora para rodar todos os testes, o que costuma esgotar os recursos da máquina. Esse processo de escolha teve o apoio do *website* <https://sorteador.com.br/embaralhador>.

Para coletar as vulnerabilidades foi utilizada uma máquina virtual com o sistema operacional Linux SEEDUbuntu12.04 disponibilizado pelo Departamento de Computação da Universidade Syracuse, que cria um ambiente controlado específico para estudo de fatores relacionados à segurança da informação <http://www.cis.syr.edu/wedu/seed/labs.html>.

A análise do número e do nível das vulnerabilidades coletadas foram realizadas considerando apenas o apontamento das ferramentas *Vega* e *Skipfish*, desconsiderando o mérito de falso-positivos, que poderiam ser verificados a partir de disparos de ataques nos *gaps* apontados pelas ferramentas em uma cópia dos sistemas considerados alimentados por uma base de dados genérica, mas para isso, cada IF deveria disponibilizar tal cópia de seu respectivo sistema.

A métrica para classificação das instituições financeiras considerou diretamente a quantidade e a severidade das vulnerabilidades encontradas em cada uma das ferramentas utilizadas. Dessa forma, quanto mais vulnerabilidades, e quanto maior a severidade das mesmas, mais baixa seria a sua classificação.

A classificação por tipo se deu pelo nível de severidade das vulnerabilidades coletadas em cada uma das ferramentas, e encontradas na lista OWASP Top 10 - 2017.

Arelada ao nível de severidade de cada vulnerabilidade, a frequência de cada uma delas também foi considerada. Dessa forma, foi possível saber em quais vulnerabilidades os sistemas das instituições, de um modo geral, mais falham.

3.2 Resultados

A partir dos últimos meses de 2016 iniciou-se a seleção das ferramentas e IFs. Nos primeiros meses de 2017 começaram os primeiros experimentos, cada ferramenta e o ambiente controlado – OWASP Mutillidae foram configurados na máquina virtual com o sistema operacional Linux SEEDUbuntu12.04 para teste de tais ferramentas. No mês de Abril desse mesmo ano foram iniciados os experimentos onde cada ferramenta foi executada para os 20 sistemas de Internet *Banking*. Limitações da rede e de poder de processamento da máquina utilizada, resultaram no atraso na conclusão das varreduras.

Por conta das limitações de rede e poder de processamento da máquina, foi encontrado um tempo médio de 20 horas no *Skipfish* e 23 horas no *Vega* por sistema de Internet *Banking*. O *Skipfish* totalizou assim 400 horas ou 16 dias e 16 horas ininterruptas de execução. Já o *Vega* totalizou então 460 horas ou 19 dias e 4 horas ininterruptas de execução. O processo total soma 860 horas ou 35 dias e 20 horas de varredura ininterrupta.

De modo geral, a ferramenta *Skipfish* detectou uma maior quantidade de ocorrências de vulnerabilidades (Tabela 2), em comparação com a ferramenta *Vega*, a qual detectou uma quantidade maior de vulnerabilidades apenas nos sistemas de Internet *Banking* de 5 IFs: F, L, M, O e P.

<i>Instituição Financeira</i>	<i>Nº de ocorrências de vulnerabilidades</i>		<i>Total</i>
	<i>Vega</i>	<i>Skipfish</i>	
Instituição C	48	339	387
Instituição M	257	41	298
Instituição L	153	82	235
Instituição H	61	104	165
Instituição B	2	143	148
Instituição E	1	129	130
Instituição G	6	98	104
Instituição F	74	4	78
Instituição O	51	16	67
Instituição J	11	55	66
Instituição Q	2	61	63
Instituição I	1	56	57
Instituição N	0	47	47
Instituição P	46	1	47
Instituição T	2	21	23
Instituição R	1	8	9
Instituição S	4	3	7
Instituição K	2	3	5
Instituição A	0	3	3
Instituição D	0	2	2

Tabela 2 – Ocorrências de vulnerabilidades detectadas por ferramenta.

<i>Vulnerabilidades</i>	<i>Vega</i>	<i>Skipfish</i>	<i>Total</i>
Injeção (SQL, OS, LDAP)	231	59	290
Quebra de Autenticação e Controle de Sessão	14	-	14
Exposição de Dados Sensíveis	173	-	173
Entidades Externas XML (XXE)	-	568	568
Quebra de Controle de Acesso	2	-	2
Configuração Incorreta de Segurança	48	578	626
<i>Cross-Site Scripting</i> (XSS)	252	11	263
Desserialização Insegura	-	-	-
Utilização de Componentes Vulneráveis Conhecidos	2	-	2
Registro e Monitoramento insuficientes	-	-	-

Tabela 3 – Top 10 vulnerabilidades de segurança de 2017 segundo OWASP.

A Tabela 3 contempla as vulnerabilidades encontradas por cada ferramenta dentre a lista das Top 10 vulnerabilidades de 2017 de acordo com OWASP. Nessa tabela, pode-se observar que a ferramenta *Vega* conseguiu detectar uma maior diversidade de vulnerabilidades diferentemente da quantidade de ocorrências das mesmas (Tabela 2).

Ambas as ferramentas apresentam classificações detalhadas sobre cada vulnerabilidade. Além da vulnerabilidade em si, cada ferramenta traz consigo informações sobre o nível de criticidade de cada vulnerabilidade e o endereço de ocorrência de cada uma no sistema.

As Figuras 4 e 5 ilustram a interface de cada uma das ferramentas. Ambas classificam as vulnerabilidades encontradas por cores:

- Vermelho: representa vulnerabilidades de alto risco em ambas as ferramentas;
- Laranja: representa vulnerabilidades de médio risco em ambas as ferramentas;
- Verde: representa vulnerabilidades de baixo risco na ferramenta *Vega*;
- Azul: representa vulnerabilidades de baixo risco na ferramenta *Skipfish*;

Os números presentes logo após o nome das vulnerabilidades encontradas se tratam do número de ocorrência de cada uma delas. Os altos números de vulnerabilidades encontradas, como mostrado na Tabela 2, se devem a possíveis repetições dado que as ferramentas podem contabilizar uma mesma vulnerabilidade mais de uma vez, dado que elas classificam vulnerabilidades em um ou mais grupos (alto e médio risco, por exemplo). Isso ocorre, pois, a severidade das vulnerabilidades não foi medida, apenas sua ocorrência.

3.3 Análise dos resultados

Como demonstrado nas tabelas das seções anteriores, os experimentos resultaram em um grande volume de dados. Foram detectadas um total de 1938 vulnerabilidades dos tipos: Configuração Incorreta de Segurança, Entidades Externas XML (XXE), Injeção (*Injection*), *Cross-Site Scripting* (XSS), Exposição de Dados Sensíveis, Quebra de Autenticação e Controle de Sessão, Quebra de Controle de Acesso e Utilização de Componentes Vulneráveis Conhecidos.

Mais da metade das vulnerabilidades encontradas foram do tipo “Configuração Incorreta de Segurança”, com 32,7%, seguida de “XXE” com 29,7%, “Injeção” representou 13,9% das ocorrências, seguida por XSS com 13,7%, “Exposição de Dados Sensíveis” com 9,0%, “Quebra de Autenticação e Controle de Sessão” com 0,7% e os menos frequentes foram “Quebra de Controle de Acesso” e “Utilização de Componentes Vulneráveis Conhecidos”, ambas com 0,1%, como se pode observar na Figura 7.

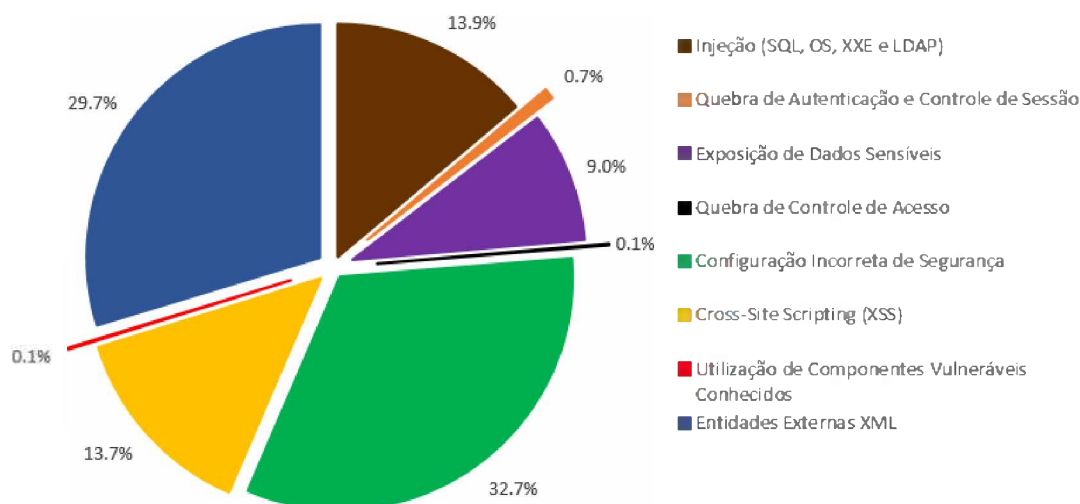


Figura 7 – Distribuição do total de vulnerabilidades encontradas por ferramenta.

3.3.1 Configuração Incorreta de Segurança

A vulnerabilidade “Configuração Incorreta de Segurança”, que possui a sexta posição na classificação da [OWASP \(2017\)](#) foi a mais frequente, conforme informado anteriormente e com 626 ocorrências, foi encontrada em quase todos os sistemas de Internet *Banking* analisados. O sistema que apresentou maior ocorrência dessa vulnerabilidade foi da Instituição C, correspondendo aproximadamente à 29,07%. A Tabela 4 mostra os portais ordenados pelo número de vulnerabilidades desse tipo.

Outras IFs que tiveram quantidades consideráveis de ocorrências desse tipo foram a H e a E, correspondendo a aproximadamente 16,29% e 15,65% das ocorrências, respectivamente. Apesar dos números encontrados, a possível ocorrência de falso-positivos pode reduzi-los.

Um exemplo de “Configuração Incorreta de Segurança” foi uma provável árvore de diretórios no sistema da Instituição H. Um processo de engenharia reversa poderia ser feito para descompilar os códigos da aplicação se essa árvore realmente se referir à diretórios do sistema.

3.3.2 Entidades Externas XML

Sobre a segunda vulnerabilidade mais encontrada, “Entidades Externas XML”, verificou-se que o maior percentual de ocorrência pertence também à Instituição C, com aproximadamente 29,57% do total de 568 ocorrências, seguida pela Instituição B, com 24,82%, e a Instituição Q, com 10,56%. A Tabela 5 mostra os portais ordenados pelo número de vulnerabilidades desse tipo.

Como exemplo de “Entidades Externas XML”, foi observada uma inserção de da-

<i>Instituição Financeira</i>	<i>Configuração Incorreta de Segurança</i>		<i>Total</i>
	<i>Vega</i>	<i>Skipfish</i>	
Instituição C	12	170	182
Instituição H	3	99	102
Instituição E	0	98	98
Instituição G	1	76	77
Instituição I	0	50	50
Instituição L	3	42	45
Instituição M	5	23	28
Instituição P	11	1	12
Instituição J	0	7	7
Instituição F	4	2	6
Instituição O	6	0	6
Instituição N	0	4	4
Instituição S	2	2	4
Instituição B	1	2	3
Instituição K	0	1	1
Instituição Q	0	1	1

Tabela 4 – Instituições Financeiras ordenadas pelo número de vulnerabilidades do tipo Configuração Incorreta de Segurança.

dos em documentos XML no sistema da Instituição E. Tais documentos provavelmente são analisados por um processador XML. Ataques como DDoS poderiam então, ser aplicados a esse sistema se ele não validar os dados recebidos ou aceitá-los de fontes desconhecidas.

Com números semelhantes à vulnerabilidade anterior e ocupando a quarta posição na classificação da [OWASP \(2017\)](#), a “Entidades Externas XML (XXE)” foi a única encontrada exclusivamente pela ferramenta *Skipfish*.

3.3.3 Injeção

O terceiro tipo mais numeroso de vulnerabilidade encontrada foi “Injeção”, com 290 aparições, o que corresponde a 13,9% de todas as ocorrências encontradas. O que chama atenção nessa vulnerabilidade é a sua classificação na [OWASP \(2017\)](#), onde a mesma é a primeira colocada devido a sua alta frequência em aplicações *web* e foi majoritariamente classificada com alto grau de criticidade por ambas as ferramentas utilizadas.

Nas análises efetuadas, tal vulnerabilidade foi encontrada nos sistemas de Internet *Banking* de 11 IFs, onde algumas destas obtiveram quantidades expressivas dessa vulnerabilidade, e a Instituição L liderou em número de ocorrências desse tipo, sendo responsável por 60% das ocorrências encontradas. Esse resultado fez com que o sistema de Internet *Banking* dessa IF seja, dentro do escopo de avaliação proposto, um dos que podem oferecer mais riscos aos usuários no que tange ao grau de vulnerabilidade dos sistemas analisados.

A Instituição H com 14,82% foi a segunda colocada em relação à quantidade encon-

<i>Instituição Financeira</i>	<i>Entidades Externas XML</i>		<i>Total</i>
	<i>Vega</i>	<i>Skipfish</i>	
Instituição C	0	168	168
Instituição B	0	141	141
Instituição Q	0	60	60
Instituição J	0	48	48
Instituição N	0	41	41
Instituição E	0	31	31
Instituição G	0	22	22
Instituição T	0	21	21
Instituição O	0	16	16
Instituição R	0	8	8
Instituição H	0	4	4
Instituição A	0	3	3
Instituição D	0	2	2
Instituição F	0	2	2
Instituição L	0	1	1

Tabela 5 – Instituições Financeiras ordenadas pelo número de vulnerabilidades do tipo Entidades Externas XML.

<i>Instituição Financeira</i>	<i>Injeção</i>		<i>Total</i>
	<i>Vega</i>	<i>Skipfish</i>	
Instituição L	135	39	174
Instituição H	42	1	43
Instituição M	0	17	17
Instituição O	16	0	16
Instituição C	15	0	15
Instituição P	13	0	13
Instituição J	5	0	5
Instituição G	4	0	4
Instituição F	1	0	1
Instituição I	0	1	1
Instituição N	0	1	1

Tabela 6 – Instituições Financeiras ordenadas pelo número de vulnerabilidades do tipo Injeção.

trada desse tipo de vulnerabilidade, seguida pela Instituição M com 5,86% e a Instituição O com 5,51%. A Tabela 6 mostra todos os 11 portais ordenados pelo número de vulnerabilidades desse tipo.

Como exemplo de “Injeção”, foi observada uma URL em uma das JSPs do sistema da Instituição L. Nela pôde-se notar um comando semelhante a uma consulta SQL. Se essa URL é composta por parâmetros, então é possível inserir o comando `' ou '1'='1`. Esse comando retorna todos os registros da tabela, comprometendo a confidencialidade dos dados armazenados pelo sistema.

<i>Instituição Financeira</i>	<i>Cross-Site Scripting</i>		<i>Total</i>
	<i>Vega</i>	<i>Skipfish</i>	
Instituição M	241	1	242
Instituição H	5	0	5
Instituição I	0	5	5
Instituição C	2	1	3
Instituição K	0	2	2
Instituição O	2	0	2
Instituição J	1	0	1
Instituição N	0	1	1
Instituição S	0	1	1
Instituição T	1	0	1

Tabela 7 – Instituições Financeiras ordenadas pelo número de vulnerabilidades do tipo *Cross-Site Scripting*.

3.3.4 *Cross-Site Scripting*

Com 263 aparições, o que corresponde a 13,7% do total de ocorrências encontradas, a vulnerabilidade “*Cross-Site Scripting*” foi a quarta mais frequente, e o Internet *Banking* da Instituição M foi o maior responsável por todas essas ocorrências, com cerca de 92,01% do total, seguida pelas IFs H e I, cada um contribuindo com aproximadamente 1,9% das ocorrências.

Essa vulnerabilidade pode ser exemplificada pela detecção de um JavaScript que se refere à um domínio diferente do sistema da Instituição J. Mesmo que a fonte do script seja de confiança do sistema, um código malicioso pode ser introduzido se o servidor da aplicação estiver comprometido.

Todas as instituições onde essa vulnerabilidade foi verificada podem ser observadas na Tabela 7, que as ordena por número de ocorrências (IFs com o mesmo número de vulnerabilidades foram ordenadas alfabeticamente).

Mesmo os números da “*Cross-Site Scripting*” estando entre os maiores verificados nesse estudo, o fator falso-positivo também incide sobre eles, conforme explicado nas vulnerabilidades anteriores, o que pode tornar os resultados desta vulnerabilidade menos críticos, apesar de muito recorrente.

3.3.5 Exposição de Dados Sensíveis

Sobre a quinta vulnerabilidade mais comum dentre os resultados obtidos, “Exposição de Dados Sensíveis”, verificou-se o maior percentual na Instituição F, com cerca de 37.57% de todas as 173 aparições. Presente também em 11 IFs, como pode ser observado na Tabela 8, os resultados dessa vulnerabilidade apontaram ainda as IFs O e P como o segundo e terceiro maiores responsáveis pelo volume de aparições da mesma, com 15,6%

e 12,71% respectivamente.

<i>Instituição Financeira</i>	<i>Exposição de Dados Sensíveis</i>		<i>Total</i>
	<i>Vega</i>	<i>Skipfish</i>	
Instituição F	65	0	65
Instituição O	27	0	27
Instituição P	22	0	22
Instituição C	17	0	17
Instituição L	13	0	13
Instituição H	11	0	11
Instituição M	11	0	11
Instituição J	3	0	3
Instituição Q	2	0	2
Instituição B	1	0	1
Instituição S	1	0	1

Tabela 8 – Instituições Financeiras ordenadas pelo número de vulnerabilidades do tipo Exposição de Dados Sensíveis.

Exemplificando essa vulnerabilidade, foi observada uma sequência de números na Instituição O que se assemelha com a estrutura de um identificador pessoal. Se essa sequência tiver sido cifrada e decifrada automaticamente pelo banco de dados quando inserida e consultada, abre precedente para que uma falha de Injeção de SQL a recupere.

Não diferente das demais, podem ocorrer também para esta vulnerabilidade, possíveis resultados falso-positivos, porém mesmo tendo uma quantidade de resultados menor que as demais, cerca de 9% do total de aparições, ela é considerada a terceira vulnerabilidade mais frequente segundo [OWASP \(2017\)](#).

3.3.6 Quebra de Autenticação e Controle de Sessão

Presente em 9 IFs, a sexta vulnerabilidade mais frequente nos testes e segunda mais frequente na classificação da [OWASP \(2017\)](#) foi “Quebra de Autenticação e Controle de Sessão”, com 14 ocorrências, o que corresponde a 0,7% de todas as vulnerabilidades encontradas.

A IF o com maior número de aparições de “Quebra de Autenticação e Controle de Sessão” foi a Instituição F, com cerca de 21,43%, seguida pelas IFs J, K e L, responsáveis por cerca de 14,28% cada e as IFs E, I, R, S e T, com participação de 7,14% cada. Os bancos com o mesmo número de tal vulnerabilidade são ordenados alfabeticamente, como pode ser observado na Tabela 9.

Um exemplo de “Quebra de Autenticação e Controle de Sessão” foi uma provável falta de implementação de um validador de senhas conhecidas no sistema da Instituição S. A falta desse tipo de validação implica em fragilidade da aplicação se códigos maliciosos forem utilizados para validar credenciais.

<i>Instituição Financeira</i>	<i>Quebra de Autenticação e Controle de Sessão</i>		<i>Total</i>
	<i>Vega</i>	<i>Skipfish</i>	
Instituição F	3	0	3
Instituição J	2	0	2
Instituição K	2	0	2
Instituição L	2	0	2
Instituição E	1	0	1
Instituição I	1	0	1
Instituição R	1	0	1
Instituição S	1	0	1
Instituição T	1	0	1

Tabela 9 – Instituições Financeiras ordenadas pelo número de vulnerabilidades do tipo Quebra de Autenticação e Controle de Sessão.

3.3.7 Quebra de Controle de Acesso e Utilização de Componentes Vulneráveis Conhecidos

Com 2 aparições cada e, portanto, classificadas por ordem alfabética, as vulnerabilidades “Quebra de Controle de Acesso” e “Utilização de Componentes Vulneráveis Conhecidos” foram a sétima e oitava vulnerabilidades mais comuns, respectivamente. As aparições da primeira foram todas na Instituição C. Já a segunda foi igualmente numerosa nas IFs F e G. As Tabelas 10 e 11 mostram os portais ordenados pelo número de vulnerabilidades desse tipo.

Como exemplo de “Quebra de Controle de Acesso”, foi observado um parâmetro que possivelmente é um número de conta em uma URL da Instituição C. Se realmente esse dado for o número de uma conta, um invasor pode modificar o parâmetro no navegador para o número de conta do alvo. Se não houver uma devida validação, o atacante ganha acesso a tal conta, podendo, então consultar saldo, copiar dados etc.

Já “Utilização de Componentes Vulneráveis Conhecidos” pode ser exemplificado pela detecção de prováveis falhas em bibliotecas de validação no sistema de Internet *Banking* da Instituição G. Se tais falhas realmente estiverem presentes em tais bibliotecas, o sistema da IF em questão possivelmente se torna suscetível à vulnerabilidade “Quebra de Autenticação e Controle de Sessão”.

Ocupando a quinta posição na classificação OWASP (2017), a “Quebra de Controle de Acesso” não ocupa nenhum dos polos de alta ou baixa frequência. Por outro lado, a “Utilização de Componentes Vulneráveis Conhecidos” ocupa a nona posição, sendo esse um baixo nível de frequência dentro do conjunto considerado. Isso acontece pela facilidade de identificar e corrigir tal deficiência nos portais.

<i>Instituição Financeira</i>	<i>Quebra de Controle de Acesso</i>		<i>Total</i>
	<i>Vega</i>	<i>Skipfish</i>	
Instituição C	2	0	2

Tabela 10 – Instituições Financeiras ordenadas pelo número de vulnerabilidades do tipo Quebra de Controle de Acesso.

<i>Instituição Financeira</i>	<i>Utilização de Componentes Vulneráveis Conhecidos</i>		<i>Total</i>
	<i>Vega</i>	<i>Skipfish</i>	
Instituição F	1	0	1
Instituição G	1	0	1

Tabela 11 – Instituições Financeiras ordenadas pelo número de vulnerabilidades do tipo Utilização de Componentes Vulneráveis Conhecidos.

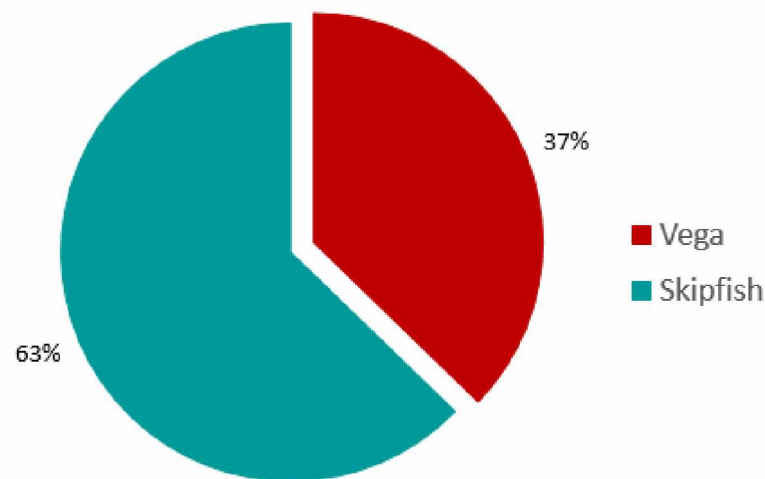


Figura 8 – Vulnerabilidades encontradas por ferramenta.

3.3.8 Vulnerabilidades por ferramenta

A ferramenta *Vega* conseguiu detectar 70% das vulnerabilidades da lista Top 10 da OWASP, já o *Skipfish* detectou 40% das vulnerabilidades presentes na mesma lista. A execução dos procedimentos resultou na detecção de 1938 ocorrências de vulnerabilidades, sendo cerca de 37% dessas no *Vega* e 63% no *Skipfish*, como visto na Figura 8.

Nenhuma das ferramentas detectou as vulnerabilidades “Desserialização Insegura” e “Registro e Monitoramento Insuficientes”. Tais deficiências estão entre as menos frequentes, segundo OWASP (2017), ocupando a oitava e a décima posição, respectivamente.

3.3.9 Discussão

A Tabela 12 mostra a quantidade de cada vulnerabilidade por sistema de cada IF. Cada vulnerabilidade recebeu um código, sendo eles:

- A1: Injeção;
- A2: Quebra de Autenticação e Controle de Sessão;
- A3: Exposição de Dados Sensíveis;
- A4: Entidades Externas XML;
- A5: Quebra de Controle de Acesso;
- A6: Configuração Incorreta de Segurança;
- A7: *Cross-Site Scripting* (XSS) e;
- A9: Utilização de Componentes Vulneráveis Conhecidos.

<i>Instituição Financeira</i>	<i>Vulnerabilidades</i>								<i>Total</i>
	<i>A1</i>	<i>A2</i>	<i>A3</i>	<i>A4</i>	<i>A5</i>	<i>A6</i>	<i>A7</i>	<i>A9</i>	
Instituição L	174	2	13	1	0	45	0	0	235
Instituição H	43	0	11	4	0	102	5	0	165
Instituição M	17	0	11	0	0	28	242	0	298
Instituição O	16	0	27	16	0	6	2	0	67
Instituição C	15	0	17	168	2	182	3	0	387
Instituição P	13	0	22	0	0	12	0	0	47
Instituição J	5	2	3	48	0	7	1	0	66
Instituição G	4	0	0	22	0	77	0	1	104
Instituição F	1	3	65	2	0	6	0	1	78
Instituição I	1	1	0	0	0	50	5	0	57
Instituição N	1	0	0	41	0	4	1	0	47
Instituição K	0	2	0	0	0	1	2	0	5
Instituição S	0	1	1	0	0	4	1	0	7
Instituição E	0	1	0	31	0	98	0	0	130
Instituição T	0	1	0	21	0	0	1	0	23
Instituição R	0	1	0	8	0	0	0	0	9
Instituição Q	0	0	2	60	0	1	0	0	63
Instituição B	0	0	1	141	0	3	0	0	145
Instituição A	0	0	0	3	0	0	0	0	3
Instituição D	0	0	0	2	0	0	0	0	2
Total	290	14	173	568	2	626	263	2	1938

Tabela 12 – Instituições Financeiras ordenadas pelo nível de criticidade de seus portais de Internet *Banking*.

Os sistemas de Internet *Banking* das 20 IFs estudadas nesse trabalho foram classificados seguindo o parâmetro de quantidade de vulnerabilidades mais e menos severas, de acordo com cada uma das ferramentas.

Cada portal foi ordenado seguindo o quão críticas são as vulnerabilidades encontradas em relação ao seu número de acessos e o seu patrimônio líquido, conforme dados sintetizados nas Tabelas 13 e 14, respectivamente.

É importante salientar que, a análise realizada considerou os valores discretos de patrimônio líquido e de número de acessos específicos de cada IF, e não os respectivos grupos criados.

<i>Instituição Financeira</i>	<i>Criticidade de Vulnerabilidades</i>				<i>Grupo Número de Acessos</i>
	<i>Alta</i>	<i>Média</i>	<i>Baixa</i>	<i>Total</i>	
Instituição L	185	37	13	235	C1
Instituição H	53	95	17	165	C1
Instituição C	36	180	171	387	C3
Instituição E	33	92	5	130	C1
Instituição O	23	18	26	67	C4
Instituição P	22	4	21	47	C4
Instituição M	19	25	254	298	C2
Instituição J	15	51	0	66	C4
Instituição F	7	7	64	78	C3
Instituição G	4	58	42	104	C2
Instituição Q	2	48	13	63	C3
Instituição I	2	43	12	57	C3
Instituição T	2	5	16	23	C2
Instituição K	2	2	1	5	C3
Instituição N	1	28	18	47	C3
Instituição B	1	20	124	145	C1
Instituição S	1	4	2	7	C1
Instituição R	1	0	8	9	C2
Instituição D	0	1	1	2	C3
Instituição A	0	0	3	3	C3
Total	409	718	811	1938	-

Tabela 13 – Instituições Financeiras pelo Número de Vulnerabilidades e Grupo de Número de Acessos. Extraído de [Alexa \(2017\)](#).

Foi feita então, uma análise de Correlação Linear de Pearson entre a quantidade de acessos e o número de vulnerabilidades de alta severidade de cada sistema e aplicada a interpretação de escala qualitativa do coeficiente de correlação de ([GUILFORD; FRUCHTER, 1965](#)):

- Entre -0,2 e 0,2: sem correlação;
- Entre -0,4 e -0,2 ou entre 0,2 e 0,4: baixa correlação;
- Entre -0,7 e -0,4 ou entre 0,4 e 0,7: correlação moderada;
- Entre -0,9 e -0,7 ou entre 0,7 e 0,9: alta correlação, e;

<i>Instituição Financeira</i>	<i>Criticidade de Vulnerabilidades</i>				<i>Categoria Patri. Líquido</i>
	<i>Alta</i>	<i>Média</i>	<i>Baixa</i>	<i>Total</i>	
Instituição L	185	37	13	235	B1
Instituição H	53	95	17	165	B1
Instituição C	36	180	171	387	B3
Instituição E	33	92	5	130	B1
Instituição O	23	18	26	67	B3
Instituição P	22	4	21	47	B2
Instituição M	19	25	254	298	B2
Instituição J	15	51	0	66	B3
Instituição F	7	7	64	78	B2
Instituição G	4	58	42	104	B3
Instituição Q	2	48	13	63	B3
Instituição I	2	43	12	57	B3
Instituição T	2	5	16	23	B2
Instituição K	2	2	1	5	B2
Instituição N	1	28	18	47	B3
Instituição B	1	20	124	145	B1
Instituição S	1	4	2	7	B1
Instituição R	1	0	8	9	B2
Instituição D	0	1	1	2	B3
Instituição A	0	0	3	3	B3

Tabela 14 – Instituições Financeiras pelo Número de Vulnerabilidades e Categoria de Patrimônio Líquido. Extraído de [Banco Central do Brasil \(2017\)](#).

- Entre -1,0 e -0,9 ou entre 0,9 e 1,0: correlação muito alta.

Verificou-se um coeficiente de correlação de 0,73, o que representa uma correlação linear forte e positiva entre elas. Neste caso, se o valor da primeira variável sofrer um aumento, o valor da segunda também aumenta. Em contrapartida, é observado uma correlação positiva baixa de 0,3, entre o total de vulnerabilidades encontradas a quantidade de acessos de cada IF, o que não pode ser considerado uma tendência entre elas.

Foi possível verificar que, sistemas bancários com maior quantidade de acessos tendem a ter mais vulnerabilidades. Uma possível explicação para essa tendência seria o tamanho do portfólio de serviços fornecidos pela IF, que pode aumentar o número de acessos ao seu sistema e torná-lo mais suscetível a vulnerabilidades. Essa mesma tendência não é observada no que se refere ao número de vulnerabilidades de alta criticidade.

Já a análise de Correlação Linear de Pearson mostra uma correlação de 0,27 entre o total das vulnerabilidades encontradas nos sistemas de Internet *Banking* analisados e o patrimônio líquido de suas respectivas IFs (Tabela 14), o que representa uma correlação linear positiva fraca entre elas. Similarmente, é observada uma correlação positiva e moderada de 0,4 entre o patrimônio líquido e a quantidade de vulnerabilidades de alta criticidade.

<i>Categoria</i>	<i>Média da quantidade de Vulnerabilidades</i>	<i>Média de Vulnerabilidades Severas</i>
B1	136,4	54,6
B2	76,6	8,8
B3	88,4	9,2

Tabela 15 – Categoria por Média de Vulnerabilidades Totais e Vulnerabilidades Severas.

Tais dados mostram, então, que é possível identificar uma baixa tendência entre o patrimônio líquido da IF e o total de vulnerabilidades encontradas e uma tendência moderada entre o patrimônio líquido e o número de vulnerabilidades de alta criticidade.

Como podemos verificar, as IFs E, H e L, que estão no grupo B1, estão entre as primeiras colocadas, tanto no que se refere ao nível de criticidade quanto ao total de vulnerabilidades presentes em seus sistemas. Isso não pôde ser observado nos sistemas das IFs B e S, que também pertencem ao mesmo grupo, mas que tiveram seus sistemas posicionados dentre os que podem oferecer menos riscos aos usuários.

Equitativamente, alguns sistemas de IFs pertencentes aos grupos B2 e B3 apresentaram um número maior de vulnerabilidades severas, o que pode representar riscos aos usuários, sendo a Instituição L (do grupo B1) responsável pelo sistema com mais vulnerabilidades severas, e a Instituição A (do grupo B3) responsável pelo sistema com menos vulnerabilidades desse tipo.

Evidenciando a discrepância de vulnerabilidades dentre os grupos analisados, a Tabela 15 mostra uma média aritmética da quantidade de vulnerabilidades totais encontradas em cada grupo dos bancos e a quantidade de vulnerabilidades e severas em cada um deles. O grupo B1 se mostrou o grupo tanto com a maior média de vulnerabilidades críticas quanto com a maior média de vulnerabilidades totais, seguindo pelo grupo B3 e B2, sendo esse último o grupo cujas IFs tem os sistemas com menos vulnerabilidades severas.

4 Conclusão

Esse trabalho investigou as vulnerabilidades presentes nos sistemas de Internet *Banking* de 20 IFs atuantes no Brasil, seguindo o modelo de trabalho de Lau (2006), Correia et al. (2008), Bau et al. (2010), Fonseca et al. (2014), Antunes e Vieira (2015) e Gallao e Junior (2016), afim de verificar uma correlação entre o patrimônio líquido das instituições e o número de vulnerabilidades em seus respectivos sistemas.

O objetivo foi concluído com êxito, e a análise do material resultante da investigação permitiu negar a hipótese, isto é, nenhum indicativo mostrou que exista uma forte correlação entre o patrimônio líquido de uma IF e a segurança de seu portal de Internet *Banking*, porém foi possível encontrar uma correlação forte entre o número de acessos e a alta gravidade das vulnerabilidades encontradas nos respectivos sistemas. Assim sendo, um banco maior ou com um maior acesso ao seus serviços pela Internet, não necessariamente possui um sistema de Internet *Banking* mais seguro (principalmente no que se refere ao grau de severidade das mesmas) que um banco menor ou com menos acessos.

Em um trabalho futuro, espera-se investigar um maior número de sistemas, contemplando um maior número de bancos, bem como utilizar novas ferramentas e verificar a ocorrência de falso-positivos. Também seria interessante elaborar uma nova perspectiva que permitisse formular uma nova hipótese baseada na linguagem de programação utilizada no desenvolvimento ou diversidade de serviços oferecidos pelos sistemas, visto que quanto mais diversificado os serviços oferecidos, mais propensos a ataques o sistema se torna.

Referências

- AL-SANEA, M. S.; AL-DARASEH, A. A. Security evaluation of saudi arabia's websites using open source tools. In: IEEE. *Anti-Cybercrime (ICACC), 2015 First International Conference on*. [S.l.], 2015. p. 1–5. Citado na página 13.
- ALEXA. *Website Traffic, Statistics and Analytics*. 2017. Disponível em: <<https://www.alexa.com/siteinfo/>>. Acesso em: 21 set. 2017. Citado 2 vezes nas páginas 7 e 38.
- ALMEIDA, M. B. Aplicação de ontologias em segurança da informação. *Diretoria da Prodemege*, 2007. Citado na página 13.
- ALMUKAYNIZI, M. et al. Predicting cyber threats through hacker social networks in darkweb and deepweb forums. In: ACM. *Proceedings of the 2017 International Conference of The Computational Social Science Society of the Americas*. [S.l.], 2017. p. 12. Citado na página 14.
- ANTUNES, N.; VIEIRA, M. Assessing and comparing vulnerability detection tools for web services: Benchmarking approach and examples. *IEEE Transactions on Services Computing*, IEEE, v. 8, n. 2, p. 269–283, 2015. Citado 3 vezes nas páginas 11, 22 e 41.
- AVELLAREDUARTE. *Internet no Brasil 2017 (estatísticas)*. 2017. Disponível em: <<http://www.avellareduarte.com.br/fases-projetos/conceituacao/demandas-do-publico/pesquisas-de-usuarios-atividades-2/dados-sobre-o-publico-alvo/internet-no-brasil-2017-estatisticas/>>. Acesso em: 16 jul. 2017. Citado na página 12.
- BANCO CENTRAL DO BRASIL. *Maiores Bancos e o Consolidado do Sistema Financeiro Nacional*. 2017. Disponível em: <<https://www3.bcb.gov.br/ifdata/>>. Acesso em: 15 jul. 2017. Citado 4 vezes nas páginas 7, 24, 26 e 39.
- BAU, J. et al. State of the art: Automated black-box web application vulnerability testing. In: IEEE. *Security and Privacy (SP), 2010 IEEE Symposium on*. [S.l.], 2010. p. 332–345. Citado 4 vezes nas páginas 19, 21, 22 e 41.
- BRAR, T. P. S.; SHARMA, D.; KHURMI, S. S. Vulnerabilities in e-banking: A study of various security aspects in e-banking. *International Journal of Computing & Business Research*, 6, 127, v. 132, 2012. Citado na página 20.
- BREI, V. A. Antecedentes e conseqüências da confiança do consumidor final em trocas relacionadas com empresas de serviço: um estudo com o usuário de internet banking no brasil. 2001. Citado na página 11.
- BROWN, G. G. et al. Analyzing the vulnerability of critical infrastructure to attack and planning defenses. In: *Emerging Theory, Methods, and Applications*. [S.l.]: INFORMS, 2005. p. 102–123. Citado na página 19.
- CARVALHO, F. R. de et al. Vulnerabilidades em aplicações web. *RE3C-Revista Eletrônica Científica de Ciência da Computação*, v. 8, n. 1, 2013. Citado na página 24.

CORREIA, M. A. et al. Segurança em internet banking. In: *XIII Simposio Brasileiro em Seguranca da Informacao e Sistemas Computacionais (SBseg)*, Gramado-RS. [S.l.: s.n.], 2008. p. 291–299. Citado 5 vezes nas páginas 11, 20, 21, 22 e 41.

DELOITTE. *Pesquisa da FEBRABAN mostra que transações com mobile banking cresceram 138% em 2015*. 2016. Disponível em: <<https://www2.deloitte.com/br/pt/footerlinks/pressreleasespage/Febraban-2016.html>>. Acesso em: 16 jul. 2017. Citado na página 11.

DIGITAL, C. Bancos perdem US\$ 1,8 milhão por falhas de segurança no internet banking. 2017. Disponível em: <http://sis-publique.convergenciadigital.com.br/cgi/cgilua.exe/sys/start.htm?from_info_index=301&infoid=45455&sid=18>. Acesso em: 29 abr. 2017. Citado na página 11.

ESTADÃO. Cibercrime faz bancos perderem R\$ 1,8 bilhão. 2015. Disponível em: <<http://link.estadao.com.br/noticias/cultura-digital,cibercrime-faz-bancos-perderem-r-18-bilhao,10000028721>>. Acesso em: 05 jun. 2016. Citado na página 11.

FEBRABAN. *O que é um banco*. 2011. Disponível em: <<http://www.febrabanopportunidades.com.br/oquee.asp>>. Acesso em: 12 jun. 2016. Citado na página 10.

FONSECA, J. et al. Analysis of field data on web security vulnerabilities. *IEEE transactions on dependable and secure computing*, IEEE, v. 11, n. 2, p. 89–100, 2014. Citado 3 vezes nas páginas 22, 23 e 41.

FORTIFY TAXONOMY. *Cross-Site Scripting: Inter-Component Communication*. 2016. Disponível em: <https://vulncat.fortify.com/en/detail?id=desc.dataflow.java.cross_site_scripting_inter_component_communication>. Acesso em: 20 jun. 2016. Citado 2 vezes nas páginas 6 e 17.

GALLAO, R. V.; JUNIOR, A. M. Segurança do internet banking no brasil. *Revista Tecnológica da Fatec Americana*, v. 1, n. 1, p. 15, 2016. Citado 4 vezes nas páginas 12, 21, 22 e 41.

GUAMÁN, D. et al. Implementation of techniques and owasp security recommendations to avoid sql and xss attacks using j2ee and ws-security. In: IEEE. *Information Systems and Technologies (CISTI)*, 2017 12th Iberian Conference on. [S.l.], 2017. p. 1–7. Citado na página 24.

GUIDE TO CRYPTOGRAPHY - OWASP. *Top 10 2013-A2-Broken Authentication and Session Management*. 2016. Disponível em: <https://www.owasp.org/index.php/Top_10_2013-A2-Broken_Authentication_and_Session_Management>. Acesso em: 20 jun. 2016. Citado 2 vezes nas páginas 6 e 18.

GUILFORD, J.; FRUCHTER, B. Fundamental statistics in psychology and education, mcgraw-hill. *New York*, 1965. Citado na página 38.

HOLE, K. J.; MOEN, V.; TJOSTHEIM, T. Case study: Online banking security. *IEEE Security & Privacy*, IEEE, v. 4, n. 2, p. 14–20, 2006. Citado na página 10.

HULUKA, D.; POPOV, O. Root cause analysis of session management and broken authentication vulnerabilities. In: IEEE. *Internet Security (WorldCIS), 2012 World Congress on*. [S.l.], 2012. p. 82–86. Citado na página 19.

IDRISSI, S. et al. Performance evaluation of web application security scanners for prevention and protection against vulnerabilities. *International Journal of Applied Engineering Research*, v. 12, n. 21, p. 11068–11076, 2017. Citado na página 24.

JORNAL ESTADO DE MINAS. *BC divulga lista de bancos conforme segmentação do SFN*. 2017. Disponível em: <https://www.em.com.br/app/noticia/economia/2017/01/30/internas_economia,843548/bc-divulga-lista-de-bancos-conforme-segmentacao-do-sfn.shtml>. Acesso em: 27 fev. 2017. Citado 3 vezes nas páginas 7, 25 e 26.

LAU, M. *Análise das fraudes aplicadas sobre o ambiente Internet Banking*. Tese (Doutorado) — Universidade de São Paulo, 2006. Citado 3 vezes nas páginas 20, 22 e 41.

MAKINO, Y.; KLYUEV, V. Evaluation of web vulnerability scanners. In: IEEE. *Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), 2015 IEEE 8th International Conference on*. [S.l.], 2015. v. 1, p. 399–402. Citado na página 14.

MOCELIN, S. S.; GONCALVES, P. R. R. Funcionamento e controles de injeção de sql em formulário de autenticação, focado em um estudo de caso. *Anais SULCOMP*, v. 4, 2008. Citado 2 vezes nas páginas 6 e 17.

NETTO, A. d. S.; SILVEIRA, M. A. Pinheiro da. Gestão da segurança da informação: fatores que influenciam sua adoção em pequenas e médias empresas. *JISTEM: Journal of Information Systems and Technology Management*, Universidade de São Paulo, v. 4, n. 3, 2007. Citado na página 13.

OWASP. *OWASP Top 10 Application Security Risks - 2017*. 2017. Disponível em: <https://www.owasp.org/index.php/Top_10-2017_Top_10>. Acesso em: 16 set. 2017. Citado 7 vezes nas páginas 15, 16, 30, 31, 34, 35 e 36.

PIRES, P. J.; FILHO, B. Alves da C. Fatores do índice de prontidão à tecnologia (tri) como elementos diferenciadores entre usuários e não usuários de internet banking e como antecedentes do modelo de aceitação de tecnologia (tam). *RAC-Revista de Administração Contemporânea*, SciELO Brasil, v. 12, n. 2, 2008. Citado na página 11.

QUEIROZ, A.; KEEGAN, B.; MTENZI, F. Predicting software vulnerability using security discussion in social media. Dublin Institute of Technology, 2017. Citado na página 14.

RODRIGUEZ, L. G. A. et al. Analysis of vulnerability disclosure delays from the national vulnerability database. In: *Workshop de Segurança Cibernética em Dispositivos Conectados (WSCDC_SBRC)*. [S.l.: s.n.], 2018. v. 1. Citado na página 14.

SAGAR, D. et al. Studying open source vulnerability scanners for vulnerabilities in web applications. *HIOAB JOURNAL*, v. 9, n. 2, p. 43–49, 2018. Citado na página 24.

SOLMS, R. V.; NIEKERK, J. V. From information security to cyber security. *computers & security*, Elsevier, v. 38, p. 97–102, 2013. Citado na página 14.

TECMUNDO. Internet banking é alvo principal de ataques hackers no Brasil, diz estudo. 2015. Disponível em: <<https://www.tecmundo.com.br/crime-virtual/79752-internet-banking-alvo-principal-ataques-hackers-brasil-diz-estudo.htm>>. Acesso em: 04 ago. 2017. Citado na página 12.

TJØSTHEIM, T.; MOEN, V. Vulnerabilities in online banks. p. 105, 2005. Citado na página 11.

DURIĆ, Z. Waptt-web application penetration testing tool. *Advances in Electrical and Computer Engineering*, v. 14, n. 1, 2014. Citado na página 19.

VIEIRA, M.; ANTUNES, N.; MADEIRA, H. Using web security scanners to detect vulnerabilities in web services. 2009. Citado 2 vezes nas páginas 11 e 22.

VIEIRA, T. M. P. et al. Análise de aplicações no sector financeiro: vulnerabilidades e mitigações. Instituto Universitário de Lisboa, 2016. Citado na página 20.

VOGT, P. et al. Cross site scripting prevention with dynamic data tainting and static analysis. v. 2007, 2007. Citado na página 17.