Pós-Graduação em Ciência da Computação

Carlo Marcelo Revoredo da Silva

Piracema.io: um sistema especialista baseado em heurística direcionada por características estáticas e dinâmicas para a detecção gradual de phishing direcionados

Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
http://cin.ufpe.br/~posgraduacao

Recife

2020

Carlo Marcelo Revoredo da Silva

Piracema.io: um sistema especialista baseado em heurística direcionada por características estáticas e dinâmicas para a detecção gradual de phishing direcionados

Este trabalho foi apresentado à Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do título de Doutor em Ciência da Computação.

Área de Concentração: Engenharia de Software e Linguagens de Programação

Orientador: Vinícius Cardoso Garcia
Coorientador: Eduardo Luzeiro Feitosa

Recife
2020

Catalogação na fonte
Bibliotecária Mariana de Souza Alves CRB4-2105

S586p Silva, Carlo Marcelo Revoredo da
Piracema.io: um sistema especialista baseado em heurística direcionada por características estáticas e dinâmicas para a detecção gradual de phishing direcionados / Carlo Marcelo Revoredo da Silva. – 2020.
161 f.: il., fig., tab.

Orientador: Vinícius Cardoso Garcia.
Tese (Doutorado) – Universidade Federal de Pernambuco. CIn, Ciência da Computação. Recife, 2020.
Inclui referências e apêndices.

1. Engenharia de Software e Linguagens de Programação. 2. Phishing. 3. Proteção de marcas. 4. Heurística baseada em características. I. Garcia, Vinícius Cardoso (orientador). II. Título.

005.1 CDD (22. ed.) UFPE-CCEN 2020-137

Carlo Marcelo Revoredo da Silva

"**Piracema.io:** um sistema especialista baseado em heurística direcionada por características estáticas e dinâmicas para a detecção gradual de phishing direcionados"

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Doutor em Ciência da Computação.

Aprovado em: 08/07/2020.

Orientador: Vinícius Cardoso Garcia

Coorientador: Eduardo Luzeiro Feitosa

BANCA EXAMINADORA

Prof. Dr. Sérgio Castelo Branco Soares
Centro de Informática (CIn) / UFPE

Profa. Dra. Edna Dias Canedo
Departamento de Ciência da Computação (CIC) / UnB

Prof. Dr. Diego de Freitas Aranha
Aarhus University, Dinamarca

Prof. Dr. Eduardo James Pereira Souto
Instituto de Computação (IComp) / UFAM

Profa. Dra. Carla Taciana Lima Lourenço Silva Schuenemann
Centro de Informática (CIn) / UFPE

Dedico este trabalho a minha família e amigos próximos que foram porto seguro perante as dificuldades durante este percurso tão difícil nos últimos meses. Dias melhorias virão.

AGRADECIMENTOS

Primeiramente a Deus, a minha mãe Arimar, meu pai José, meu irmão Adriano e meu filho Daniel, são os alicerces na minha vida. Ao meu orientador Vinícius, um cara inteligente e preciso. Ao meu Co-orientador flamenguista Eduardo, inteligente e sempre disponível, gratidão pelos ensinamentos, muito honrado por essa parceira. A professora Tayana por suas contribuições e simpatia. Aos membros da Banca, as professoras Edna e Carla e os professores Sérgio, Diego e Eduardo Souto, por suas disponibilidades e ensinamentos. A Universidade de Pernambuco (UPE) por fornecer toda a infraestrutura necessária para a realização desse estudo. Ao amigo e parceiro em pesquisas Péricles da UFRPE, um exemplo de pessoa. Aos meus orientandos de Iniciação Científica Mateus e Muryllo, que me ajudaram com a mão na massa. Ao Gerson da Tempest, por sua disponibilidade e sugestões. Ao pessoal do DefesaDigital e *PhishTank Community (mailing list)* pelo conhecimento disseminado e dicas valiosas. Aos amigos de batalha no doutorado Jackson e Flávio por todo apoio e incentivo. Aos demais familiares, amigos e colegas de trabalho que sempre me deram muita força e acreditaram nos meus objetivos.

RESUMO

Não é incomum encontrar estudos que investigaram abordagens que analisam características estáticas (i) e dinâmicas (ii) de uma página Web para detectar ataques de *phishing* através de predição. Em (i), as características não consideram aspectos como tempo ou de terceiros envolvidos, o que propõem benefícios em termos de desempenho, contudo, tem baixa precisão devido ao escopo ser limitado porque não considera a página em execução. Diante disso, estudos são impulsionados a adotarem uma abordagem (ii) que considera aspectos temporais e contexto da atuação do *phishing*, oferecendo maior eficiência nas soluções anti-phishing. Todavia, a análise em (ii) tem limitações devido a mudança contínua de conteúdo. Não obstante, alguns *phishing*, a exemplo dos direcionados a uma marca específica, possuem um alto grau de fidedignidade com a página genuína, portanto, ao mesmo tempo que aumentam a exploração da suscetibilidade do usuário final, a riqueza em detalhes dificulta a predição de algo malicioso. Há também o desafio em identificar quais características são mais e menos relevantes devido as novas tendências, considerando o cenário dinâmico de atuação do *phishing*. Diante disso, o estudo faz uma investigação sobre a relevância, relação e similaridade entre (i) e (ii) através de uma regressão logística sobre amostras de *phishing* reais. O intuito é contribuir com os rumos para novas abordagens baseadas em predição, considerando aspectos como fidedignidade, ofuscação, propagação, sazonalidade e volatilidade, que podem dificultar a identificação dos padrões em páginas maliciosas. A proposta apresenta-se como uma solução complementar, ou seja, atuar em conjunto com soluções já existentes que são baseadas em lista de bloqueio. Adicionalmente, também é pretendido disponibilizar um mecanismo de lista de permissão baseado em um protocolo de autenticação e autorização para fortalecer a proposta contra falsos positivos. O presente estudo propõe um sistema especialista como mecanismo *anti-phishing* baseado em regras. Sua detecção é tida como gradual porque sua máquina de inferência processa as regras em profundidade gradativa, visando reduzir o custo computacional e ser menos invasivo no contexto da privacidade durante o processamento da predição. A proposta demonstra maior eficiência em *phishing* com maior riqueza em detalhes, e com isso, se caracterizando como uma alternativa de proteção de marcas. Como prova de conceito (PoC), o estudo faz uma avaliação de possíveis falsos positivos e negativos da proposta quando a mesma é submetida à amostras reais de *phishing* e de páginas genuínas. Também é pretendido avaliar o caráter complementar da proposta com as soluções nativas em navegadores Web e também avaliar os impactos positivos atingidos pela análise gradual da proposta. Como resultado, além dos dados quantitativos, essa pesquisa também realizou uma análise qualitativa da proteção, identificando contribuições e limitações.

Palavras-chaves: Phishing. Proteção de marcas. Heurística baseada em características. Sistema Especialista. Classificador baseado em Árvore de regras.

ABSTRACT

It is not uncommon to find studies that have investigated approaches that analyze static (i) and dynamic (ii) features of a Web page to detect *phishing* attacks through prediction. In (i), the features do not consider aspects such as time or third parties involved, which propose benefits in terms of performance, however, it has low precision due to the scope being limited because it does not consider the page in execution. Therefore, studies are encouraged to adopt an approach (ii) that considers temporal aspects and context of *phishing* 's performance, offering greater efficiency in anti-phishing solutions. However, the analysis in (ii) has limitations due to the continuous change of content. Nevertheless, some *phishing*, like those targeting a specific brand, have a high degree of reliability with the genuine page, therefore, while increasing the exploitation of the end user's susceptibility, the wealth of details makes it difficult the prediction of something malicious. There is also the challenge of identifying which features are more and less relevant due to new trends, considering the dynamic scenario of *phishing*. Therefore, the study investigates the relevance, relationship and similarity between (i) and (ii) through a logistic regression on real *phishing* samples. The aim is to contribute with directions for new approaches based on prediction, considering aspects such as reliability, obfuscation, propagation, seasonality and volatility, which can make it difficult to identify patterns on malicious pages. The proposal presents itself as a complementary solution, that is, to act in conjunction with existing solutions that are based on a blacklist. In addition, it is also intended to provide a whitelist mechanism based on an authentication and authorization protocol to strengthen the proposal against false positives. The present study proposes a specialist system as a rule-based *anti-phishing* mechanism. Its detection is considered to be gradual because its inference machine processes the rules in gradual depth, aiming to reduce the computational cost and be less invasive in the context of privacy during the prediction processing. The proposal demonstrates greater efficiency in *phishing* with greater richness in details, and with that, it is characterized as an alternative for brand protection. As a proof of concept (PoC), the study makes an assessment of possible false positives and negatives of the proposal when it is submitted to real samples of *phishing* and genuine pages. It is also intended to assess the complementary character of the proposal with native solutions in Web browsers and also to evaluate the positive impacts achieved by the gradual analysis of the proposal. As a result, in addition to the quantitative data, this research also carried out a qualitative analysis of protection, identifying contributions and limitations.

Keywords: Phishing. Brand protection. Feature-based heuristics. Expert System. Decision Tree Classifier.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

De acordo com (KONDUTO, 2019), metade dos golpes relacionados a cartões de crédito são aplicados através de ataques de *phishing*, e protagonizados em ambientes de *e-commerce*. No mesmo estudo, é descrito que no Brasil, o *e-commerce* em 2018 sofria um ataque de *phishing* a cada 6.5 segundos. Em 2019, o índice de tentativas teve um intervalo maior, 7 segundos. Apesar da aparente diminuição, não houve uma redução efetiva porque a quantidade de transações em 2019 foi superior à 2018. Além disso, é possível observar que ataques de *phishing* também exploram aspectos sazonais para enganar suas vítimas. Como exemplo, pode-se citar os sites que, durante o distanciamento social causado pelo COVID-19, supostamente sugerem disponibilizar o auxilio emergencial do banco Caixa Econômica Federal[1].

Um dado intrigante levantado pela KnowBe4[2], empresa especializada na conscientização de ataques de *phishing* em ambientes corporativos, revela que 37.9% dos funcionários de variados setores em diversas empresas são propensos a cair em ataques dessa natureza. Outro fato curioso é que muitos profissionais associam o ataque de *phishing* como uma investida bem sucedida apenas a usuários pouco familiarizados com a tecnologia. Contudo, o GitLab[3] fez um experimento com seus funcionários e constatou que 20% deles forneceram dados sensíveis.

É notório que na literatura se associe ataques de *phishing* à engenharia social, ou seja, essencialmente sustentados na exploração do elo humano no processo. Um exemplo são as fraudes que exploram um nicho de proprietários de dispositivos móveis, propagando transações fraudulentas por SMS[4]. Todavia, recentemente foi registrada uma vulnerabilidade no navegador *Internet Explorer* em que possibilitava fraudadores compartilharem links maliciosos, obtendo controle do dispositivo com privilégios administrativos. Também há casos em que o subterfúgio é explorado não apenas pelo aspecto humano. Além disso, o *phishing* em questão foi combinado com ataques de *malware*, a exemplo de incidentes com *ransomware*[5], demonstrando que os ataques de *phishing* vem se aperfeiçoando continuamente.

O estudo de (ABDALLAH; MAAROF; ZAINAL, 2016) destaca que, apesar das organizações mantenedoras dos serviços oferecerem meios seguros no tráfego, a ação causa pouco ou nenhum impacto aos ataques de *phishing* porque esses atuam diretamente nos usuários finais. Além disso, as fraudes recentemente vem apresentando maior riquezas em detalhes visuais e textuais (**fidedignidade**), baseiam-se em estratégias de curta atividade (**volatilidade**), são publicadas na Web em grande volume (**propagação**), são cada vez mais sensíveis a eventos anuais (**sazonalidade**) e, por fim, adotam manobras que dificultam a percepção do crime pelo crivo humano (**ofuscação**), conforme descrito no estudo de (SILVA; FEITOSA; GARCIA, 2019a).

[1] Golpe do auxílio emergencial faz vítimas em todo o Brasil. https://glo.bo/2Whmxgd
[2] https://bit.ly/37ef31h
[3] https://siliconangle.com/2020/05/21/gitlab-runs-phishing-test-employees-20-handing-credentials/
[4] PayPal SMS scams – don't fall for them! https://bit.ly/3bGEUAF
[5] CVE-2020-0674. https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-0674

1.1 MOTIVAÇÃO

Conforme destacam alguns estudos, (ALEROUD; ZHOU, 2017; VAYANSKY; KUMAR, 2018; CARTA et al., 2019), os *phishing* evoluíram, parando de atuar de forma genérica (atacando todo e qualquer alvo) para atuar em alvos direcionados, em ambientes onde ocorrem transações financeiras ou o volume de recursos financeiros é maior. Tipicamente, tais ambientes estão relacionados com uma organização, que por sua vez está associada a uma marca. Uma marca é uma identidade visual que pode sugerir autoria sobre determinados recursos de uma organização. Portanto, através da engenharia social, o atacante traça perfis das partes envolvidas e adota certas estratégias, como registrar domínios com palavras-chave, prática conhecida como *Cybersquatting*, ou faz uso de termos homográficos e combinações com jogo de palavras, manobra conhecida como *Typosquatting*, conforme pontuado em (SILVA; FEITOSA; GARCIA, 2019a), tornando o *phishing* mais fidedigno, ou seja, direcionado a um determinado contexto. Exemplos de *phishing* direcionados são o *Spear* phishing e o *SMiShing*.

Além das características textuais, os *phishing* também se enviesam em características visuais, como o cadeado HTTPS (tunelamento) e similaridade entre o *template* legítimo e o fraudado (KHONJI; IRAQI; JONES, 2013). Essa engenharia demonstra que muitos desses *phishing* são direcionados a explorar a suscetibilidade através da forja da fidedignidade. Portanto, geralmente, a fidedignidade é baseada nos detalhes de uma marca, aumentando as chances de uma exploração bem sucedida.

Todavia, estudos *anti-phishing*: (ALKHOZAE; BATARFI, 2011; CHELLIAH; ARUNA, 2014; KIRDA; KRUGEL, 2005; NARESH; SAGAR; REDDY, 2013; CHAUDHRY; CHAUDRY; RITTENHOUSE, 2016; ALEROUD; ZHOU, 2017; ABURROUS et al., 2008), norteiam suas heurísticas com base na ausência de detalhes textuais e visuais, ou seja, quanto menor a fidedignidade do *phishing*, maiores as chances de sucesso na detecção. Na mesma linha, a detecção é comprometida à medida que o *phishing* tem maior fidedignidade, sendo um raciocínio semelhante ao crivo humano. Diante disso, a presente Tese tem como motivação apresentar uma heurística que responda incidentes à ataques de *phishing* direcionados. Considerando que boa parte das propostas da literatura apresentam limitações em *phishing* rico em detalhes, a proposta visa contribuir atenuando essa ineficiência.

Atualmente, existem muitas iniciativas na literatura que são baseadas em técnicas de Inteligência Artificial (IA) para detectar padrões maliciosos presentes na página, a exemplo da Aprendizagem de Máquina (do inglês, *Machine Learning*, ML) ou Sistemas Especialistas (do inglês, *Expert Systems*, ES). Contudo, diante a tanto volume de informações e recursos compartilhados pela Web, teoricamente, em termos computacionais, talvez seja inviável processar um grande número de características. Portanto, decidir as características candidatas é uma atividade que merece cautela. Além disso, aspectos da privacidade precisam ser preservados.

1.2 OBJETIVOS

Esse estudo de doutorado tem como objetivo propor um mecanismo *anti-phishing* capaz de identificar e mitigar ataques de *phishing* direcionados, com foco na fidedignidade de uma página genuína, através de uma heurística baseada em regras, a fim de fazer predição e informar ao usuário final durante a tentativa de exibir uma determinada página. Considerando que as soluções atuais tendem a ter maiores dificuldades em detectar *phishing* direcionados, a proposta visa atenuar essa lacuna apresentando-se como um mecanismo adicional em conjunto com as soluções existentes.

Vale ressaltar que o processo de predição de página *phishing* é realizado através de uma *heurística baseada em regras*, que adota uma *análise gradual* em sua execução. A abordagem é denominada gradual porque executa em profundidade, processando o mínimo possível de características para obter um veredito, atenuando o custo computacional e intrusão à privacidade do usuário final. Adicionalmente, também é apresentado um experimento controlado como prova de conceito sobre a eficiência da solução, comparando a mesma com soluções nativas existentes em navegadores Web da atualidade.

1.2.1 Objetivos específicos

Com relação aos objetivos específicos, foram definidas atividades que complementam a solução proposta. Primeiramente, foi necessário realizar uma investigação no cenário real de atuação de *phishing* sobre comportamentos existentes, no intuito de obter **características estáticas e dinâmicas** presentes em páginas maliciosas. Posteriormente, os resultados obtidos fundamentam as decisões de uma **árvore baseada em regras** como modelo para predição de *phishing* durante a navegação na Web.

Tal modelo atua no ato da tentativa de renderização da página para o usuário final e é regido por uma **análise gradual**, que percorre prioritariamente as características mais relevantes e menos invasivas ao conteúdo, preocupando-se com a desempenho e privacidade. Também foi proposto um **serviço externo baseado em lista branca**, que atua como um repositório que registra entradas válidas para aplicar exceções ao acesso de páginas previamente cadastradas.

1.3 METODOLOGIA

A metodologia aplicada nesta Tese é ilustrada na Figura 1, que segmenta o processo em 3 etapas. A primeira etapa (**1**) é um levantamento teórico baseado em evidências e que possui dois artefatos (1.1 e 1.2). O artefato 1.1 representa uma revisão literária *ad-hoc*, que resulta em analisar trabalhos correlatos presentes na literatura e fundamentar a construção de uma taxonomia de *phishing* direcionados. Já o artefato 1.2 remete a elaboração de duas *pesquisas empíricas* que observam comportamentos estáticos e dinâmicos. Tais abordagens possuem fundamentações baseadas na investigação *ad-hoc* do artefato 1.1.

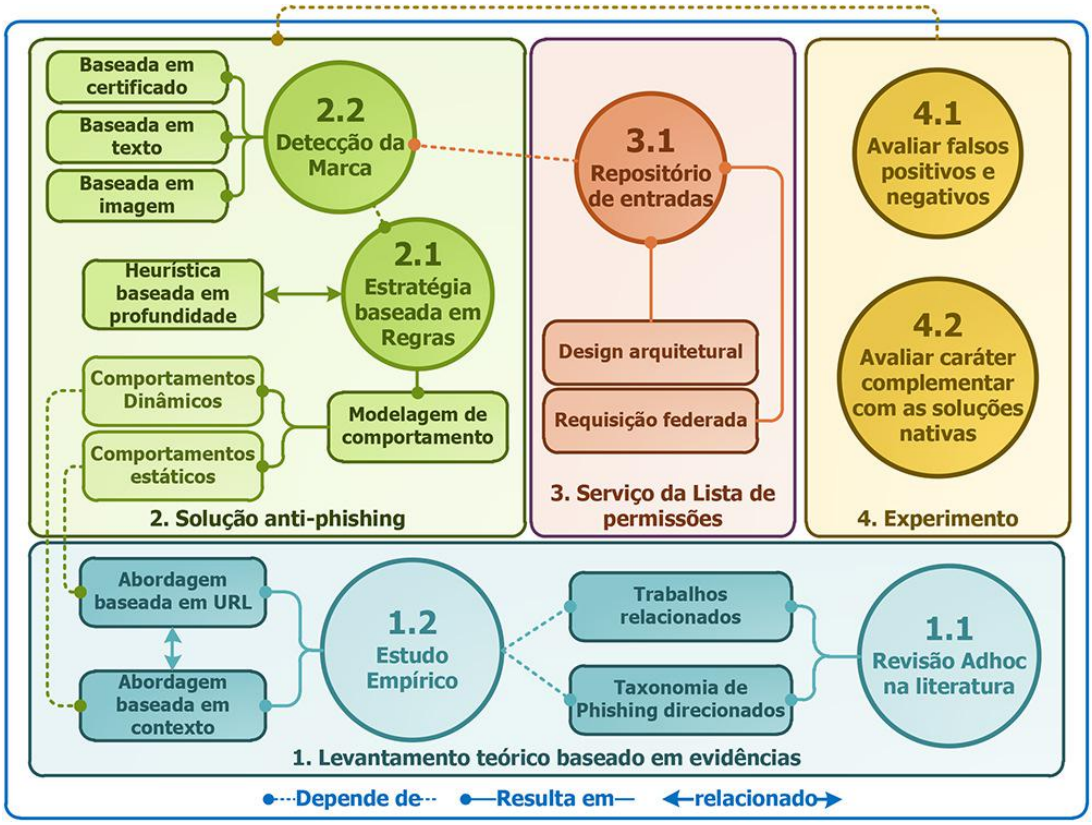

Figura 1 – Estrutura da solução proposta

Dando sequência, a etapa (**2**) descreve o desenvolvimento do mecanismo *anti-phishing*, em que a preocupação será identificar comportamentos suspeitos na camada mais superficial do ataque, que muitas vezes tem como ponto de partida a navegação na Web do usuário final. Esse ambiente majoritariamente é representado pelas atividades realizadas em um navegador Web (2.1), do qual é embasado pelos resultados obtidos no artefato 1.2. Também é proposto um reconhecimento de marcas (2.2), que utiliza estratégias distintas.

Dando continuidade, na etapa (**3**), a solução representada por um serviço de provimento de lista de permissões, esse se enviesa em propor um modelo de repositório de entradas (3.1), que apresenta um protocolo para o processo de requisição federada, estabelecendo uma troca segura de leitura de entradas previamente registras ou de escrita para novas inclusões, alimentando assim a lista de permissão. Por fim, na etapa (**4**) é descrito o planejamento de dois experimentos controlados, o primeiro visa analisar o tratamento de falsos positivos e negativos da proposta, já o segundo visa identificar as contribuições da proposta quando a mesma é submetida a uma atuação em conjunto com as soluções nativas dos navegadores Web.

1.4 ESTRUTURA DO DOCUMENTO

A presente Tese de doutorado segue estruturada tendo como base a metodologia descrita na Figura 1. No Capítulo 1 é definido o cenário de atuação, descrevendo uma síntese da problemática e pretensões da Tese. No Capítulo 2 é apresentada a contextualização resultante da

revisão *ad hoc* (1.2) e seus respectivos artefatos resultantes. No Capítulo 3 é apresentado um estudo empírico (1.2) baseado em regressão logística, seu intuito é observar comportamentos estáticos e dinâmicos do *phishing*, resultando em um conjunto de características candidatas para um modelo de classificação. No Capítulo 4 é apresentado o arcabouço conceitual da proposta, apresentando uma solução *Client-side* (2) e *Server-side* (3). No Capítulo 5 é definido um protótipo para uma abordagem prática do modelo conceitual anteriormente apresentado, apresentando as decisões e desafios no processo de desenvolvimento. No Capítulo 6 é apresentada uma prova de conceito (PoC) do protótipo desenvolvido, analisando os objetivos específicos alcançados. Por fim, no Capítulo 7 é apresentada as considerações finais com base no estado atual, relatando os resultados até o momento e o planejamento dos trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Este Capítulo trata da fundamentação teórica desta Tese, apresentando os conceitos básicos para seu entendimento e os trabalhos relacionados existentes na literatura. Primeiro, é feita a apresentação dos conceitos e aspectos sobre o *phishing*, seu funcionamento e estratégias de atuação. Depois são apresentadas as abordagens existentes para detecção de *phishing*, especialmente as baseadas em heurísticas que serão utilizadas nesta Tese. Em seguida, uma discussão sobre sistemas especialista é feita, para explicar como será empregado na Tese. Por fim, são apresentados os trabalhos relacionado a *phishing* que consideram as estratégias de atuação apresentadas, incluindo uma discussão.

2.1 CONCEITOS BÁSICOS

A fraude é uma ação que representa qualquer tipo de ato intencional ou deliberado de privação de bens ou dinheiro através da astúcia, engano ou outros atos injustos (ACFE, 2018). Dentre os diferentes tipos de fraude, *phishing* é a de maior destaque. A proposta apresenta uma proteção contra fraudes em recursos computacionais na Web e, apesar da existência de diversas modalidades, a pesquisa, em seu estado atual tem como escopo os ataques de *phishing*.

Por sinal, a *Kaspersky* aponta que o *phishing* é um ataque caracterizado por tentativas fraudulentas contra usuários da Internet (KASPERSKY, 2014). Nesse cerne, o atacante cria uma página falsa que apresenta-se como um ambiente confiável, induzindo suas vítimas à submeterem dados sensíveis, como credencias de acesso a um determinado serviço genuíno (MOHAMMAD; THABTAH; MCCLUSKEY, 2015). De acordo com relatórios publicados pela *Anti-Phishing Working Group* (APWG) nos últimos anos[1], *phishing* é um dos ataques que vem ganhando destaque, inclusive no Brasil.

2.1.1 Phishing

O *phishing* é o tipo de fraude por meio da qual um golpista tenta obter dados pessoais e financeiros de um usuário, pela utilização combinada de meios técnicos e engenharia social (CERT.BR, 2012b; MITNICK; SIMON, 2003). Um ponto importante sobre *phishing* é que sua investida emerge de forma situacional, como, por exemplo, o aumento expressivo de ataques de *phishing* no Brasil devido a liberação de saques do FGTS de contas inativas[2]. Não obstante, o *phishing* também impulsiona o crime cibernético, pois apresenta-se como uma ferramenta para propagação de *malware* na Internet (CERT.BR, 2012a). Um bom exemplo é o *ransomware WannaCry*, que gerou problemas em escala global.

[1] https://apwg.org/resources/apwg-reports/

[2] Ataques de *phishing* em massa visam roubar informações do FGTS: https://goo.gl/nMS3Wi

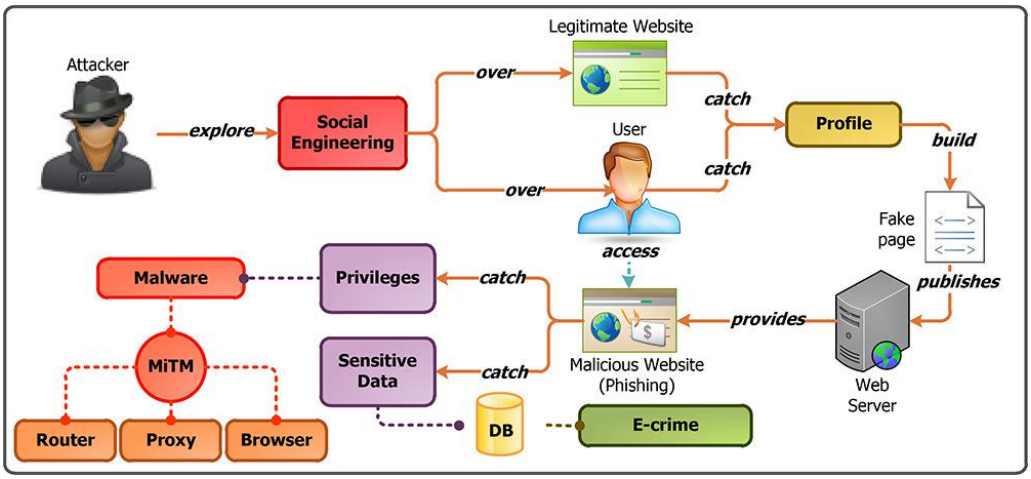

Figura 2 – Fluxograma padrão de um ataque convencional de *phishing*

Os ataques de *phishing* podem assumir diferentes formas. Por exemplo, em sua forma mais básica, os atacantes tentam extrair informações confidenciais diretamente de uma vítima por meio de uma resposta por e-mail ou por um formulário online. No entanto, o *phishing* não é apenas uma técnica para roubar informações de uma vítima individual. Os ataques de *phishing* também podem ser usados para desativar sistemas por meio de URLs maliciosas ou anexos de e-mail infectados ou até mesmo como um meio de obter acesso prolongado a dispositivos, sistemas e/ou bancos de dados para realizar campanhas sofisticadas de reconhecimento e exfiltração (VERIZON, 2015). A Figura 2 ilustra o ataque de *phishing* convencional.

Tudo se inicia através da engenharia social, onde o atacante explora a confiança de suas vítimas. Através dela, ele observa aspectos do serviço, como uma senha de acesso com 6 ou 8 dígitos, e também dos usuários finais, como analisar meios de autenticação, resultando em um perfil de cada alvo envolvido (KHONJI; IRAQI; JONES, 2013). Esse perfil se traduz como um conjunto de comportamentos que serve de subsídio para a construção de um site malicioso. Quanto maior a qualidade do perfil maior será a fidedignidade, diminuindo as suspeitas.

Uma vez que a exploração do *phishing* tem o cerne focado no contexto da exploração de aspectos humanos, analisar tais comportamentos demonstra que o ecossistema de atuação do *phishing* possui atores bem definidos, como os usuários, organizações e os recursos compartilhados na Web (URI). Uma análise sobre os comportamentos é ilustrada na Figura 3.

Percebe-se, na Figura 3, que as URI, quando exploradas, podem propiciar, através da URL ou conteúdo, a obtenção de informações sobre a morfologia da URL, bem como o conteúdo a ser exibido ao usuário final. Além disso, existem terceiros, que são organizações com papéis distintos, como prestadores de hospedagem, serviços de DNS, plataformas de denúncia de *phishing* (que alimentam listas de bloqueio) e navegadores Web. Por fim, usuários finais oferecem informações que podem alimentar a fidedignidade de um *phishing*. Por exemplo, através da engenharia social, o atacante analisa uma senha de acesso com 6 ou 8 dígitos solicitada por um serviço mantenedor, resultando em um conjunto de perfis (KHONJI; IRAQI; JONES, 2013). Além disso, com o ataque bem sucedido, o mal-intencionado armena as informações em uma

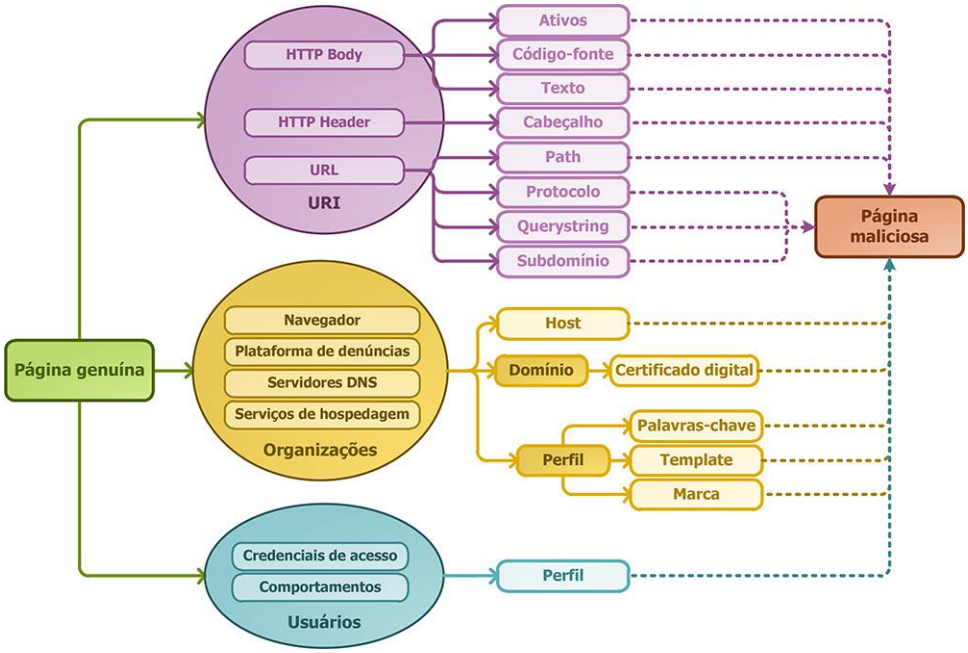

Figura 3 – Aspectos sensíveis ao contexto de atuação do *phishing*

base para posteriormente fazer uso ou mesmo vender tais informações sensíveis no mercado de crimes cibernéticos (e-crime). Outra possibilidade é obter privilégios do dispositivo da vítima, podendo combinar o ataque com *malware* para interceptar ou espionar atividades da vítima, atuando como homem do meio (MiTM).

2.1.1.1 *Phishing* Direcionado

Fazendo uma análise da linha de tempo sobre *phishing*, percebe-se que o ataque foi se moldando de acordo com as tendências e oportunidades inerentes ao seu cenário de atuação. Em outras palavras, os atacantes perceberam que quanto mais valioso é o alvo, mais recursos (principalmente dinheiro) eles arrecadam em uma campanha de ataque. Nesse contexto, surgem os ***phishing* direcionados**, comumente chamados de *spear phishing*, que são modalidades do *phishing* que adotam uma engenharia que preza pela riqueza nos detalhes textuais e visuais, exigindo mais recursos, mas que resultam em uma estratégia que explora a suscetibilidade do usuário final através da fidedignidade.

Phishing direcionado tem um raio de sucesso mais estreito, mas uma taxa de sucesso maior em geral, sendo observados em cenários corporativos. Segundo relatório da Kaspersky[3], em 2019, ataques desse tipo causaram prejuízos financeiros que beiraram 1,7 bilhão de dólares. De acordo com a TrendMicro[4], *phishing* dessa natureza são mais difíceis de serem detectados.

Uma citação que merece destaque é a influência de um ataque bem sucedido de *Spear Phishing* contra John Podesta, na ocasião, chefe de gabinete da Casa Branca nos Estados

[3] https://www.kaspersky.com.br/blog/what-is-bec-attack/14811/
[4] https://www.trendmicro.de/cloud-content/us/pdfs/security-intelligence/white-papers/wp-spear-phishing-email-most-favored-apt-attack-bait.pdf

Unidos. Podesta teve sua conta de e-mail particular invadida através do ataque, e com isso, algumas mensagens pessoais foram vazadas. Muitos especialistas atribuem esse vazamento como determinante para a vitória de Donald Trump (oposição) nas eleições (WIKILEAKS, 2017). Um *phishing* de tal natureza foi capaz de influenciar as eleições de maior impacto sócio-econômico global. Mas por que o *phishing* direcionado é tão poderoso?

Alguns comportamentos presentes no *phishing* são amplamente debatidos na literatura, como a propagação e volatilidade. Contudo, outros carecem um pouco de formalidade nas termologias, portanto, a presente tese elenca os comportamentos seguindo uma própria estrutura categórica. O intuito é agrupar as estratégias de cada comportamento a ser explorado, além de facilitar as particularidades entre um *phishing* convencional e um *phishing* direcionado. Com base no estudo de (SILVA; FEITOSA; GARCIA, 2019a), seguem as seguintes categorias de comportamento:

- **Fidedignidade**, que descreve a alta riqueza de detalhes da fraude em relação à página genuína. É uma estratégia que nem sempre é considerada.
- **Ofuscação**, que descreve as investidas do fraudador em ocultar informações que poderiam ser visíveis ao usuário final, mas devido a quantidade alta ou baixa de caracteres, alguns detalhes podem não ser observados.
- **Propagação**, que descreve alguns comportamentos que visam aumentar o alcance das fraudes em um grande número de usuários, como técnicas de *bypass* em listas de bloqueio.
- **Sazonalidade**, que descreve a sensibilidade do *phishing* aos eventos do calendário anual.
- **Volatilidade**, que remete ao tempo de vida curto do mesmo, evidenciando que a fraude é rapidamente abandonada por seu criador.

Um fato interessante é que esses comportamentos são bem aderentes entre si. Por exemplo, a sazonalidade remete a eventos que aquecem o *e-commerce*, como natal e *Black Friday*, e estes aumentam significativamente a propagação. A alta propagação é uma estratégia de ofuscação para burlar mecanismos baseados em listas de bloqueio e tal prática é impulsionada pelo ciclo de vida curto do *phishing* (SILVA; FEITOSA; GARCIA, 2019a). Não obstante, a ofuscação e a propagação atuam em conjunto como subterfúgio para enganar o usuário.

A única estratégia que não é aderente as outras é a fidedignidade, uma vez que seu propósito é trazer identidade ao alvo da fraude. A fidedignidade investe nos detalhes textuais e visuais, resultando em *phishing* direcionados, por serem norteados para um público-alvo (HO et al., 2017; SRINIVASA; ALWYN; PAIS, 2019). Um exemplo desse tipo de ataque é a especialidade de *Spear Phishing* denominada *Business Email Compromise* (BEC), quando o fraudador forja um e-mail se passando por algum membro da organização, para obter informações sensíveis.

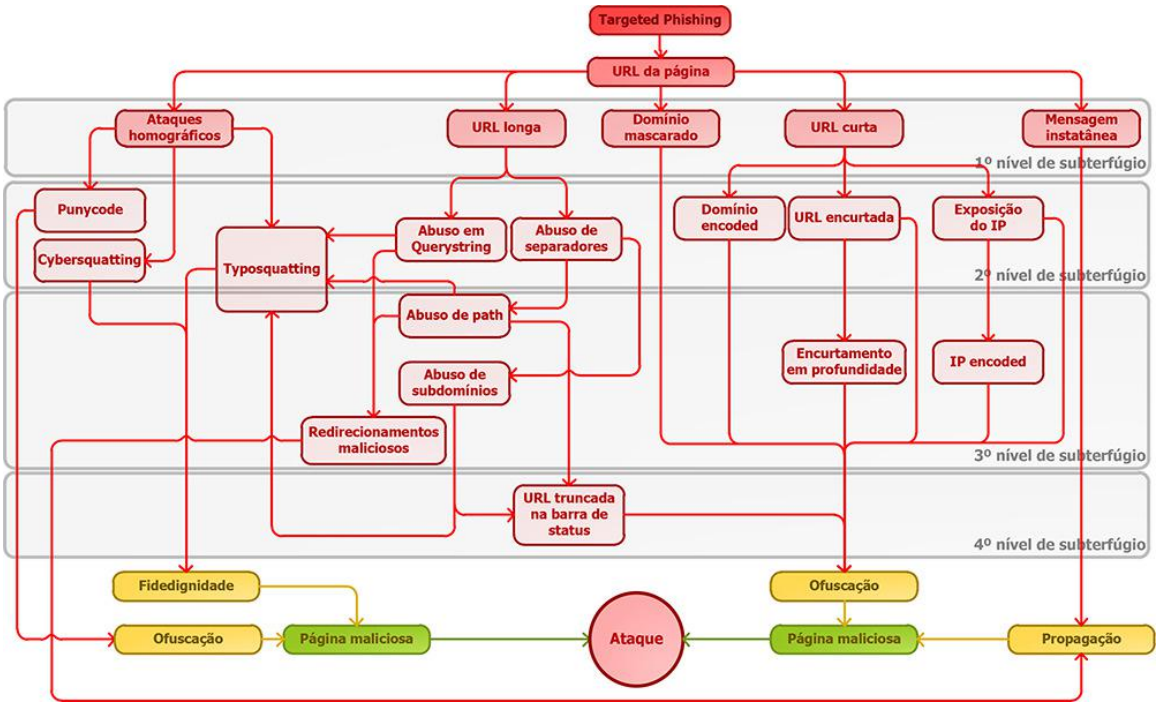

Figura 4 – Taxonomia de phishing baseados em URL

2.1.2 Estratégias para Mitigação de *Phishing*

Para identificar as estratégias empregadas para a mitigação de *phishing*, especialmente o direcionado, foi realizada uma pesquisa $ad-hoc$ que seguiu uma estrutura similar a investigação da literatura do tipo *snowball* (WOHLIN, 2014). Foram encontradas inúmeras abordagens debatidas na literatura com intuito de otimizar a predição de *phishing* e esta Tese destaca as que foram incorporadas para a solução sugerida.

2.1.3 Estratégia de Profundidade

Uma vez que a proposta tem como intuito detectar *phishing* através de uma análise gradual, é necessário contextualizar a estrutura que descreve um processo em profundidade entre as características e comportamentos existentes no contexto de atuação de um *phishing*. Diante disso, é preciso considerar essa relação de forma categórica e, que faça uso de uma sensibilidade ao ambiente de atuação, considerando dispositivos, funcionalidades e recursos envolvidos. Essa ótica pode ser observada conforme a taxonomia ilustrada nas Figuras 4 e 5, que estabelecem níveis de subterfúgio, bem como ações e resultados consequentes.

Os níveis de subterfúgio representam a profundidade das técnicas maliciosas (destacadas em vermelho na Figura 4) categorizadas por tipo de exploração generalizada, que pode ou não resultar em derivações. Por exemplo, uma URL maliciosa propagada através de mensagem instantânea tem um contato unicamente de primeiro nível porque atua diretamente com o usuário final, redirecionando o mesmo para uma página maliciosa, ou seja, um recurso fraudulento (destacado em verde).

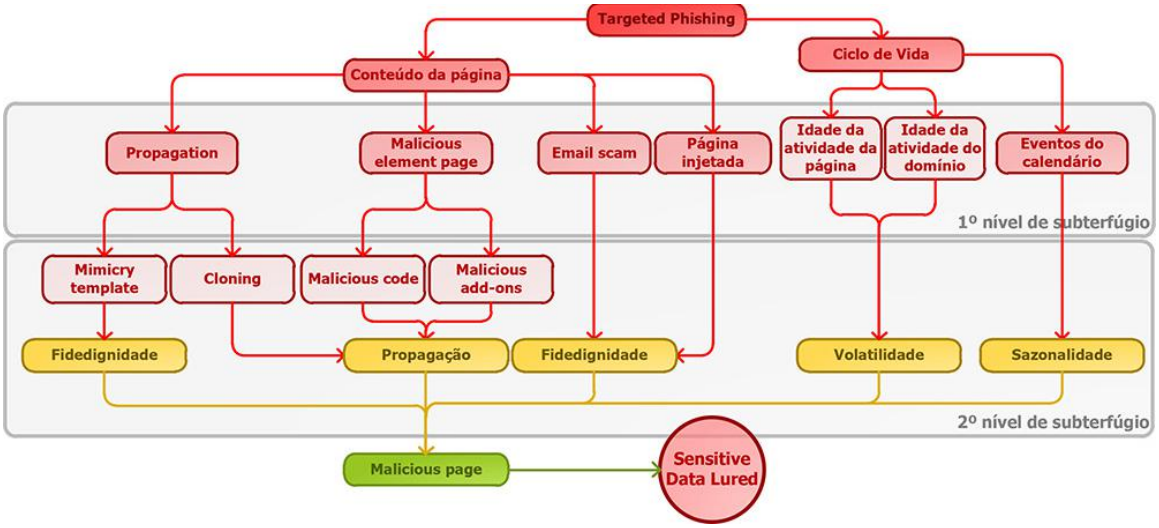

Figura 5 – Taxonomia de phishing baseados em Conteúdo e Ciclo de Vida

Em contrapartida, existem investidas de natureza mais indireta, que por definição, inicialmente são generalizadas e seus impactos desencadeiam para algo mais específico, atingindo segundo, terceiro ou quarto nível. Por exemplo, um subterfúgio de URL longa (Figura 4) bifurca em dois tipos de estratégia: por *querystring* ou separadores. Um ataque com separadores é bifurcado entre explorar o *path* ou subdomínio. Diante disso, através de uma exploração por subdomínio, o atacante tem intenção de fazer subterfúgio baseado em ofuscação da URL na barra de status do navegador ou mesmo fazer uso de *typosquatting* para resultar na forja.

Originalmente, *phishing* direcionados tem a marca como ponto de partida (Figura 5). Contudo, a fraude em si pode ter diversas facetas, a exemplo de um boleto falso, um *malware*, um envenenamento de DNS, uma clonagem de certificado ou um ataque de *phishing*. Um boleto falso se define como um boleto forjado por um fraudador, fazendo uso da fidedignidade para explorar a suscetibilidade. Não obstante, certas fraudes podem considerar um determinado recurso computacional, a exemplo de um *smartphone* ou um protocolo específico, explorando características intrínsecas ao recurso, originando em fraudes específicas. Dentre o exposto, o que chama atenção é a combinação de *phishing* e *malware*, a exemplo do ataque BEC.

Através da taxonomia exposta, é possível observar que recursos como um *host*, um *domínio*, um *banner*, um *DNS*, servem como artifício para explorar indiretamente a suscetibilidade do usuário final. Por exemplo, um *host* precisa hospedar uma página forjada para ter sucesso na investida. São recursos que atuam como intermediários para um determinado golpe, e sozinhos não se torna possível a concretização de um ataque. Diferentemente, recursos como certificado *SSL/TLS*, *e-mail* e *webpage* possibilitam uma investida de forma direta com a vítima, ou seja, são recursos atuantes na camada mais superficial entre a fraude e o usuário final. Além disso, as estratégias de intervenção (rótulos em amarelo) do mal intencionado podem ser variadas.

2.1.3.1 Abordagens baseadas em Heurística

A literatura relata técnicas para a construção de heurísticas de predição de *phishing*. Muitas dessas estratégias se enviesam em observar a semântica e o conteúdo das páginas maliciosas, em busca de extrair padrões. Existem diversas estratégias de heurística *anti-phishing* que são debatidas na literatura. Contudo, comumente, as abordagens são divididas em 3 tipos (AMIRI; AKANBI; FAZELDEHKORDI, 2014; AFROZ; GREENSTADT, 2011). São elas:

- **Abordagens que não consideram o conteúdo**, que não necessitam analisar o conteúdo da página suspeita para classificá-la como legítima ou fraudulenta. Em geral, tais soluções empregam filtros com base em listas de bloqueio ou de permissões, análise de padrões léxicos na URL, ou ambas as abordagens. Diante das limitações existente sobre o uso de lista de bloqueio em responder a *phishing* recém-criados, as estratégias de padrões léxicos ganharam força. Exemplos de padrões dessa natureza seriam verificar o tamanho da URL, a quantidade de caracteres como separadores, a reputação ou localização geográfica do *host* da página, entre outros (MA et al., 2009). Essa abordagem se faz interessante devido (i) a velocidade de resposta em termos de processamento, visto que o único elemento a ser analisado é a URL; e (ii) a ausência de intrusão sobre o conteúdo da página, oferecendo menos riscos a privacidade do usuário final. Contudo, por não avaliar o conteúdo, a abordagem é menos sensível ao contexto (Elwell; Polikar, 2011).

- **Abordagens baseadas em conteúdo**, que baseiam na predição de elementos ou informações constantes na página suspeita. Elementos dessa natureza incluem campos de senha, erros ortográficos, formatação do *CSS*, além de código *HTML* ou *JavaScript*. Apesar da possibilidade de descartar o uso de listas de bloqueios, devido ao ambiente do *phishing* ser dinâmico, a heurística desse tipo de solução precisa realizar revisões constantes para se adequar à à mudanças dos padrões. Não obstante, Aburrous et al. (ABURROUS et al., 2008) destacam a utilização de *hash* para identificar sites maliciosos duplicados na Web. Contudo, essa abordagem pode ser facilmente quebrada caso qualquer modificação seja realizada no site malicioso, já que seu *hash* resultante também seria modificado. Além disso, as soluções dessa abordagem, em comparação as baseadas em lista de bloqueio, precisam ter uma preocupação maior sobre casos de falsos positivos, uma vez que o processo de desfazer um bloqueio indevido envolve reavaliar a métrica de precisão do modelo, ou seja, algo mais custoso que simplesmente retirar uma URL de uma listagem.

- **Abordagens baseadas em similaridade visual**, que realizam a predição com base em captura de telas das páginas suspeitas. É possível realizar predição de *phishing* por similaridade através do contraste da imagem e agrupamentos de pontos-chave com algoritmo *k-mean* (WANAWE; AWASARE; PURI, 2014). Além disso, o cálculo da distância

euclidiana também pode ser utilizada para identificar semelhanças (JAIN; B., 2017). Outro meio de similaridade pode ser realizado através do reconhecimento ótico de caracteres, convertendo as imagens em texto e comparando o texto resultante. Não obstante, existe a estratégia de identificar similaridade através do *layout* CSS. Assim como na abordagem baseada no conteúdo da página, essa metodologia também soluciona melhor a questão de *phishing zero-day*. Todavia, o classificador de similaridade também precisa ser treinado constantemente. Outro fator é que certas páginas maliciosas não reproduzem fielmente o visual da página genuína, seja por ausência no carregamento de imagens ou de folhas de estilo CSS, resultando em uma página esteticamente diferente da genuína, propiciando falsos negativos.

2.1.4 Sistemas Especialistas

Um Sistema Especialista (SE) é um conceito aplicado na Inteligência Artificial com objetivo de resolver problemas através de decisões que simulam o comportamento de um especialista humano em uma determinada área de conhecimento. Jazi et al. (JAZI; MOBARAKEH; LYASHENKO, 2015) descrevem SE como um software que recebe informações como entrada, através de interação com humano (conversacional) ou de forma automatizada (solitária), a serem processadas por seu **motor de inferência**. O motor de inferência é o núcleo do SE. É nele que toda a heurística de regras são aplicadas junto ao processamento das informações de entrada, no intuito de obter a melhor solução especialista sobre um determinado problema. O raciocínio para atingir o diagnóstico pode ocorrer de modo progressivo ou regressivo.

O modo progressivo ocorre quando o motor de inferência processa de modo incremental, a medida que as informações são fornecidas, a heurística vai buscando fatos conclusivos que melhor se aplicam a cada situação, ou seja, ocorre uma interação com o usuário até que uma solução seja encontrada. Já no modelo regressivo, o sistema assume uma opinião conclusiva sobre o problema, mas posteriormente realiza um processamento das informações e regras visando confirmar se a decisão em questão é a mais adequada.

Para todo SE, deve existir uma base de conhecimento, que representa um conjunto de sub-bases previamente estabelecidas para um determinado segmento. Por exemplo, uma base de conhecimento para o setor de vendas, crédito pessoal, contabilidade, ou seja, um artefato que deve ser elaborado por um especialista no assunto. Além disso, o SE também adota uma base de modelo estratégico, que é responsável em aperfeiçoar a eficiência do SE de forma contínua. Tais modelos estratégicos podem ser baseados em regras gramaticais ou conjunto de padrões de comportamentos observados pelo especialista proprietário. O processo para a construção de um SE pode ser dividido em 7 etapas (JAZI; MOBARAKEH; LYASHENKO, 2015), conforme ilustrado na Figura 6.

Na etapa 1, o projetista do SE precisa definir bem o problema em questão. Já na etapa 2, é preciso estabelecer como e qual será a fonte de informações necessárias para a construção do conhecimento. Na etapa 3, é importante descrever como o conhecimento será estruturado

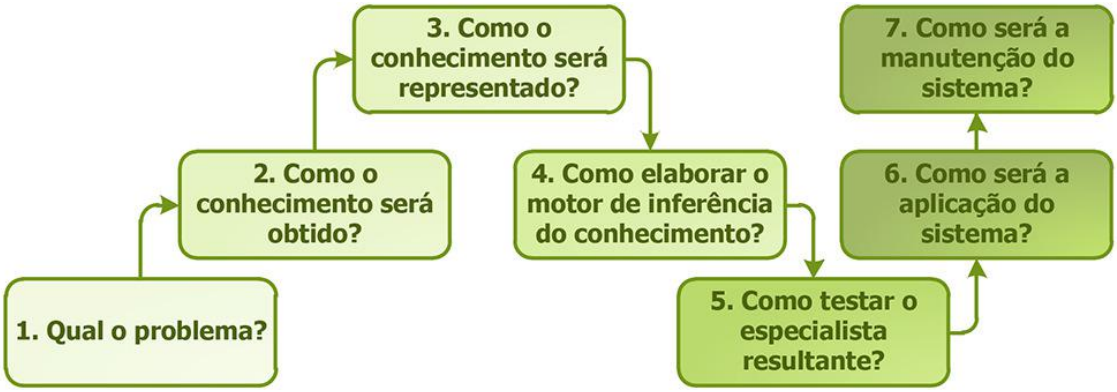

Figura 6 – Passos para a elaboração de um Sistema Especialista (SE)

para seu processamento. Na etapa 4, é definida a elaboração do motor de inferência do SE, que tem como base o conhecimento obtido na fase 2 e estruturado na fase 3. Na etapa 5, é definido como o motor de inferência será testado para avaliar o desempenho de sua precisão. Na etapa 6, deve ser descrito o *design* arquitetural (componentes e conectores) e de interface (interação com o usuário) do sistema especialista. Por fim, na etapa 7 é definido o processo de evolução e manutenção do SE, evitando que o mesmo fique obsoleto com o passar do tempo.

Diante o contexto da Tese, a área de conhecimento é a detecção de *phishing* e o software especialista fará alguns questionamentos sobre informações contidas em uma página maliciosa. Diante disso, a solução dará um parecer sobre a melhor decisão no caso em questão, liberando ou bloqueando a renderização da página. Os questionamentos irão respeitar uma árvore de decisão, representando a máquina de inferência da solução. Esse artefato será descrito em detalhes no Capítulo 4 através de fluxogramas. Já as informações (conhecimento) serão extraídas da URL ou do conteúdo da página, através de abordagens estáticas e dinâmicas.

Uma questão importante de ser enfatizada é a importância e justificativa pela adoção de sistemas especialistas. Em sistemas convencionais, o conhecimento sobre o domínio é encapsulado em suas instruções de controle, bem como em sua estrutura de dados da aplicação, ou seja, fica intrínseco ao código-fonte (JAZI; MOBARAKEH; LYASHENKO, 2015). Diferentemente, no escopo de sistema especialista, o conhecimento relativo ao domínio do problema em questão fica encapsulado em estruturas de dados independente do código-fonte da aplicação, ou seja, o algorítimo da aplicação é independente do conhecimento para a solução do problema de domínio (JAZI; MOBARAKEH; LYASHENKO, 2015).

2.2 TRABALHOS RELACIONADOS

Nesta seção, serão descritos trabalhos da literatura que possuem soluções correlatas à proposta desta pesquisa. A Figura 7 ilustra os trabalhos relacionados e a relação dos mesmos sobre as contribuições correlatas aos objetivos dessa pesquisa. Todos os trabalhos tem em comum a proteção em ambientes *e-commerce*, *internet banking* e *redes sociais* contra ataques de *phishing*. Contudo, dentre os trabalhos relacionados, foram observados demais aspectos de

similaridade com base em 5 eixos estratégicos, a saber: (i) Análise dinâmica, (ii) Análise estática, (iii) Phishing direcionado, (iv) Proteção à Marca e (v) Sistema especialista.

Estudos	Ano	Análise dinâmica	Análise estática	Phishing direcionado	Proteção a Marca	Sistema especialista
Santiago et al.	2015	✓	✓	✓	✗	✗
Rajeshwari & Babu	2016	✗	✓	✗	✓	✓
Jog et al.	2018	✓	✓	✗	✗	✗
Carta et al.	2019	✓	✓	✗	✓	✗
ElOrche et al.	2019	✗	✓	✗	✗	✓
Este estudo	-	✓	✓	✓	✓	✓

Figura 7 – Trabalhos relacionados e a relação com os 5 eixos da proposta

Em (SANTIAGO; PEREIRA; HIRATA JR., 2015) abordam diversas maneiras de aplicabilidade de fraudes em recursos na Web, resultando em um conjunto de métodos e técnicas de prevenção em fraudes de cartão de crédito (i e ii) e ataques à sistemas de pagamento eletrônico que exploram identidade visual (iii). A pesquisa apresenta os papéis existentes no ecossistema de *e-commerce*, destacando comportamentos e ameaças relacionadas. Na mesma linha, o estudo apresenta técnicas de detecção de fraudes com técnicas de aprendizagem de máquina supervisionada ou não-supervisionada, e avaliam a proposta através de um experimento com um grande serviço latino americano de pagamentos.

Segundo o estudo de (Rajeshwari U; Babu, 2016), é proposto uma proteção em tempo real, ou seja, que monitora a navegação do usuário contra fraudes em *e-commerce*. Sua estratégia se baseia em processamento de dados através de análise de *streaming*, que visa detectar padrões maliciosos com base em históricos das transações (v) em determinadas operadoras financeiras (iv), considerando aspectos como ciclo de vida e tempo (i), além de palavras-chave na detecção (ii). Em (JOG; CHANDAVALE, 2018) é proposta uma solução de combate a fraudes analisando comportamentos durante a transação com cartões de crédito (i e ii) considerando as mudanças de ambiente (i) e URL (ii). O estudo se baseia em detectar características relacionadas aos papéis envolvidos no ecossistema de pagamentos. O algoritmo da proposta se baseia em atualizações periódicas através de uma abordagem ativa e passiva.

Em (CARTA et al., 2019) os autores evidenciam uma tendência na literatura sobre soluções anti-fraude com base em inteligência artificial, diante de tal lacuna, os autores propõem uma nova abordagem baseada em modelo de consenso múltiplo prudencial, técnica que combina a eficácia de diversos algoritmos de classificação que considera algumas marcas (iv). O modelo se baseia em uma estratégia probabilística na detecção de padrões maliciosos (i e ii). Em (ORCHE; BAHAJ, 2019), os autores utilizam ontologia para o reconhecimento de padrões fraudulentos (v), provendo dessa maneira uma proteção em padrões maliciosos (ii) no lado cliente durante a navegação do usuário final.

Em relação as estratégias mencionadas na Seção 2.1.1.1, muitos trabalhos abordam preocupações sobre a propagação e volatilidade do *phishing*, tais características são presentes em qualquer natureza de *phishing*, consequentemente, são estratégias mais debatidas e consolidadas na literatura. Outras estratégias, como a ofuscação, são menos debatidas, mas ainda

sim, são mencionadas e exploradas. Todavia, se faz interessante investigar com mais interesse as estratégias de fidedignidade e sazonalidade, visando obter maior precisão sobre a relevância das características que configuram uma página como maliciosa. Tais estratégias tem caráter mais qualitativo e dinâmico em relação as outras, tornando mais complexa suas investigações.

A presente Tese tem como proposta fortalecer o elo entre as 5 características mencionadas, visando estruturar um modelo baseado em regras, resultando em uma heurística de predição que melhor responda aos incidentes de *phishing* direcionados.

Por fim, foi possível observar que os trabalhos relacionados não descrevem alguma sensibilidade explícita no contexto das 5 categorias de comportamento apresentadas sobre o *phishing*. Implicitamente, talvez, todos os trabalhos abordem sobre a ofuscação, propagação e volatilidade, já que muitas vezes atuam em conjunto e quase sempre são inerentes a natureza do *phishing* convencional. Contudo, aspectos mais particulares, como a fidedignidade e a sazonalidade, que são mais atuantes em *phishing* direcionados, apresentam uma cerca carência literária, ou mesmo uma ausência na formalização desses comportamentos. Diante disso, além das contribuições descritas nos objetivos específicos, também assumimos que a divisão categórica sobre os comportamentos apresentada por nosso estudo acaba sendo uma contribuição. Tal formalização visa auxiliar na definição de termos e nomenclaturas nos comportamentos tão diversificados de um *phishing*, seja o convencional ou o direcionado.

2.3 CONSIDERAÇÕES FINAIS

Diante da fundamentação teórica exposta, a presente Tese de doutorado tem como objetivo adotar técnicas descritas que serviram de apoio para a elaboração da proposta da solução. Para tanto, a Tese adota uma abordagem heurística baseada em regras, propondo assim um sistema especialista dirigido por uma conjunto de regras estruturadas em árvore. O processamento e execução dessas regras respeitam uma estratégia em profundidade, visando atenuar problemas de desempenho e privacidade. A heurística pretende ser sensível para detecção de determinados comportamentos (características) presentes em uma página Web, com o intuito de julgar a mesma como maliciosa ou não, com base no padrão de sua composição. Adicionalmente, a solução visa oferecer uma heurística que seja capaz de nortear a mitigação em *phishing* direcionados, ou seja, que adotam a estratégia de riqueza em detalhes.

Por fim, esse capítulo teve o intuito de apresentar um referencial teórico sobre o cenário abordado pelo estudo, bem como uma ótica que delimita escopo e apresenta o estado de arte atual do cenário de atuação de propostas de natureza correlata ao que é proposto na pesquisa dessa Tese. O próximo capítulo visa apresentar o estudo empírico que trouxe fundamentação teórica para a observação de características estáticas e dinâmicas no contexto de atuação do *phishing*, resultando em uma regressão logística para propor um modelo de classificação para a predição de *phishing*.

3 ESTUDO EMPÍRICO

Esse Capítulo tem o intuito de apresentar um estudo empírico que investigou padrões de comportamentos existentes em *phishing* reais. A abordagem é baseada em **regressão logística**, que segundo (HILBE, 2016), é uma técnica da estática capaz de definir um modelo responsável em realizar predições através de variáveis categóricas. Essas variáveis são oriundas de variáveis explicativas contínuas e/ou binárias, extraídas no processo de observação/interpretação dos dados coletados. O intuito desse modelo é tomar decisões com base de dados multivariados.

A regressão logística é capaz de medir o relacionamento de uma determinada variável dependente e categórica com outras variáveis independentes (HILBE, 2016). Considerando o escopo da presente Tese, o relacionamento é medido através da relevância das ocorrências (análise quantitativa) e observando a relação e similaridade entre as características (análise qualitativa). Diante disso, torna-se possível propor um modelo de classificação com base em hipóteses levantadas através de observações em amostras de cenários reais.

Segundo Wohlin et al.(WOHLIN et al., 2000), tal prática entende-se um estudo primário, ou seja, uma abordagem da engenharia de software experimental que observa comportamentos em um contexto específico. Nesta Tese, tais comportamentos são traduzidos como as características empregadas na heurística proposta para classificar uma página como maliciosa. As características são observadas em páginas confirmadas como fraudulentas e páginas consideradas legítimas, no intuito de testar hipóteses que investigam se um determinado comportamento ocorre apenas em fraudes.

Como as amostras e variáveis (dependentes e independentes) não podem ser controladas e o ambiente a ser observado trata-se de um ambiente real, o estudo adotou a metodologia *survey*, conforme debatidos nas obras de Molleri et. al (MOLLERI; PETERSEN; MENDES, 2016), Robson (ROBSON, 2002) e Babbie (BABBIE, 2019). Geralmente, um *survey* apresenta-se como uma investigação em retrospecto, analisando comportamentos através de dados coletados em uma amostra representativa da população do contexto. Com os resultados dessa coleta, extrai-se conclusões que possam ser generalizadas a população da qual a amostra foi derivada.

3.1 DEFINIÇÃO DAS CARACTERÍSTICAS

Essa seção descreve o escopo deste estudo empírico, no caso as caraterísticas existentes e extraíveis de páginas Web fraudulentas ou não. Adicionalmente, a composição da relação e similaridade entre as características resultou em uma taxonomia.

3.1.1 Características

Cada característica representa um meio de medir um comportamento do *phishing*. Contudo, nem sempre a presença da característica pode representar uma página maliciosa. Devido a isso,

modelos de predição de *phishing* adotam estratégias em que o conjunto de determinadas características existentes podem representar uma determinada classe no modelo de classificação. Essa decisão reflete no balanceamento entre sensibilidade e especificidade da heurística.

Como o intuito dessa pesquisa é avaliar as ocorrências individuais de cada característica em *phishing* reais e em páginas legitimas, foram avaliadas dezenas de características existentes na literatura categorizadas por abordagens distintas, a saber: (i) padrões léxicos do código-fonte e da URL, que analisam comportamentos estáticos; e (ii) padrões de comportamentos no conteúdo e contexto, classificadas como dinâmicas porque consideram o aspecto de tempo e mudanças. Todas as características foram extraídas de estudos publicados na literatura, várias delas definidas com diferentes nomes, mas com o mesmo funcionamento. O importante é que essas características são bastante utilizadas em heurísticas para modelos de classificação, a exemplo de *dataset* disponibilizados em sites como *Kaggle* [1] e *UCI* [2]. Por questões de praticidade, os detalhes sobre cada características serão descritos na seção que apresenta os resultados dos dados e interpretação.

Importante salientar que certas características, comumente presentes em obras literárias, não constam nesse estudo por fatores variados. Algumas tornaram-se obsoletas devido ao término de um serviço, a exemplo do *Google Page Rank*, ou necessitavam de extração de dados com certa limitação, a exemplo do protocolo *WHOIS*[3] em casos de domínios privados. Maiores detalhes serão descritos na Seção 3.5. A Tabela 1 apresenta as 30 características definidas neste estudo.

3.1.2 URL

Esta Tese propõe a análise da URL, considerando a estratégia ilustrada na Figura 8, em que divide a URL em duas partes no intuito de observar a anatomia da URL com estratégias distintas e necessárias para investigar certas características.

Figure

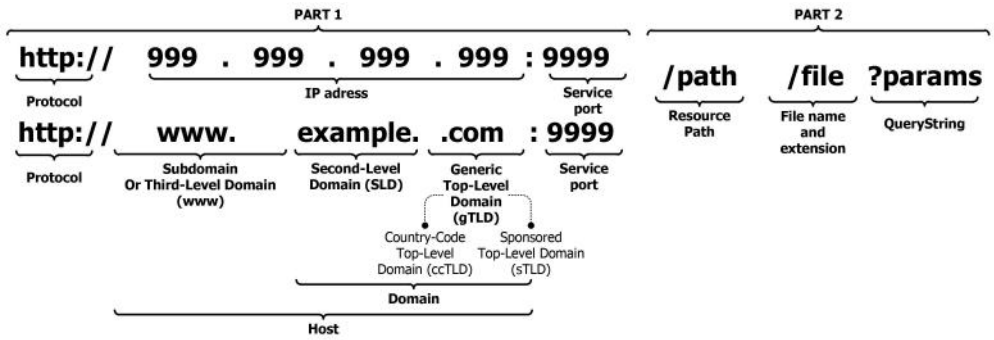

Figura 8 – Anatomia da URL

[1] https://www.kaggle.com/akashkr/phishing-website-dataset
[2] https://archive.ics.uci.edu/ml/datasets/phishing+websites
[3] WHOIS: https://tools.ietf.org/html/rfc3912

Tabela 1 – Nome e descrição das 30 características

Nome	**Descrição**
C01. Default HTTP port explosure	Exposição da porta 80 ou 403 na URL
C02. Double slash path	Presença de // na URL
C03. Duplicate URL entry	URL duplicadas na plataforma
C04. Incident Response for community	Intervalo para um *phishing* ser confirmado na plataforma
C05. Precision for community	Presença de falsos positivos e negativos na plataforma
C06. Activity time	Tempo de atividade (uptime) do *phishing*
C07. Age of Domain	Tempo de atividade de um domínio malicioso
C08. Cloning strategy	Clonagem de páginas maliciosas
C09. SEO score	Reputação de um determinado HOST
C10. Volatility	Descreve o curto tempo de atividade do *phishing*
C11. Content page most exploit	Tipo de conteúdo mais explorado
C12. Host most exploit	Serviço de hospedagem mais explorado
C13. Language most exploit	Idioma mais explorado
C14. Seasonality most exploit	Eventos do ano-calendário mais explorados
C15. Service most exploit	Serviços computacionais mais explorados
C16. TLD most exploit	Registro de domínio TLD mais explorado
C17. Encoded exploit	Quando ocorre codificação no domínio
C18. IP address explosure	Quando o IP é exposto (ausência de domínio)
C19. Shortened URL	Exploração de práticas através de encurtadores de URL
C20. URL with variables	Exploração no *path* e *querystring* da URL
C21. Amount of separators	Abuso no uso de separadores em uma URL
C22. HTTP with specification port	URL que utilizam portas diferentes da padrão
C23. URL size	Analisa as URL curtas e longas
C24. Browser punycode exploit	Explorações através do protocolo *punycode*
C25. Concatenate subdomains	Abuso na quantidade de subdomínios
C26. Domain with reputation	Domínio sequestrado ou com páginas maliciosas injetadas
C27. HTTP tunneling	Quando existe ou não certificado digital
C28. Malicious browser-based code	Quando existe código malicioso focado no dispositivo
C29. URL redirection	Quando a URL possui alguma investida de redirecionamento
C30. URL spoofing	Explora a confiança do usuário através de palavras-chave

A parte 1 é composta pelo protocolo, endereço IP ou host, e a porta do serviço Web. O host é composto por um subdomínio, também chamado de terceiro nível, e um domínio. Já o domínio é a combinação de um domínio de topo (gTLD) e segundo nível (SLD). Domínios gTLD tem o propósito de especificar o país ou segmento geral de atuação do site. Já o domínio SLD é o qual o proprietário do site nomeia no registro de servidores DNS. A parte 2 da URL é composta pelo *path*, nome do arquivo, extensão do arquivo e parâmetros do método GET.

Elementos como protocolo, endereço IP e porta podem ser definidos pelo atacante, porém, terão manipulações um pouco mais previsíveis. Por exemplo, na parte 1 é possível prever que no protocolo haverá valores como HTTP ou FTP precedidos dos caracteres "://", e a porta será representada por números inteiros precedidos de dois pontos ":". Em contrapartida, elementos como endereço IP ou host oferecem maior flexibilidade para o atacante explorar alguma tática, considerando que o mesmo tem liberdade de definir os valores como desejar. Na parte 2, o elemento parâmetros terá seu valor precedido pelo carácter "?" ou "&" para múltiplos valores. Contudo, o atacante tem liberdade de atribuir os respectivos valores. Os demais elementos da parte 2 podem receber valores arbitrários.

3.1.3 Taxonomia

Uma taxonomia é um artefato responsável em apresentar uma estrutura de conceitos de forma categorizada. Tal artefato tem potencial para oferecer um entendimento sobre a sistemática e as similaridades compartilhadas entre comportamentos debatidos, pois sua classificação proporciona um olhar clínico sobre os comportamentos e suas relações. As características definidas na seção anterior permitem a elaboração de uma taxonomia que reflete o modelo heurístico de predição de *phishing*.

A fundamentação para a distribuição das características em categoria é sustentada pela proposta de agrupamento (*clustering*) não-paramétrico e particional, que define modelos intrínsecos de modo que seus elementos sejam semelhantes com base em algum critério. Contudo, os respectivos subgrupos não possuem um relacionamento hierárquico entre eles. Em suma, é um processo de classificação não-supervisionada que aglomera dados baseando-se em similaridades específicas. Para a definição da taxonomia, foram levantados 4 questionamentos, a saber: (i) Como estabelecer a similaridade entre as características? (ii) Como segmentar os agrupamentos? (iii) Qual será a quantidade de agrupamentos? (iv) Como avaliar os agrupamentos definidos?

Para responder os questionamentos, foi realizada uma investigação na literatura como embasamento para estruturar, justificar e testar a taxonomia proposta. Para tanto, foi utilizada a estratégia de Lough (LOUGH, 2001), que compila argumentos extraídos de cinco estudos similares (HOWARD, 1998; AMOROSO, 1994; BISHOP, 1999; LINDQVIST; JONSSON, 1997; KRSUL, 1998). Estes trabalhos enumeram requisitos necessários para a definição de uma taxonomia, conforme ilustrado na Figura 9, que encontram-se agrupados por autoria. Tais requisitos podem ser traduzidos como diretivas para atingir um nível de eficiência ao modelo de classificação da

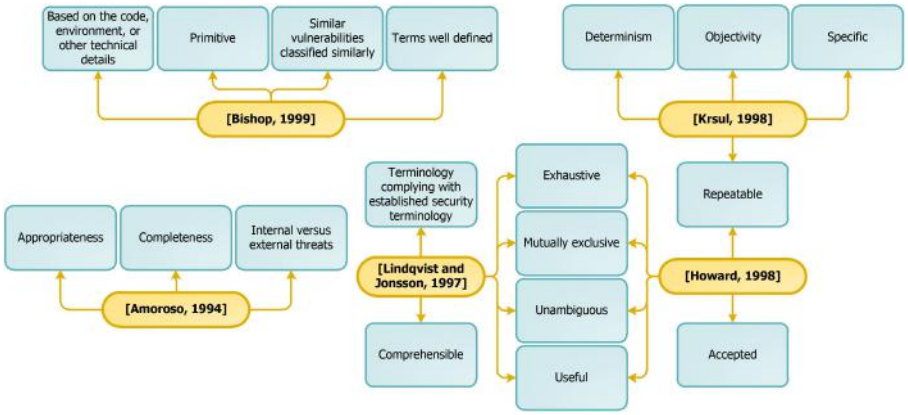

Figura 9 – Os requisitos para a construção de uma taxonomia

taxonomia. Os requisitos são definidos como:

- **Accepted**: Com base nas considerações de (HOWARD, 1998), o conjunto de hipóteses levantadas por uma taxonomia deve ser tecnicamente aceito.
- **Appropriateness**: Um meio de atender as características é estabelecer categorias construídas por fundamentos baseados em obras relevantes sobre o tema, consolidando a mesma de caráter apropriado (AMOROSO, 1994).
- **Based on the code, environment, or other technical details**: Segundo (BISHOP, 1999) uma taxonomia deve cobrir aspectos de implementação e infraestrutura envolvidos no respectivo cenário.
- **Comprehensible**: Uma taxonomia deve garantir um nível satisfatório de entendimento tanto para especialistas como para aspirantes ao tema relacionado (LINDQVIST; JONSSON, 1997). Ela deve ser intuitiva o bastante para o leitor entender que as características se encaixam de forma padronizada.
- **Completeness**: A taxonomia precisa ser suscetível a uma possível expansão. A justificativa é que não seria incomum surgir novos tipos de *phishing*, portanto afirmar que uma taxonomia é imutável e ao mesmo tempo infalível seria algo temerário ou mesmo questionável, porém a mesma precisa garantir um nível de plenitude e precisão (AMOROSO, 1994).
- **Determinism**: Uma taxonomia precisa garantir que cada situação deve ser respondida com SIM ou NÃO, caracterizando-a como determinista (KRSUL, 1998).
- **Exhaustive**: Uma taxonomia é considerada exaustiva quando todas as características e situações são cobertas por seu modelo de classificação (HOWARD, 1998; LINDQVIST; JONSSON, 1997).

- **Internal versus external threats**: Uma taxonomia precisa considerar aspectos que, apesar de não serem controlados por um domínio específico, possam ser relacionados (AMOROSO, 1994). A taxonomia proposta visa atender com baixo nível de abstração todas as possíveis ameaças e suas respectivas variações.
- **Mutually exclusive**: Segundo (HOWARD, 1998; LINDQVIST; JONSSON, 1997), toda taxonomia deve ser mutuamente exclusiva entre suas categorias.
- **Objectivity**: A taxonomia precisa se preocupar em expor de forma clara as características que diferem entre as categorias, expressando assim objetividade (KRSUL, 1998).
- **Primitive**: Os comportamentos descritos na taxonomia precisam coincidir com o contexto em questão, possibilitando descrever diversas situações procedentes (BISHOP; BISHOP, 1995).
- **Repeatable**: Segundo (HOWARD, 1998; KRSUL, 1998), uma taxonomia precisa estruturar suas decisões através de uma arvore de decisão.
- **Similar vulnerabilities classified similarly**: A taxonomia proposta faz uma classificação com base na similaridade, agrupando com base em comportamentos compartilhados, evitando ambiguidades (BISHOP, 1999).
- **Specific**: A taxonomia deve ser específica e com adequação ao domínio (KRSUL, 1998).
- **Terminology complying with established security terminology**: A taxonomia proposta precisa possuir termos bem definidos, ou seja, que respeite as nomenclaturas técnicas do respectivo domínio (BISHOP; BISHOP, 1995; LINDQVIST; JONSSON, 1997).
- **Terms well defined**: A taxonomia proposta deve se preocupar em atender o cenário específico para ataques de *phishing* e fazer uso de terminologias comumente utilizadas (BISHOP, 1999).
- **Unambiguous**: A taxonomia não deve apresentar redundância em suas categorias ou ramificações, ou seja, proporcionar duplo sentido em alguma ramificação da sua árvore de decisão (HOWARD, 1998; LINDQVIST; JONSSON, 1997). caracterizando um aspecto inequívoco (HOWARD, 1998).
- **Useful**: Segundo (HOWARD, 1998; LINDQVIST; JONSSON, 1997), a taxonomia deve ser útil para o cenário de atuação, de fácil entendimento e baixa complexidade para maior adesão.

A Figura 10 apresenta a taxonomia proposta com as 30 características separadas em aspectos do contexto e padrões léxicos. A taxonomia segmenta as características em 6 categorias distintas, segmentadas por: comunidades que atuam como plataformas de denúncia de incidentes de *phishing*; o ciclo de vida do *phishing* e o perfil-alvo explorado pelos mal intencionados,

representando assim 16 características. Além dessas, também existem as categorias que analisam comportamentos que visam aplicar "by-pass" em listas de bloqueios; outras que analisam a morfologia da URL; e por fim, a exploração da fidedignidade através de recursos na URL ou conteúdo da página, dos quais são representados por 14 características.

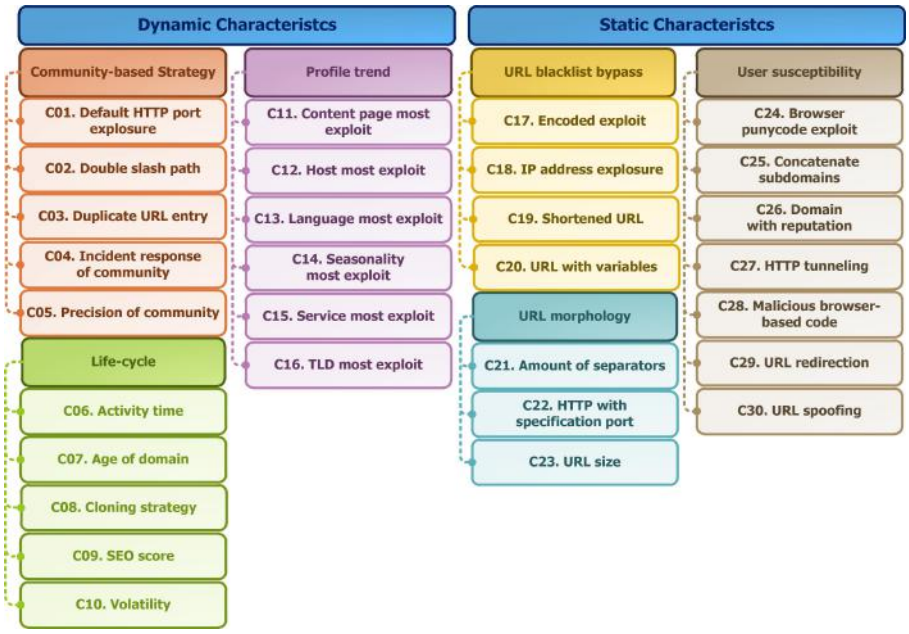

Figura 10 – Taxonomia de phishing baseada em contexto

Um fato que merece destaque é que, até o momento, não foi possível observar na literatura qualquer taxonomia que trouxesse uma abordagem considerando aspectos estáticos e dinâmicos, conforme descrito pela presente Tese, configurando-se assim em uma contribuição para trabalhos que visam analisar comportamentos em características.

Vale destacar que foi realizada uma investigação na literatura para estruturar, justificar e testar a taxonomia proposta. Para tanto, foi utilizada a estratégia de Lough (LOUGH, 2001), que compila argumentos extraídos de cinco estudos similares (HOWARD, 1998; AMOROSO, 1994; BISHOP, 1999; LINDQVIST; JONSSON, 1997; KRSUL, 1998). Estes trabalhos enumeram requisitos necessários para a definição de uma taxonomia. Todo o detalhamento sobre o processo de elaboração da taxonomia encontra-se em (LOUGH, 2001).

3.2 POPULAÇÃO E AMOSTRA

A população utilizada nesse estudo empírico é formada por páginas legítimas e páginas fraudulentas relacionadas com *phishing*. Para tanto, o estudo adotou a plataforma *PhishTank* como base de dados tanto para as amostras de *phishing* reais (*phishing* válidos) como para sites legítimos (*phishing* inválidos). Apesar da possibilidade de redução na precisão, ainda sim, fazer uso de amostras viabiliza a obtenção dos objetivos de uma metodologia dessa natureza.

Embora existam outros serviços, como o OpenPhish[4] e o SafeBrowsing[5], a opção pelo *PhishTank* foi devido sua base aberta e com maior volume, além de suas funcionalidades estarem disponíveis gratuitamente.

3.2.1 A plataforma PhishTank

O PhishTank é uma plataforma comunitária e gratuita onde qualquer pessoa pode enviar, verificar, rastrear e compartilhar dados de *phishing*[6]. Também fornece uma API aberta para compartilhar com aplicações terceiras seus dados *anti-phishing* de forma gratuíta. É importante salientar que a equipe do *PhishTank* não considera sua plataforma como uma medida de proteção[7], contudo, as informações fornecidas pela mesma servem de subsídio para mecanismos de resposta a incidente em diversas organizações[8], a exemplo da *Yahoo!*, *McAfee*, *APWG*, *Mozilla*, *Kaspersky*, *Opera* e *Avira*.

O *PhishTank* é descrito como uma **comunidade** porque comporta um grande número de usuários que colaboram entre sim dados de *phishing* na Web. Seu caráter **colaborativo** remete ao fato de todos usuários cadastrados terem a possibilidade de alimentar a base de dados de *phishing*. Cada registro de *phishing* é alimentado através de denúncias voluntárias de usuários. O ciclo de vida entre o *phishing*, a plataforma e seus usuários é dividido em 5 etapas, conforme ilustrado na Figura 11.

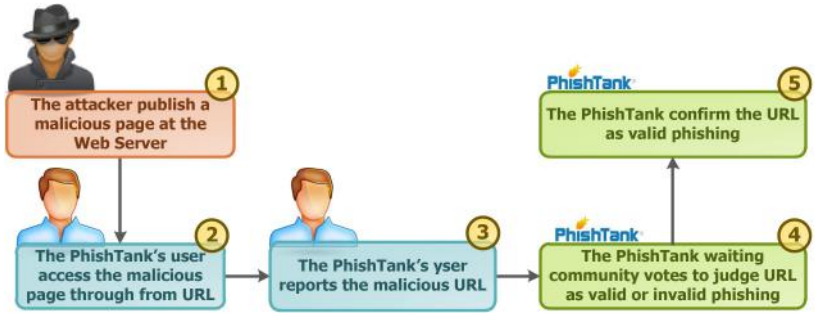

Figura 11 – Ciclo de vida da comunidade *PhishTank*

Na etapa 1 o atacante publica sua página maliciosa em um servidor web, disponibilizada para ser propagada pela Web. A etapa 2 remete o descobrimento da URL maliciosa por um usuário. Posteriormente, o mesmo acessa o *PhishTank* e denuncia a URL, realizando assim a etapa 3. Já a etapa 4 descreve o momento em que a plataforma aguarda as votações da comunidade sobre a URL recém denunciada. Por fim, na etapa 5 ocorre quando o sistema de votos recebe uma quantidade satisfatória para considerar a URL maliciosa ou não. Vale

[4] https://openphish.com/
[5] https://safebrowsing.google.com
[6] https://www.phishtank.com/faq.php#whatisphishtank
[7] https://www.phishtank.com/faq.php#doesphishtankprotect
[8] http://www.phishtank.com/friends.php

salientar que a quantidade "suficiente" de votos não é explicitada, a plataforma declara que pode variar de acordo com o histórico das denúncias[9].

3.2.2 Amostras

Para a definição das amostras, foi necessário obter uma quantidade significativa de *phishing* "válidos", *online* ou *offline*. Como alternativa, a plataforma disponibiliza um *web service* que fornece um **arquivo JSON**[10]. O mesmo é atualizado a cada 1 hora e tem em média 15,000 registros. Além da URL, status, confirmação e data de publicação, também fornece a **data de confirmação** e a **marca alvo da fraude**. A data de confirmação é o momento que a votação se encerra e a URL é confirmada como *phishing*.

Para extração, foi desenvolvida uma aplicação para o consumo periódico do arquivo JSON via API da plataforma. Como a plataforma não estabelece um prazo de conclusão da votação, o estudo adotou uma margem de **intervalo de coleta** de 1 mês adjacente para cada mês corrente, ou seja, a coleta do mês de janeiro foi encerrada até o último dia de fevereiro, e seguiu sucessivamente, concluindo assim a coleta do mês de dezembro de 2018 no dia 31/01/2019.

Contudo, o processo teve alguns entraves. Cerca de 90% das URL eram mantidas nos demais arquivos subsequentes. Considerando que cada JSON compactado possuía 9MB, acabava por sobrecarregar a plataforma, proporcionando erros *509 bandwidth limit exceeded*, indicando que a chave na requisição havia sido banida. Visando contornar esse problema, foram realizados cadastros prévios de diversas chaves para serem substitutas cada vez que alguma chave fosse banida, conforme o fluxograma da esquerda na Figura 12.

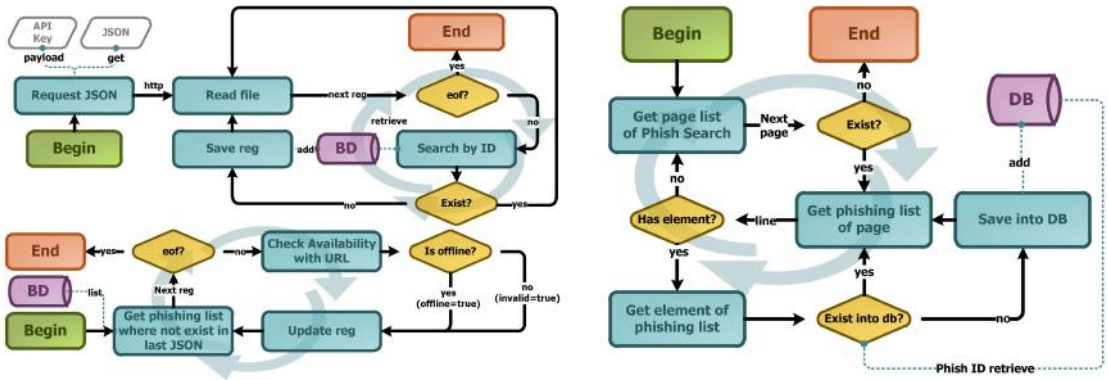

Figura 12 – Fluxograma das extrações baseadas em "JSON" e "Phish Search"

Por ser bastante informativo, o JSON seria uma excelente amostra para analisar os comportamentos, todavia, o mesmo tem algumas limitações. Por registrar apenas *phishing* "válidos" e "online", características **temporais**, como C14 (**Seasonality Most Exploit**), teriam resultados enviesados. Um exemplo disso foi observado em um arquivo baixado no dia 15/01/2019, em que os meses de janeiro e fevereiro de 2018 possuíam, respectivamente, 358 e 617 regis-

[9] https://www.phishtank.com/faq.php#howmanypeoplehavetov
[10] http://data.phishtank.com/data/online-valid.json.bz2

tros. No mesmo arquivo, meses mais recentes, como novembro e dezembro do mesmo ano, constavam, respectivamente, 1.524 e 1.791 registros.

Como alternativa, a plataforma disponibiliza a funcionalidade **phish archive**[11] no menu **phish search**, que contém o histórico de todas as URL submetidas, possibilitando filtrar os registros por "validos" ou "inválidos" e "online" ou "offline". Infelizmente, o *PhishTank* não oferece um *download* de todos os registros do relatório resultante da opção **phish search**. Diante disso, houveram diversas tentativas de contato com a equipe do *PhishTank* no intuito de obter essa informação. Porém, como as solicitações não tiveram retorno, foi desenvolvido um *Web Crawler* para coletar e armazenar as informações dos registros listados de forma automatizada, o processo é ilustrado no fluxograma da direita na Figura 12.

Outra dificuldade é que a página da listagem exibia apenas os 70 primeiros caracteres da URL. Caso fosse superior, o restante dos caracteres eram trocados por "...". Desta forma, nos casos em que a quantidade de caracteres na URL ultrapasse esse limite, o *Web crawler* precisava acessar a página de detalhes para obter a URL completa. Contudo, para alguns registros a página de detalhes estava inacessível e, por ser um número pouco expressivo, estes foram descartados. Além disso, a tentativa de obter *phishing* muito antigos, com mais de 10 anos, frequentemente apresentava erros HTTP 504 (*Gateway time-out*), caracterizando uma limitação no acesso ao servidor.

O relatório da página **phish search** oferece informações, como o **ID do *phishing*** na plataforma, a **URL**, a **data de submissão**, sua **confirmação** (se é "valido" ou "inválido") e sua **disponibilidade** (se está "online" ou "offline"). A data de submissão registra o momento em que um determinado usuário realiza a denúncia da URL possivelmente maliciosa. O fluxo de definição das amostras é ilustrado na Figura 13.

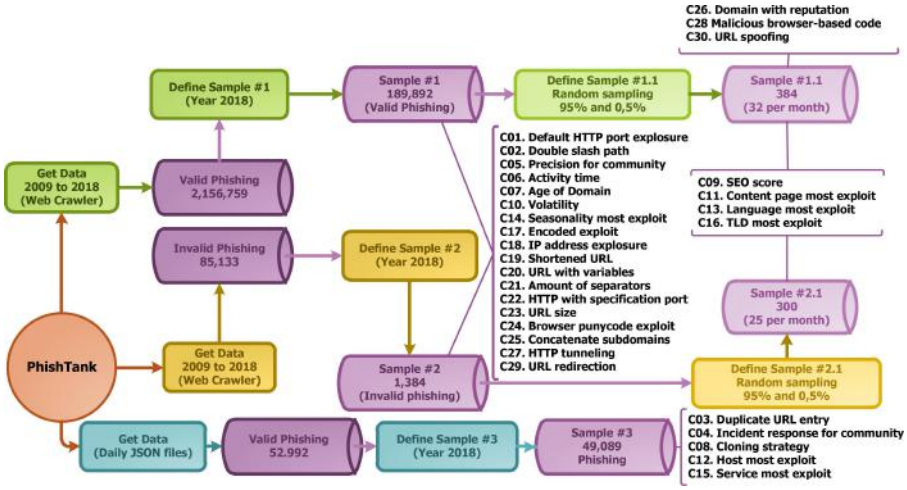

Figura 13 – Estratégia para definição de população e amostras

Diante disso, o Web Crawler extraiu 2.156.759 registros de *phishing* durante o período de 10 anos, de 2009 até 2018. De posse desses dados, foi possível definir a **Amostra #1**, que

[11] https://www.phishtank.com/phish_archive.php

considerou apenas *phishing* válido e do ano de 2018, resultando assim em 189.892 registros. Paralelamente, foi definida a **Amostra #2**, seguindo a mesma regra, só que para *phishing* "inválidos", resultando em 1.384 registros de sites legítimos. O intuito é fazer um comparativo objetivando analisar se há diferenças de comportamento entre *phishing* válidos e inválidos.

Mesmo com a população reduzida, conforme a amostra #1, ainda sim o processo de extração foi inviável para características em que o processo automatizado não era possível. Diante disso, conforme as Equações 3.1 e 3.2, foi definida a **Amostra #1.1**, aleatória (SINGH; MANGAT, 1996; LUMLEY, 2011). Da mesma forma, foi definida a **Amostra #2.1** visando realizar comparativos de hipóteses entre *phishing* válidos e inválidos. Considerando-se variabilidade máxima na população, com uma confiança de 95% e com uma precisão de 5%, o tamanho da amostra será de 384 para *phishing* válidos e 300 para *phishing* inválidos. Como o objetivo dessas amostras também é ser temporal, a quantidade de registros foi dividida proporcionalmente pelos meses do ano, resultando em 32 registros para cada mês da amostra #1.1 e 25 registros para cada mês da amostra #2.1.

$$\frac{z^2 \times p(1-p)}{e^2} \tag{3.1}$$

$$1 + \left(\frac{z^2 \times p(1-p)}{e^2 N} \right) \tag{3.2}$$

O cálculo possui duas etapas, na (3.1) a grandeza *z* representa o grau de confiança em desvios padrões (95% ou 1.96 na escala Z), o *e* representa a margem de erro (5% ou 0.05), o *N* representa a população total de URL (189,892 para os válidos e 1,384 para os inválidos). Por fim, o *p* é a verdadeira probabilidade do evento, ou seja, a chance probabilística individual de uma URL ser escolhida, representada por uma constante 0.50 (50% de probabilidade). Por fim, a **Amostra #3** é o resultado da coleta dos JSON periódicos, representando 49.089 registros válidos, confirmados e com conteúdo preservado. Vale destacar que a **Amostra #3** existe devido o relatório da página **phish search** não informar a data de confirmação, obrigando o uso do JSON para responder certas características temporais.

3.3 EXTRAÇÃO DE DADOS

Como o processo precisou ser constante no ano de 2018, foi preciso realizar uma rotina com início e fim bem definidos. Portanto, existem dois marcos importantes de extração de dados, a saber: os momentos de **extração de dados da plataforma para as amostras atemporais**, realizada a cada hora e diariamente durante o ano de 2018; e um único momento de **extração de dados da plataforma para as amostras temporais**, realizada no dia 01/02/2019.

A Figura 14 ilustra cada amostra definida pelo estudo e seu conjunto de dados distintos. Por exemplo, A amostra #1 contém dados pertencentes ao escopo HTTP como a URL e código fonte, porém não armazena cabeçalho e corpo da página. A coluna *PhishTank Info* é

Samples	HTTP				PhishTank Info							Transitional	Extraction type	
					Submission time	Verification Time	Status		Confirmation		Target			
	URL	Code	Header	Body			Online	Offline	Valid	Invalid			Auto	Manual
Sample #1	✓	✓	✗	✗	✓	✗	✓	✓	✓	✗	✗	✓	✓	✗
Sample #1.1	✓	✓	✓	✓	✓	✗	✓	✗	✓	✗	✗	✓	✗	✓
Sample #2	✓	✓	✗	✗	✓	✗	✓	✓	✗	✓	✗	✓	✓	✗
Sample #2.1	✓	✓	✓	✓	✓	✗	✓	✗	✗	✓	✗	✓	✗	✓
Sample #3	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	✓	✗	✓	✗

Figura 14 – Definição de *phishing* info

Community-based strategy	Extract approach	Extract element	Sample #1	Sample #1.1	Sample #2	Sample #2.1	Sample #3
C01. Default HTTP port explosure	URL-based	URL part 1	✓	✗	✓	✗	✗
C02. Double slash path	URL-based	URL part 2	✓	✗	✓	✗	✗
C03. Duplicate URL entry	URL-based	URL part 1 & 2	✗	✗	✗	✗	✓
C04. Incident response for community	Context-based	Submission & Verification Time	✗	✗	✗	✗	✓
C05. Precision for community	Context-based	HTTP Status Code	✓	✗	✓	✗	✗
Life-cycle	**Extract approach**	**Extract Element**	**Sample #1**	**Sample #1.1**	**Sample #2**	**Sample #2.1**	**Sample #3**
C06. Activity time	Context-based	HTTP Status Code	✓	✗	✓	✗	✗
C07. Age of Domain	Context-based	WHOIS	✓	✗	✓	✗	✗
C08. Cloning strategy	Content-based	HTTP Status Code	✗	✗	✗	✗	✓
C09. SEO score	Context-based	Alexa Service	✗	✓	✗	✓	✗
C10. Volatility	Context-based	HTTP Status Code	✓	✗	✓	✗	✗
Target Profile	**Extract approach**	**Extract Element**	**Sample #1**	**Sample #1.1**	**Sample #2**	**Sample #2.1**	**Sample #3**
C11. Content page most exploit	Content-based	HTTP Body	✗	✓	✗	✓	✗
C12. Host most exploit	URL-based	URL Domain (Host)	✗	✗	✗	✗	✓
C13. Language most exploit	Content-based	HTTP Body & Header	✗	✓	✗	✓	✗
C14. Seasonality most exploit	Context-based	Submission Time	✓	✗	✓	✗	✗
C15. Service most exploit	Content-based	HTTP Body	✗	✗	✗	✗	✓
C16. TLD most exploit	URL-based	URL Domain (TLD)	✗	✓	✗	✓	✗
URL black list bypass	**Extract approach**	**Extract Element**	**Sample #1**	**Sample #1.1**	**Sample #2**	**Sample #2.1**	**Sample #3**
C17. Encoded exploit	URL-based	URL part 1	✓	✗	✓	✗	✗
C18. IP address explosure	URL-based	URL part 1	✓	✗	✓	✗	✗
C19. Shortened URL	URL-based	URL part 1 & 2	✓	✗	✓	✗	✗
C20. URL with variables	URL-based	URL part 2	✓	✗	✓	✗	✗
URL morphology	**Extract approach**	**Extract Element**	**Sample #1**	**Sample #1.1**	**Sample #2**	**Sample #2.1**	**Sample #3**
C21. Amount of separators	URL-based	URL part 1 & 2	✓	✗	✓	✗	✗
C22. HTTP with specification port	URL-based	URL part 1	✓	✗	✓	✗	✗
C23. URL size	URL-based	URL part 1 & 2	✓	✗	✓	✗	✗
User susceptibility	**Extract approach**	**Extract Element**	**Sample #1**	**Sample #1.1**	**Sample #2**	**Sample #2.1**	**Sample #3**
C24. Browser punycode exploit	URL-based	URL part 1	✓	✗	✓	✗	✗
C25. Concatenate subdomains	URL-based	URL part 1	✓	✗	✓	✗	✗
C26. Domain with reputation	URL-based	URL part 1	✗	✓	✗	✗	✗
C27. HTTP tunneling	URL-based	URL part 1	✓	✗	✓	✗	✗
C28. Malicious browser-based code	Content-based	HTTP Body & Header	✗	✓	✗	✗	✗
C29. URL redirection	URL-based	URL part 2	✓	✗	✓	✗	✗
C30. URL spoofing	URL-based	URL part 1 & 2	✗	✓	✗	✗	✗

Figura 15 – Relação de categorias e respectivas amostras

composta por dados fornecidos pela plataforma *PhishTank* como a data de submissão, data de verificação, status, confirmação e marca alvo, conforme na Seção 3.2.2.

A coluna Transitional indica se os dados são temporais ou atemporais. A medida que o tempo se passava, um *phishing* recém capturado como *online* poderia ficar *offline*, e esse tipo de comportamento precisava ser considerado em características temporais, a exemplo da *C10. Volatility* que investiga quantos *phishing* estão *online* e *offline* no final de 2018. Todavia, outras características, a exemplo de C10. Clone strategy, é tida como atemporal porque contabiliza ocorrências de clone durante 2018. Ou seja, aspectos temporais poderiam enviesar certos resultados, por isso a necessidade de amostras distintas sobre as mudanças de status. Por fim, o método de extração teve dois tipos: manualmente, de forma subjetiva; e automatizada, de forma objetiva através de um algoritmo.

A Figura 15 ilustra as características com seus respectivos contextos de extração e amostras utilizadas para a obtenção dos dados. A ordem de exibição e análise dos dados serão apresentados respeitando a estrutura da taxonomia da Seção 3.1.3.

Community-based strategy	Scale			URL black list bypass	Scale		
	weak	moderate	strong		weak	moderate	strong
C01. Default HTTP port explosure	✓	✗	✗				
C02. Double slash path	✗	✓	✗	C17. Encoded exploit	✗	✗	✓
C03. Duplicate URL entry	✓	✗	✗	C18. IP address explosure	✓	✗	✗
C04. Incident response for community	✗	✗	✓	C19. Shortened URL	✗	✗	✓
C05. Precision for community	✗	✓	✗	C20. URL with variables	✗	✓	✗
Life-cycle	**weak**	**moderate**	**strong**	**URL morphology**	**Scale**		
C06. Activity time	✗	✗	✓		weak	moderate	strong
C07. Age of Domain	✗	✗	✓				
C08. Cloning strategy	✗	✗	✓	C21. Amount of separators	✗	✓	✗
C09. SEO score	✗	✓	✗	C22. HTTP with specification port	✓	✗	✗
C10. Volatility	✗	✗	✓	C23. URL size	✗	✓	✗
Target profile	**Scale**			**User susceptibility**	**Scale**		
	weak	moderate	strong		weak	moderate	strong
C11. Content page most exploit	✗	✓	✗	C24. Browser resource exploit	✗	✗	✓
C12. Host most exploit	✗	✗	✓	C25. Concatenate subdomains	✗	✗	✓
C13. Language most exploit	✓	✗	✗	C26. Domain with reputation	✗	✗	✓
C14. Seasonality most exploit	✗	✗	✓	C27. HTTP tunneling	✗	✓	✗
C15. Service most exploit	✗	✗	✓	C28. Malicious browser-based code	✗	✗	✓
C16. TLD most exploit	✗	✗	✓	C29. URL redirection	✓	✗	✗
				C30. URL spoofing	✗	✗	✓

Figura 16 – Escala de relevância das características

3.4 RESULTADOS OBTIDOS

Para tanto, foi adotada a metodologia *Goal Question Metric* (GQM), proposta por Basili et. al (BASILI; CALDIERA; ROMBACH, 1994), com o intuito de proporcionar um formalismo e planejamento quanto à medição dos resultados a serem extraídos e interpretados. Cada uma das 6 categorias definidas pela taxonomia representa um tipo de **objetivo** a ser avaliado sobre as amostras, que atuam como **objetos de medição**. Na mesma linha, cada característica representa um **questionamento**, e cada comportamento foi uma **métrica**, totalizando 85.

O estudo empírico conseguiu obter informações sobre comportamentos das 30 características exploradas, das quais servirão de fundamentação para as considerações e decisões a serem tomadas pela solução proposta da Tese, com resultados quantitativos e qualitativos.

3.4.1 Qualitativos

Essa seção descreve os resultados obtidos através da análise qualitativa obtida pelo estudo empírico, observando aspectos subjetivos e de interpretações obtidas no processo de extração.

3.4.1.1 Relevância das Características

O estudo busca, com base nos dados obtidos, expor um veredito sobre a **relevância** da característica, em uma escala entre *WEAK*, *MODERATE* e *STRONG*, conforme ilustrado na Figura 16. O critério para classificar a relevância considera aspectos quantitativos e qualitativos, por exemplo, um resultado quantitativo, sendo discreto ou expressivo, pode influenciar consideravelmente a relevância, contudo, comportamentos temporais ou aspectos do contexto, quando aplicáveis, podem equilibrar essa classificação.

Diante dos resultados obtidos, o estudo ponderou o nível de relevância de cada característica entre *WEAK*, *MODERATE* e *STRONG*. Essa análise não se limita apenas a aspectos

quantitativos, já que, em determinadas situações, aspectos subjetivos, como o conteúdo e contexto, foram determinantes, resultando assim em uma análise objetiva e subjetiva. Através dos dados apresentados, foi possível observar que boa parte da relevância se concentra nas categorias "Life-cycle", "Target profile" e "User susceptibility".

As características de "Life-cycle" trouxeram à tona dados críticos como a atividade, idade do domínio, clonagem e a volatilidade, comportamentos que podem ser determinantes para classificar uma página suspeita como fraude. Na mesma linha, a categoria "Target profile" evidenciou que aspectos como host, sazonalidade, tipo de serviço e registro de domínio esboçam tendências nos ataques. Por fim, a categoria "User susceptibility" destaca que o abuso de recursos do navegador, concatenação de subdomínios, sequestro de domínio, código malicioso e *typosquatting* na URL são fatores que podem ser decisivos ao veredito de uma página.

Em relação a categoria "URL blacklist bypass", houve caso de fraca relevância devido o comportamento em questão não ter sido muito explorado nos registros da amostra, como os casos de exposição de IP. Já as ocorrências de manipulação de *path* e *querystring* são situações que tem como principal resultado a duplicidade de URL, sendo sanado com políticas que identificam a fraude pelo domínio da URL.

Em relação a "Community-base strategy", de fato são problemáticas específicas de um cenário, ou seja, não significa que suas consequências serão propagadas em uma escala global ou mesmo que os aspecto abordados também ocorram em uma outra plataforma, justificando assim sua baixa média de relevância. Já as características de "URL morphology" tem sua importância, contudo, suas particularidades não representem diretamente uma ação maliciosa, mas sim, consequências de outras investidas, justificando assim seu baixo índice de relevância.

3.4.1.2 Relações e Similaridades

Essa seção descreve as relações observadas entre as características. Tais aspectos podem influenciar diretamente ou indiretamente o resultado de cada característica, além de impactar em uma ou mais características distintas. Não obstante, a relação pode ser cruzada com categorias distintas, apresentando maior sensibilidade da taxonomia em relação aos aspectos de similaridades entre as categorias. Na Figura 17 é possível observar a ocorrência de relações e similaridades entre as 30 características.

Certas combinações descrevem a utilização de palavras-chave de um idioma para atrair a atenção das vítimas (C13 e C30), seja utilizando valores arbitrários em parâmetros da URL (C20) ou praticando *typosquatting* na URL (C30) fazendo alusão a serviços populares (C15 e C30). Além disso, devido a pouca acuidade dos registradores de domínio, os mal intencionados registram domínios visando vantagens. Uma delas seria a redução do tamanho da URL (C16, C23), e consequentemente, aumentar o score SEO (C09). A prática de registrar domínios também oferece liberdade para explorar a composição das palavras no domínio de uma URL (C16, C25) com *cybersquatting*. Todavia, certas características podem aumentar as suspeitas, como o uso exaustivo de separadores (C21) e subdomínios (C25), impactando no tamanho da

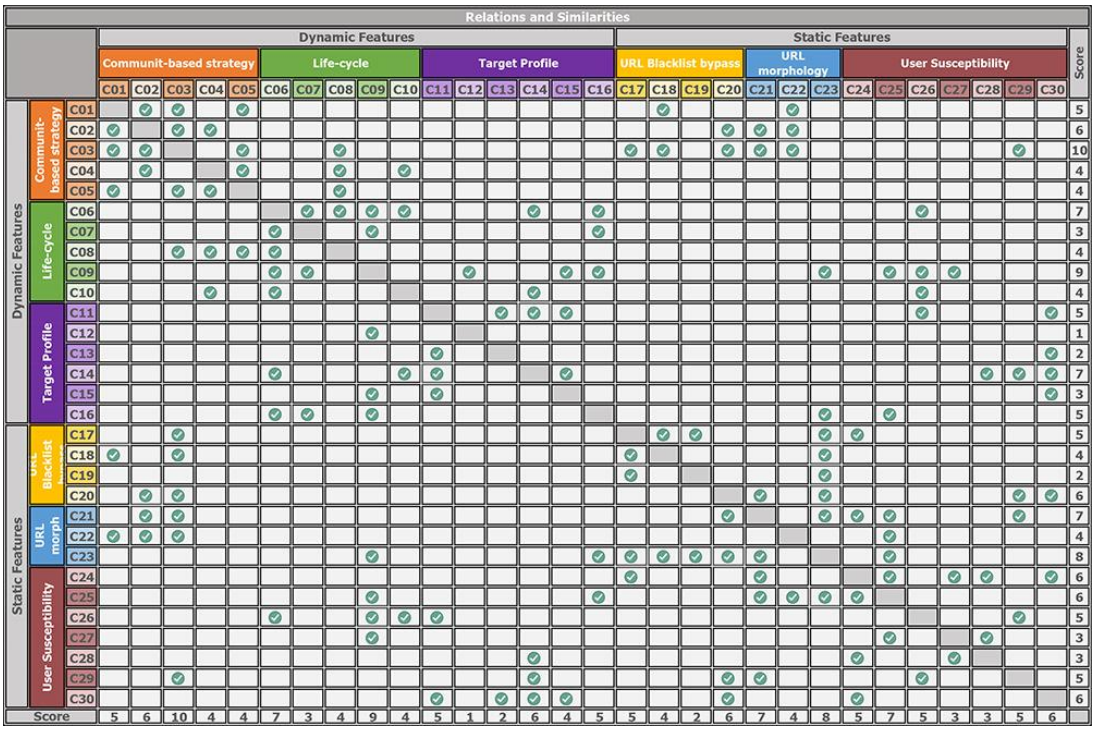

Figura 17 – Relações e similaridade entre as 30 características

URL (C23), e os mal intencionados parecem se preocupar com esse balanceamento.

Outras combinações relatam que parte dos *phishing* confirmados não duram o tempo de vida suficiente para receberem o veredito final sobre a confirmação de sua denúncia. Ou seja, em muitos casos, a denúncia submetida na plataforma acaba por receber um veredito quando a URL da mesma já encontra-se *offline*. Por exemplo, a C04 descreveu que 12.15% dos *phishing* válidos levam entre 7 e 15 dias para serem confirmados na plataforma. Todavia, a característica C06 evidenciou que 24.87% dos *phishing* válidos possuem um tempo de atividade entre 15 dias a 1 mês, ou seja, essa relação apresenta um problema crônico. Consequentemente, isso implica dizer que a lista de bloqueio acaba por armazenar um amontoado de *phishing* offline (C04, C10) e que a contribuição da comunidade precisa ser melhorada (C04, C05).

Além disso, seria interessante a adoção de melhores estratégias na plataforma de denúncias no intuito de evitar votações desnecessárias, como os casos de duplicidade (C03) que foram observados e outros (C01, C02, C08, C17, C18, C20) que trazem o aumento de duplicidades. Por exemplo, para as ocorrências das características C01 e C02, ao gerar o *hash*, a plataforma *PhishTank* considera a URL completa, portanto, qualquer variação na URL torna-se suscetível a *bypass* e que de certa forma, representa duplicidade, gerando esforço desnecessário para uma votação anteriormente confirmada.

Demais combinações visam explorar os aspectos da reputação do serviço, com uma engenharia social sobre o perfil alvo (C11, C13), aliada a técnicas que aumentam a fidedignidade (C24, C27). Essa prática fortalece os ataques que são direcionados (C28) a uma organização, conhecidos como *Spear Phishing*. Não obstante, ao invés de registrar, os atacantes podem

utilizar uma URL legítima que dispara um redirecionamento (C29) para uma determina URL que encontra-se como valor de algum parâmetro GET, se aproveitando assim da reputação do domínio (C26) da URL legítima em questão.

Todavia, o atacante pode sequestrar um domínio existente através de alguma vulnerabilidade, a exemplo de injeção de arquivos em uma sessão de *upload*. O atacante utiliza o domínio e se beneficia das mesmas vantagens anteriormente mencionadas mas também se beneficia com outras, como o prestígio (C26), o tempo de atividade (C06) e baixa volatilidade (C10) contribuindo assim com o score SEO (C09), e a confiança do usuário em determinados períodos do ano (C11) em casos que o proprietário do domínio possua prestígio em algum segmento de atuação, a exemplo do e-commerce.

Outras observações remetem estratégias dos mal intencionados que consideram aspectos dinâmicos como tempo e contexto, como manter a atividade de uma fraude em servidores distintos (C06, C08), o curto tempo de vida do *phishing* (C06, C10) prevendo que o mesmo irá cair em alguma lista de bloqueio e os esforços atuantes em um determinado *phishing* possuam um curto período, fazendo o mesmo ser brevemente abandonado, justificando sua natureza volátil. Outro aspecto é o conteúdo (C11, C14) e serviço (C14, C15) que é explorado em um determinado período do ano, direcionando os ataques por sazonalidade. Por fim, demais ameaças remetem a aspectos que exploram os recursos do navegador e que poderiam ser minimizados por seus respectivos mantenedores (C24), como a política de determinados códigos javascript ou instalação de *plug-in* e extensões (C28).

3.4.2 Quantitativo

Conforme descrito anteriormente, o presente estudo empírico também resultou em dados quantitativos que foram importantes para as decisões da solução proposta, como observar aspectos relacionados a tendência e frequência nos comportamentos de cada investida maliciosa.

Esta seção detalha os resultados obtidos na categoria **Community-based Strategy**, respeitando a estrutura taxonômica proposta na Figura 10, considerando aspectos objetivos e dados estocásticos resultantes. Todavia, devido a extensa quantidade de resultados descritos em tabelas e gráficos, as outras categorias estão apresentadas no Apêndice A.

3.4.2.1 Community-based Strategy

Essa categoria agrupa características direcionadas nas falhas das **estratégias adotadas pela plataforma de denúncias**, ou seja, possíveis explorações por parte dos atacantes em eventos relacionados ao controle *anti-phishing* que a plataforma oferece.

C01. Default HTTP port explosure

Essa característica avalia casos de URL que expõem a porta padrão nos caracteres. Alguns mecanismos de lista de bloqueio armazenam o hash considerando todos os caracteres da URL. O problema desse comportamento é que se o usuário que faz a denúncia da URL informa a

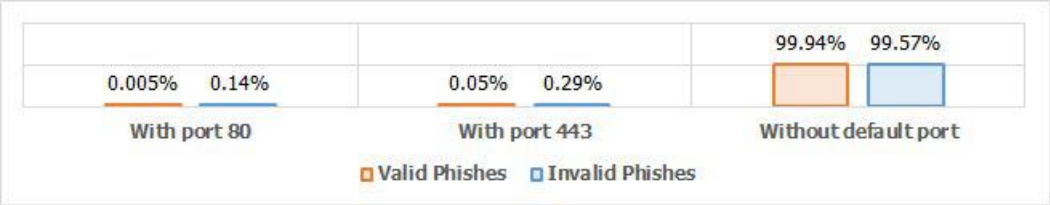

Figura 18 – C01 - ocorrências

mesma com a porta padrão em seus caracteres, a exemplo de ":80" ou ":443", faz com que o hash resultante seja diferente de um hash em que as portas não são especificadas, propiciando a ocorrência de "bypass".

Diante disso, a plataforma precisa fazer um tratamento para esse tipo de situação. Em contrapartida, existem casos em que o mecanismo bloqueia a URL considerando apenas sua primeira parte, mais especificamente o domínio da mesma. Apesar dessa estratégia atenuar o problema das portas, e outros como C18, contudo, por condenar a URL por domínio, tem sérios problemas com a característica C26, que descreve casos de sequestro. Os dados estão ilustrados na Figura 18 e o GQM na Tabela 2.

Tabela 2 – GQM de C01. Default HTTP port explosure

Objetivo 1	Analisar a confiabilidade da resposta ao incidente de *phishing* sobre o ponto de vista das abordagens baseada em comunidades.				
Questão	Q01. Como é o tratamento da comunidade sobre as entradas de URL com exposição da porta padrão?				
Métricas	[M01] Contabilizar registros que explicitam a porta 80 em *phishing* válidos.				
	[M02] Contabilizar registros que explicitam a porta 443 em *phishing* válidos.				
	[M03] Contabilizar registros que explicitam a porta 80 em *phishing* inválidos.				
	[M04] Contabilizar registros que explicitam a porta 443 em *phishing* inválidos.				
Hipóteses	Existe um interesse por parte dos mal-intencionados em propagar as URL maliciosas exibindo a porta padrão, no intuito de realizar mudanças no hash gerado, a comunidade não prevê esse tipo de situação.				
Amostras	1 e 2	**Relevance**	MODERATE	**Relations**	C02, C03, e C22
Extração	Obter a URL e analisar a presença de porta 80 ou 443.				
Limitações	-				
Observações	Pode haver casos em que a abordagem despreze a presença de porta padrão e gere o hash sem o mesmo, evitando assim um possível "by-pass" através dessa técnica. Essa característica em questão apenas analisa a presença e não a efetividade dos mecanismos de proteção.				
Análise	A exposição de porta padrão é quase que inexistente nas URL maliciosas. Ocorrências mais significativas foram encontradas em URL legítimas que expõe a porta 443.				

Conforme a Figura 18, o eixo X discrimina a porta utilizada, já o eixo Y representa a quantidade de ocorrências (em unidade e percentual). Diante disso, ficou evidente que os casos dessa característica são poucos expressivos, com apenas 0.005% de *phishing* válidos com essa exploração. Contudo, a existência propicia os problemas descritos anteriormente.

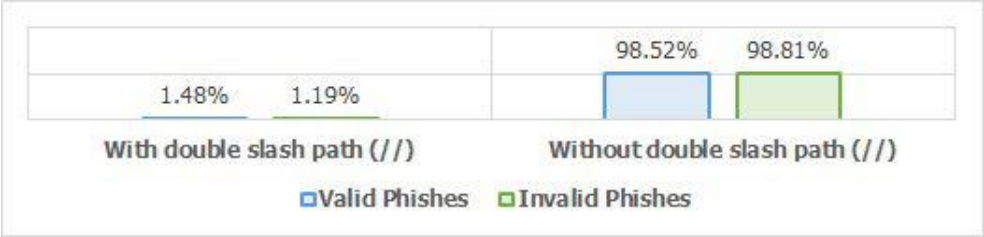

Figura 19 – C02 - ocorrências

Diante disso, a característica foi considerada com relevância *WEAK*.

Não obstante, o *SafeBrowsing* bloqueia considerando o domínio, diante disso, foi possível observar que em casos de sequestro de domínio, conforme a característica C26, todas as páginas do respectivo domínio foram bloqueadas pelos navegadores que utilizam o *SafeBrowsing*, gerando problemas de falsos positivos.

C02. Double slash path

Essa característica avalia casos em que o atacante dissemina a sua URL maliciosa com um *path* vazio, ou seja, os caracteres "//" na parte 2 da URL. Diante disso, em alguns casos, a exemplo de aplicações que não são *RESTful*, o navegador irá redirecionar o usuário desprezando o path "vazio", burlando assim a lista de bloqueio, já que a URL sem o *double slash* teria um *hash* diferente, técnica similar a característica C01. Casos dessa natureza precisam ser tratados pelo mantenedor da lista de bloqueio, evitando falsos negativos. Os dados estão ilustrados na Figura 19 e o GQM na Tabela 3.

Tabela 3 – GQM de C02. Double slash path

Objetivo 1	Analisar a confiabilidade da resposta ao incidente de *phishing* sobre o ponto de vista das abordagens baseada em comunidades.				
Questão	Q02. Como é o tratamento da comunidade sobre as entradas de URL com double slash?				
Métricas	[M05] Quantidade de URL com double slash path nos *phishing válidos*.				
Métricas	[M06] Quantidade de URL com double slash path nos *phishing inválidos*.				
Hipóteses	A comunidade não se preocupa em checar se existe alguma exploração por double slash path nas URL confirmadas.				
Samples	1 e 2	**Relevance**	MODERATE	**Relations**	C03
Extração	Verificar double slash path na String da URL.				
Limitações	-				
Observações	-				
Análise	Com base na amostra, existe um número modesto nos registros de *phishing* válidos e inválidos, com 1.48% e 1.19% respectivamente. Esses dados demonstram que a ocorrência pode ocorrer independente de ser uma ameaça ou não, contudo, a plataforma parece não prever o comportamento.				

Conforme a Figura 19, existe semelhança entre válidos e inválidos, o que chama atenção é que a plataforma não faz um tratamento prevendo esse tipo de situação, possibilitando a

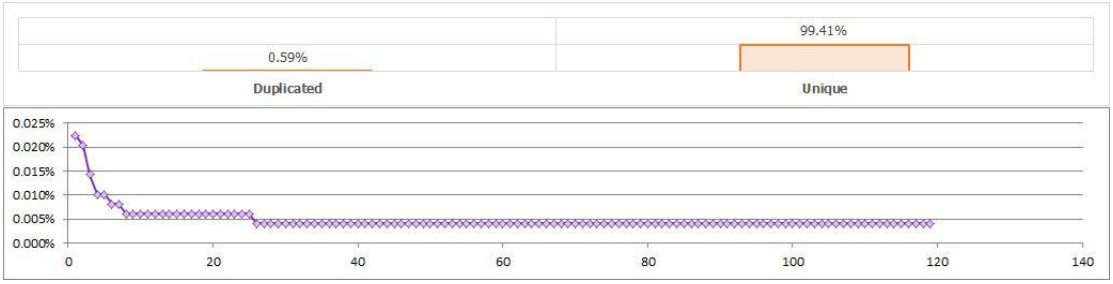

Figura 20 – C03 - ocorrências

Tabela 4 – GQM de C03. Duplicate URL entry

Objetivo 1	Analisar a confiabilidade da resposta ao incidente de *phishing* sobre o ponto de vista das abordagens baseada em comunidades.				
Questão	Q03. Como é o tratamento da comunidade sobre as entradas de URL já denunciadas?				
Métricas	[M07] Quantidade de URL duplicadas no repositório.				
Hipóteses	A comunidade gasta esforço com URL anteriormente já denunciadas.				
Samples	3	**Relevance**	WEAK	**Relations**	C01, C02, C20
Extração	Verificar duplicidade de URL por colisão da String da URL.				
Limitações	-				
Observações	-				
Análise	Com base na amostra, existe um número modesto de URL repetidas (0.24%). Uma única URL foi repetida 10 vezes nos registros da plataforma.				

ocorrência de falsos negativos, portanto, a característica foi considerada *MODERATE*.

C03. Duplicate URL entry

Essa característica remete a duplicidade de uma mesma URL na lista de bloqueio mantida pela comunidade. O intuito é evidenciar a ocorrência de votação de forma desnecessária para uma determinada URL, uma vez que a mesma já tenha sido submetida a uma votação. Os dados extraídos estão ilustrados na Figura 20 e a análise GQM na Tabela 4.

Conforme ilustrado na Figura 20, o gráfico de colunas descreve que a comunidade apresenta poucos registros de *phishing* duplicados, 0.59% ou 288 duplicidades. Um comportamento que talvez justifique a ocorrência seria os casos de *phishing* confirmados que, posteriormente ficam *offline*, sendo novamente denunciados e submetidos ao sistema de votação. Na mesma figura, o gráfico de dispersão contabiliza a repetição individual de cada *phishing* duplicado. O eixo X descreve o quantitativo em percentual, já o eixo Y apresenta o montante de URL que foram duplicadas, totalizando 119. A soma dessas duplicações individuais resulta em 288.

Devido aos números de duplicados não serem expressivos (0.59%), considerando o total da amostra, a característica foi considerada com relevância *WEAK*, contudo, fica evidente um esforço desnecessário desempenhado na plataforma, e, considerando que a votação não tem prazo estabelecido de término, consequentemente haverá atrasos desnecessários na confirmação.

C04. Incident response of community

Conforme as denúncias e votações, apresentadas na Seção 3.2.1, a diferença entre a data de confirmação e de denúncia possibilita analisar o tempo de resposta da comunidade. O fluxo do intervalo da comunidade está ilustrado na Figura 21, os dados extraídos na Figura 22 e a análise GQM consta na Tabela 5.

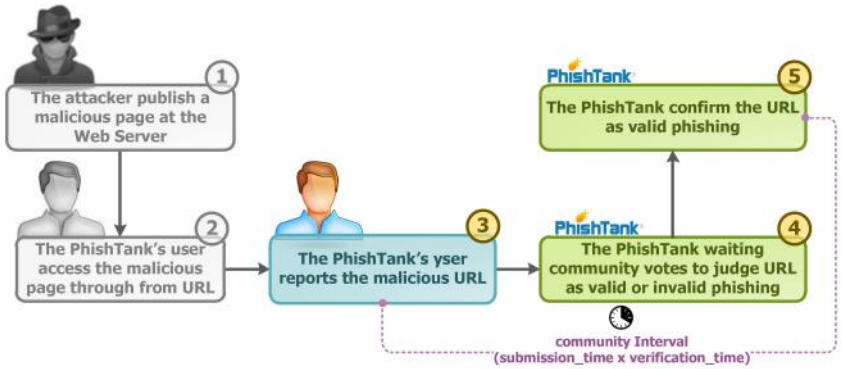

Figura 21 – Fluxograma do intervalo de denúncias da comunidade

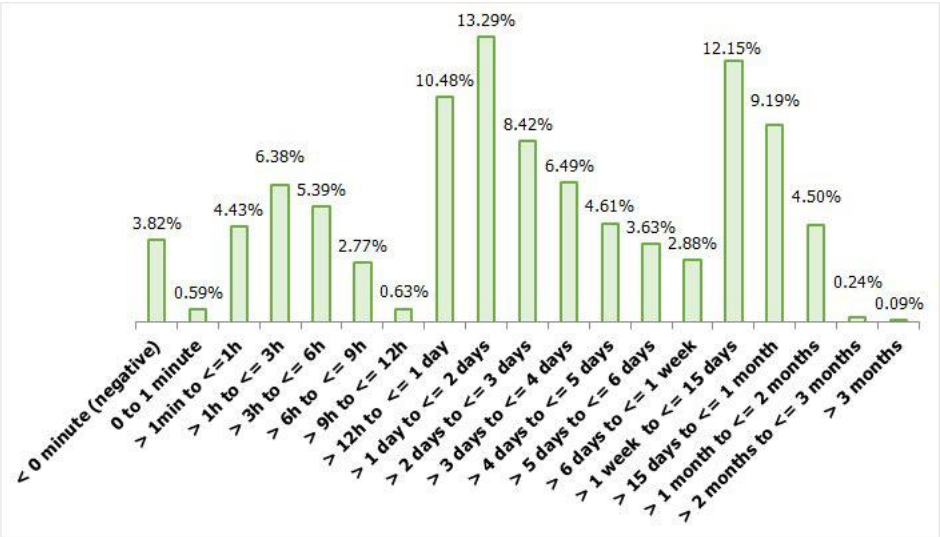

Figura 22 – C04 - ocorrências

Conforme a Figura 22, o eixo X discrimina os intervalos de tempo em escala, já o eixo Y representa a quantidade de ocorrências em percentual considerando a totalidade da amostra. Diante disso, a plataforma *PhishTank*, por não estabelecer um prazo no sistema de votação, faz com que a comunidade apresente uma fraca política de resposta ao incidente. De todos os *phishing* denunciados, apenas 20% são confirmados antes das 6 primeiras horas, contando a partir da denúncia, evidenciando que essa característica tem uma relevância *STRONG*.

Foi possível observar que 50% dos *phishing* válidos levam entre 9 e 168 horas (1 semana) para serem confirmados, caracterizando uma janela de vulnerabilidade. Outro fato que chama

Tabela 5 – GQM de C04. Incident response for community

Objetivo 1	Analisar a confiabilidade da resposta ao incidente de *phishing* sobre o ponto de vista das abordagens baseada em comunidades.				
Questão	Q04. Qual o intervalo de transição de *phishing* submetido para confirmado?				
Métricas	[M08] Tempo de resposta da comunidade para concluir a votação de um possível *phishing*.				
Hipóteses	A comunidade apresenta uma considerável janela de vulnerabilidade quanto a *phishing* 0-day.				
Amostras	3	**Relevance**	STRONG	**Relations**	C06, C10
Extração	Subtrair as datas extraídas.				
Limitações	-				
Observações	Alguns resultados tinham valores negativos, esses foram descartados, uma vez que não faz sentido o mesmo ser confirmado antes de ser submetido.				
Análise	Foi possível observar um considerável *delay* por parte da plataforma em resposta a um determinado *phishing* existente. Alguns dados são bastante preocupantes, como o caso de 10.67% levarem de 12 a 24h para terem um parecer na sua votação. Na mesma linha, 13.48% levaram 2 dias, e o que mais merece destaque é que 11.78% levam entre 7 a 15 dias para serem confirmados.				

atenção é que quase 4% dos registros estão com a data de confirmação menor que a data de submissão, resultando em valores anômalos no gráfico. Por fim, os 26% restantes levaram mais de uma semana ou até mesmo meses para serem confirmados.

C05. Precision of community

Essa característica avalia a precisão da comunidade em relação a falsos positivos das URL que são submetidas. Esse aspecto é importante para a plataforma, uma vez que representa a precisão com relação aos *phishing* válidos e inválidos que são denunciados. Os dados extraídos seguem na Figura 23 e a análise GQM é descrita na Tabela 6.

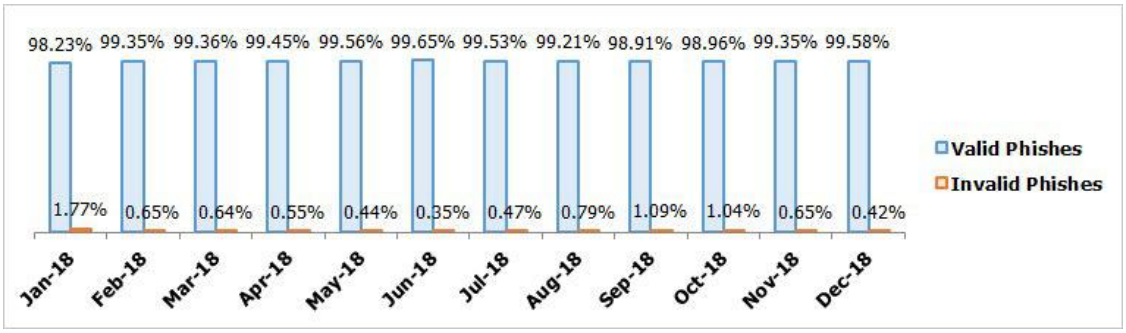

Figura 23 – C05 - ocorrências

Conforme descreve a Figura 23, o eixo X discrimina o intervalo em meses durante o ano de 2018, já o eixo Y representa a quantidade de ocorrências em percentual considerando a totalidade da amostra. Diante disso, a plataforma apresentou uma boa precisão sobre as denúncias que são recebidas, com um desvio padrão de 0.54 sobre os resultados de *phishing* confirmados. Contudo, apesar de discretos, ainda sim a plataforma não apresenta um sólido

Tabela 6 – GQM de C05. Precision of community

Objetivo 1	Analisar a confiabilidade da resposta ao incidente de *phishing* sobre o ponto de vista das abordagens baseada em comunidades.				
Questão	Q05. Qual o índice da comunidade sobre os falsos positivos que são denunciados?				
Métricas	[M09] Número de denúncias de *phishing* válidos.				
	[M10] Número de denúncias de *phishing* inválidos.				
Hipóteses	A comunidade submete um considerável número de falsos positivos.				
Amostras	1 e 2	**Relevance**	MODERATE	**Relations**	C10
Extração	Subtrair a quantidade de válidos e inválidos.				
Limitações	-				
Observações	Alguns registros constavam tanto em válidos como em inválidos, esses foram excluídos dos válidos durante a definição das amostras.				
Análise	Diante as denúncias válidas e inválidas, a taxa de denúncias inválidas é relativamente baixa, apresentando média de 0.74%. Além disso, a base da plataforma demonstra valores constantes quanto a taxa de válidos submetidos.				

processo quanto as denúncias indevidas, necessitando que um voluntário identifique e registre os casos, portanto, o peso desse comportamento pode ser considerado *MODERATE*.

3.5 AMEAÇAS E LIMITAÇÕES DO ESTUDO EMPÍRICO

Essa seção descreve as ameaças e limitações a serem consideradas pelo estudo, as mesmas foram agrupadas por objetivos e fases da metodologia.

3.5.1 Ameaças sobre as características

Certas considerações adotadas pelo estudo fizeram com que algumas características, presentes na literatura, não fossem abordadas pelo estudo. Primeiramente, por definição de escopo, o estudo limitou a quantidade de características se adequando para o tempo planejado de extração e adotou como critério as características que fossem mais influenciadas pelas tendências com o passar dos anos. Por exemplo, a característica *Google page rank* é bastante presente em obras literárias, contudo, tornou-se obsoleta desde que o mantenedor do serviço decidiu não mais disponibilizar essa informação, no dia 18 de abril de 2016(Dunlop; Groat; Shelly, 2010).

Além disso, certas características, a exemplo da C07. Age of Domain, necessitou consultar dados através do protocolo *WHOIS*, e tal processo pode haver alguns entraves, a exemplo de domínios que são privados ou que não pertencem ao mal intencionado, como os casos de sequestros de domínio ou *host* oferecidos como serviço de hospedagem. Devido a esse comportamento, foi preciso realizar uma extração mais subjetiva, analisando o conteúdo de forma manual, justificando assim a utilização de uma amostra reduzida.

3.5.2 Ameaças sobre a taxonomia

Analisando a Figura 10, a mesma classifica as categorias com o propósito de analisar, separadamente, o contexto e os caracteres da URL. Apesar de desmembrados, é possível observar uma intersecção entre elas, casos em que uma determinada característica influencia em outras. Esses comportamentos foram identificados no campo "Relations" das tabelas da Seção A, a exemplo de C17, C19 e C25 que uma vez presentes, influenciam consideravelmente a característica C23. Importante frisar que apesar de segmentados, ambas não são sobrepostas na estrutura da taxonomia, atendendo assim aos pré-requisitos apresentados na Seção 3.1.3.

3.5.3 Ameaças sobre a definição de dados e amostras

O processo de geração de novos JSON considerou eventos dos quais o *PhishTank* **removia** ou **adicionava** registros de forma periódica. Não é possível responder com precisão os motivos dessas atividades, contudo, alguns comportamentos podem justificar.

Em relação as remoções, o sistema baseado em votação visa minimizar a ocorrência de falsos positivos, contudo, ainda não isenta essa possibilidade, portanto, a plataforma disponibiliza uma seção em que um usuário pode alertar sobre um julgamento inadequado. A própria plataforma declara que esse tipo de denúncia é levada muito a sério[12], e caso seja confirmado o equivoco, a URL em questão é mudada de *phishing* válido para inválido, ou seja, o registro acaba por ser **removido** na próxima geração do arquivo JSON. Até porque, um falso positivo de um site genuíno e muito popular afetaria a credibilidade da plataforma.

Já em relação as adições, além do processo do surgimento de novos *phishing* na Web, o arquivo JSON pode receber *phishing* remanescentes de votações pendentes anteriormente. Por exemplo, uma URL foi submetida na plataforma, mas a mesma só será considerada no arquivo JSON quando a plataforma tiver um veredito sobre a votação da mesma, o que pode levar horas ou até mesmo dias. Ou seja, o tempo de transição do status "inválido" para "válido" pode variar, ocasionando a presença de novos registros em arquivos futuros gradativamente.

Esses comportamentos significam que, em comparação aos meses mais recentes, os mais antigos acabam sendo mais suscetíveis a remoções, conforme explicado na Seção 3.2.2. Como forma de constatação, realizando uma pesquisa em "phish search", conforme apresentada na Seção 3.3, foi possível observar que os meses de janeiro e fevereiro possuíam, respectivamente, 11,503 e 18,953 registros. Portanto, esse estudo realizou um monitoramento periódico durante todo o ano de 2018, observando cada atualização do arquivo JSON e observando a adição de novos *phishing* e analisando a remoção de antigos.

3.5.4 Ameaças sobre o resultado dos dados

Por conter aspectos subjetivos, a escala proposta pelo estudo, *WEAK*, *MODERATE* e *STRONG*, resultam em vereditos definidos por interpretações oriundas de observações analisadas. O es-

[12] https://www.phishtank.com/developer_info.php

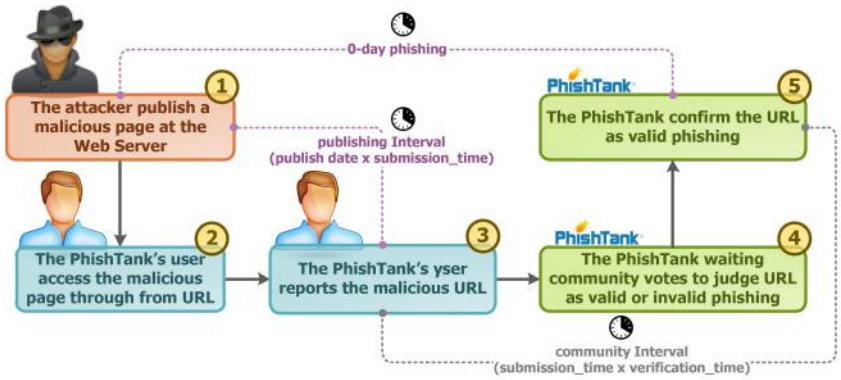

Figura 24 – Outros marcos observados no ciclo de vida do phishing

tudo cobriu contextos que pertencem a escopo do mesmo, podendo haver interpretações distintas caso outro tipo de contexto fosse considerado. Essas limitações serão apresentadas segmentadas seguindo a estrutura proposta pela taxonomia.

3.5.4.1 Ameaças nos resultados de Community-based strategy

Conforme ilustrado na Figura 24, caso houvesse como extrair as informações do domínio, existiriam alguns marcos importantes a serem analisados, como o intervalo entre as etapas 1 e 3, que seriam a janela da publicação do *phishing* na Web até o momento em que o mesmo foi denunciado na plataforma, estimando assim o **tempo médio de denúncia**. Na mesma linha, o intervalo entre as etapas 1 e 5 que seriam o **período do *phishing* 0-day**, ou seja, o tempo médio que um *phishing* fica imune da lista de bloqueio.

Contudo, para maior precisão sobre o tempo de denúncia e *0-day*, seria necessário observar a data do início da atividade do *phishing*. Uma maneira de obter essa informação seria através do "Creation Date" obtido pelo protocolo *WHOIS*. Contudo, esse evento não pode garantir precisão quanto a atividade maliciosa, ou seja, o "Creation Date" não sugere um marco de práticas ilícitas de uma página, por exemplo, poderia haver um momento de ociosidade até iniciar com as atividades fraudulentas.

Outra característica interessante para investigação seria a **sincronização entre a base de dados da plataforma e o respectivo navegador** que utiliza a plataforma como apoio ao mecanismo de proteção. Por exemplo, na página do *PhishTank*, determinados *phishing* estavam confirmados, mas não foram reconhecidos como uma ameaça quando acessados através do navegador Opera. O motivo seria o atraso na sincronização da lista de bloqueio do navegador com os registros do respectivo repositório, evidenciando assim um problema crônico que merece ser investigado. Contudo, para maior precisão sobre o atraso na sincronização, seria necessário comparar em cada *phishing* a data e hora do acesso contra a data e hora da confirmação no repositório, obtendo uma diferença.

3.5.4.2 Ameaças nos resultados de Life-cycle

Uma vez que o *phishing* assume o status "offline", o mesmo é desprezado por JSON futuros, porém, o motivo da indisponibilidade pode ter sido por algo temporário, e se o mesmo retornar a atividade, a plataforma continuará a assumir o *phishing* como "offline". Provavelmente, essa mesma URL maliciosa terá que novamente ser submetida a todo o processo de denúncia e votação para ser considerada no JSON novamente. Diante disso, as ferramentas que fazem interoperabilidade com a API precisam armazenar inúmeros *phishing* antigos, evitando assim a possibilidade de falsos negativos. O resultado é uma imensa base de *phishing* consultada a todo instante, mesmo quando muitos são inativos, resultando em um **esforço desnecessário e que poderia ser evitado**, tal comportamento poderia ser investigado futuramente.

Outro comportamento seria avaliar a *frequência de mudanças do* phishing durante o tempo, através do campo "Updated Date" do protocolo *WHOIS*. Todavia, o escopo dessas operações remetem ao DNS e não a página propriamente dita, ou seja, só aplicável para casos de *phishing* com domínio registrado. Ainda sim, não teria tanta precisão, já que a atividade ilícita poderia ocorrer antes ou depois do registro. Da mesma forma, uma mudança em "Update Date" não implica dizer que o conteúdo da página foi modificado, possibilitando enviesar os resultados.

Em relação a monitorar as mudanças do *phishing*, a Figura 53 descreve os surgimentos dos clones durante o ano através do *hash*, porém, não observa as mudanças dos respectivos clones. Uma alternativa seria observar as mudanças e monitorar a atividade do *phishing* também pelo *hash*, porém, a mesma não foi contemplada no estado atual do estudo, por definição de escopo.

Todavia, em certos casos, controlar eventos de um *phishing* através de seu *hash* pode ter suas limitações. Por exemplo, qualquer informação adicional, como por exemplo um *banner* publicitário, que não necessariamente seria inserido pelo autor da fraude, mas sim pelo servidor de hospedagem, já seria suficiente para modificar o *hash* gerado do contéudo da página. Além disso, informações dinâmicas relacionadas, por exemplo, informar a data atual ou dados extraídos de *cookies*, também poderia modificar o *hash*. Diante disso, tais comportamentos podem reduzir o quantitativo de clones capturados. Em contrapartida, também existe a chance de existir colisão de hash, ou seja, páginas distintas com um mesmo hash, aumentando assim a quantidade de clones erroneamente. Contudo, a chance disso acontecer é desprezível (2^128) com o algoritimo SHA-256.

3.5.4.3 Ameaças nos resultados de Target profile

Um fator oportuno para os crimes cibernéticos é a diversidade na conduta quanto a punições por crimes digitais que são ocasionadas pelas políticas de um determinado estado ou país (ZDNET, 2016), diante disso, seria interessante analisar a **região de atividade** mais explorada pelos mal intencionados. Diversos servidores que armazenam páginas maliciosas encontram-se geograficamente em países onde não existe quase ou nenhuma legislação que responda por crimes dessa natureza. Isso justifica o fato de existir *phishing* que, mesmo já denunciados,

ainda encontram-se ativos por mais de 1 ano.

Apesar de analisar o idioma e TLD da região, identificando assim o público alvo, ainda sim não significa que a fraude esteja em atuação na respectiva região. Muito provável que haja diversos casos de *phishing* que tem como alvo uma localidade, mas esteja hospedado em um servidor com operação em outra região. Diante disso, por questões de escopo, não foi possível identificar a região de atividade do *phishing*. Por fim, a característica C12, que avalia o serviço de hospedagem mais explorado, existiam casos em que o *phishing* possuía domínio registrado e configurado com mascaramento[13]. Casos desse tipo impossibilitavam saber o servidor de hospedagem da fraude, portanto, acabaram não sendo considerados.

3.5.4.4 Ameaças nos resultados de URL blacklist bypass

Algumas URL utilizam um serviço de encurtamento que não tinha mais qualquer atividade, nesses casos não era possível desencurtar a URL para visitar a página genuína ou analisar o nível de profundidade aplicado. Portanto, esses casos foram computados apenas nas ocorrências de URL encurtadas. No caso do "goo.gle", apesar de não ter mais a opção de criar encurtamentos, ainda sim, o serviço faz o desencurtamento de URL já encurtadas anteriormente, possibilitando assim fazer análise de profundidade.

3.5.4.5 Ameaças nos resultados de URL morphology

Quanto aos caracteres separadores, seria interessante investigar outros, menos presentes, porém, recorrentes, observando assim padrões não analisados. Com relação ao tamanho da URL, estabelecer que uma quantidade de caracteres define como grande ou pequena pode ser subjetivo. Diante disso, o estudo se baseou em uma média obtida através da literatura, transformando assim essa característica de contínua para categórica.

3.5.4.6 Ameaças nos resultados de User susceptibility

Uma característica interessante para análise seria mensurar a política de detecção de *malware* adotada pelo navegador, observando os elementos da página e as ações do usuário, como os cliques para navegação e *download*. Não obstante, outra política a observar seria quanto a instalação e uso de *plug-ins* e extensões. Muitos serviços, principalmente os bancários, adotam estratégias de *hardening* no lado cliente através de *plug-ins* ou extensões, sejam de autoria própria ou mantida por terceiros, a exemplo dos bancos brasileiros que utilizam a solução Warsaw desenvolvida pela GAS Tecnologia [14]. Nesse contexto, seria interessante analisar a eficiência dessas soluções.

Não obstante, em relação aos redirecionamentos maliciosos, outro meio explorado, e que não fez parte do escopo desse estudo, seria o caso do atacante preencher perfis falsos em algum

[13] http://support.godaddy.com/help/article/422/forwarding-or-masking-your-domain-name
[14] https://www.dieboldnixdorf.com.br/gas-antifraude

ambiente que compartilha perfis pessoais, como redes sociais, fóruns, etc. Não é incomum esse tipo de ambiente permitir que o usuário informe uma URL em seu perfil, indicando uma possível página pessoal. Nesses casos, geralmente a respectiva aplicação não checa a procedência da URL informada, possibilitando propagar uma URL maliciosa no ambiente compartilhado.

3.6 CONSIDERAÇÕES FINAIS

Esse estudo apresentou uma metodologia para obtenção de evidências sobre determinados comportamentos a serem analisados em amostras extraídas de ambiente reais de atuação de *phishing*. Tais evidências foram descritas através de gráficos e argumentadas baseando-se em métricas GQM. Considerando que não são poucas as propostas de predição de *phishing*, contudo, o problema continua crônico nos dias atuais, justificando avaliar a aplicabilidade dessas soluções. Como muitas das soluções são guiadas por um conjunto de características, o presente estudo, como reflexão, analisa a relevância de características comumente utilizadas no processo de predição. O estudo segmentou as características em tipos e nível de classificação que consideram a estrutura léxica da URL, contexto, conteúdo e similaridades.

3.6.1 Desafios em Community-based strategy

Em relação as características baseadas em "Community-based strategy", foi possível observar um problema crônico sobre a resposta ao incidente do *PhishTank*. O sistema de votação, por não possuir um prazo estimado e nem informar a quantidade de votos para conclusão, apresenta um *delay* quanto a conclusão do veredito, em que 49.80% dos registros levaram entre 1 a 7 dias, evidenciando uma janela de vulnerabilidade. Seria razoável que mesmo o *phishing* ainda não confirmado, os mecanismos de proteção que utilizam o *PhishTank*, a exemplo do Opera, considerassem a página como suspeita, alertando que o ambiente pode ser hostil. Apesar da lacuna, a plataforma apresentou um alto índice de precisão das denúncias submetidas, apresentando um desvio padrão de 0.54 de confiabilidade.

3.6.2 Desafios em Life-cycle

Em relação as características baseadas em "Life-cycle", foi evidenciado a vida curta do *phishing*, inclusive o tempo de atividade dos *phishing* válidos foram confrontados com os *phishing* inválidos, sendo possível observar um tempo de vida muito mais curto por parte dos sites fraudadores, em que 74.89% dos *válidos* possuem um tempo de atividade entre 12h e 2 meses. Consequentemente, a volatilidade do *phishing* na transição de "online" para "offline" é bastante alta, quase 80% dos *phishing* existentes tendem a ficar *offline*. Já os inválidos, 59.82% concentram-se entre 10 e 13 meses de atividade. Não obstante, existe um número significativos de *phishing* que são clonados, demonstrando que a prática é bastante explorada e um tratamento sobre a mesma poderia reduzir esforços nas estratégias adotadas pela comunidade.

3.6.3 Desafios em Target profile

Em relação as características em "Target profile", esse agrupamento evidenciou recursos que são mais explorados pelos atacantes, possibilitando também analisar aspectos com maior suscetibilidade aos ataques. Foi possível observar que o conteúdo do tipo informações bancárias e transações financeiras são os mais explorados. Outro aspecto é que o serviço de hospedagem *000webhostapp* representou 10.09% de todas as fraudes catalogadas na amostra #3, indicando que é o serviço mais explorado. Alguns resultados talvez fossem previsíveis, como o idioma inglês americano ser o mais explorado, mas ainda sim, foi uma característica interessante porque evidenciou que o segundo idioma mais explorado é o português brasileiro, descrevendo assim que o Brasil apresenta-se de fato como uma região bastante explorada pelos fraudulentos.

Outro comportamento que reforça a teoria sobre os ataques direcionados ao Brasil foi a sazonalidade no ano calendário de incidentes de *phishing*. Os últimos meses do ano possuem bons picos de ocorrências com seus eventos fixos e programados, como *Black Friday* e natal. Contudo, casos esporádicos, como os saques de FGTS no Brasil, trouxeram números alarmantes. Na mesma linha, quanto aos serviços mais explorados, ficou evidente que os bancos brasileiros são alvos constantes. Já os domínios registrados, existe um considerável número de domínios ".br" sendo registrados ou mesmo sequestrados.

3.6.4 Desafios em URL blacklist bypass

Em relação as características baseadas em "URL blacklist bypass", foi apresentado um número muito modesto para os casos em que o *phishing* válido estava expondo a porta padrão ou o endereço IP, contudo, o mesmo se repetia em casos de *phishing* inválidos, representando assim características com relevância *WEAK*. Em contrapartida, apesar da exploração por codificação possuir poucas ocorrências, ainda sim, por se tratar algo fora do convencional, foi considerado como *STRONG*. Na mesma linha, com números razoáveis de ocorrências, as URL encurtadas e o abuso na utilização de variáveis no *path* ou *querystring*, apesar de também serem utilizadas em casos de *phishing* inválidos, em uma determinada quantidade ou profundidade essas URL tendem a sair do convencional, portanto a característica foi considerada *MODERATE*.

3.6.5 Desafios em URL morphology

Em relação as características baseadas em "URL morphology", comportamentos como a quantidade de separadores e tamanho da URL levantam suspeitas em URL de acordo com a quantidade de utilização. Em contrapartida, o número de URL que utilizavam uma porta diferente da padrão foi quase que inexistente, sendo assim, uma característica de baixa relevância.

3.6.6 Desafios em User susceptibility

Em relação as características baseadas em "User susceptibility", foi possível observar um grande número de investidas através de recursos do navegador Web que eram disparados pela própria

URL. Um dos comportamentos que mais chamaram atenção foi a concatenação de subdomínios, *cybersquatting* e *typosquatting* em domínios, aspectos que são cada vez mais explorados nos últimos anos. Em contrapartida, decidir se uma página é maliciosa ou não pela ausência ou presença do cadeado acabou perdendo a força com o passar do tempo, bem como se apoiar em score de SEO de certos serviços disponíveis. E conforme já mencionado, os casos de sequestros trouxe como destaque o Brasil como bastante explorado pelos ataques de *phishing*.

3.6.7 Proposta para novas abordagens

Por ter extraído um considerável número de *phishing* reais, a análise realizada por esse estudo considera aspectos quantitativos, a exemplo dos gráficos expostos. Não obstante, por considerar o conteúdo e contexto, bem como identificar relevância e similaridades, o estudo também oferece resultados qualitativos. Com esses dados, fica possível concluir que aspectos temporais, na perspectiva desse estudo, influenciaram na relevância das características comumente adotadas em heurísticas para modelos de predição.

O estudo pode oferecer apoio para o desenvolvimento de um avaliador de modelos dessa natureza, utilizando métricas de avaliação conforme já apresentadas como sensibilidade, especificidade e eficiência, bem como outras métricas como valor da predição e coeficiente de variação, no intuito de julgar a maturidade da precisão do novo modelo proposto. O avaliador teria o papel de atribuir pesos nas características da heurística, representando assim os fatores de relevância, propondo algo como um modelo de maturidade para novos modelos de predição. A definição de um modelo de classificação possui desafios que podem ser atenuados com os resultados desse estudo, a exemplo da (i) **categorização**, (ii) **análise de relevância** e (iii) **agrupamento das características**.

Diante de (i), uma característica pode ser considerada como uma variável que possui valor contínuo ou categórico. Um exemplo de valor categórico seria a URL ter ou não ter tunelamento (C27). Já uma variável com valor contínuo seria, por exemplo, o resultado do tamanho da URL (C23), ou seja, apresenta valores diversos para cada URL. Como o modelo de classificação precisa lidar com valores categóricos, seria necessário transformar as variáveis contínuas em categóricas. Através da estatística descritiva, os resultados desse estudo podem trazer um norte ao processo de conversão dessas variáveis, a exemplo da escala de intervalos dos tamanhos apresentada na Figura 73.

Com base em (ii), considerando o contexto *anti-phishing*, é importante atribuir um peso, conforme debatido na Seção 3.4.1.1, para que um conjunto de variáveis possa definir a classe de uma página acessada. Como o ambiente do *phishing* é suscetível a mudanças, é importante considerar, além de aspectos estáticos, como padrões léxicos do código-fonte e da URL, também observar o conteúdo e contexto do *phishing*. Em relação ao conteúdo, remete a tendências e padrões de comportamento. Já em relação ao contexto, consiste em estratégias *anti-phishing*, atividade, tempo, recursos computacionais e sazonalidade. Não obstante, outro critério importante é considerar a ocorrência da combinação de características.

Por fim, em (iii) aborda a problemática do agrupamento das características atendida pelo modelo de classificação. Nesse contexto, o (iii) serve como base de apoio para o (i) e (ii), ou seja, de posse dos dados desse estudo, é possível realizar uma análise de agrupamento (*cluster*) para garantir maior sensibilidade quanto aos aspectos de semântica e similaridades das características, conforme debatido na Seção 3.4.1.2. Na mesma linha, um agrupamento bem definido evita problemas de sobreposição de atributos que trabalham com avaliação ponderada.

Diante o exposto, esse Capítulo teve o intuito de apresentar um conjunto de características como fundamentação para a construção de um modelo de classificação de *phishing*. A metodologia foi apresentada como um estudo empírico sustentado por uma regressão logística. O resultado do estudo serviu como base para a definição das características a serem consideradas na proposta quanto a predição de *phishing*. O próximo Capítulo visa apresentar a proposta da pesquisa, descrevendo sua metodologia e justificativa das estratégias para atingir os objetivos.

4 PROPOSTA

Esse Capítulo apresenta detalhes e decisões sobre a solução proposta, intitulada Piracema.io. A metodologia é baseada em uma árvore de regras a ser processada pela análise gradual, considerando uma estrutura organizada por semântica e similaridade, e priorizando a relevância das características.

Diferente das soluções para predição de *phishing* baseadas na ausência de características, que tendem a resultar em falsos negativos, principalmente na presença de *phishing direcionados*, esta Tese visa detectar o *phishing* direcionado adotando as estratégias quantitativas e qualitativas descritas em (SILVA; FEITOSA; GARCIA, 2019a) e no Capítulo 3, visando uma maior riqueza de detalhes no intuito de explorar a suscetibilidade com base na fidedignidade.

Sendo assim, a presente Tese emprega uma **estratégia baseada em alta fidedignidade**, ou seja, quanto maior a quantidade de elementos verossímeis, maiores são as chances de uma predição bem sucedida. Vale destacar que esta proposta não substitui as abordagens existentes, mas sim torna-se aderente ao cenário como uma solução complementar para atenuar as lacunas das soluções atuais em incidentes de *phishing* direcionados.

Primeiramente, serão esclarecidas algumas **premissas**, que especificam termos e responsabilidades de cada informação, bem como regras e restrições pré-estabelecidas. Posteriormente, é apresentado o processo de **filtragem por lista de permissão**, uma funcionalidade opcional que tem papel de atenuar ocorrências de falsos positivos. Logo em seguida, é apresentada a técnica de **busca textual e visual**, processo responsável em extrair informações para detectar ataques homográficos (manobras como *typosquatting* e *cybersquatting*) e também a detecção de marcas envolvidas. Com isso, é apresentado o motor de inferência, a estrutura baseada em árvore que representa o **modelo de predição** baseado em sistema especialista. Complementando, é apresentada a solução responsável pela **gestão da filtragem por lista de permissão**, e por fim, a seção reúne discussões e considerações sobre as limitações do estado atual da proposta.

4.1 VISÃO GERAL

Este Capítulo descreve a estratégia baseada em alta fidedignidade, dirigida por um conjunto de regras, que resulta em um Sistema Especialista (SE) para detecção de *phishing*, em especial os direcionados. A Figura 25 ilustra uma visão geral (fluxograma de interação) da proposta.

Antes de explicar em detalhes a estratégia, se faz necessário apresentar algumas premissas assumidas que são essenciais para o funcionamento da solução. São elas:

1. A estratégia faz uso de uma **lista de permissão**, para aplicar exceções de checagem em certos domínios, no qual exige um **processo** para alimentá-la, visando auxiliar em acessos futuros, evitando latência e falsos positivos.

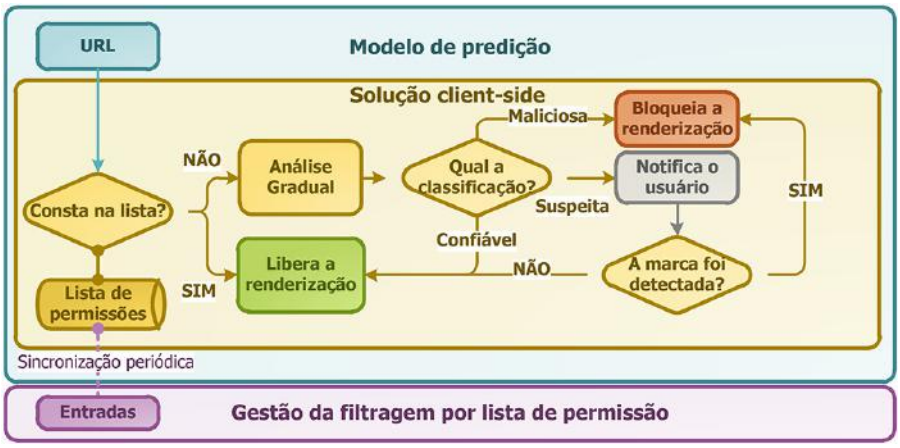

Figura 25 – Fluxograma da abordagem proposta

2. Existem **25 comportamentos** a serem analisados na página;
3. Caso exista atividade suspeita, a **entrada** é avaliada pela solução;
4. Uma entrada é o endereço de **HOST** de uma determinada marca. Para tanto, o HOST deve ter um **domínio** registrado e **certificado digital** válido (emitido e em vigor), tais informações são importantes para **detecção da marca**;
5. Características e marcas são detectadas por **análise textual e/ou visual na URL ou conteúdo da página**;
6. O processo inicia uma página com 0 pontos, e cada vez que um comportamento for detectado, à mesma é **atribuída uma pontuação**, o valor pode variar entre -3 (alto prestígio), -1 (médio prestígio), 1 (baixo prestígio) e 3 pontos (ausência de prestígio), de forma acumulativa;
7. Caso o processo de checagem seja concluído e contabilize a página com menos de 2 pontos, a mesma será classificada como **confiável**;
8. Caso o processo de checagem seja concluído e contabilize a página entre 2 a 5 pontos, a mesma será classificada como **suspeita** e o usuário alertado na renderização;
9. Durante o processo de checagem, no momento que a página atingir 6 ou mais pontos, a mesma será classificada como **não confiável**;
10. Quando o processo finalizar com a página classificada como suspeita, **caso alguma marca seja identificada**, o processo mudará a classificação da página para não confiável;
11. Quando a página é tida como não confiável, o processo de **renderização é interrompido**;

12. Visando minimizar problemas de **desempenho e privacidade**, a solução adota uma **estratégia gradual**, partindo por características estáticas e dinâmicas presentes na **URL** para então, se necessário, seguir para o **conteúdo da página**;

13. Páginas armazenadas em **HOST com muitos anos de atividade** tem penalidade reduzida (de $+3$ para $+1$) no caso de ausência do certificado digital;

14. Algumas páginas **não permitem a captura de informações do certificado**, nesses casos a punição também é reduzida (de $+3$ para $+1$);

15. Características temporais que **prestigiam a página** (-3), como o tempo de atividade do HOST ou domínio, não são aplicadas para HOSTs que estejam na *lista suspeita* ou que atuem como serviços de hospedagem;

16. A lista suspeita também é composta por domínios de serviços de hospedagem, contudo, o que faz um serviço pertencer a essa lista não é apenas a ocorrência de explorações, mas também a conduta de cada mantenedor em relação a flexibilização e mitigações pontuais em casos de ataques que partem de seus recursos;

17. Uma lista suspeita é um **conjunto de domínios que não necessariamente são fraudulentos**, mas, por terem atrativa adesão e política de uso flexível, são explorados por mal intencionados;

18. Conteúdo entende-se como as **informações** do *HEADER* e *BODY* do *HTTP response*;

19. Todas as pontuações e limiares assumidos pelo conjunto de regras foram baseados no experimento apresentado na Seção 6.

Um ponto que merece destaque é a justificativa nas atribuições de valores, como -3, -1, 1, 3, ou mesmo os limiares que classificam uma página analisada, como menor que 2 ser confiável, entre 2 e 5 suspeito e acima de 5 como não confiável. O valor 6 representa a confirmação de uma ameaça (100%). Diante disso, o valor 3 representa 50% da página já ser considerada como uma ameaça, na mesma linha, a atribuição 1 ou uma pontuação 2 representa 33% da página ser uma ameaça. Tais valores são baseados nos resultados obtidos na avaliação, através da análise de variância descrita na Seção 6.2.1 do Capítulo 6.

As 30 características extraídas do estudo empírico (Capítulo 3) serviram como base para a fundamentação dos 25 comportamentos mapeados no modelo de predição proposto. A Figura 26 ilustra a relação de cada comportamento com suas respectivas características que possuem alguma semântica ou similaridade quanto a fundamentação em sua definição.

		Comportamentos no Modelo de Predição	Características extraídas no Estudo Empírico
Escopo	URL	1. Tem ausência de domínio registrado?	C07, C09, C26
		1.1. Ausente da lista de permissões?	-
		1.2. Tem certificado inválido?	C26, C27
		1.3. O certificado foi recém validado?	C06, C07, C09
		1.4. Tem simulação TLD?	C26, C30
		1.5. Tem TLD na lista suspeita?	C15, C16
		1.6. Tem tentativa homográfica?	C30
		1.7. Tem nome muito grande?	C19, C21, C23, C30
		1.8. Tem nome encoded?	C17, C24
		1.9. Registro recém criado?	C03, C04, C05, C06, C07, C08, C09
		2. Tem subdomínio?	C25
		2.1. Tem simulação TLD?	C26, C30
		2.2. Tem tentativa homográfica?	C30
		2.3. A sintaxe é muito grande?	C19, C21, C23, C30
		2.4. Qual a quantidade de subdomínios?	C21, C23
		3. O HOST consta na lista suspeita?	C12, C15
		4. Tem IP exposto?	C18
		5. Tem tentativa homográfica no path ou querystring?	C20, C30
		6. Tem investida de redirect?	C02, C20, C29
		7. Tem uptime recente?	C03, C04, C05, C06, C07, C08, C09
		8. Tem porta específica?	C01, C22
		9. Tem referência a formulário?	C11, C13
	Conteúdo	10. O código-fonte é acessível?	C11, C28
		10.1. Tem elementos x-origin?	C28, C29
		10.2. Tem comportamento forjado?	C11, C12, C14, C28, C29
		10.3. Tem submissão de formulário?	C28, C29
		11. Tem ausência de favicon?	C09
		12. Tem conteúdo sazonal?	C11, C14, C15
		13. Foi possível obter a marca?	C11
		13.1. Tem conflito de identidade?	C11

Figura 26 – Relação entre as características do modelo de predição e as características extraídas no estudo empírico

4.2 SOLUÇÃO CLIENT-SIDE

De posse das premissas sobre a predição, o desenvolvimento da proposta inicia pelo *client-side*. Importante salientar que as fundamentações observadas no estudo empírico do Capítulo 3 tem como escopo qualquer tipo de *phishing*, mas os resultados obtidos (características observáveis) estão presentes em *phishing* direcionados e foram importantes para a fundamentação da solução *client-side*. Assim, a ideia é projetar a solução como um mecanismo que atua internamente em um navegador Web, a exemplo de outras soluções, que faz requisições remotas a um serviço externo, visando uma interoperabilidade que ofereça escalabilidade e melhorias contínuas na solução. O fluxo é ilustrado na Figura 27.

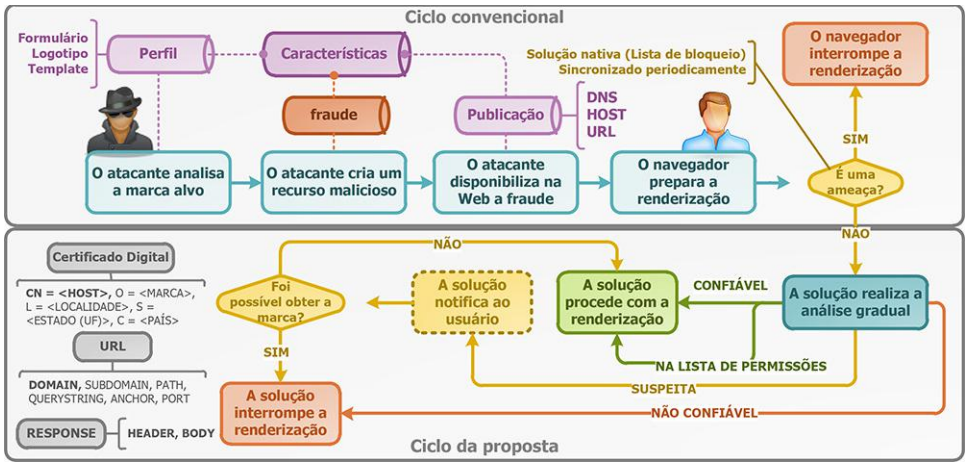

Figura 27 – Cenário de atuação da proposta

O comportamento das soluções atuais é ilustrado na parte superior da Figura 27, intitulada ciclo convencional, que se inicia na tentativa de renderizar uma página Web no navegador. Na Figura é descrita apenas a abordagem de lista de bloqueio, contudo entende-se que tal lista de bloqueio muitas vezes seja alimentada por heurísticas de predição mantida por seus respectivos criadores. Por exemplo, o *SafeBrowsing* declara que faz varreduras pela Web (GOOGLE, 2019) investigando padrões e, caso julgue como não confiável, a página é inserida em uma lista. Esse tipo de solução adota a lista de bloqueio por desempenho, mas é alimentada com heurística. Além disso, a alimentação dessas listas também é realizada por denúncias voluntárias da comunidade (WHITTAKER; RYNER; NAZIF, 2010; OPENDNS, 2019).

Por ter natureza baseada em lista de bloqueio, o ciclo convencional tem alto desempenho e por isso a solução da Tese concede prioridade a mesma no processo como um todo. Caso o ciclo convencional não tenha sido bem sucedido, o próximo passo é ser submetido ao ciclo da proposta, conforme ilustrado na parte inferior da Figura 27. A solução tem como principal atividade a análise gradual, que tem como resposta 4 possíveis classificações para a página em questão, a saber: (i) na lista de permissão, (ii) confiável, (iii) não confiável e (iv) suspeita.

A renderização da página estará condicionada ao resultado dessa classificação. No caso de (i) e (ii), ambas irão permitir a procedência da renderização. A diferença é que em (i), pelo fato da página estar hospedada em um *host* previamente cadastrado na lista de permissão, a mesma não será submetida ao processo de análise das características, reduzindo o **custo computacional** e evitando possíveis **falsos positivos**. Já em (ii), a página passou por todas as etapas da análise gradual e ao término, a mesma contabilizou um total inferior a 2 pontos. Contudo, em (iii) indica que em algum momento da análise gradual, o total chegou a 6 ou mais pontos, sendo suficiente para encerrar a análise e interromper o processo de renderização.

Por fim, em (iv) indica que a página foi submetida a todas as etapas e acabou concluída como suspeita, ou seja, um total entre 2 à 5 pontos. Neste caso, o próximo passo é identificar a marca em questão. Em caso positivo, por medida de segurança, a solução irá assumir a página como não confiável e encerrará o processo. Caso contrário, a página será renderizada, porém o usuário será notificado com um alerta sobre possível ameaça sem confirmação. O esquema de pontos atribuídos com base nas decisões é detalhado na Seção 4.2.3.

É importante explicar a decisão da solução em relação às marcas detectadas em página suspeitas. A proposta assume que as marcas detectadas são bastante visadas por mal intencionados e, consequentemente, os mantenedores das mesmas habitualmente tomam medidas mais rígidas em relação a sua identidade, como não abrir mão do uso de certificados digitais e domínios registrados, o que em tese, dificultaria a mesma ser classificada como suspeita. Portanto, a solução acha mais prudente reduzir o nível de confiança em páginas que, anteriormente classificadas como suspeitas, estejam utilizando uma marca supostamente de forma indevida. A solução também espera que páginas dessa natureza sejam previamente registradas na lista de permissão, uma vez que a exploração desses ambientes seja muito iminente.

Conforme ilustrado na Figura 27, alguns artefatos presentes na página, como o certificado

digital, a URL e o *response* HTTP são considerados na análise gradual, bem como no processo de obtenção da marca. No certificado, um campo obrigatório é o *Canonical Name* (CN), que representa o domínio correspondente que foi registrado pelo proprietário. Os demais campos não são avaliados. Já na URL, existem diversos elementos a serem considerados, como o domínio, subdomínio, *path*, *querystring*, âncoras e porta da aplicação. Em relação ao conteúdo da página, serão considerados o cabeçalho e corpo da resposta. Por fim, é importante destacar que apesar da solução *client-side* fazer interoperabilidade com um serviço *server-side*, a troca de informação limita-se ao host acessado no momento. Nenhuma outra informação é enviada e todo o processo de detecção é realizado no lado do cliente, garantindo o sigilo dos dados analisados durante o processo de proteção.

Diante o exposto, a Tese agora descreve com mais detalhes sobre cada abordagem aplicada na solução proposta, baseando-se no fluxo de processamento realizado pela análise gradual.

4.2.1 Processo de filtragem por lista de permissão

A checagem por lista de permissão tem como propósito oferecer exceções de checagem em *hosts* previamente cadastrados. O intuito é reduzir o custo computacional em processamentos desnecessários, bem como reduzir os índices de possíveis falsos positivos. Não é incomum que em repositórios de denúncia de URL maliciosas, a exemplo do *PhishTank*, usuários de forma equivocada ou proposital submetam URL legítimas e, consequentemente, os mecanismos de lista de bloqueio acabem bloqueando o acesso de forma indevida à uma página não maliciosa, ocasionando em falso positivo.

4.2.2 Busca textual e visual

Essa técnica inicialmente define alguns termos como palavras-chave para a detecção de comportamentos, denominados como *token*. Exemplo dessa natureza seria um formulário de autenticação, analisado através do código-fonte, ou termos no texto renderizado ao usuário que indicam algum tipo de procedimento com dados sensíveis. O contexto a ser analisado de tais informações pode conter um volume considerável de dados, portanto a técnica de busca por *token* garante algo direcionado, bem como estabelecer apenas o necessário a ser investigado, atenuando a quebra da privacidade. A indexação é um processo que converte determinados textos em um formato que facilite sua procura de forma mais otimizada, ou seja, trocando os textos por índices previamente definidos, que apontam para o local de um texto.

Portanto, o processo de indexação agrupa *tokens* mais importantes visando acelerar a busca por um conteúdo de interesse, assim como o funcionamento de um índice de um livro. E mais além, as informações de entrada são estruturadas através de uma abordagem denominada índice invertido, que manipula a indexação em memória de forma otimizada, permitindo que a busca seja realizada por um determinado termo, sem a necessidade de percorrer termos distintos. Além disso, outro processo importante é o denominado análise, que faz a conversão do texto em unidades para buscas elementares, denominada termos. Durante esse processo, o

texto é submetido a diversas operações, a saber: a extração de palavras relevantes, a remoção de palavras comuns, descarte de pontuação, redundâncias, **case sensitive**, entre outros.

Com isso, a análise converte determinados textos de interesse em *token*, que posteriormente são considerados como termos para o índice da busca, a exemplo de palavras-chave que fazem menção a marca ou padrões de estilização presentes na página. Um dos benefícios desse procedimento é a criação de um dicionário de sinônimos para buscar por termos que possam se referir a uma determinada marca, a exemplo da prática de *Cybersquatting* e *Typosquatting* (SILVA; FEITOSA; GARCIA, 2019a), que utilizam termos reservados ou homônimos para persuadir o usuário final. Tais comportamentos podem ser previstos através da busca indexada. No apêndice desse documento, Capítulo B, consta a Tabela 41, que descreve alguns exemplos de sinônimos referente a uma marca, não se limitando a elas.

Já o reconhecimento visual tem como foco analisar arquivos de imagem. Adicionalmente, o processo também faz uso de reconhecimento óptico de caracteres, do inglês *Optical Character Recognition* (OCR), para obter maiores informações em busca de mais informações, como em casos de *phishing* com pouco ou quase nenhum conteúdo no código-fonte, porque sua estética visual é estruturada de forma dinâmica, mais detalhes serão descritos na Seção 4.2.3.10. Outra alternativa é reconhecer padrões existentes em imagens distribuídas na página, sendo possível até mesmo detectar marcas visadas. O processo é ilustrado na Figura 28.

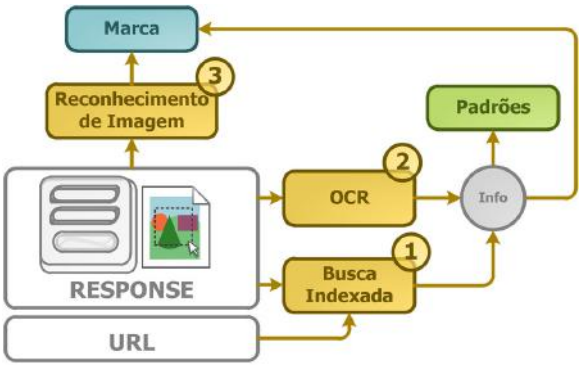

Figura 28 – Estratégias para a detecção de reconhecimento de marcas

Conforme pode ser observado, a estratégia tem como prioridade a análise textual com a busca indexada. Essa abordagem se baseia em palavras-chave, também definidas como *token*, a serem investigados em um determinado conjunto de palavras. Essa abordagem visa uma alto desempenho no resultado da busca em um grande volume de informação. Além disso, por ter uma busca direcionada e com escopo definido por palavras-chave reservadas, o processo de análise torna-se menos intrusivo aos dados pessoais que eventualmente possam existir no volume de dados a ser analisado. Exemplos de *token* a serem analisados são padrões presentes na anatomia da URL, como o domínio, subdomínios, e tags HTML no conteúdo, como <*title*> ou <*body*>. Na Figura 29 é possível observar a busca textual como ponto inicial e, posteriormente, a busca visual através de OCR ou reconhecimento de imagem.

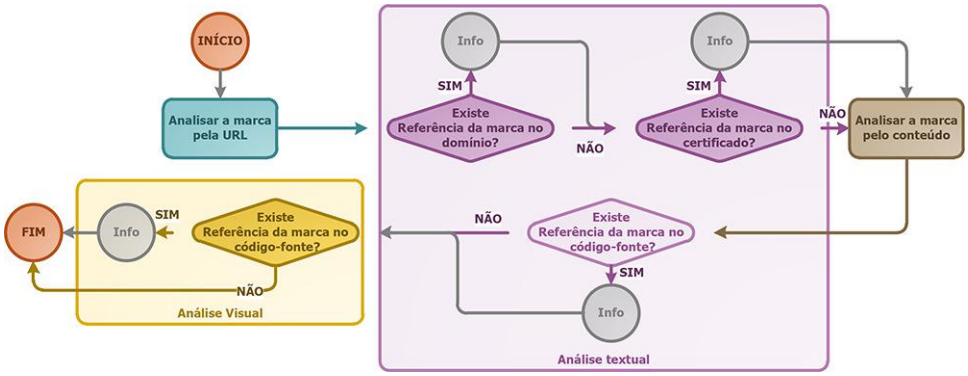

Figura 29 – Representação da detecção de marca

O ponto inicial ocorrerá pela análise textual, que terá como fonte de informações a URL, mais especificamente o domínio e certificado digital. Caso não seja suficiente, o próximo passo será realizado durante a renderização da página. A investigação pela marca será feita no conteúdo da página, que também utiliza a abordagem de análise léxica e sintática anteriormente utilizada na análise da URL. Além disso, caso o conteúdo de texto seja muito limitado e a presença de imagens no conteúdo seja considerável, técnica de OCR será adotada a fim de obter maior quantidade de informação, convertendo as imagens em texto, resultando em informações candidatas para uma análise léxica e sintática.

4.2.3 Sistema especialista (SE)

Por focar a suscetibilidade do usuário final, as chances de sucesso de um ataque de *phishing* são maiores quando as vítimas em questão possuem menor perspicácia técnica para perceber que a página é fraudulenta. Portanto, é razoável dizer que ataques dessa natureza seriam menos eficientes se todo o usuário inexperiente tivesse um especialista no assunto ao lado, prontamente disponível para notificá-lo do perigo. Diante dessa ótica, a proposta em questão adotou na proposta uma abordagem de SE para representar a proteção *client-side* durante a navegação do usuário.

Conforme a Figura 30, a metodologia tem como base os princípios e evidências encontrados no **estudo empírico** do Capítulo 3, obtendo uma fundamentação para **técnicas de predição**. O próximo passo foi elaborar o **conhecimento especializado** como base da estruturação das regras de distinção entre malicioso, suspeito ou confiável, obtendo assim uma **heurística para predição**, caracterizando um **sistema de apoio à decisão**. Posteriormente, foi definido um **analisador de comportamentos** para identificar elementos que propiciam a análise dos comportamentos da heurística, resultando então no **SE**. As etapas da estruturação do SE seguem conforme a Figura 31.

As características foram agrupadas por uma categorização de comportamentos, conforme ilustrada na Figura 32, o que facilita a identificação dos comportamentos. A estrutura de classificação considera escopo (URL ou Conteúdo), estratégia (conforme o Capítulo 2), tipo

Figura 30 – A proposta aplicada como um sistema especialista (SE)

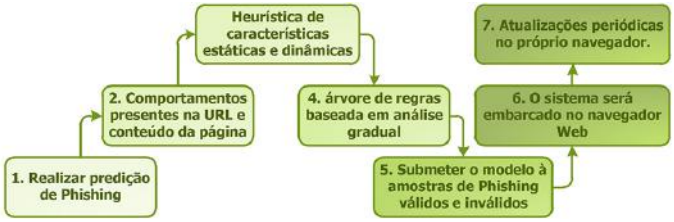

Figura 31 – A proposta como um Sistema Especialista (SE)

		Características	Estratégia					Tipo		Relevância
			Fidedignidade	Ofuscação	Propagação	Sazonalidade	Volatilidade	Dinâmica	Estática	
Escopo	URL	1. Tem ausência de domínio registrado?	✓	✗	✗	✗	✗	✓	✗	☆
		1.1. Ausente da lista de permissão?	✓	✗	✗	✗	✗	✓	✗	☆
		1.2. Tem certificado inválido?	✓	✗	✗	✗	✗	✓	✗	★
		1.3. O certificado foi recém validado?	✓	✗	✗	✗	✓	✓	✗	★
		1.4. Tem simulação TLD?	✓	✓	✓	✓	✗	✗	✓	★
		1.5. Tem TLD na lista suspeita?	✓	✗	✓	✗	✗	✗	✓	★
		1.6. Tem tentativa homográfica?	✓	✓	✗	✓	✗	✗	✓	★
		1.7. Tem nome muito grande?	✓	✓	✓	✓	✗	✗	✓	★
		1.8. Tem nome encoded?	✗	✓	✓	✗	✗	✗	✓	★
		1.9. Registro recém criado?	✓	✗	✗	✗	✓	✓	✗	★
		2. Tem subdomínio?	✓	✓	✓	✓	✗	✗	✓	☆
		2.1. Tem simulação TLD?	✓	✓	✗	✓	✗	✗	✓	★
		2.2. Tem tentativa homográfica?	✓	✓	✗	✓	✗	✗	✓	★
		2.3. A sintaxe é muito grande?	✓	✓	✓	✓	✗	✗	✓	★
		2.4. Qual a quantidade de subdomínios?	✓	✓	✓	✓	✗	✗	✓	★
		3. O HOST consta na lista suspeita?	✗	✗	✓	✗	✗	✗	✓	★
		4. Tem IP exposto?	✗	✓	✓	✗	✗	✗	✓	☆
		5. Tem tentativa homográfica no path ou querystring?	✓	✓	✓	✓	✗	✗	✓	☆
		6. Tem investida de redirect?	✓	✓	✓	✗	✗	✗	✓	☆
		7. Tem uptime recente?	✗	✗	✓	✓	✓	✓	✗	★
		8. Tem porta específica?	✗	✗	✓	✗	✗	✗	✓	☆
		9. Tem referência a formulário?	✓	✗	✗	✓	✗	✗	✓	☆
	Conteúdo	10. O código-fonte é acessível?	✗	✓	✗	✗	✓	✓	✗	☆
		10.1. Tem elementos x-origin?	✗	✓	✓	✗	✗	✗	✓	★
		10.2. Tem comportamento forjado?	✓	✓	✗	✓	✗	✗	✓	★
		10.3. Tem submissão de formulário?	✓	✗	✗	✓	✗	✗	✓	★
		11. Tem ausência de favicon?	✓	✗	✗	✗	✗	✗	✓	☆
		12. Tem conteúdo sazonal?	✗	✗	✗	✓	✗	✓	✗	☆
		13. Foi possível obter a marca?	✓	✗	✗	✗	✗	✗	✓	☆
		13.1. Tem conflito de identidade?	✓	✓	✗	✓	✗	✗	✓	☆

Figura 32 – Agrupamento dos comportamentos

(Estática ou Dinâmica) e a relevância (-3, -1, 0, 1 ou 3). Além disso, características antes representadas por constantes, a exemplo da quantidade de subdomínios na URL, foram transformadas em variáveis categóricas, assumindo valores em uma escala de 3 níveis, sendo até 1 subdomínio assumindo o valor 0, maior que 1 e menor ou igual a 2 assume-se o valor 1, já 3 ou mais será assumido o valor 3.

A coluna representada com estrelas remete a relevância da característica. Conforme ilustrado, as características de baixa (estrela apagada), de média (estrela parcialmente iluminada) e alta (estrela totalmente iluminada) relevância estão classificadas com base na análise de relevância resultante do estudo empírico do Capítulo 3.

Importante esclarecer que um valor 0 significa que o comportamento não representa perigo,

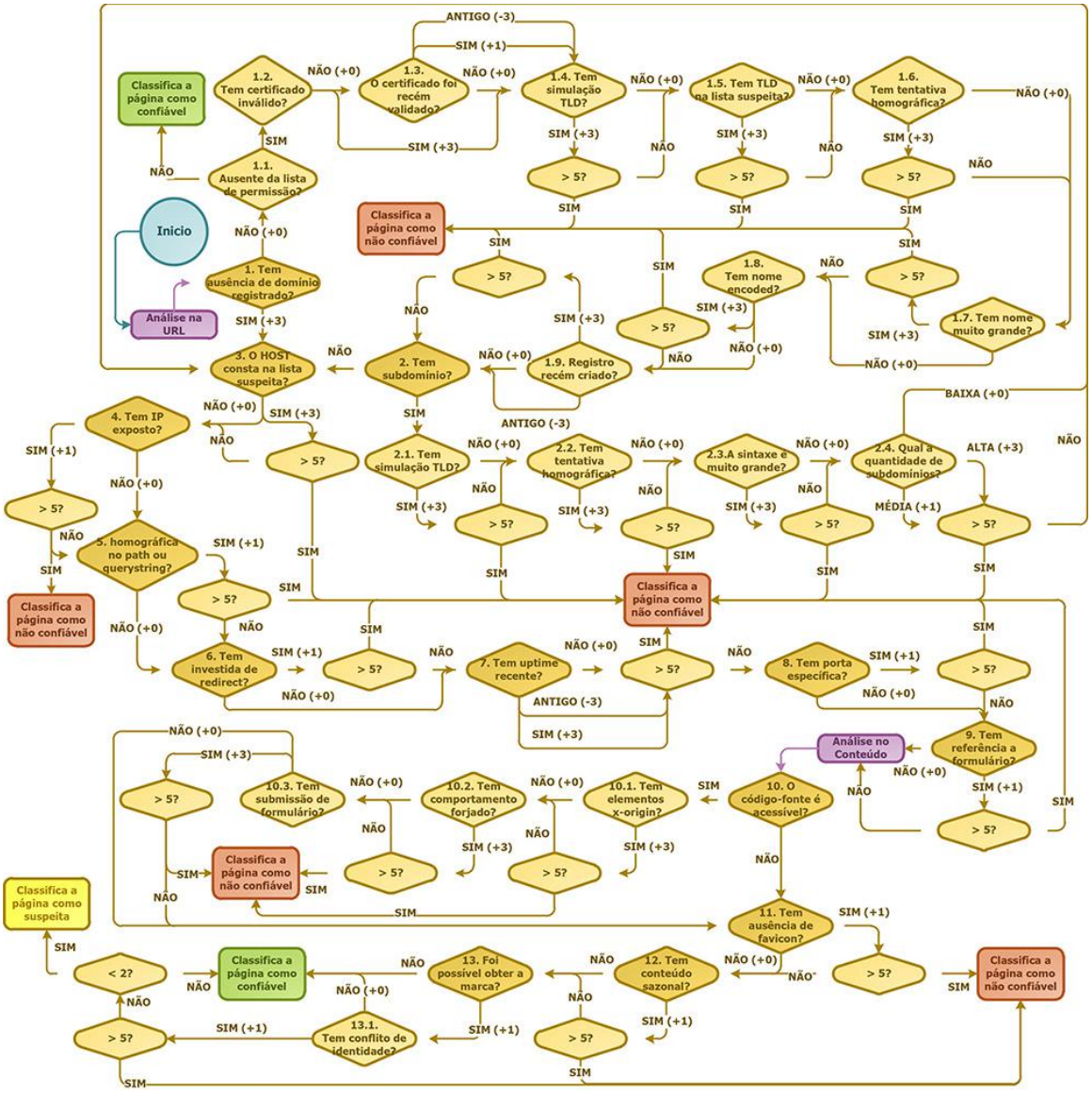

Figura 33 – Representação da máquina de inferência

já o valor 1 sugere suspeitas sobre o comportamento, e um valor 3 remete que o perigo precisa ser considerado. Também podem haver casos de valores negativos, tal situação caracteriza a proposta focada em fidedignidade, porque essas atribuições estabelecem que um determinado comportamento descreve prestígio no sentido de ser fidedigno, como em casos de -1 para relativamente fidedignos e -3 para muito fidedignos.

Por fim, é possível observar que as características de maior peso tem relação com a estratégia de fidedignidade. Com isso, tal valoração auxilia nos resultados obtidos pelo motor de inferência, representada como uma árvore de regras, conforme a Figura 33.

Outro aspecto importante é quanto ao tipo da *estratégia das categorias*, divididas entre **estática e dinâmica**. O estudo assume como estática a característica que não depende de aspectos do tempo ou de terceiros. Além disso, a estratégia pode atuar em dois **escopos distintos**, na **URL** ou no **conteúdo da página**. Por exemplo, analisar comportamentos na

URL, como nomenclaturas no domínio e subdomínio ou identificar sintaxes maliciosas no código-fonte do conteúdo, são aspectos que não consideram o tempo, portanto são estáticas. Esse tipo de característica, por ter menos custo computacional, tendem a ter prioridade no processamento da análise gradual. Já características dinâmicas são diretamente afetadas pelo tempo ou de agentes terceiros, a exemplo do tempo de atividade de um domínio (que é checado no escopo da URL através do protocolo *WHOIS*) ou mesmo identificar no conteúdo termos referentes à aspectos sazonais (que são impactadas pelo calendário anual).

Com base na Figura 33, algumas observações merecem destaque, a saber: cada checagem de característica é um fluxo de controle (representado por um losango), e pode haver um fluxo pai e seus respectivos filhos (representado por uma cor escura para o pai e mais clara para os filhos), a exemplo do fluxo principal 1, que possui 9 filhos, numerados entre 1.1 até 1.9. Há casos em que um fluxo pai computa valor na página (-3, -1, 0, 1, 3) ou não computa, servindo apenas de estrutura de controle, já os fluxos filhos sempre computam valor na página. Apesar de haver 30 losangos (fluxos), 5 são apenas estruturas de controle (1, 1.1, 2, 10 e 13) e 25 são fluxos que pontuam a página. Como forma de elucidar os passos da máquina de inferência, segue a descrição de cada um dos 30 fluxos de controle da árvore de regras.

4.2.3.1 1. Tem ausência de domínio registrado?

Esse fluxo verifica se a página em questão está desprovida de um domínio registrado pelo proprietário. Uma grande dificuldade em analisar essa questão ocorre em casos que o malicioso hospeda sua fraude através de um serviço de hospedagem, como *sites.google.com* ou *000webhostapp*, já que o resultado do protocolo WHOIS irá indicar que a página tem um registro de domínio, apesar do mesmo não ter sido realizado pelo proprietário da fraude. Para esses casos, uma lista de exceções foi criada. Além da lista, também foi observado casos em que o tempo de registro do domínio fosse superior a 3 anos, visando minimizar o problema mencionado. Caso conste esse comportamento, será adicionado $+3$ na pontuação da página e o fluxo seguirá para o item 3, uma vez que o item 2 analisa o subdomínio, elemento do qual teria como pré-requisito um domínio registrado. Caso exista um domínio, demais características serão avaliadas, a saber:

- **1.1. Ausente da lista de permissão?** como ele possuí domínio, talvez esteja previamente registrado como entrada no **serviço server-side**, esse processo realiza uma consulta na lista de permissão, e caso confirmado, a página é marcada como confiável e o processo de análise gradual é abortado. Caso contrário, o fluxo segue normalmente.

- **1.2. Tem certificado inválido?** Verifica se o domínio tem um certificado digital válido, ou seja, emitido e não expirado. Domínios que tiverem seu certificado expirado terão a mesma reputação de um domínio sem certificado digital. É possível obter essa informação porque todo certificado digital tem uma data de expiração. Caso conste esse comportamento, será adicionado $+3$ na pontuação da página.

- **1.3. O certificado foi recém validado?** Caso esteja válido, é checado se a emissão do mesmo foi recente. É possível obter essa informação porque todo certificado digital tem uma data de emissão e data de expiração. Importante mencionar que **características temporais** remetem a natureza volátil do *phishing*. Como o mesmo geralmente tem vida curta, o tempo de atividade torna-se um importante aliado na distinção entre a reputação da página. Essa estratégia também é assumida por muitos avaliadores de SEO (a exemplo do Alexa Page Rank), que se baseia no tempo de atividade para avaliar a reputação de um determinado site. Portanto, toda característica temporal considera algo recente com um prazo e até 20 dias, ou define como antigo algo acima de 6 meses. Essa média de tempo foi baseada nos resultados obtidos no Capítulo 3. Caso conste esse comportamento, será adicionado $+3$ para recém criados ou -3 para emissões antigas, indicando que a página tem uma certa reputação.

- **1.4. Tem simulação TLD?** É um comportamento malicioso que ocorre quando um determinado domínio faz uma distribuição de caracteres com intenção de simular um domínio de topo (TLD), geralmente precedido de hífen ("-"), por exemplo, *http://magazine luiza-com.info*. Esse tipo de investida visa oferecer fidedignidade através da nomenclatura do domínio da página e em muitos casos, a URL é exibida truncada na barra de status do navegador. Nesses casos, o usuário pode acabar não observando o trecho ".info", passando a impressão que o TLD da página é ".com". Caso conste esse comportamento, será adicionado $+3$ na pontuação da página.

- **1.5. Tem TLD na lista suspeita?** Existe uma lista de domínios que são gratuitos na internet (a exemplo do .tk), o que torna os mesmos muito visados para o crime. No apêndice, Capítulo B, consta uma lista dos domínios que são considerados nessa característica 40. Caso conste esse comportamento, será adicionado $+3$ na pontuação da página.

- **1.6. Tem tentativa homográfica?** Nesses casos, são domínios com palavras-chave para representar uma marca, como por exemplo, *http://auxilio-emergencia.tk* com objetivo de passar a ideia que trata-se de uma página da caixa econômica federal.

- **1.7. Tem nome muito grande?** Não são incomuns os casos de *phishing* com um grande número de caracteres na composição do nome do domínio. Internacionalmente, um domínio tem como tamanho mínimo de 3 e máximo de 63 caracteres (incluindo os 4 referentes a extensão, como .com, .net, .gov, etc), portanto, o atacante pode arbitrariamente inserir palavras-chave com uma certa liberdade. O estudo considerou acima de 40 caracteres como tamanho grande. Caso conste esse comportamento, será adicionado $+3$ na pontuação da página

- **1.8. Tem nome encoded?** Esse recurso remete a codificar o nome do domínio em um charset diferente do convencional, a exemplo de investidas baseadas em *punycode* ou

URL *encoding*, conforme descrito no Capítulo 3. Caso conste esse comportamento, será adicionado +3 na pontuação da página

- **1.9. Registro recém criado?** Esse é outra característica temporal que remete a natureza volátil do *phishing*. Por ser temporal, essa característica segue a regra descrita no item 1.3, só que se baseia no tempo de criação da página através do protocolo WHOIS. É possível obter essa informação porque o WHOIS informa a data de criação do registro, data de atualização e data de expiração. Caso conste esse comportamento, será adicionado +3 para recém criados ou -3 para registros antigos, indicando que a página tem uma certa reputação.

4.2.3.2 2. Tem subdomínio?

O subdomínio é outro elemento da URL em que o atacante tem uma certa arbitrariedade na sua composição, não sendo incomum casos de exploração similares realizadas no domínio. Caso a URL não tenha subdomínios, o fluxo segue para o item 3 sem modificações na pontuação da página, caso contrário, algumas particularidades serão analisadas, a saber:

- **2.1. Tem simulação TLD?** Semelhante ao item 1.4, porém o atacante não tem a necessidade de fazer uso de um separador, sendo possível até mesmo fazer uma combinação com uma composição também no domínio, tornando o ataque mais elaborado, a exemplo da URL *http://paypal.com.secure-transaction.tk*. Caso conste esse comportamento, será adicionado +3 na pontuação da página.

- **2.2. Tem tentativa homográfica?** Semelhante ao item 1.6, contudo, com subdomínios o atacante tem maior liberdade em explorar palavras-chave, devido a maior capacidade de caracteres, em comparação a composição do domínio. Por exemplo, *http://bradesco.net.empresas.transacoes-seguras.tk*. Caso conste esse comportamento, será adicionado +3 na pontuação da página. Também é um recurso que pode gerar *bypass* em alguma política de lista de permissão que autorize URL com termos de domínios conhecidos, como "facebook.com" ou "paypal.com".

- **2.3.A sintaxe é muito grande?** Internacionalmente, um domínio pode ter até 127 subdomínios e cada subdomínio pode ter até 63 caracteres. Contudo, com base nos resultado descritos no Capítulo 3, o estudo define que toda a sua sintaxe, ou seja, a junção dos subdomínios existentes incluindo os pontos ("."), acima de 14 caracteres é considerada uma sintaxe longa. Caso conste esse comportamento, será adicionado +3 na pontuação da página.

- **2.4. Qual a quantidade de subdomínios?** Conforme mencionado anteriormente, um domínio pode comportar até 127 subdomínios, contudo, o estudo considerou até 1 subdomínio uma quantidade baixa, até 2 uma quantidade média e 3 em diante quantidade alta. Caso conste esse comportamento, será adicionado +3 na pontuação.

4.2.3.3 3. O HOST consta na lista suspeita?

Assim como o item 1.5, o estudo também elaborou um dicionário de host que são bastante visados para o crime, a exemplo do *000webhostapp*. São serviços de hospedagem bastante atrativos, por ter uma mensalidade muito baixa ou mesmo gratuita. No apêndice, Capítulo B, consta uma lista de HOSTs considerados nessa característica 40. Caso conste na lista, será adicionado $+3$ na pontuação da página e o fluxo segue para o item 4, caso contrário, o fluxo segue para o item 4 sem modificações na pontuação.

4.2.3.4 4. Tem IP exposto?

São casos em que além de não conter domínio registrado, a página ainda é acessada fazendo uso do IP público do respectivo servidor que hospeda a página, como esse tipo de informação não oferece nenhum tipo de identidade, a solução considera essa característica suspeita, atribuindo $+1$ na pontuação da página em situações existentes.

4.2.3.5 5. Tem tentativa homográfica no path ou querystring?

São casos em que a exploração é realizada no *path* ou *querystring* da URL. Esses valores são atribuídos arbitrariamente pelos mal intencionados, servindo como elementos que podem garantir fidedignidade ou mesmo propagação. A fidedignidade é resultante de ataques homográficos que utilizam palavras-chave que possam trazer alguma confiança. Já no aspecto da propagação, o mal intencionado pode gerar URL diferentes para uma mesma página, isso para os casos que mecanismos de lista de bloqueio geram o *hash* da URL considerando todo o conteúdo de caracteres, o que resultaria em um hash diferente para cada URL com valores distintos em seu *path* ou *querystring*, possibilitando um *bypass* nesse tipo de solução.

4.2.3.6 6. Tem investida de redirect?

São casos em que a aplicação ao carregar uma URL, espera que em algum setor valorado da mesma, como um *path* ou *querystring* conste como valor uma outra URL, fazendo um redirecionamento, conforme explicado no Capítulo 3. Muitos atacantes utilizam o prestígio de um determinado domínio, com SEO baixo, para realizar esse tipo de investida. Caso conste o comportamento, será adicionado $+1$ na pontuação da página e o fluxo segue para o item 6.

4.2.3.7 7. Tem uptime recente?

Esse é outra característica temporal que remete a natureza volátil do *phishing*. Por ser temporal, essa característica segue a mesma regra descrita no item 1.3, porém, é baseada no tempo de atividade do HOST, informação obtida através do protocolo WHOIS. É possível obter essa informação porque o WHOIS informa a data de criação (inicio de atividade) e data de ultima atualização (último uptime) Caso conste esse comportamento, será adicionado $+3$ para recém ativos ou -3 para com atividades antigas, indicando que a página tem uma certa reputação.

4.2.3.8 8. Tem porta específica?

São casos de páginas que não rodam na porta padrão 80 ou 443. Essa característica levanta suspeitas porque muitas vezes o serviço da página é disponibilizado fazendo uso de portas adotadas como padrão por certas ferramentas, a exemplo da porta 8080 do tomcat. Caso exista a presença desse comportamento, o fluxo atribuirá $+1$ na pontuação da página.

4.2.3.9 9. Tem referência a formulário?

Essa característica analisa o sufixo da URL, que muitas vezes possui referência ao arquivo que é carregado no respectivo acesso, a exemplo de sufixos como index.html. Frequentemente, o ataque utiliza um formulário para submeter dados sensíveis, e eventualmente a página que comporta esse formulário possui expressões com nomes sugestivos, como cart, form, login ou auth. Caso exista a presença desse comportamento, o fluxo atribuirá $+1$ na pontuação.

4.2.3.10 10. O código-fonte é acessível?

Em alguns cenários, pode haver pouco ou nenhum conteúdo no código-fonte da página quando é carregada. A explicação é que em muitos casos o conteúdo é exibido ao usuário final de forma dinâmica, e esses são renderizados através de funções que são encapsuladas em arquivos externos, a exemplo de funções *javascript* que desenham o layout da página e constam em arquivos com extensão ".js". Outro comportamento são páginas que possuem sua estética montada através de imagens como plano de fundo através de tags HTML como o *<map>*. Caso o código-fonte seja significativo, algumas características serão analisadas, a saber:

- **10.1. Tem elementos x-origin?** Uma característica presente em ataques de *phishing* é a grande quantidade de links e outros elementos que oferecem navegabilidade para o usuário final. Uma particularidade que merece suspeitas são casos de encaminhamentos de domínios cruzados, ou seja, encaminhar de um HOST para outro distinto de forma demasiada. O estudo considera mais de 25% dos links da página uma quantidade demasiada. Esse item também sugere observar elementos como *<form>*, *<iframe>* e tentativas de *clickjacking*, que são roubos de clique através de elementos que são exibidos em sobreposição a outros. Caso exista o comportamento, o fluxo atribuirá $+3$.
- **10.2. Tem comportamento forjado?** Uma investida comum em *phishing* direcionados é simular erros HTTP ou exibir uma interface ao usuário se baseado na resolução do dispositivo. Não são incomum os casos de páginas que abrem normalmente em dispositivos móveis (através de links compartilhados via SMS, característico do *SMiShing*), mas que se o mesmo for aberto em um browser de desktop, é exibido um erro forjado ao usuário. Portanto, manipulações *javascript* através do *navigator.userAgent* visando detectar a resolução do dispositivo baseadas no cabeçalho *User-Agent* são investidas que caracterizam essa característica. Uma vez presente, o fluxo atribuirá $+3$.

- **10.3. Tem submissão insegura de formulário?** Essa característica analisa se a página em questão submete dados de formulário, caso positivo, se o mesmo tiver ausência de um certificado válido, a solução levanta suspeitas sobre a página, atribuindo $+1$.

4.2.3.11 11. Tem ausência de favicon?

Da mesma forma descrita no item 7, alguns mal intencionados não se preocupam tanto com os detalhes, deixando a sua aplicação com o favicon padrão da ferramenta utilizada no desenvolvimento da página, detalhe que também poderia fortalecer a identidade visual de sua marca. Portanto, caso exista o comportamento, o fluxo atribuirá $+1$ na pontuação da página.

4.2.3.12 12. Tem conteúdo sazonal?

Essa característica remete aos fenômenos que podem ocorrer em um determinado período do calendário, a exemplo do natal e *blackfriday*, que movimentando significativamente o *e-commerce*. Além de calendário definido, esses eventos sazonais também possuem palavras-chave que facilitam sua identificação. Contudo, assim como descrito no item 9.3, essa característica não é comum apenas em fraudes, portanto o fluxo atribui $+1$ ao total da página.

4.2.3.13 13. Foi possível obter a marca?

Teoricamente, muitos dos fluxos anteriores possibilitam a obtenção de informações sobre a marca envolvida, a exemplo do item 1 e seus filhos, item 2 e seus filhos, item 9 e seus filhos, e também os itens 10 e 11. Portanto, a última investida da análise será observar particularidades referente a marca-alvo do ataque. Em casos que a marca seja possível de ser reconhecida, o fluxo segue para o item 12.1, caso contrário o processo é encerrado com algum veredito.

- **13.1. Tem conflito de identidade?** Após obter as informações da marca através dos itens anteriores, é observado se existe algum conflito de identidade, por exemplo, domínio não correspondente ao nome da companhia no certificado. Além disso, caso o host em questão esteja na lista suspeita, será observado se o título, certificado ou domínio registrado possui referência a alguma marca. Caso exista essa divergência, será atribuído $+3$.

Em relação a árvore de regras, a fundamentação é de natureza estocástica, ou seja, baseada em resultados estatísticos extraídos na pesquisa empírica do Capítulo 3. Contudo, certas situações podem ter mais ou menos peso, por isso a variação na pontuação atribuída a página. Como a heurística considera a fidedignidade, características que são intrínsecas à forja da identidade possuem maior peso. A análise inicia o processo no escopo da URL, agrupando as características semanticamente e priorizando por peso. Posteriormente, segue no conteúdo agrupando e priorizando conforme o raciocínio anterior, caracterizando-se como gradual.

Diante o exposto, é correto dizer que a solução será menos eficiente em casos que a fidedignidade seja baixa, por exemplo, casos que não existam certificado digital ou mesmo um domínio registrado, muitas das etapas são ignoradas. Um ponto importante da proposta é assumir como mandatório a presença de certificado digital (HTTPS) na página, e a justificativa se enviesa em duas questões. A primeira é o **escopo de atuação**, que por ser aderente à soluções existentes, a proposta não se preocupa em identificar fraudes sem fidedignidade, que teoricamente são mais previsíveis às soluções existentes, fortalecendo o aspecto complementar da presente Tese. Além disso, o certificado digital possui informações que podem ajudar na obtenção do suposto proprietário em questão. No intuito de evitar interação com o usuário final, essa alternativa foi adotada pela proposta como meio de obter informações sobre a marca e atividades da navegação, ou seja, automatizada, sem precisar de intervenções do usuário.

A outra questão é a **responsabilidade com os dados sensíveis**. Trafegar informações sensíveis sem tunelamento é uma atitude muito temerária. Além disso, a presença de um certificado garante o registro de um domínio, outro elemento importante que fortalece a identidade da marca. A solução classificará o prestígio da página de acordo com os comportamentos encontrados, portanto, situações que representem risco aos dados sensíveis e o tunelamento esteja ausente, a solução irá definir a página como não confiável. Além disso, nos dias atuais existem alternativas sem custo para adotar tunelamento, a exemplo da autoridade certificadora *Let's Encrypt*[1], que oferece gratuitamente certificados TLS X.509, fortalecendo a justificativa.

4.3 SOLUÇÃO SERVER-SIDE

Essa seção descreve a solução baseada em serviço que visa prover um repositório de informações para validar as entradas sugeridas pela solução *client-side*, bem como oferecer melhorias através de atualizações periódicas. O fluxo dessa estratégia pode ser conferido na Figura 34.

Uma entrada é uma informação fornecida pelo proprietário de uma marca para notificar previamente sobre a apropriação de um determinado recurso disponível na Web, como por exemplo um *host* que hospeda uma aplicação que faz transações financeiras, alimentando assim uma lista de permissão para evitar checagem desnecessárias em sites genuínos. Diante disso, surge uma preocupação para que pessoas sem a devida autorização não acabem alimentando a lista de permissão com falsas entradas, o que resultaria em falsos negativos. Visando minimizar esse problema, a solução adota um modelo de requisições federadas.

O processo descreve que solicitações entre clientes e servidores sejam devidamente assinadas e autenticadas (SCIARRETTA et al., 2017; NACER et al., 2017). Essa comunicação é realizada através de um protocolo de troca de chaves. A proposta faz uso desse modelo para garantir a autenticação e autorização durante o processo de leitura e escrita no repositório de entradas.

De posse da requisição, o serviço *server-side* valida o token enviado. Caso seja inválido, a requisição é rejeitada com erro HTTP 403 (requisição inválida). Caso seja válido, o próximo

[1] https://letsencrypt.org/documents/isrg-cps-v2.7/

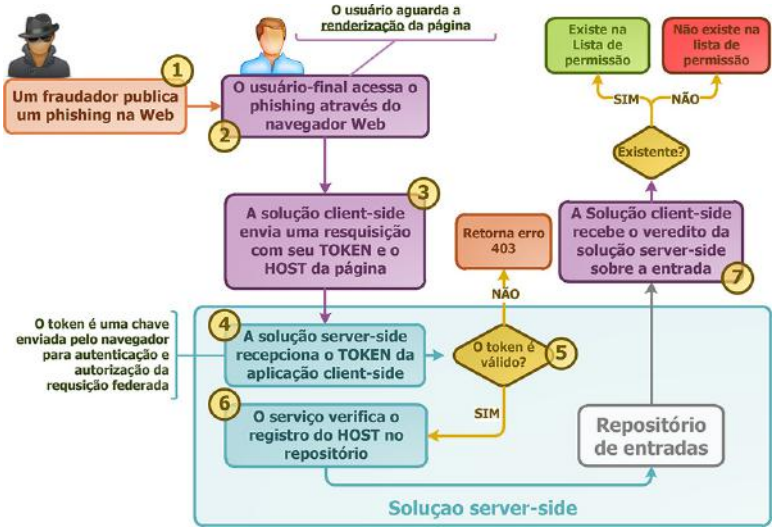

Figura 34 – Fluxograma da solução server-side

passo será verificar se as credenciais informadas são válidas, caso inválida, a requisição é rejeitada com erro HTTP 401 (não autorizado). Caso válida, será verificado se o host em questão já existe no repositório, em caso de inexistência, a entrada é registrada. O processo que garante uma troca de *token* entre as requisições é o que estudo denomina como requisição federada, que utiliza um protocolo para compartilhamento de recursos de forma limitada. O processo dessa comunicação é ilustrado na Figura 35.

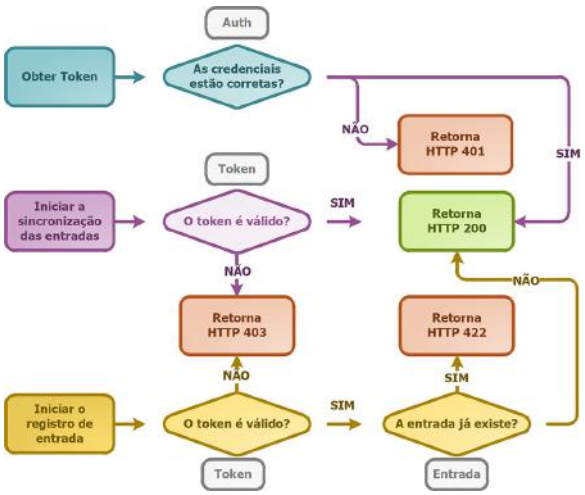

Figura 35 – Representação da validação e registro de entradas

Existem 3 operações (recursos) que serão compartilhadas com terceiros (atores externos da aplicação), a saber: (i) obter token, (ii) sincronizar entradas e (iii) registrar entrada. Em (i), remete ao processo que o ator precisará fornecer suas credencias para a criação de seu *token* que concede permissão para as operações (ii) e (iii). Em (ii), os navegadores Web poderão sincronizar sua lista de permissão de entradas confiáveis durante a navegação do usuário final, visando identificar exceções no processo da análise gradual. Já em (iii), os proprietários de

marcas poderão registrar seus domínios registrados, alimentando a lista de permissão.

A requisição federada garante um protocolo de autenticação e autorização aos recursos, isso remete dizer que tanto os navegadores Web como os proprietários de marcas serão atores que precisarão se autenticar (credenciais de acesso) e cada um terá um papel que especifica suas autorizações (*token*). Com isso, a proposta *server-side* visa garantir o não repúdio das informações persistidas, além disso, regras são estabelecidas no processo de registro, ou seja, limitar o controle ao recurso através de políticas de autorização. Por fim, outro papel do serviço *server-side* é sincronizar periodicamente no navegador Web o arquivo que representa o conjunto de regras e também a lista de permissão. Tais arquivos ficam armazenados no navegador para evitar latência durante o processo de análise. Os registros na lista serão prefixos de *hash* (os primeiros 32 dos 64 bits de um SHA-256) das entradas previamente registradas.

4.4 AMEAÇAS DA PROPOSTA

Essa seção descreve as ameaças e limitações consideradas pela pesquisa sobre as decisões na fundamentação da heurística e a árvore de regras, bem como seus aspectos arquiteturais.

4.4.1 Ameaças na calibragem da heurística

Importante salientar a estratégia de fidedignidade intrínseca na proposta. Caso a página em questão não possua riqueza em seus detalhes, a proposta não poderá detectar comportamentos suspeitos ou a marca alvo em questão. Diante disso, quanto maior a fidedignidade, maiores serão as chances de sucesso da proposta. Em contrapartida, quanto menor a fidedignidade, menores serão as chances de sucesso. Contudo, é esperado pela proposta que quanto menor for a fidedignidade, maiores serão as chances da proteção nativa detectar a fraude, caracterizando assim uma coexistência e benefício mútuo entre a solução convencional e a Tese.

4.4.2 Ameaças na identificação de comportamentos

O processo de busca indexada pode ter ruídos, caso o conteúdo não seja preciso ou muito diversificado, isso caracteriza ainda mais a proposta baseada em fidedignidade. Contudo, certos resultados podem resultar em falsos positivos, casos com textos muito heterogêneos.

Outra preocupação é na natureza intrusiva durante os processos de identificação de comportamentos e a marca durante a navegação. Além disso, o processo de traçar perfil para atenuar falsos negativos pode acabar ferindo a privacidade. Como forma de atenuar essa ameaça, o processo de investigação de evidencias é baseado em uma **análise em profundidade gradual**, ou seja, priorizando informações contidas na URL e posteriormente o conteúdo. Contudo, ainda sim, devido a natureza subjetiva, a privacidade pode acabar sendo violada.

4.4.3 Ameaças no reconhecimento da marca

Uma ameaça que merece destaque é a presença de diversas logomarcas em uma mesma página. Esse comportamento pode dificultar uma identificação precisa da marca, a exemplo de páginas com entidades parceiras, divulgações publicitárias ou mesmo formulários com diversas bandeiras de cartões de crédito. Uma forma de atenuar a situação é obter informações relacionadas em elementos da página que são destinados a identificação da mesma, a exemplo do "<title>" da página ou figuras existentes no cabeçalho ou rodapé da página. Contudo, essa engenharia de usabilidade e diagramação não foi muito explorada pela Tese.

4.4.4 Ameaças sobre o sistema especialista

Devido a estratégia de lista de permissão, poucas são as chances de ocorrer problemas relacionados a falsos positivos, e caso ocorra, por via de regra, é preciso apresentar uma opção ao usuário final que o mesmo siga com o acesso e assuma algum risco. Todavia, limitações referentes a falsos negativos merecem ser evidenciadas, uma vez que essa ameaça de amplitude pode intensificar essa ameaça. E caso a solução não consiga identificar comportamento suspeito ou a marca, não poderá proteger o usuário, remetendo as ameaças relacionadas a fidedignidade.

Outro ponto, é que a análise gradual visa finalizar o processo no momento que a página atinge a pontuação de uma ameaça, em prol do desempenho e privacidade. Contudo, visando menor ocorrência de falsos positivos, talvez algumas características fossem imprescindíveis de avaliação, porque seu resultado podem trazer um equilíbrio no julgamento. Por exemplo, um domínio que tenha um certificado recentemente renovado (+1), uma possível tentativa homográfica (+3) e um domínio com nome grande (+3) já atingiria o quantitativo necessário para determinar como ameaça. Todavia, se o HOST em questão apresenta muitos anos de atividade (-3), seria interessante analisar outros aspectos antes de decretar o veredito precocemente, como verificar a idade do domínio registrado (-3).

Com isso, é importante discutir sobre a decisão da adoção da abordagem de sistema especialista diante de outras alternativas, a exemplo do aprendizagem de máquina (do inglês, *Machine Learning*, ML). Como a proposta adotou a estratégia de utilizar gatilhos, que interrompem todo o processo quando atinge um quantitativo, isso inviabilizou as práticas de ML, uma vez que esse tipo de estratégia recebe como entrada o valor de todas as características, sendo inviável predizer de forma parcial. Contudo, cada abordagem tem sua vantagem e desvantagem, destacando o *trade-off* da maior a precisão, maior será o custo computacional.

Todavia, pelo fato do sistema especialista se basear de forma empírica sobre os comportamentos a serem investigados, talvez seja necessária uma coleta contínua de antecedentes de incidentes, observando mudanças no cenário que possam baixar a relevância de certas características. Por fim, também se faz necessário observar, através de uma análise de desempenho (*benchmark*), o comprometimento do desempenho (o tempo de transação), comparando entre

a abordagem de sistema especialista e ML, observando o custo por maior precisão.

4.4.5 Ameaças da solução server-side

Apesar de atenuar os casos de falsos positivos, é importante considerar que o processo de lista de permissão é algo opcional, podendo ocasionar em um escopo limitado de atuação. Diante disso, é esperado que os proprietários de marcas que desejem se proteger, em que os mesmos receberam um *token* para registrar suas entradas, bem como seus respectivos parceiros. É esperado que tal procedimento faça parte do processo interno a medida que a solução esteja se consolidando em seu cenário de atuação.

Outra preocupação seria a existência de supostos proprietários com intenções maliciosas, que possam se passar por entidades sérias e tentem obter a permissão de acesso as requisições federadas para realizar entradas, aplicando *by-pass* na solução através de engenharia social. Contudo, algumas medidas podem ser adotadas para aumentar o rigor de informações da entidade que demonstra interesse, fortalecendo o não repúdio das ações realizadas como meio de intimidar motivações maliciosas no repositório.

4.5 CONSIDERAÇÕES FINAIS

Diante o exposto, esse Capítulo apresentou a metodologia e justificativa que compõem a solução proposta pela Tese. O próximo capítulo visa apresentar a implementação da proposta, ou seja, um protótipo como artefato com menor nível de abstração, considerando decisões tecnológicas e maiores detalhes na especificação de certos componentes e decisões arquiteturais.

5 IMPLEMENTAÇÃO DO PROTÓTIPO

Esse Capítulo descreve a implementação de um protótipo empregado para avaliar, de forma empírica, da solução apresentada no Capítulo anterior. A ideia é descrever algo com menor abstração, ou seja, com decisões tecnológicas definidas para a obtenção de um modelo concreto da proposta, no intuito de realizar uma avaliação das hipóteses levantadas pela pesquisa. Adicionalmente, é pretendido submeter o protótipo a uma metodologia experimental formalizada, a ser descrita no Capítulo 6.

5.1 MODELO DE PREDIÇÃO

Conforme mencionado no Capítulo 4, o modelo de predição possui estratégias distintas para cada objetivo específico a ser atingido. Nas seções a seguir são descritas as estratégias e respectivas tecnologias para o desenvolvimento do protótipo. Importante mencionar que o estudo elaborou protótipos pensando em abordagens da inteligência artificial (Sistemas Especialistas e Aprendizagem de Máquina). O motivo foi realizar uma comparação e justificar porque a solução proposta optou pela estratégia baseada em SE.

5.1.1 Processo de extração dos comportamentos

Conforme descrito no Capítulo 4, foi possível observar que muitos comportamentos utilizam recursos externos para checagem de informação, como saber informações do certificado, a idade do domínio e a atividade do *HOST*. Para esses casos, foi utilizada a API *WhoisApiXML*[1]. Já para os comportamentos que analisam o certificado digital foram utilizadas as bibliotecas *javax.net.ssl* e *java.security.cert*[2]. Por fim, o conteúdo da página foi obtido pela biblioteca *java.net.URLConnection*. Essas bibliotecas são nativas da API padrão do Java. A Figura 36 descreve o passo a passo da extração dos comportamentos da página durante o processo de obtenção das informações para a análise gradual.

Cada elemento de informação (esfera verde) é necessário para atender um determinado número de fluxos (relacionados logo abaixo). Já os retângulos amarelos representam ações que utilizam as biblioteca nativas do Java ou um serviço específico para recuperar a informação (a exemplo do WHOIS e do desencurtador de URL). Cada URL que esteja encurtada, como o *bit.ly*[3], são "desencurtadas" pela API do serviço *unshorten.me*[4], obtendo assim a URL que de fato será analisada. De posse dessas informações, foi possível atingir os requisitos necessários para o processamento da análise gradual em uma página Web.

[1] https://whois.whoisxmlapi.com
[2] https://docs.oracle.com/javase/8/docs/api/javax/net/ssl/package-frame.html
[3] https://bit.ly
[4] https://unshorten.me

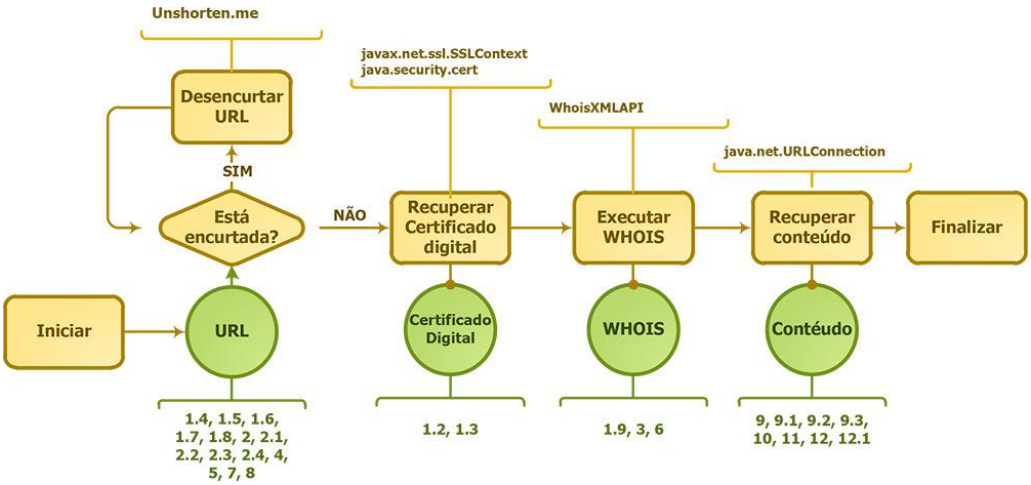

Figura 36 – Fluxograma da extração de informações na página Web

5.1.1.1 Busca textual indexada

Para a implementação da busca indexada foi utilizado o projeto Apache Lucene[5], o qual fornece uma API com diversas funcionalidades para atividades relacionadas a indexação e e procura por texto. Com o *Lucene*, é possível indexar qualquer tipo de dado disponível no formato textual, a exemplo do conteúdo de uma página Web ou o código-fonte da mesma, com tags HTML. Além disso, outros aspectos importantes que o *Lucene* oferece é a ordenação e agrupamento nas buscas, procedimento denominado de *faceting*. Com ele, é possível definir um agrupamento categorizado com base nos termos indexados. Isso oferece a possibilidade de explorar os resultados de acordo com a categoria de interesse, mais precisamente, ordenando os resultados de acordo com critério de relevância.

Outra possibilidade interessante é adoção de técnicas como *Named Entity Recognition* (NER), que identificam similaridades e relações entre um termo e outro, o que pode resultar em maior precisão durante a identificação de uma marca. Um exemplo dessa funcionalidade é quando se pesquisa em algum motor de busca, como o do google, a palavra "pascal", que resulta não apenas uma página na wikipedia sobre matemático francês, mas também informações relacionadas, como vídeos, imagens e páginas sobre a linguagem de programação pascal, ou seja, realiza um tratamento de desambiguação dos termos pesquisados.

Por fim, outra estratégia interessante adotada é o *suggest*, que é um sistema de sugestões de correção ortográfica. Retomando o exemplo anteriormente mencionado, pesquisando no google pela palavra "pascau", são retornados resultados sobre um jornalista canadense, porém também é sugerido pelo motor de busca a pesquisa pelo termo "pascal", para o caso de um possível erro de ortografia na intenção de pesquisar sobre o matemático francês. Esse tipo de situação é bem útil para detectar investidas através de *typosquatting*.

[5] https://lucene.apache.org/

5.1.1.2 Busca visual de informações em imagens

No estado atual, o protótipo propõe simular o funcionamento de um navegador para a interação entre o SE e a navegação do usuário final. Para tanto, foi desenvolvida uma aplicação *Desktop* em *JavaFX*, que utiliza em *background* a ferramenta Tesseract[6] (para o processo de OCR) e faz uso da API *Google Vision*[7] (para o reconhecimento por imagem na detecção de marcas). A justificativa de utilização dessas ferramentas é concentrar na proposta do protótipo em si, sem ter que dispor de esforços em conceitos abstraídos pelas mesmas, como reconhecimento visuais de caracteres e imagens.

O procedimento de reconhecimento de imagem pelo *Google Vision* foi feito através de uma requisição remota. Como o processo é posterior a tentativa de OCR, por questões de desempenho, a requisição sempre utiliza uma imagem já existente (criada pelo OCR), que segue em uma requisição no *Querystring* em formato Base64. Essa imagem representa o conteúdo aberto na aba do navegador com uma resolução de 720x450 (50% de 1440x900), resultando em média uma imagem com 13.4KB. A decisão para tal resolução foi definida após alguns testes realizados, visando garantir uma resolução razoável para análise.

5.1.2 Sistema Especialista (SE) x Aprendizagem de Máquina (ML)

Esta Tese de doutorado adotou a abordagem de SE como modelo de predição da solução proposta. Contudo, antes de tomar tal decisão, foi feita uma comparação com outra abordagem similar, visando investigar, no contexto proposto, qual seria a mais adequada considerando preocupações como desempenho e privacidade. Portanto, com os resultados obtidos, o estudo decidiu optar pela estratégia de SE ao invés de ML e nessa seção serão descritos os dados comparativos e suas respectivas justificativas.

A predição em ML foi implementada em *Python*, com a biblioteca *sklearn*[8], e assim como definido no modelo de SE, também foi adotado classificador supervisionado Árvore de Decisão (do inglês, *Decision Tree Classifier*, DTC). Essa abordagem é interessante porque possibilita um modelo, de classificação ou regressão, ser auto explicável visualmente. Por ser *transparente*, toda a estrutura lógica de suas decisões se torna acessível, possibilitando visualizar uma representação da árvore. Com isso, informações importantes podem ser reveladas, como as características mais relevantes (no topo da árvore)[9]. Para tanto foi utilizada a biblioteca *Graphviz*[10]. Devido as dimensões da árvore, a mesma não consta na estrutura do presente documento, mas pode ser visualizada através de um link[11].

Além disso, o gráfico também ilustra alguns aspectos estocásticos que merecem destaque, como o coeficiente de Gini (RAILEANU; STOFFEL, 2004), instrumento responsável em medir

[6] https://github.com/tesseract-ocr/
[7] https://cloud.google.com/vision/
[8] https://scikit-learn.org/stable/
[9] https://scikit-learn.org/stable/modules/tree.html
[10] https://www.graphviz.org/
[11] Árvore gerada no Graphviz: https://www.dropbox.com/s/c0pp1ykwgrwv3q1/dtc.png?dl=0

a igualdade ou desigualdade de uma distribuição. Em suma, esse coeficiente revela padrões existentes em determinados registros (instâncias) correlacionadas com certas características (*features*) e suas respectivas classificações (*labels*). No Apêndice B, consta na Tabela 39 um link com as amostras utilizadas para o experimento a ser apresentado no Capítulo 6.

Por ser um modelo de classificação, a aprendizagem de máquina do tipo supervisionada possui um conjunto de dados rotulados e previamente definidos. Desta forma, a abordagem ML requer que a máquina seja submetida a um processo de treino, atividade que estabelece valores estocásticos necessários para realizar a predição. Foi elaborado um arquivo CSV com 5822 instâncias, com 3 rótulos definidos: $no\_phishing$ (-1), $suspect\_phishing$ (0) e $yes\_phishing$ (1); constituída por 25 atributos. O referido arquivo também está disponível na Tabela 39. O algoritmo implementado em *python* é executado no projeto Java através da biblioteca Jython[12].

Como resposta ao treino, a taxa de precisão ficou em 97.68%. Essa taxa é calculada através da função *cross_val*, que faz uma medição probabilística considerando a quantidade de rótulos (-1), (0) e (1) e as respectivas instâncias (quantidade de registros). A quantidade de instâncias (5822) foram criadas através de um algoritmo que gerou aleatoriamente valores arbitrários entre as características e ao final foi definida uma classificação com base nos resultados.

Por exemplo, o vetor de valores *0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0* é um exemplo das 5822 instâncias. Os primeiros 25 dígitos representam os valores de atribuição para cada característica. No exemplo ilustrado, é possível observar que a respectiva página apresentou um registro de domínio relativamente recente (fluxo 8, atribuído o valor $+1$), uma atividade de *uptime* relativamente recente (fluxo 17, atribuído o valor $+1$) e também uma porta diferente da padrão (fluxo 18, atribuído $+1$). Como a soma resulta em 3, a página em questão é classificada como suspeita (rótulo zero - 0), conforme o último dígito da sequência.

5.2 GESTÃO DA FILTRAGEM POR LISTA DE PERMISSÃO

A solução *server-side* é descrita como um serviço que disponibiliza uma interface programável (API) que favorece a interoperabilidade, seja a comunicação do repositório com o navegador Web, bem como possibilitar que o proprietário das marcas e seus parceiros se comuniquem com o repositório sem a intervenção de terceiros.

5.2.1 Arquitetura Orientada à Serviços (SOA)

O serviço em questão é uma aplicação que adota o paradigma *Service Oriented Architecture* (SOA), ou seja, suas funcionalidades são disponibilizadas baseadas em serviço. O estilo arquitetural adotado foi o REST, que possibilita a interação com diversos dispositivos distintos que se comunicam pela Web. Em termos tecnológicos, foi adotado o Jersey [13]e o Hibernate[14] como

[12] https://www.jython.org/
[13] https://eclipse-ee4j.github.io/jersey/
[14] https://hibernate.org/

mecanismo *Object Relation Modeling* (ORM) para persistência e o AspectJ[15] para realização da auditoria das atividades do serviço.

5.2.2 Requisições federadas

Conforme descrito na Seção 5.2.1, a comunicação entre o *client-side* e *server-side* será realizado através de requisições em uma API REST. Para assegurar autenticação e autorização, o paradigma adotado faz uso de *frameworks* como *Spring Security*[16], *Spring Boot*[17] e *OAuth2*[18].

5.3 PLATAFORMA PIRACEMA.IO E PROTEÇÃO CLIENT-SIDE

Adicionalmente, a presente Tese também elaborou um serviço Web, que assume o domínio registrado piracema.io[19]. Em sua página principal existe um campo que solicita ao usuário a entrada de uma URL. Uma vez submetida, o serviço sugere um veredito sobre a mesma. Esse serviço faz uso das mesmas tecnologias descritas na Seção 5.2.1.

Além disso, também foi proposta uma proteção *client-side* em forma de extensão para navegadores (Chrome e Firefox), que ao clicar no ícone da aplicação, faz uma requisição ao serviço Web enviando a URL acessada na aba do navegador. Essa extensão foi desenvolvida conforme as especificações descritas na documentação *Chrome Developers*[20], que limitam o desenvolvimento das aplicações em linguagens interpretadas, e não compiladas, pelo navegador, como HTML, JavaScript e CSS. Devido a isso, não foi possível armazenar a lógica da árvore no lado do cliente.

5.4 CONSIDERAÇÕES FINAIS

Diante o exposto, este Capítulo teve o intuito de apresentar a abordagem arquitetural com um menor nível de abstração, uma vez que especifica ferramentas e certos *frameworks* utilizados para a elaboração do protótipo, artefato resultante da abordagem do Capítulo. O próximo Capítulo apresenta uma avaliação, ou seja, uma metodologia formal para avaliar os resultados preliminares do protótipo apresentado.

[15] https://www.eclipse.org/aspectj/
[16] https://spring.io/projects/spring-security
[17] https://spring.io/projects/spring-boot
[18] https://oauth.net/2/
[19] https://piracema.io
[20] https://developer.chrome.com/extensions

6 AVALIAÇÃO DO PROTÓTIPO

Esse Capítulo descreve uma metodologia para avaliar os resultados do protótipo conforme exposto no Capítulo anterior, visando submeter o mesmo a um ambiente de avaliação. A metodologia adota foi baseada em um experimento controlado. Maiores detalhes são descritos nas seções que se seguem.

6.1 EXPERIMENTO CONTROLADO

Com base na Engenharia de Software Experimental (ESE), um experimento controlado sugere um modelo de avaliação qualitativo e/ou quantitativo, visando medir e analisar um artefato em questão. Além disso, o processo precisa seguir um protocolo de execução que possibilite que terceiros possam replicar o cenário de teste proposto (Basili; Selby; Hutchens, 1986).

A ideia é fazer uso de uma abordagem evolucionária, fazendo com que o artefato possa ser melhorado a medida que o mesmo é submetido a um processo de experimentação. Portanto, a pesquisa adotou a abordagem experimental como meio de testar hipóteses. Conforme Wohlin et al (WOHLIN et al., 2000), a abordagem experimental é um meio eficiente de avaliar uma determinada ferramenta ou processo, através de variáveis e amostras. Contudo, o sucesso de tal abordagem é condicionada de boas decisões relacionadas a estruturação do experimento, com uma boa definição dos objetivos.

Um experimento é tido como controlado porque todo o ambiente de experimentação foi criado pelo projetista. Todavia, apesar de simular um ambiente real, não é possível assumir que seus resultados apresentados serão idênticos na hipótese da solução ser submetida em um ambiente real. Em contrapartida, o processo de um experimento controlado tem seus benefícios por apresentar resultados que podem nortear sobre os objetivos específicos.

Como o experimento em questão tem o intuito de contabilizar os sucessos obtidos na predição da solução, a solução de predição a ser avaliada é tida como uma **variável independente** (tratamento). Já o resultado de sucesso ou falha é um exemplo de **variável dependente** (causa). Importante mencionar que tanto a conclusão das extrações, bem como a execução do experimento foi realizado no dia 08/05/2020, sendo essa a data o ponto de partida para checar as características temporais. Por fim, é assumido que os experimentos propostos são de **validade interna**, já que suas hipóteses são explicações plausíveis no cenário proposto.

De posse desses dados, o próximo passo foi estabelecer um processo automatizado que tenta renderizar cada URL da amostra, representando um processo recursivo que se repete até submeter toda a lista de URL. O procedimento inicial é verificar se a proposta consegue identificar algum comportamento suspeito. Essa formalização de métricas para captura dos dados é tida como a **medição** dos resultados.

Caso não seja suficiente, a solução solicitará o conteúdo da página. Caso seja identificado

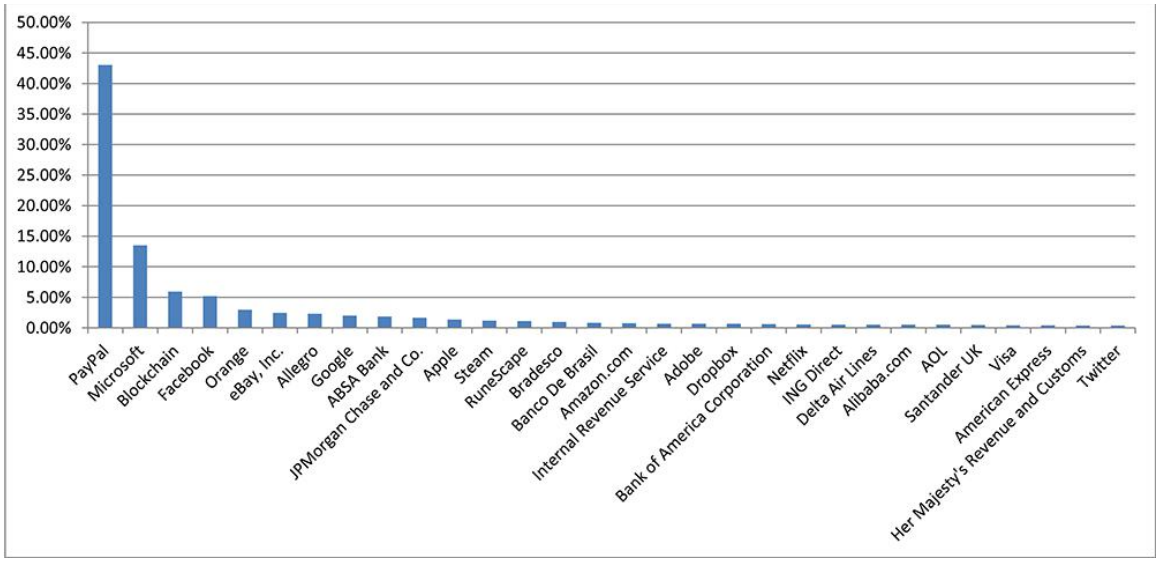

Figura 37 – Marcas envolvidas no processo de avaliação

um comportamento, o próximo passo será reconhecer a marca envolvida, seja através da URL, conteúdo, OCR ou reconhecimento de imagem (RI), seguindo essa ordem de execução e prioridade. No final, é contabilizado o total de sucessos na detecção de comportamentos suspeitos através de cada fluxo. Cada fluxo é uma etapa no processamento da árvore, que pode ter uma ou mais questionamentos, a exemplo do fluxo 1 que possui 10 questionamentos (1, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8 e 1.9).

Como meio de investigar as marcas envolvidas no processo, foi realizada uma análise das marcas em potencial presentes na amostra. O JSON do *PhishTank* possui um campo "*target*", contudo foi possível observar que o mesmo possui diversos registros atribuídos como "Other". Além disso, também foi constatado que muitos valores desse campo não condiziam com a marca alvo em questão. Diante disso, a solução extraiu as marcas envolvidas através da busca textual e visual. Com base na Figura 37, foi possível observar 30 marcas mais exploradas.

Diante o escopo da proposta, dois experimentos foram planejados. O primeiro experimento tem o objetivo de analisar os falsos positivos e negativos da proposta. Já o segundo experimento tem o objetivo de avaliar a relevância da proposta em relação as soluções nativas existentes nos navegadores e também observar a análise gradual.

6.1.1 Experimento 1

Para o primeiro experimento, foi estabelecido o protocolo conforme descrito na Tabela 7. Foram estabelecidas duas fases, a saber: (i) a obtenção de uma amostra de páginas reais para serem submetidas a solução *client-side*, que atua como um navegador Web sensível ao contexto; e (ii) e o processo gerenciado de submissão dessa amostra ao conjuntos de regras. Em (i), o processo precisou obter um número considerável de *phishing* válidos e inválidos, recapitulando, válidos significam páginas confirmadas como maliciosas pela plataforma de

denúncia, e inválidos significam páginas classificadas como não maliciosas, ou seja, falsos positivos publicados na plataforma.

Tabela 7 – Protocolo do primeiro experimento

Hipóteses	A hipótese nula (H_0) assumida no primeiro experimento é: O resultado obtido da árvore de classificação na amostra de *phishing* válidos é o mesmo quando submetido em *phishing inválidos*.
	A hipótese alternativa (H_1) assumida no primeiro experimento é: A árvore de classificação proposta visa apresentar baixos falsos positivos quando submetido à *phishing* válidos.
	A hipótese alternativa (H_2) assumida no primeiro experimento é: A árvore de classificação proposta visa apresentar baixos falsos negativos quando submetido à *phishing* inválidos.
Métricas	Todas as URL foram registradas como *snapshot* na base de dados. Por tanto, foi necessário apenas executar o algoritmo da solução diretamente na URL e código fonte de cada registro da base de dados, e contabilizar os resultados da predição. Além dos sucessos, também serão contabilizados os falsos positivos (os registros classificados como não confiáveis na amostra de inválidos) e negativos (os registros classificados como confiáveis na amostra de válidos).
Amostras	*PhishTank* Válidos e *PhishTank* Inválidos

Existem fluxos que analisam o conteúdo, todavia, a página precisaria estar online. E como o *phishing* é bastante volátil, foi decidido preservar as informações da resposta em um banco de dados local, no intuito de manter um *snapshot* do conteúdo de cada página obtida. O objetivo é **preservar o estado do *phishing***, como uma fotografia do mesmo em um determinado momento, devido sua natureza volátil, possibilitando que o experimento possa ser replicado a qualquer momento por terceiros.

Com relação ao primeiro experimento, a pesquisa adotou o *PhishTank*, que disponibiliza um **arquivo JSON**[1] para a composição dos *phishing* válidos. O mesmo foi baixado no dia 08/05/2020 e constavam com 14.768 registros confirmados. Além da URL, status, confirmação e data de publicação, também fornece a **data de confirmação** e a **marca alvo da fraude**. A data de confirmação é o momento que a votação se encerra e a URL tida como *phishing* confirmado. Contudo, por ser necessário obter o conteúdo, para o experimento foram considerados 4039 registros. Já a função *"phish search"* da plataforma serviu para a composição dos *phishing* inválidos, processo similar à extração descrita no Capítulo 3.

Foi possível obter 1210 registros, porém apenas 661 registros estavam online. A construção da base *snapshot* é ilustrada na Figura 38. E o processamento de cada fluxo é ilustrado na Figura 39. Somando os 8 fluxos no escopo da URL com os 4 fluxos

6.1.2 Experimento 2

Para o segundo experimento, foi estabelecido o protocolo conforme descrito na Tabela 8. Foi pretendido fazer uma comparação da proposta do estudo com certas soluções nativas disponíveis nos navegadores mais populares. Diante disso, foi proposto um fluxograma para

[1] http://data.phishtank.com/data/online-valid.json.bz2

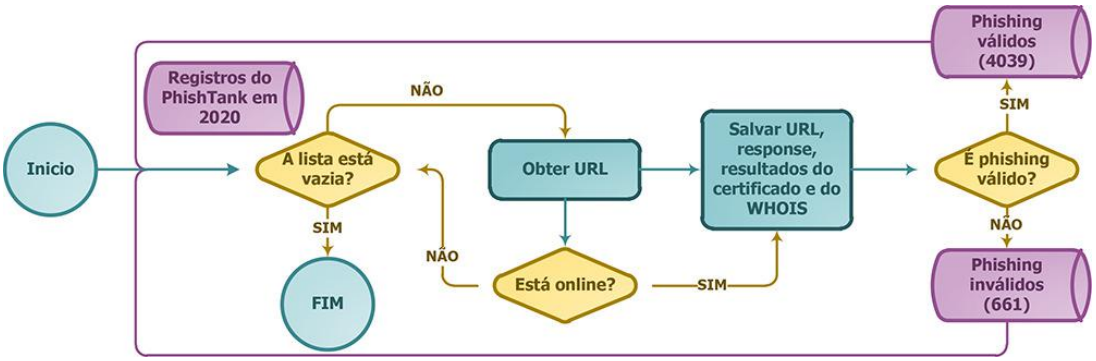

Figura 38 – Fluxograma da construção da base *snapshot*

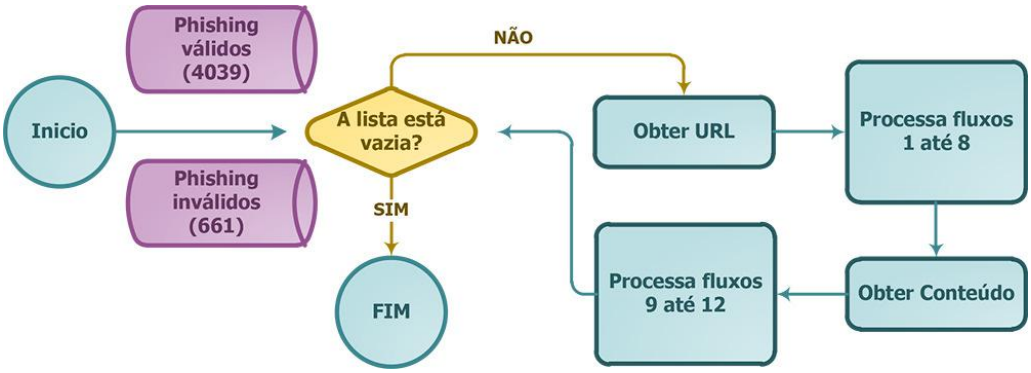

Figura 39 – Etapas do processamento dos fluxos

coleta de informações, conforme ilustrado na Figura 40. Os navegadores Chrome, Firefox e Safari utilizam um mesmo mecanismo, o *SafeBrowsing*.

Tabela 8 – Protocolo do segundo experimento

Hipóteses	A hipótese nula (H_0) assumida no segundo experimento é: Não há mudanças significativas quando a solução proposta é aplicada ao cenário de atuação.
	A hipótese alternativa (H_1) assumida no segundo experimento é: A proposta, em seu escopo atual, melhora a proteção do usuário-final quando combinada as soluções nativas dos navegadores.
	A hipótese alternativa (H_2) assumida no segundo experimento é: A análise gradual, em seu escopo atual, apresenta-se como uma estratégia que oferece melhor desempenho e menor violação da privacidade durante a predição de *phishing*.
Métricas	Um algoritmo que faz uso do *Selenium WebDriver* executa um laço de repetição, abrindo individualmente cada URL em uma janela do navegador. Com isso, observa se a página em questão foi bloqueada ou não pela solução avaliada.
Amostras	*OpenPhish* Válidos

Diante disso, foi adotado ao experimento o uso do Chrome, representando assim os 3 navegadores mencionados. Além dele, os navegadores da Microsoft, Edge e Internet Explorer, também compartilham um mesmo mecanismo, o *SmartScreen*. Por fim, o Opera faz uso da base do PhishTank. Conforme o último mencionado, foi preciso buscar uma outra amostra de *phishing* que não fosse oriunda do *PhishTank*, evitando obter resultados enviesados. Portanto, a amostra para esse comparativo entre os navegadores teve que ser diferente, e foi adotada

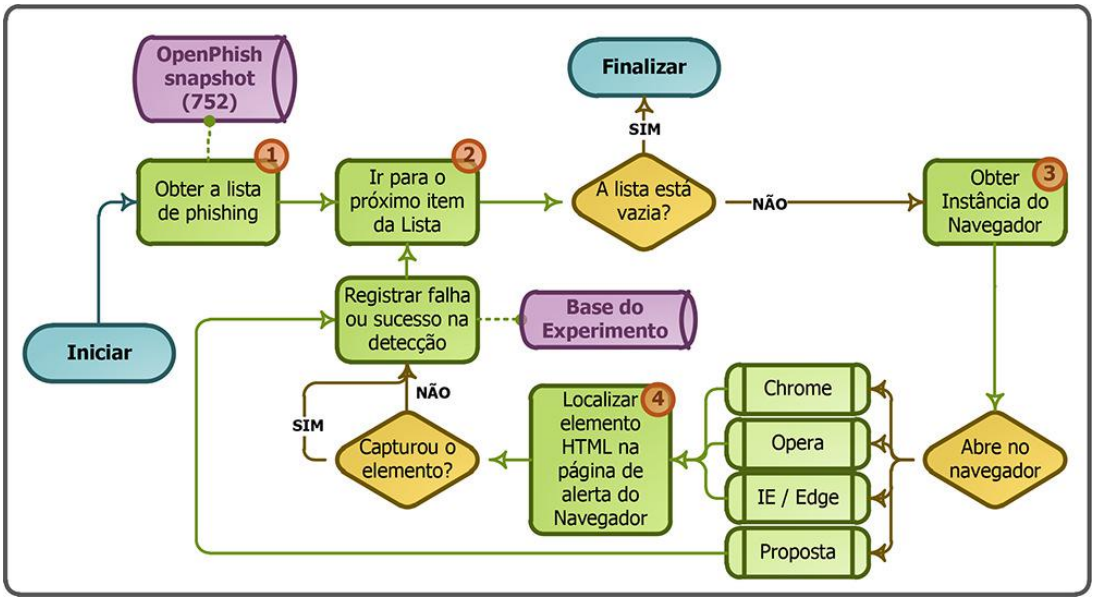

Figura 40 – Experimento de comparação com as soluções nativas dos navegadores

uma amostra obtida pelo *OpenPhish*. Concluindo as variáveis independentes, a proposta é representada pelo protótipo que simula um navegador.

Considerando que o cenário irá acessar milhares de *phishing* simultaneamente, foi preciso estabelecer testes automatizados. A ferramenta escolhida para proporcionar essa automação foi a *Selenium Server*[2], que oferece automação de testes diretamente no navegador. Também foi utilizada a biblioteca *Selenium WebDriver*, um mecanismo que realiza os testes definidos em uma instância do navegador, dando suporte a praticamente todos os navegadores disponíveis. Para garantir maior precisão nos resultados, após a execução dos testes, foi realizada uma triagem no *LOG* na conclusão da aplicação que invoca as instâncias do navegador, verificando assim o motivo das falhas nos métodos, para confirmar se o teste ocorreu como esperado.

O próximo passo é identificar se, ao invés da página do *phishing*, foi exibida a página de alerta. Esse tratamento precisou ser adaptado de acordo com o navegador utilizado, pois cada um possui ter particularidades na resposta à ameaça. Porém, ambos compartilham o mesmo comportamento de sobrepor a página maliciosa com uma página de alerta. É possível obter elementos dessa página de alerta com a biblioteca *Selenium WebDriver*, tal busca pode ser feita pelo conteúdo do HTML existente. Como exemplo, no Opera a checagem foi feita pelo texto *"Aviso de Fraude"* no conteúdo da página. Por outro lado, pelo Chrome a checagem foi no *<title>* da página, pelo texto *"Erro de Segurança"*[3]. Por fim, se o elemento ou título fosse reconhecido, significaria que a tela de alerta havia sido exibida, indicando que o filtro foi bem sucedido. Caso contrário, seria registrada uma ocorrência de falha na filtragem do navegador.

Diante desse cenário, foi possível extrair 3664 URLs no arquivo *feed.txt* disponibilizado[4]

[2] http://docs.seleniumhq.org/download/

[3] A busca foi feita pelo título da página porque o navegador impedia o acesso aos detalhes sobre o conteúdo da página de alerta.

[4] https://openphish.com/feed.txt

na página do *OpenPhish* e baixado no dia 08/05/2020. Contudo, dos mencionados, apenas 752 estavam online. Com isso, foi definida a amostra para avaliar os mecanismos nativos.

6.2 RESULTADOS OBTIDOS

Essa seção descreve os resultados obtidos nos dois experimentos. Os registros de *phishing* válidos (ameaça confirmada) são oriundos de denúncias de usuários registradas em plataformas como o *PhishTank* e *OpenPhish*. Já os registros de *phishing* inválidos (alarme falso) foram obtidos no *PhishTank*, considerados pela mesma como denúncias indevidas. Os arquivos criados por esse estudo, bem como seu protótipo estão disponibilizados para conferência, maiores informações no Apêndice da Seção B.

6.2.1 Métricas de Variação

Outro destaque são as métricas na variação dos valores atribuídos na pontuação das páginas analisadas. Essa informação se faz importante para oferecer justificativa e embasamento para a definição de valores que correspondem ao veredito da página (o porque do > 5 na árvore de decisão). Esses resultados podem ser visualizados nos gráficos da Figura 41.

A média aritmética, por ser um dado que representa um padrão existente, pode ser adotada como um indicador para definir a fronteira entre o que é ou não uma página maliciosa. Conforme ilustrado no primeiro gráfico da esquerda, é possível observar que as médias apresentadas nos registros de *phishing* válidos nas amostras do *OpenPhish* e *PhishTank* possuem valores aproximados, 5.97 e 7.25. Já na amostra de *phishing* inválidos, a média aritmética foi 1.68. Contudo, para maior precisão, é interessante calcular o desvio padrão (DP) visando observar a dispersão dos valores individualmente em torno da média amostral. Conforme no gráfico citado, as amostras do *OpenPhish* (válidos), *PhishTank* válidos e inválidos apresentaram, respectivamente, os seguintes valores de DP: 1.32, 1.69 e 0.48.

Considerando a grandeza das variáveis, os desvios padrões apresentados foram relativamente baixos. Contudo, o processo de classificar como alto ou baixo pode ser subjetivo. Porém, um meio de julgar essa variedade como alta ou baixo pode ser realizado através do cálculo do coeficiente de variação (CV), sugerindo que quanto menor o CV, mais homogêneo é o conjunto de dados. Um CV tido como baixo pode variar de acordo com a aplicabilidade,

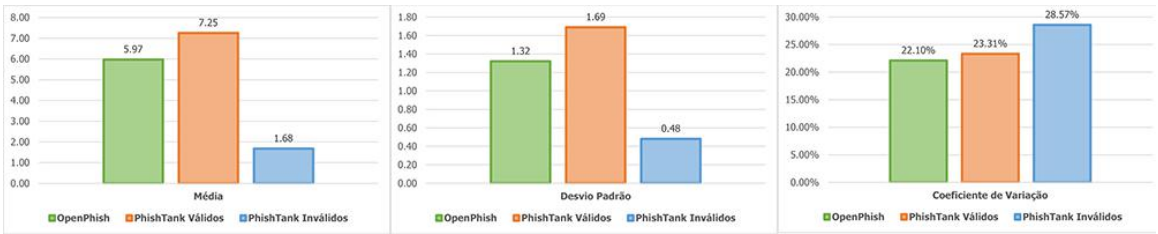

Figura 41 – Média e Desvio padrão nas duas abordagens de classificação

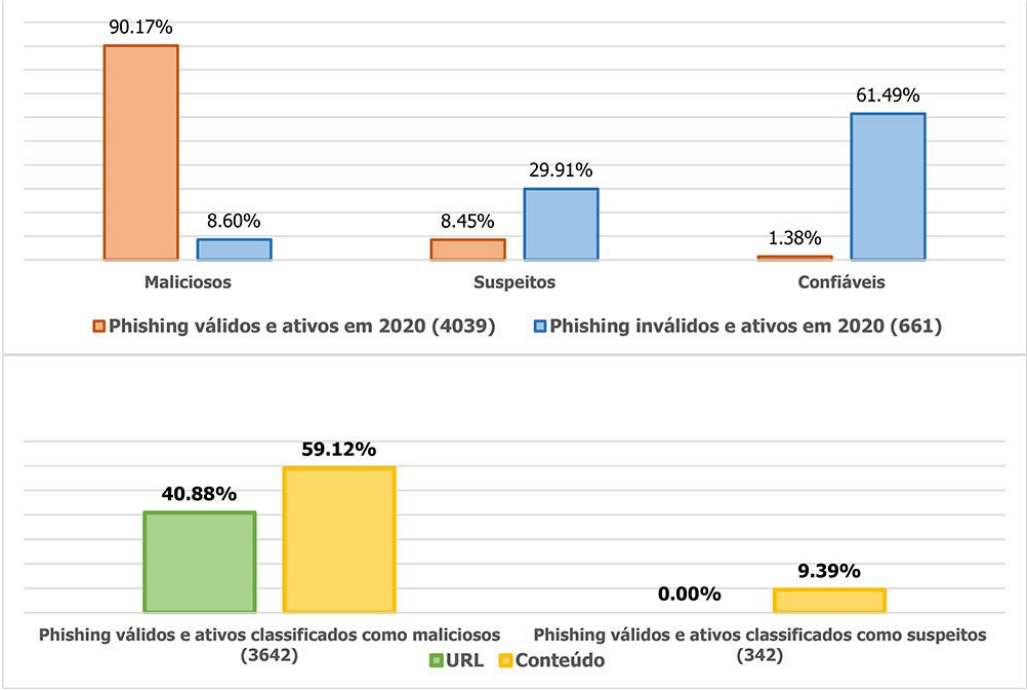

Figura 42 – Resultados no Experimento 1 (H_1 e H_2)

mas um senso comum na literatura é considerar aceitável um percentual inferior a 30% (CUI, 1989). Diante disso, conforme ilustrado no gráfico mais a direita da Figura 41, as taxas de CV apresentadas foram: 22.10%, 23.31% e 28.57%.

6.2.2 Resultados do Experimento 1

Com base no exposto, foi executado o experimento considerando analisar a precisão da solução proposta, submetendo a mesma em uma avaliação que considera a possibilidade de falsos positivos e negativos, metodologia comparativa da qual adotou duas amostras distintas (*PhishTank*), uma com *phishing* reais e outra com páginas não maliciosas. Os resultados obtidos no experimento são ilustrados na Figura 42.

Além de uma boa resposta na detecção de ameaças, é importante que a solução também saiba distinguir uma página não maliciosa, evitando assim falsos positivos. Com base nos resultados, foi possível observar que existe uma tendência contraria nos resultados entre as amostra de válidos e inválidos, isso reflete como ponto positivo na proposta. Na amostra de válidos, a solução apresentou 90.17% de taxa de acerto e 9.83% de erro. Diante ao resultado obtido, é possível evidenciar a hipótese (H_1). Já na amostra de inválidos, a solução teve uma taxa de acerto de 61.49% contra 38.51%. Contudo, da taxa de erro nos válidos, 8.45% ainda foram considerados suspeitos e apenas 1.38% tido como confiáveis. Já na amostra de inválidos, 29.91% foram considerados suspeitos e apenas 8.60% erroneamente como maliciosos, sendo esse último de fato a margem de erro. Diante ao resultado obtido, é possível evidenciar a hipótese (H_2) e descartar a hipótese alternativa (H_0).

Também foi verificada a incidência das categorias de característica em cada registro avali-

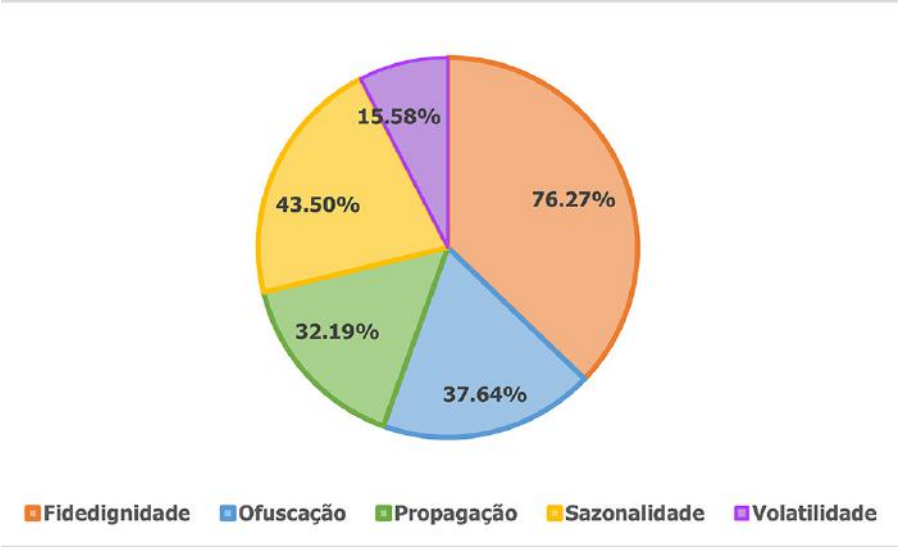

Figura 43 – Incidência das categorias na amostra de válidos do *PhishTank*

ado. Primeiramente, Considerando a amostra com 4039 *phishing* válidos do *PhishTank*, foram detectadas 20487 ocorrências de categorias. É importante salientar que cada registro pode conter 25 comportamentos e cada comportamento pode apresentar mais de uma categoria diferente, por isso o valor tão elevado de ocorrências de categorias. Conforme ilustrado no gráfico de pizza da Figura 43, dentre essas ocorrências, 76.27% são relacionadas à comportamentos da Fidedignidade, demonstrando a relevância no contexto de atuação do *phishing*.

Já na Figura 44 é possível observar o resultado individual para cada característica por registro. Todas as 4 características temporais se destacaram nos resultados, como a idade de emissão do certificado, a idade do registro do domínio, o tempo de atividade do host e conteúdo sazonal. Tornando evidente que aspetos temporais, apesar de serem mais custosos, tem resultados eficientes. Na mesma linha, 12 características atemporais também se destacaram, a saber: domínio com simulação TLD, domínio com TLD na lista suspeita, domínio com tentativa homográfica, domínio com nome muito grande, subdomínio com simulação TLD, subdomínio com tentativa homográfica, subdomínio com sintaxe grande, domínio com muitos subdomínio, host na lista suspeita, conteúdo com elementos x-origin, domínio com comportamento forjado e conflito de autoria.

O critério de destaque nesse contexto refere-se a diferença entre as ocorrências em válidos e inválidos. Das características mencionadas, a ocorrência de comportamentos suspeitos na amostra de válidos é, ao menos, 5 vezes maior em comparação as ocorrências na amostra de inválidos. Em alguns casos essa diferença chega a ser mais de 80 vezes. Esse comportamento demonstra-se positivo porque destaca as características mais e menos relevantes, ou seja, teoricamente, essas características são determinantes na redução de falsos positivos e negativos, tornando-se candidatas a serem executadas com prioridade em qualquer modelo de predição, sendo peças-chave para aumentar a precisão da solução.

Contudo, observações precisam ser levantadas sobre os resultados da Figura 44. Por exem-

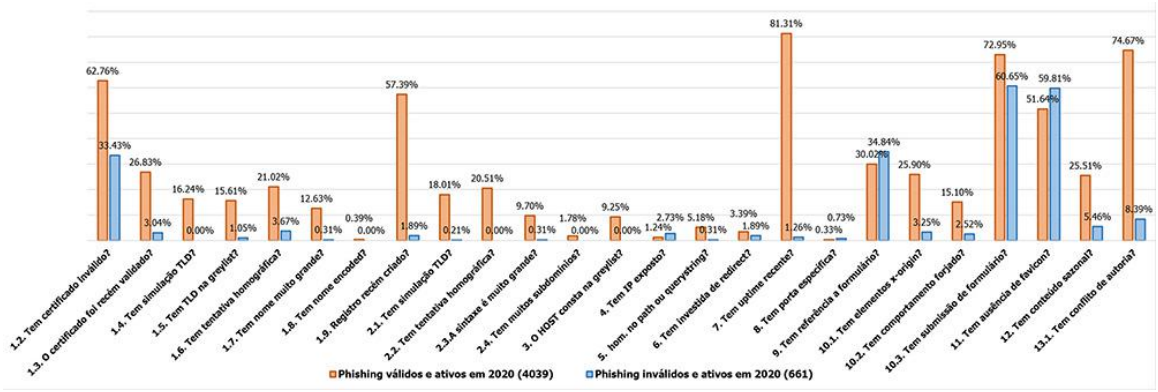

Figura 44 – Resultados por características na amostra de válidos do *PhishTank*

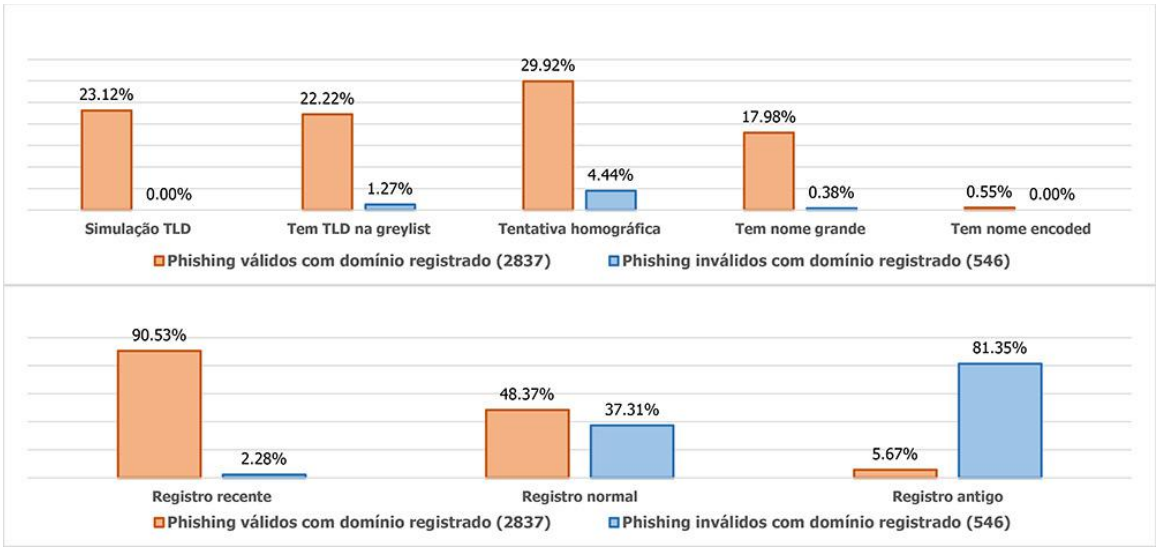

Figura 45 – Resultados relacionados ao domínio

plo, em relação a taxa de certificados recém validados na amostra de *phishing* válidos, foi apresentado 26.83%, mas é importante elucidar que esse percentual é com base na quantidade global, ou seja, de um total de 4039 registros. Contudo, dos 4039 registros, apenas 1504 apresentaram certificados válidos, portanto, do total de registros com certificados válidos, a quantidade de certificados recém-validados é de 72.03%.

Isso revela que é bastante típico que um domínio registrado em um *phishing* possui um tempo de atividade muito curto. Em contrapartida, páginas genuínas, quando possuem domínio registrado, geralmente o domínio tem um longo período de atividade. Esse comportamento também ocorre em casos de URL que não possuíam domínio registrado. Diante disso, os gráficos 45, 46 e 47 são ilustrados para destacar, respectivamente, o domínio, subdomínio, certificado e uptime.

Com base na Figura 45, é possível considerar que todas as características referente a domínio são relevantes, indicando que *phishing* direcionados vem se tornando mais frequentes nos dias atuais, que os mal intencionados buscam elaborar suas fraudes com mais riqueza em

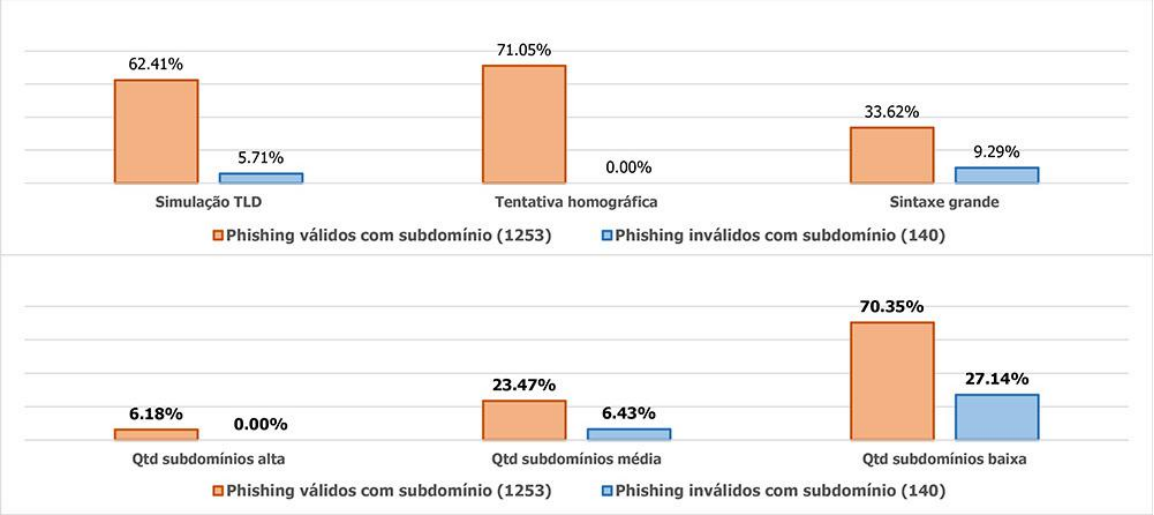

Figura 46 – Resultados relacionados ao subdomínio

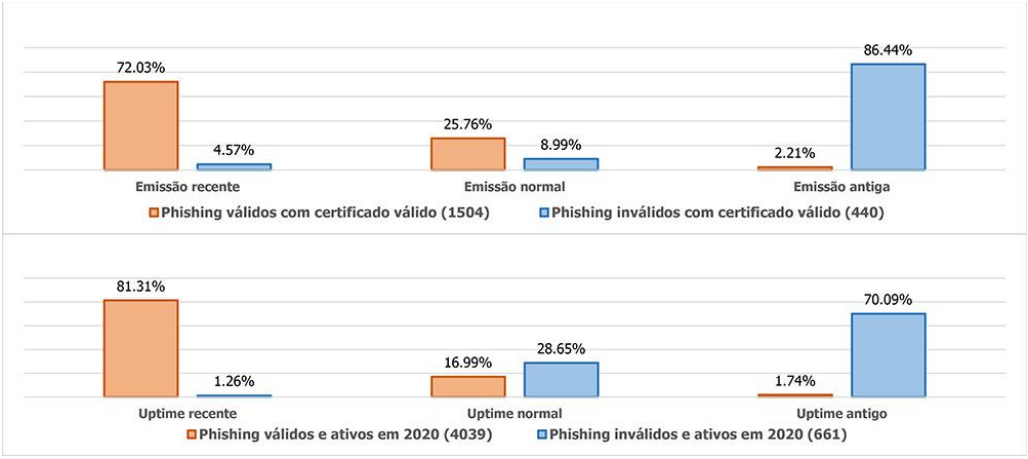

Figura 47 – Resultados relacionados ao certificado digital e uptime

detalhes, não abrindo mão de registrar um domínio para aumentar a fidedignidade. Na mesma linha, o tempo de vida do domínio denuncia a natureza volátil do *phishing*, 90.53% dos 4532 *phishing* válidos com domínio registrado possuem domínio com pouco tempo de atividade. Em contrapartida, *phishing* inválidos registram apenas 2.28% de seus domínios classificados com pouco tempo de atividade.

Em relação a Figura 46, assim como descrito sobre domínio, as características de subdomínio também apresentam-se relevantes. Tais elementos representam a identidade visual de uma marca, portanto, são comportamentos que merecem destaque em relação a forja da fidedignidade. Na comparação entre válidos e inválidos, a ocorrência desses comportamentos é ao menos 5 vezes maior, apresentando-se como um parâmetro eficiente para distinguir o que é confiável ou malicioso. Apesar da quantidade registros com muitos subdomínios na mesma URL ser ainda modesta nos registros de válidos (6.18%), ainda sim, é importante destacar que não houve qualquer ocorrência dessa característica na amostra de inválidos.

Sobre a Figura 47, em relação ao certificado digital, essa característica temporal se apre-

sentou muito eficiente para distinguir uma fraude de uma página genuína, isso fica evidente na quantidade de emissões recentes na amostra de válidos e a quantidade de emissões antigas na amostra de inválidos. Na mesma linha, a análise de uptime do host segue o mesmo raciocínio, destacando ambas as características com alta relevância.

Em relação aos resultados obtidos na amostra de válidos, apesar da alta taxa de precisão, ainda sim houveram casos de páginas maliciosas serem consideradas suspeitas ou mesmo confiáveis. Uma explicação é que em certos casos podem ocorrer sequestro de domínio ou mesmo uma página maliciosa ser injetada em um HOST legítimo. Para esses casos, a presente solução não garante proteções. Em relação a amostra de inválidos, apesar de baixa, ainda sim é considerável a taxa de erro apresentada. Uma explicação é que muitos proprietários não adotam boas práticas na distribuição de seus recursos, seja com ausência de certificado ou adoção de padrões não recomendados na composição da URL.

Portanto, situações que mal intencionados exploram a reputação de serviços com prestígio podem ter seus desafios. Exemplos dessa prática: criação de perfis *fakes* para aplicar golpes em *Twitter*, *Facebook*, *Instagram*, bem como hospedar páginas ou formulários em serviços do *Google* (docs, sites) ou da *Microsoft* (como o forms do office.com).

Outro ponto interessante é a necessidade da avaliar o conteúdo do *phishing*. Como anteriormente ilustrado na Figura 42, mais da metade: 59.12% precisaram ter o conteúdo analisado para receber a confirmação. Contudo, 40.88% dos registros receberam veredito apenas pela URL, o que não deixa de ser um bom número, justificando a eficiência e importância da analise gradual. Apenas salientando, no gráfico da Figura 42, que a primeira coluna dos registros suspeitos está zerada porque obviamente nenhuma característica é considerada suspeita analisando apenas a URL, todos os casos de suspeitos e confiáveis sofreram todos os saltos disponíveis, tanto na URL como no conteúdo.

6.2.3 Resultados do Experimento 2

Com base no exposto, foi executado o experimento que faz um comparativo entre as soluções correlatas existentes nos navegadores populares, fazendo uso de amostra extraída do *OpenPhish*. Os resultados obtidos no experimento são ilustrados na Figura 48.

Conforme ilustrado na Figura 48, o gráfico em barras do topo descreve a taxa de detecção de páginas confirmadas como ameaças (descartando os suspeitos ou confiáveis). Foi possível observar uma taxa de 92.69% de acerto da proposta, já as demais opções, tiveram taxas de erro superiores a 20%. Uma explicação é que essas soluções são baseadas em lista de bloqueio, elas dependem da sincronização periódica de suas listas para terem maior efetividade. A grande problemática é que em questões de horas, diversas pessoas podem cair em golpes, evidenciando a importância da estratégia preditiva da proposta.

No gráfico da parte inferior, descreve os casos de falhas apresentados pelos navegadores e a ferramenta atuando como uma solução complementar, realizando uma proteção nos casos que o navegador acabou falhando. Por exemplo, das 752 páginas, 165 o Chrome não reconheceu

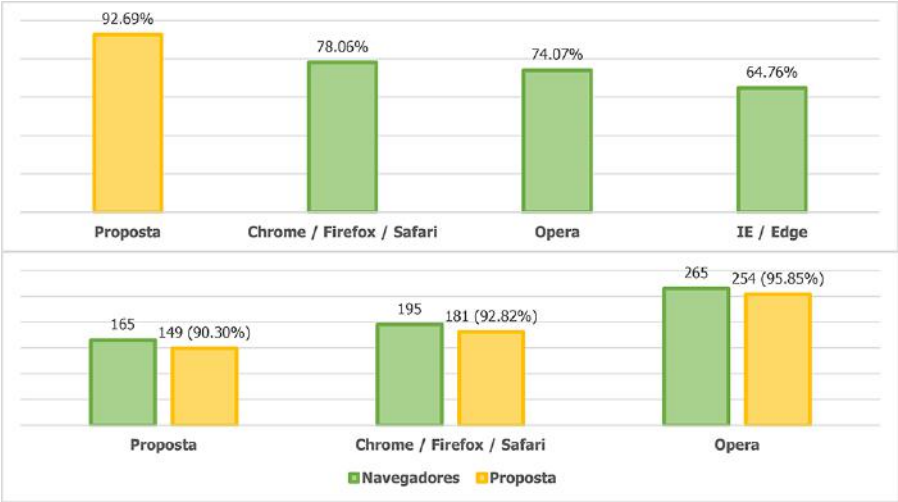

Figura 48 – Resultados em comparação e cooperação com os navegadores (H_1)

como ameaça, não protegendo o usuário ao acesso. Diante disso, a proposta foi submetida nessas 165 páginas e conseguiu impedir 149 dessas (90.30%), minimizando os casos sem resposta, evidenciando assim a hipótese (H_1). Os demais casos também apresentaram taxas superiores a 90%, destacando que a proposta aumentou consideravelmente a superfície de proteção para o usuário-final.

6.2.4 Comparativo entre as abordagens de (SE) e (ML)

A motivação de comparar as duas abordagens é fazer uma avaliação sobre o custo benefício da decisão da análise gradual possuir ou não gatilhos. O fato é que os gatilhos que interrompem o processo podem acabar resultando em falsos positivos, contudo, uma redução no custo computacional e da privacidade é notoriamente obtida através dessa abordagem, que é possível apenas no contexto de SE, já que no contexto de ML exige como entrada o valor resultante de cada característica, ou seja, nela é necessário analisar todas as características para tomar uma decisão, conforme já discutido na Seção 4.4.4.

Um comportamento detectado foi a quantidade de fluxos necessária para cada o motor de inferência julgar uma página como maliciosa. Com base na Figura 49 esse comportamento é ilustrado. O eixo X do gráfico descreve a quantidade de fluxos (saltos) processados para atingir os vereditos anteriormente confirmados (4039). É possível observar que boa parte (70.68%) concentra-se entre 20 até 25 saltos efetuados. Esse comportamento evidência a relevância da presença das 25 característica, uma vez que 21.03% precisaram de 20 dos 25 fluxos.

Todavia, no mesmo gráfico, foi possível observar que apenas 4.12% precisaram entre 3 até 8 saltos para atingir uma confirmação de ameaça, descrevendo um equilíbrio sobre a pontuação definida pelo estudo para julgar a página, evitando ocorrências de falsos positivos. Também é importante destacar que a média de saltos foi 18.36, o que resultou em uma redução ao comparar os 25 saltos sempre realizados na abordagem ML (que não é possível adotar a análise gradual), evidenciando a hipótese (H_2) e descartando a hipótese alternativa (H_0).

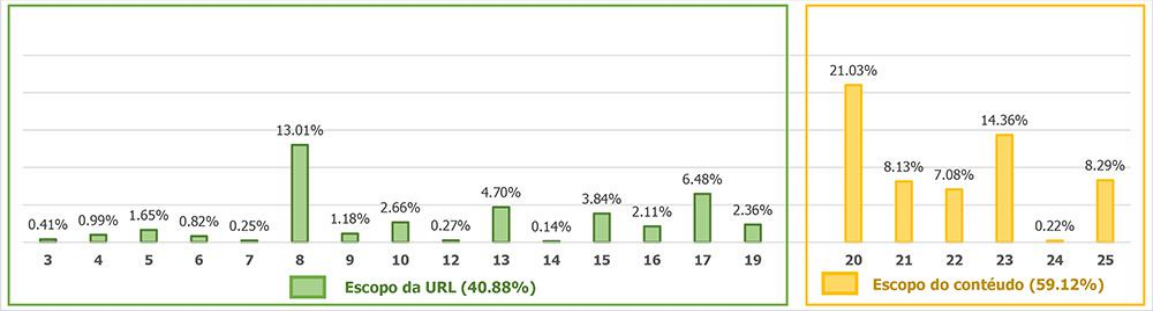

Figura 49 – Fluxos processados necessários para um veredito (H_2)

Tabela 9 – Registros de falsos positivos entre as abordagens SE e ML

SE	**ML**	**%**
julgados como 1	julgados como -1	0.25%
julgados como 1	julgados como 0	2.62%
julgados como 0	julgados como -1	0.05%
julgados como 0	julgados como 1	0.00%
julgados como -1	julgados como 0	0.02%
julgados como -1	julgados como 1	0.00%

O fato é que no presente contexto, devido a natureza da obtenção das informações, o processo de extração de características envolvem requisições remotas e um volume considerável de informação, portanto, medidas que possam atenuar esse problema são bastante relevantes. O desempenho se enviesa no custo computacional da operação (WANAWE; AWASARE; PURI, 2014; LI et al., 2019). Diante a tantas opções de navegadores, um alto desempenho no carregamento das páginas torna-se um critério relevante quanto a escolha do navegador, ou seja, soluções heurísticas precisam balancear a quantidade de características adotadas, caso contrário podem inviabilizar sua aplicabilidade. Além disso, a privacidade do usuário é menos violada.

Diante da presente avaliação, foi observado que nas páginas julgadas precocemente como maliciosas, apenas 2.62% apresentou um falso positivo, uma ameaça que faz valer o custo benefício envolvido, com base nos números apresentados. Além disso, esse comportamento também demonstra que a ordem das características a serem analisadas estão com certo grau de aceitação no quesito relevância. A Tabela 9 detalha o comportamento de cada incongruência detectada entre as abordagens.

Por fim, é importante destacar que cada modelo utilizou uma tecnologia distinta em sua implementação. No caso do modelo SE foi utilizada a tecnologia Java e o modelo ML fez uso do Python. Devido a isso, o estudo não fez uma comparação de desempenho sobre o tempo de resposta (em milissegundos), porque iria sugerir um resultado enviesado, uma vez que o ML executou um processo adicionou de geração de *bytecode* para ser executado em ambiente Java (através do Jython). Contudo, as atividades que extraem informações externas são as mais críticas no sentido de desempenho e privacidade, e como elas são realizadas a medida que

a árvore é percorrida, é sensato assumir que a quantidade de saltos seja fator determinante.

Portanto, por possibilitar os gatilhos, em tese, a abordagem de SE já se faz a mais recomendada para esse contexto de aplicação, contudo, foi importante considerar a hipótese de um possível cenário em que a abordagem ML seria mais precisa, então foi investigada a possível margem de erro assumida por optar um SE, o que resultou em 2.62%. Considerando a baixa taxa de erro diante ao benefício de menor custo e intrusão à privacidade, a presente Tese optou pelo modelo SE.

Outra questão é a adesão da abordagem de classificação. O estudo adotou a árvore de decisão (e exemplificou com DTC), contudo é importante salientar que esse modelo também tem seus riscos em relação a superadequação (*over-fitting*). Esse problema remete ao fato do modelo proposto ter excelentes resultados apenas na amostra que foi submetida. Apesar do modelo ter sido submetido à 3 amostras distintas (válidos e inválidos do *Phishtank* e válidos do *OpenPhish*), ainda sim é temerário afirmar que o mesmo não possua essa limitação.

Outro fator é a sensibilidade da árvore com o peso de suas características. Em seu estado atual, a proposta já possui valores previamente pesos atribuídos, contudo, posteriormente pode acontecer uma reengenharia das características, seja a adição de novas (surgimento de novas tendências), o aumento ou diminuição do peso das existentes (tornou-se mais ou menos relevante) ou mesmo remoção plena de uma ou mais características (tornou-se obsoleta).

Diante disso, uma única mudança de tal natureza será suficiente para resultar em uma árvore totalmente diferente, e esse cenário pode ser caótico em situações de relacionamento lineares, ou seja, quando uma característica aumenta e a outra diminui, algo bastante cogitável em um ambiente suscetível à mudanças como é o cenário de atuação do *phishing*.

6.3 AMEAÇAS DA AVALIAÇÃO

Essa seção descreve ameças aos resultados obtidos que podem oferecer variações nos números apresentados. As ameaças foram agrupadas por semântica para maior entendimento.

6.3.1 Ameaças no contexto geral

As ameaças descritas nessa seção são compartilhadas por todos os dois experimentos elaborados, basicamente descrevem limitações sobre a elaboração das amostras ou processo de reconhecimento heurístico.

6.3.1.1 Ameaças na detecção de marcas

Em relação a detecção de comportamentos, a solução adota medidas e contramedidas de iniciativa própria para investigar padrões suspeitos na atividade da navegação. Contudo, em relação ao reconhecimento de marcas, a heurística do procedimento precisa ser acompanhada periodicamente, uma vez que mudanças podem ser adotas por seus proprietários.

Por exemplo, o proprietário da marca pode modificar padrões visuais e textuais da identificação de sua marca. Apesar da pesquisa fazer um levantamento preliminar e exploratório sobre algumas marcas, a fim de prover um cenário minimamente operacional, a clareza na definição dos padrões visuais é indispensável se for considerar o cenário real de atuação. Ou seja, o levantamento de padrões visuais e textuais é uma atividade que precisa ser definida com o apoio do proprietário da marca, e essa dependência torna-se crítica.

Também foi possível evidenciar o problema da heterogeneidade ou falta de detalhes no conteúdo, o que dificulta detectar o direcionamento do ataque para uma marca. No caso da heterogeneidade remete aos casos que duas ou mais marcas ficam em evidencia. Já a falta de detalhes dificulta a conclusão em alguma marca específica. Além disso, muitos elementos embutidos (*mashups*) ou demais referências cruzadas podem dificultar a detecção da marca.

Outra limitação foi o processamento interno de extração de informações por OCR e Reconhecimento de Imagem (RI) com o Google Vision. Devido a natureza complexa em termos de custos em processamento, essas atividades são executadas em último caso, visando não comprometer o tempo de resposta de cada página. Os testes com OCR foram executados em situações que a página continha pouca informação no código-fonte, como casos de *templates* construídos com arquivos embutidos ou elementos *mashups* anteriormente citados.

Já os testes de RI, como sempre são executados após uma investida não suficiente de OCR, tem o atenuante de reutilizar uma imagem já existente, além disso, a imagem criada tem um tamanho relativamente pequeno, como dito anteriormente, em torno de 13.4KB. Todavia, ainda sim é um processo custoso e que acaba comprometendo o tempo de resposta. Foi possível observar no experimento que as requisições que percorriam toda a árvore, ou seja, que extraíam todas as informações possíveis, apresentaram em média um tempo de resposta de 3.8s, isso em casos que não era necessário realizar os procedimentos de OCR e RI. Já nos casos que haviam a necessidade de realizar tais procedimentos, o tempo subiu para 9.18s.

Diante disso, também é preciso considerar outros fatores subjetivos, como a velocidade de conexão do ambiente de execução do experimento e a banda de transferência disponibilizada pelo servidor que hospeda a página maliciosa. É fato que muitas dessas páginas maliciosas são hospedadas em servidores com baixo custo, o que sugere alta latência nas requisições. Por fim, algumas preocupações, como desfocar rostos em possíveis imagens no conteúdo da página, bem como outros tratamentos em determinadas informações sensíveis precisariam ser consideradas visando a privacidade do usuário final, uma vez que nesse modelo proposto, imagens do conteúdo da aba do navegador seriam geradas de forma arbitrária pela solução.

6.3.1.2 Ameaças na construção das amostras

Além disso, alguns casos de *phishing* válidos eram páginas que apresentaram conta suspensa no servidor de hospedagem ou mesmo apresentar erros de HTTP amigáveis, fazendo com que o resultado do carregamento da página apresentasse uma página padrão do servidor Web e não exatamente a página maliciosa. O processo de detecção do *status code* do HTTP minimizou

esse problema, contudo, alguns casos precisaram ser tratados, devido apresentar situações improcedentes, como páginas offline que ao invés de apresentarem código 404, o servidor exibia páginas personalizadas retornando o código 200. Apesar do tratamento, ainda sim a avaliação fica suscetível a ruídos dessa natureza. O mesmo vale para o processo de detecção de páginas padronizadas informando a suspensão.

Não obstante, como em qualquer tipo de experimento, a automação pode ocasionar em riscos na precisão dos resultados obtidos. Contudo, o estudo considera que a confiabilidade da metodologia adquiriu considerável precisão com a decisão de eliminar o caráter subjetivo da avaliação, no caso, a intervenção manual.

Outro fator importante é a possibilidade da existência de clones de *phishing* no processo, o que poderia enviesar alguns resultados. Contudo, o processo de detecção desse tipo de situação foi atenuado pelo experimento da seguinte forma: criado um hash do conteúdo da página e com isso mitigar possíveis colisões. Contudo, é importante mencionar que qualquer mudança, por menor que seja, é suficiente para modificar o *hash* e, consequentemente, o mesmo não ser julgado como um clone. O conteúdo da página é algo muito dinâmico, e geralmente o mesmo é modificado periodicamente. Além disso, existe os casos do conteúdo poder ser modificado a cada acesso, nos casos dos serviços gratuitos de hospedagens que inserem anúncios de propaganda de forma aleatória, o que faria modificar o *hash* resultante.

Por fim, conforme mencionado no Capítulo 5, o protótipo utilizou a API do *unshorten.me* para "desencurtar" URL encurtadas. Contudo, uma prática relativamente comum entre os fraudulentos é realizar encurtamento em profundidade, que seria encurtar uma URL já encurtada. Esses casos foram identificados, contudo, limitando-se até 3 investidas de encurtamento. Apesar de raro, pode acontecer casos com mais saltos, todavia, estes foram descartados.

6.3.1.3 Ameaças no processo de filtragem por lista de permissão

O processo baseado em lista de permissões também é suscetível a problemas de sequestro. Além do já mencionado sobre o sequestro de *HOST*, também é importante considerar que uma vez que a entrada seja registrada, a mesma será ignorada de todo o processo de checagem, pondo em cheque a precisão da proposta. Outro ponto é que apesar da presença de certificado trazer prestígio a página, é preciso considerar a possibilidade de algum desses certificados serem forjados ou mesmo clonados, trazendo um impacto significativo na heurística.

Outro fator que merece menção é a ausência de testes no serviço de lista de permissão na presente Tese de doutorado. A justificativa é que o grande desafio em si desse recurso da proposta seria a adesão dos proprietários de marcas no processo de alimentação da lista de permissão. Outro ponto é que um mal intencionado pode forjar diversos elementos da identidade de uma marca, como criar uma empresa fictícia, clonar um certificado digital ou mesmo realizar o sequestro de um HOST legítimo, tais situações não encontram-se no escopo da proposta. Contudo, trabalhos futuros para atenuar esses problemas são descritos no Capítulo 7, na Seção 7.4.

6.3.2 Ameaças no experimento com *phishing* válidos e inválidos

Essa seção descreve limitações especificas ao primeiro experimento. Quanto à **limitação de escopo**, no estado atual, foram considerados apenas os *phishing* de 2020 com o status "confirmado", contudo, foi suficiente para ter um considerável número de URL maliciosas para análise. A justificativa de considerar os *phishing* mais recentes é que a grande relevância do filtro se dar por sua rápida resposta sobre um novo *phishing* que surge, ou seja, o menor tempo da janela de vulnerabilidade.

Outro fato que reforça a preocupação dessa dependência é que existem casos de *phishing* publicados no repositório que ainda não passaram pelo crivo da administração da plataforma, representando uma janela de vulnerabilidade no repositório. Por exemplo, o *PhishTank*, por ser uma plataforma colaborativa onde qualquer pessoa denuncia de forma irrestrita, o *phishing* uma vez cadastrado, é classificado como "não confirmado", pois será submetido à uma análise manual por parte dos administradores do *PhishTank* para então ser uma ameaça "confirmada". Esse fato constata um atraso significativo da plataforma sobre um *phishing* que representa um perigo real ao usuário final, evidenciando o problema de falsos negativos. Contudo, a análise manual é um preço que se paga para evitar falsos positivos.

6.3.3 Ameaças no experimento comparativo entre as soluções nativas

Essa seção descreve limitações especificas ao segundo experimento. É notória a dificuldade em responder à *phishing* recém-criados por mecanismos baseados em lista de bloqueio, justificando assim a inferioridade entre as opções existentes e a solução proposta. Além disso, há um atraso na sincronização da lista de bloqueio do navegador com os registros do respectivo repositório, representando assim uma janela de vulnerabilidade através do uso do navegador. Isso também confirma que o experimento ao ser rodado em um momento posterior pode não ter os mesmos resultados descritos no estado atual.

Outro ponto, é que os testes foram baseados em navegadores para *desktop*, portanto, não é possível considerar que os resultados seriam os mesmos em ambientes de dispositivos móveis. De toda forma, talvez fosse interessante comparar entre as versões de cada plataforma. Todavia, existem certos aspectos da computação móvel, como repositório de aplicativos, que merecem análises voltadas para esse cerne.

6.4 CONSIDERAÇÕES FINAIS

Diante o exposto, esse capítulo teve o intuito de apresentar uma avaliação que evidencia resultados preliminares da proposta. O intuito foi apresentar uma metodologia formal, com base nos princípios da engenharia de software experimental, um experimento controlado que visa analisar o resultado do protótipo com base nos objetivos específicos. O Capítulo 7 apresentará as conclusões e considerações finais da proposta da Tese em seu estado atual, bem como trabalhos futuros planejados com a conclusão dessa Tese.

7 CONCLUSÃO

A presente Tese propôs uma solução que visa minimizar incidentes de fraudes durante a navegação do usuário final através da Web. Conforme mencionado no Capítulo 2, *phishing direcionados* são fraudes de escopo fechado, a exemplo do *spear phishing* e *SMiShing*. Devido sua natureza morfológica, com riqueza em detalhes, esses ataques sugerem uma abordagem preditiva mais direcionada ao contexto do ataque. Diante disso, os resultados dessa pesquisa visam oferecer modelos de predição mais eficientes em relação à **proteção da marca** de uma organização. Uma solução dessa natureza visa monitorar aspectos sobre a **identidade textual** em domínios, subdomínios e uso de palavras-chave em motores de busca. Além disso, também visa proteger a **identidade visual**, ou seja, o abuso de elementos que representam visualmente a organização, como *templates* e logomarcas. Nesse contexto, a pesquisa apresentou **características intrínsecas à marca alvo**, oferecendo apoio na **capacidade de resposta** e **tempo de resposta** do modelo preditivo proposto.

A metodologia da proposta, apresentada no Capítulo 4, define a solução como um sistema especialista para proteger o usuário durante sua navegação, considerando as diversas ameaças nos cenários de *internet banking*, *e-commerce* e *redes sociais*, âmbitos que trafegam dados sensíveis e constantemente são explorados por mal intencionados. A proposta foi fundamentada por resultados obtidos em estudos empíricos, conforme descritos no Capítulo 3. A proposta também apresentou no Capítulo 5 a elaboração de um protótipo com base nos fundamentos descritos, possibilitando nortear o desenvolvimento desse tipo de solução.

Apesar das lacunas e ameças existentes, conforme expostas no Capítulo 4, a proposta, na ótica da pesquisa, ofereceu um nível satisfatório de combate a fraudes visando a proteção ma marca conforme evidenciado na avaliação de um protótipo, descrita no Capítulo 6. Foram dois experimentos controlados, no primeiro, com 4039 *phishing* reais, a solução apresentou uma resposta aos incidentes de 90.17% como taxa de acerto e 9.83% de erro, sendo que dos 9.83%, 8.45% foram considerados suspeitos e apenas 1.38% foram sugeridos erroneamente como confiáveis. Além disso, visando observar o desempenho em falsos positivos, a proposta foi submetida a uma amostra com 953 páginas legítimas (não maliciosas), e apresentou uma taxa de acerto de 61.49%, contudo, 29.91% foram consideras como suspeitas, ainda sim permitido o acesso as mesmas, portanto, a taxa de erro foi de fato 7.55%.

Já na segunda avaliação, em que confronta a solução com outras existentes, a proposta teve uma precisão de 92.45% e erro de 7.55%, em um cenário que as demais opções apresentaram resultados abaixo dos 75%. Adicionalmente, como a proposta também se apresenta como uma opção complementar as existentes, foi possível observar que a proposta minimizou consideravelmente as falhas nas soluções existentes, com cobertura acima de 90%.

Por ter extraído um considerável número de *phishing* reais, a análise realizada por essa pesquisa considera aspectos quantitativos, a exemplo dos gráficos expostos. Não obstante, por

considerar o conteúdo e contexto, bem como identificar relevância e similaridades, a pesquisa também oferece resultados qualitativos. Com esses dados, fica possível concluir que aspectos temporais, que consideram o ciclo de vida da página, bem como do domínio e respectivo certificado digital, na perspectiva dessa pesquisa, influenciaram na relevância das características comumente adotadas em heurísticas para sistemas especialistas.

Além disso, a proposta também desenvolveu protótipos que adotam estratégias distintas de técnicas de IA com o objetivo de avaliar qual melhor abordagem na aplicabilidade em questão. Foram apresentados dados que trazem fundamentos sobre a decisão da Tese em adotar a utilização da estratégia baseada em sistemas especialistas ao invés da aprendizagem de máquina. Portanto, além do cenário da segurança da informação, a discussão e reflexões presentes nessa Tese de doutorado trazem contribuições também no cerne da inteligência artificial, uma vez que apresenta experimentos empíricos que fazem uso das tecnologias e abordagens debatidas na literatura.

7.1 MEIOS DE APLICABILIDADE DA PROPOSTA

Deslumbramos como possibilidade de aplicação da proposta em cenários distintos, de acordo com a necessidade e propósito no contexto em questão. Contudo, qualquer que seja a representação, ela sempre irá depender de atualizações periódicas na heurística que constitui a árvore de regras, portanto, essa pré-requisito precisa ser considerado.

Uma das alternativas de emprego da proposta seria a mesma atuar como um *proxy*, sendo uma camada intermediária no nó entre o *gateway* e as estações de trabalho que compõem uma rede corporativa. Tal solução poderia rodar na própria estrutura da corporação ou mesmo ser provida como um serviço, a exemplo do modelo *Security as a Service*.

Uma alternativa seria a proposta atuar como um *software endpoint* instalado em uma estação de trabalho, caracterizando uma proposta no modelo *standalone*. Na mesma linha, outra representação seria um complemento do navegador Web, podendo atuar como um *plug-in* ou extensão (add-on) ou mesmo um *proxy* como serviço configurado no navegador Web.

7.2 LIMITAÇÕES DA PROPOSTA

Além das soluções descritas no Capítulo 6, que são resultantes do protótipo proposto, ainda existem limitações nas decisões arquiteturais do protótipo que merecem destaque.

As tecnologias utilizadas no processo de detecção da marca foram condicionadas ao foco exclusivamente direcionado no provimento de um mecanismo minimante executável para testar as contribuições do estudo. Consequentemente, por ter oferecido abstração em complexidades que poderiam fugir do escopo da proposta, como implementar um mecanismo para detectar padrões em imagens, a exemplo do serviço *Google Vision*, tal funcionalidade acabou gerando uma dependência com terceiros. Além disso, provavelmente a proposta teria um melhor de-

sempenho caso o mecanismo de detecção de marcas fosse implementado nativamente, sem a necessidade de requisição remota em cada processo de classificação.

Outra questão remete a análise gradual. É razoável sugerir que talvez a estratégia adotada pelo processamento em profundidade não seja o melhor caminho. Diante os resultados atuais, se faz interessante melhorar a proposta considerando as etapas do processo que apresentaram maior relevância na pontuação de cada veredito.

7.3 RESULTADOS OBTIDOS

Em seu estado atual, a Tese disponibiliza seus resultados obtidos através de uma plataforma disponível *online* intitulada *piracema.io*[1]. A principal funcionalidade é possibilitar que o usuário informe um link para que o mesmo seja julgado como confiável, suspeito ou malicioso. Também é possível que usuários contribuam enviando denúncias de páginas maliciosas. Além disso, pesquisadores podem obter dados estatísticos sobre ataques de *phishing* nos últimos 5 anos. Adicionalmente, a plataforma também disponibiliza uma base "snapshot" de *phishing*, através do endereço *raw.piracema.io*.

O propósito é manter um histórico de comportamentos no conteúdo de *phishing*, Devido a sua alta volatilidade, se faz interessante preservar seu código-fonte no decorrer dos meses para analisar características em sua morfologia, visando identificar comportamentos que possam contribuir em sua mitigação. Além disso, possibilita que outros estudos com estratégias similares possam realizar suas avaliações com uma base balanceada.

O projeto é *open source* e está disponível em repositórios do GitHub[2]. Além das ferramentas mencionadas, o estudo também resultou em artigos científicos aceitos e publicados em conferências e periódicos da área, conforme descrito na Tabela 10, bem como artigos submetidos que ainda estão em revisão por pares, conforme descrito na Tabela 11. Por fim, também houveram premiações e orientações em projetos de iniciação científica (IC), conforme descritos, respectivamente, nas Tabelas 12 e 13.

Conforme exposto no Capítulo 6, foi desenvolvido um protótipo responsável em simular o funcionamento de um navegador durante a interação entre o sistema especialista e a navegação do usuário. Contudo, visando maior viabilidade do uso, é pretendido distribuir, como um protótipo, uma **extensão para os navegadores**[3], porém, essa decisão resulta em alguns impactos. Em seu estado atual, a aplicação faz uma requisição remota, enviando a URL e recebendo como resposta um veredito do lado do servidor. Pela natureza de uma aplicação do tipo extensão (que tem seu conteúdo interpretado e não compilado), não teve como encapsular o classificador no lado do cliente. O cenário ideal seria o modelo especialista sendo processado de forma embarcada no navegador, oferecendo maior desempenho no processo.

[1] https://piracema.io
[2] Links disponíveis no Apêndice B, Tabela 39
[3] Extensão para Chrome e Firefox, disponível no Apêndice B, Tabela 39

Tabela 10 – Artigos aceitos em periódicos e anais de eventos

Tipo	**Título**	**Veículo**	**Formato**	**Trilha**	**QUALIS**
Periódico	Heuristic-based strategy for Phishing prediction: A survey of URL-based approach (SILVA; FEITOSA; GARCIA, 2019a)	Computers & Security	Artigo completo	-	A1
Simpósio	Suscetibilidade através da forja de fidedignidade: uma abordagem sobre ataques de *Phishing* (SILVA; FEITOSA; GARCIA, 2019b)	XIX Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg 2019)	Artigo completo	Principal	B3
Simpósio	Uma Modelagem de Risco Centrada em Comportamentos para o Desenvolvimento Seguro de Serviços no Ecossistema Web (SILVA; GARCIA, 2019)	XIX Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg 2019)	Artigo completo	Principal	B3
Conferência	Adoção da seleção de características como mecanismo antiphishing: aplicabilidade e impactos (BARROS; SILVA; MIRANDA, 2019a)	XVI ENCONTRO NACIONAL DE INTELIGÊNCIA ARTIFICIAL E COMPUTACIONAL (ENIAC 2019)	Artigo completo	Principal	B4
Simpósio	Uma Avaliação das plataformas de denúncia de *Phishing* (TEIXEIRA; SILVA, 2019)	XIX Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg 2019)	Poster	WTICG	-
Simpósio	Aplicabilidade e Impactos quanto a Adoção de Modelos de Classificação como Mecanismos Anti-phishing (BARROS; SILVA; MIRANDA, 2019b)	XIX Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg 2019)	Poster	WTICG	-

Tabela 11 – Artigos submetidos (aguardando avaliação)

Tipo	**Título**	**Veículo**	**Formato**	**Status**	**QUALIS**
Periódico	Heuristic-based Strategy for Phishing Prediction: An Empirical Research of Context-Aware Approach	Expert Systems With Applications	Artigo completo	Under Review	A1

Tabela 12 – Premiações

Prêmio	**Trilha**	**Título**	**Veículo**
Best Paper	Best Paper Undergrad Track	Adoção da seleção de características como mecanismo antiphishing: aplicabilidade e impactos (BARROS; SILVA; MIRANDA, 2019a)	XVI ENCONTRO NACIONAL DE INTELIGÊNCIA ARTIFICIAL E COMPUTACIONAL (ENIAC 2019)

Tabela 13 – Orientações

Tipo	Projeto	Aluno	Instituição	Atuação
Iniciação Científica	Soluções anti-phishing baseadas em aprendizagem de máquina: uma revisão da literatura	Julio Cesar Gomes de Barros	UPE	Orientador
Iniciação Científica	Mecanismos Anti-Phishing em Navegadores Web: Uma Abordagem Experimental	Lucas Candeia Teixeira	UPE	Orientador
Iniciação Científica	Taxonomia para predição e reconhecimento de *phishing*: uma abordagem experimental	José Hilton Pereira dos Santos	UPE	Orientador
Iniciação Científica	Aplicabilidade e Impactos quanto a adoção de Modelos de Classificação como Mecanismos Anti-phishing	Mateus Lins e Silva Duque de Barros	UFRPE	Co-Orientador

7.4 TRABALHOS FUTUROS

Como trabalhos futuros, é pretendido explorar ao menos quatro lacunas detectadas no estado atual da proposta, a saber:

Em relação ao processo de manutenção (leitura e escrita) de entradas na lista de permissões, seria **(i) a adoção de redes permissionadas com *blockchain* e contratos inteligentes** (ZHONG et al., 2019b; ZHONG et al., 2019a). Nesse cenário, também é possível deslumbrar maior interação entre os proprietários de marca e o serviço de proteção, promovendo benefícios na detecção da marca e também na heurística da máquina de inferência, já que permitem que terceiros confiáveis possam contribuir com o serviço definindo regras arbitradas pelos mesmos.

Diante disso, é pretendido pela pesquisa propor um mecanismo para auxiliar na definição de contratos, que serve como abstração para as preocupações mencionadas. Tal mecanismo se apresenta como uma linguagem específica de domínio, do inglês, *Domain Specific Language* (DSL) (FOWLER, 2010), que pode ser através de texto ou elementos visuais, e responsável em facilitar o processo de composição das regras a serem consideradas. O fluxo de geração é ilustrado na Figura 50.

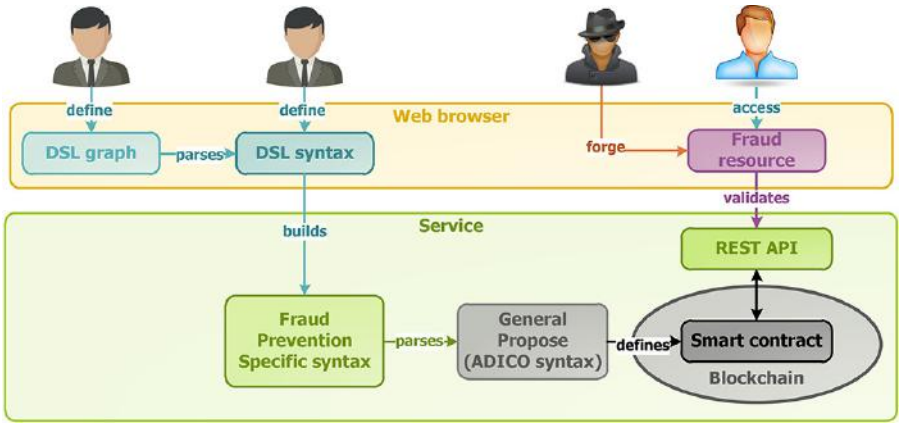

Figura 50 – Modelagem de domínio para prevenção de fraudes

Com base nesse contexto, cada contrato é criado pelo proprietário da marca, nesse contrato seriam especificadas as regras referente aos ativos da respectiva marca. Cada proprietário é responsável pelo seu contrato, bem como a liberação de parceiros para colaborarem com seus recursos. Um contrato inteligente na proposta estabelece as regras do ecossistema de recursos através de uma linguagem de programação, resultando em um algorítimo que dispara as regras pré-definidas durante transações que envolvam os recursos e marca em questão.

A proposta desse estudo futuro é apresentar uma interface intermediária que estende as funcionalidades da DSL ADICO (Frantz; Nowostawski, 2016). Ou seja, a proposta é disponibilizar uma DSL mais específica, voltada para a elaboração de contratos contra fraudes na Web, com recursos visuais que alimentam essa perspectiva. Uma vez que o proprietário conclua a composição das regras, o serviço irá converter as instruções dessa DSL específica para uma DSL com propósito mais generalizado, a exemplo de ADICO. Com isso, a geração do contrato inteligente será definida com base a especificação da DSL generalizada.

Portanto, é possível observar alguns benefícios com a adoção dos contratos inteligentes, já que permitem que terceiros confiáveis possam contribuir com o serviço definindo regras arbitradas pelos mesmos, oferecendo uma alternativa para atenuar falsos positivos. Contudo, assim como em qualquer tipo de recursos computacional, a elaboração de contratos inteligentes tem seus desafios, já que requer conhecimentos prévios, seja tecnológico ou de domínio, uma vez que o contrato inteligente trata-se de um algorítimo, que pode ser implementado em linguagens como *Solidity*, *JavaScript*, *Go*, entre outras (SINGH et al., 2019). Além disso, ameaças à segurança também são presentes durante o processo de elaboração dos contratos inteligentes. Em suma, existem desafios envolvidos que resultam em pesquisas futuras.

Dando continuidade as lacunas, como estudo futuro também é pretendido melhorias sobre as abordagens de IA envolvidas. Técnicas de *Deep Learning* são cogitadas para **(ii) uma análise por agregação de aspectos morfológicos**, já que eventuais mudanças nos padrões podem resultam em um aumento significativo de falsos negativos. Um exemplo disso são os problemas de mudanças no cenário, que podem variar devido a fidedignidade e sazonalidade. Além disso, essas mudanças também podem refletir na relevância das características candidatas, que podem se tornar mais ou menos relevantes com o passar do tempo. Nesse sentido, técnicas de agregação (*Ensembles*) de classificadores tornam-se opção para maior precisão e robustez nos resultados (OZA; TUMER, 2008).

Nesse contexto, outra pretensão seria uma melhoria em relação à busca por indexação. Essa atividade é fundamental em diversos aspectos da Tese, condicionando decisões importantes como a detecção de marcas e identificação de ataques homográficos. Apesar de certa eficiência da busca indexada em um grande volume de informações, ainda sim existe um considerável esforço em mitigar e elencar os termos e palavras-chave importantes para o processo (definição de *bag-of-words*). Outra questão é buscar maior simplicidade no sentido de detectar comportamentos através dos termos e o idioma utilizado para tal processo. Portanto, é pretendido pela Tese atenuar essas lacunas através de Processamento de Linguagem Natural (do

inglês, *Natural Language Processing*, NLP), combinado a técnicas de *Deep Learning* ou *Latent Dirichlet Allocation* (LDA), visando propor **(iii) um modelo simplificado de representação textual** para automatizar e otimizar o processo de compreensão dos termos de uma marca.

Por fim, também são planejadas melhorias na proteção de ataques direcionados a dispositivos. A presente proposta, em seu estado atual, foi sensível ao contexto de atuação, considerando a resolução de tela e o cabeçalho *User Agent* para detectar ataques de *SMiShing*. Contudo, é pretendimento refinar e expandir essa abordagem através de **(iv) uma detecção de dispositivo baseada em entropia**, a exemplo da investigação baseada em *Device fingerprinting*, contudo, mitigando os impactos com a privacidade.

REFERÊNCIAS

ABDALLAH, A.; MAAROF, M. A.; ZAINAL, A. Fraud detection system: A survey. *Journal of Network and Computer Applications*, v. 68, p. 90 – 113, 2016. ISSN 1084-8045. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1084804516300571>.

ABURROUS, M.; HOSSAIN, M.; THABTAH, F.; DAHAL, K. Intelligent phishing website detection system using fuzzy techniques. In: *Proceedings of 3rd International Conference on Information and Communication Technologies: From Theory to Applications*. [S.l.: s.n.], 2008. p. 1 – 6. ISBN 978-1-4244-1751-3.

ACFE. Report to the nations: 2018 global study on occupational fraud and abuse. *Disponível em: http://bit.ly/2MJ4zgm*, 2018.

AFROZ, S.; GREENSTADT, R. Phishzoo: Detecting phishing websites by looking at them. In: *ICSC*. IEEE Computer Society, 2011. p. 368–375. ISBN 978-1-4577-1648-5. Disponível em: <http://dblp.uni-trier.de/db/conf/semco/icsc2011.html#AfrozG11>.

ALEROUD, A.; ZHOU, L. Phishing environments, techniques, and countermeasures: A survey. *Computers & Security*, 2017.

ALKHOZAE, M. G.; BATARFI, O. A. Phishing websites detection based on phishing characteristics in the webpage source code. *International Journal of Information and Communication Technology Research*, 2011.

AMIRI, I. S.; AKANBI, O. A.; FAZELDEHKORDI, E. *A Machine-Learning Approach to Phishing Detection and Defense*. 1st. ed. [S.l.]: Syngress Publishing, 2014. ISBN 0128029277, 9780128029275.

AMOROSO, E. G. *Fundamentals of Computer Security Technology*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1994. ISBN 0-13-108929-3.

BABBIE, E. R. Survey research methods/ earl r. babbie. *SERBIULA (sistema Librum 2.0)*, 05 2019.

BARROS, M.; SILVA, C.; MIRANDA, P. Adoção da seleção de características como mecanismo antiphishing: aplicabilidade e impactos. In: *Anais do XVI Encontro Nacional de Inteligência Artificial e Computacional*. Porto Alegre, RS, Brasil: SBC, 2019. p. 214–225. ISSN 0000-0000. Disponível em: <https://sol.sbc.org.br/index.php/eniac/article/view/9285>.

BARROS, M.; SILVA, C.; MIRANDA, P. Aplicabilidade e impactos quanto a adoção de modelos de classificação como mecanismos anti-phishing. *XIX Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais*, 2019.

BASILI, V. R.; CALDIERA, G.; ROMBACH, H. D. The goal question metric approach. In: *Encyclopedia of Software Engineering*. [S.l.]: Wiley, 1994.

Basili, V. R.; Selby, R. W.; Hutchens, D. H. Experimentation in software engineering. *IEEE Transactions on Software Engineering*, SE-12, n. 7, p. 733–743, July 1986.

BISHOP, M. *How Attackers Break Programs, and How To Write Programs More Securely*. Technical Tutorial Session T1, University of California, Davis, August 24, 1999, 1999.

BISHOP, M.; BISHOP, M. *A Taxonomy of UNIX System and Network Vulnerabilities*. [S.l.], 1995.

CARTA, S.; FENU, G.; RECUPERO, D. R.; SAIA, R. Fraud detection for e-commerce transactions by employing a prudential multiple consensus model. *Journal of Information Security and Applications*, v. 46, p. 13 – 22, 2019. ISSN 2214-2126. Disponível em: <http://www.sciencedirect.com/science/article/pii/S2214212618304216>.

CERT.BR. Códigos maliciosos (malware). *Disponível em: https://cartilha.cert.br/malware/*, 2012.

CERT.BR. Golpes na internet. *Disponível em: https://cartilha.cert.br/golpes/*, 2012.

CHAUDHRY, J. A.; CHAUDRY, S. A.; RITTENHOUSE, R. G. Phishing attacks and defences. *International Journal of Security and its Application*, 2016.

CHELLIAH, G. A.; ARUNA, S. Preventing phishing attacks using anti-phishng prevention technique. *International Journal of Engineering Development and Research*, 2014.

COSTELLO, A. M. Punycode: A bootstring encoding of unicode for internationalized domain names in applications (idna). *Disponível em: https://tools.ietf.org/html/rfc3492*, 2003.

CUI, Z. C. Allowable limit of error in clinical chemistry quality control. *Clinical Chemistry*, v. 35, n. 4, p. 630–631, 04 1989. ISSN 0009-9147. Disponível em: <https://doi.org/10.1093/clinchem/35.4.630>.

Dunlop, M.; Groat, S.; Shelly, D. Goldphish: Using images for content-based phishing analysis. In: *2010 Fifth International Conference on Internet Monitoring and Protection*. [S.l.: s.n.], 2010. p. 123–128.

Elwell, R.; Polikar, R. Incremental learning of concept drift in nonstationary environments. *IEEE Transactions on Neural Networks*, v. 22, n. 10, p. 1517–1531, Oct 2011. ISSN 1045-9227.

FOWLER, M. *Domain Specific Languages*. 1st. ed. [S.l.]: Addison-Wesley Professional, 2010. ISBN 0321712943.

Frantz, C. K.; Nowostawski, M. From institutions to code: Towards automated generation of smart contracts. In: *2016 IEEE 1st International Workshops on Foundations and Applications of Self* Systems (FAS*W)*. [S.l.: s.n.], 2016. p. 210–215.

GOOGLE. Google safe browsing. *Available at: https://safebrowsing.google.com/*, 2019.

HILBE, J. M. *Practical guide to logistic regression*. [S.l.]: Taylor & Francis, 2016. ISBN 9781498709576,1498709575.

HO, G.; SHARMA, A.; JAVED, M.; PAXSON, V.; WAGNER, D. Detectimg credential spearphishing attacks in enterprise settings. *USENIX Security 2017*, 2017.

HOWARD, J. D. *An Analysis of Security Incidents on the Internet 1989-1995*. Tese (Doutorado) — Carnegie Mellon University, Pittsburgh, PA, USA, 1998. UMI Order No. GAX98-02539.

JAIN, A. K.; B., G. B. Phishing detection: Analysis of visual similarity based approaches. *Security and Communication Networks*, 2017.

JAZI, M.; MOBARAKEH, M.; LYASHENKO, V. Developing expert system in order to detect the journal phishing attacks. *Journal of Mathematics and Technology*, v. 6, p. 2078–257, 01 2015.

JOG, A.; CHANDAVALE, A. A. Implementation of credit card fraud detection system with concept drifts adaptation. In: BHALLA, S.; BHATEJA, V.; CHANDAVALE, A. A.; HIWALE, A. S.; SATAPATHY, S. C. (Ed.). *Intelligent Computing and Information and Communication*. Singapore: Springer Singapore, 2018. p. 467–477. ISBN 978-981-10-7245-1.

JOVANOVIC, G. Standardization of the old church slavonic cyrillic script and its registration in unicode. *Disponível em: http://www.unicode.org/charts/PDF/U0400.pdf*, 2009.

KASPERSKY. O que é phishing? *Disponível em: https://goo.gl/4EEtxk*, 2014.

KHONJI, M.; IRAQI, Y.; JONES, A. Phishing detection: A literature survey. *IEEE Communications Surveys and Tutorials*, v. 15, n. 4, p. 2091–2121, 2013.

KIRDA, E.; KRUGEL, C. Protecting users against phishing attacks. *The Computer Journal*, 2005.

KONDUTO. Konduto: Raio-x da fraude. terceira edição, 2019. *Disponível em: https://bit.ly/2mv2uvN*, 2019.

KRSUL, I. V. *Software Vulnerability Analysis*. Tese (Doutorado) — Purdue University, West Lafayette, IN, USA, 1998. AAI9900214.

LI, Y.; YANG, Z.; CHEN, X.; YUAN, H.; LIU, W. A stacking model using url and html features for phishing webpage detection. *Future Generation Computer Systems*, 2019.

LINDQVIST, U.; JONSSON, E. How to systematically classify computer security intrusions. In: *Security and Privacy, 1997. IEEE Symposium on*. [S.l.: s.n.], 1997. p. 154–163. ISSN 1081-6011.

LOUGH, D. L. *A Taxonomy of Computer Attacks with Applications to Wireless Networks*. Tese (Doutorado) — Virginia Polytechnic Institute and State University, 2001. AAI3006082.

LUMLEY, T. *Complex Surveys: A Guide to Analysis Using R*. Wiley, 2011. (Wiley Series in Survey Methodology). ISBN 9781118210932. Disponível em: <https://books.google.com.br/books?id=L96ludyhFBsC>.

MA, J.; SAUL, L. K.; SAVAGE, S.; VOELKER, G. M. Identifying suspicious urls: An application of large-scale online learning. In: *Proceedings of the 26th Annual International Conference on Machine Learning*. New York, NY, USA: ACM, 2009. (ICML '09), p. 681–688. ISBN 978-1-60558-516-1. Disponível em: <http://doi.acm.org/10.1145/1553374.1553462>.

MITNICK, K. D.; SIMON, W. L. *A Arte de Enganar: Ataques de Hackers: Controlando o Fator Humano na Segurança da Informação*. [S.l.]: Pearson, 2003. ISBN 8534615160.

MOHAMMAD, R. M.; THABTAH, F.; MCCLUSKEY, L. Tutorial and critical analysis of phishing websites methods. *Comput. Sci. Rev.*, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 17, n. C, p. 1–24, ago. 2015. ISSN 1574-0137.

MOLLERI, J. S.; PETERSEN, K.; MENDES, E. Survey guidelines in software engineering: An annotated review. In: *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. New York, NY, USA: ACM, 2016. (ESEM '16), p. 58:1–58:6. ISBN 978-1-4503-4427-2.

NACER, H.; DJEBARI, N.; SLIMANI, H.; AISSANI, D. A distributed authentication model for composite web services. *Computers & Security*, v. 70, p. 144 – 178, 2017. ISSN 0167-4048. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0167404817301153>.

NARESH, U.; SAGAR, U. V.; REDDY, C. V. M. Intelligent phishing website detection and prevention system by using link guard algorithm. *IOSR Journal of Computer Engineering*, 2013.

OPENDNS. Phishtank. *Available at: https://www.phishtank.com/*, 2019.

ORCHE, A. E.; BAHAJ, M. Approach to use ontology based on electronic payment system and machine learning to prevent fraud. In: *Proceedings of the 2Nd International Conference on Networking, Information Systems & Security*. New York, NY, USA: ACM, 2019. (NISS19), p. 37:1–37:6. ISBN 978-1-4503-6645-8. Disponível em: <http://doi.acm.org/10.1145/3320326.3320369>.

OZA, N. C.; TUMER, K. Classifier ensembles: Select real world applications. *Information Fusion*, v. 9, n. 1, p. 4–20, 2008.

RAILEANU, L. E.; STOFFEL, K. Theoretical comparison between the gini index and information gain criteria. *Annals of Mathematics and Artificial Intelligence*, Kluwer Academic Publishers, USA, v. 41, n. 1, p. 77–93, maio 2004. ISSN 1012-2443. Disponível em: <https://doi.org/10.1023/B:AMAI.0000018580.96245.c6>.

Rajeshwari U; Babu, B. S. Real-time credit card fraud detection using streaming analytics. In: *2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*. [S.l.: s.n.], 2016. p. 439–444.

ROBSON, C. *Real World Research - A Resource for Social Scientists and Practitioner-Researchers*. second. Malden: Blackwell Publishing, 2002.

SANTIAGO, G. P.; PEREIRA, A. C. M.; HIRATA JR., R. A modeling approach for credit card fraud detection in electronic payment services. In: *Proceedings of the 30th Annual ACM Symposium on Applied Computing*. New York, NY, USA: ACM, 2015. (SAC '15), p. 2328–2331. ISBN 978-1-4503-3196-8. Disponível em: <http://doi.acm.org/10.1145/2695664.2695990>.

SCIARRETTA, G.; CARBONE, R.; RANISE, S.; ARMANDO, A. Anatomy of the facebook solution for mobile single sign-on: Security assessment and improvements. *Computers & Security*, v. 71, p. 71 – 86, 2017. ISSN 0167-4048. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0167404817300895>.

SILVA, C. M. R.; FEITOSA, E. L.; GARCIA, V. C. Heuristic-based strategy for phishing prediction: A survey of urlbased approach. *Computers & Security*, 2019. ISSN 0167-4048.

SILVA, C. M. R.; FEITOSA, E. L.; GARCIA, V. C. Suscetibilidade através da forja de fidedignidade: uma abordagem sobre ataques de phishing. *XIX Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais*, 2019.

SILVA, C. M. R.; GARCIA, V. C. Uma modelagem de risco centrada em comportamentos para o desenvolvimento seguro de serviços no ecossistema web. *XIX Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais*, 2019.

SINGH, A.; PARIZI, R. M.; ZHANG, Q.; CHOO, K.-K. R.; DEHGHANTANHA, A. Blockchain smart contracts formalization: Approaches and challenges to address vulnerabilities. *Computers & Security*, p. 101654, 2019. ISSN 0167-4048. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0167404818310927>.

SINGH, R.; MANGAT, N. S. Stratified sampling. In: _____. *Elements of Survey Sampling*. Dordrecht: Springer Netherlands, 1996. p. 102–144. ISBN 978-94-017-1404-4. Disponível em: <https://doi.org/10.1007/978-94-017-1404-4_5>.

SRINIVASA, R.; ALWYN, R.; PAIS, R. Jail-phish: An improved search engine based phishing detection system. *Computers & Security*, 2019.

STOUT, B.; MCDOWELL, K. *United States Patent*. [S.l.], 2012. Disponível em: <https://patentimages.storage.googleapis.com/2c/d7/19/1b58c99bb246c4/US8285830.pdf>.

TEIXEIRA, L. C.; SILVA, C. M. R. Uma avaliação das plataformas de denúncia de phishing. *XIX Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais*, 2019.

VAYANSKY, I.; KUMAR, S. Phishing – challenges and solutions. *Computer Fraud & Security*, v. 2018, p. 15–20, 01 2018.

VERIZON. 2015 data breach investigations report. *Disponível em: https://vz.to/2ZAfr8H*, 2015.

WANAWE, K.; AWASARE, S.; PURI, N. V. An efficient approach to detecting phising a web using k-means and naïve- bayes algorithms. *International Journal of Research in Advent Technology*, 2014.

WHITTAKER, C.; RYNER, B.; NAZIF, M. Large-scale automatic classification of phishing pages. In: *NDSS '10*. [s.n.], 2010. Disponível em: <http://www.isoc.org/isoc/conferences/ndss/10/pdf/08.pdf>.

WIKILEAKS. How should npr report on hacked wikileaks emails? *Disponível em: https://www.npr.org/sections/publiceditor/2016/10/19/498444943/how-should-npr-report-on-hacked-wikileaks-emails*, 2017.

WOHLIN, C. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In: *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*. New York, NY, USA: Association for Computing Machinery, 2014. (EASE '14). ISBN 9781450324762. Disponível em: <https://doi.org/10.1145/2601248.2601268>.

WOHLIN, C.; RUNESON, P.; HöST, M.; OHLSSON, M. C.; REGNELL, B.; WESSLÉN, A. *Experimentation in Software Engineering: An Introduction*. Norwell, MA, USA: Kluwer Academic Publishers, 2000. ISBN 0-7923-8682-5.

ZDNET. Phishing-as-a-service is making it easier than ever for hackers to steal your data. *Disponível em: https://goo.gl/6UST6u*, 2016.

ZHONG, L.; WANG, H.; XIE, J.; QIN, B.; LIU, J. K.; WU, Q. A flexible instant payment system based on blockchain. In: JANG-JACCARD, J.; GUO, F. (Ed.). *Information Security and Privacy*. Cham: Springer International Publishing, 2019. p. 289–306. ISBN 978-3-030-21548-4.

ZHONG, L.; WU, Q.; XIE, J.; GUAN, Z.; QIN, B. A secure large-scale instant payment system based on blockchain. *Computers & Security*, v. 84, p. 349 – 364, 2019. ISSN 0167-4048. Disponível em: <http://www.sciencedirect.com/science/article/pii/S016740481831407X>.

APÊNDICE A – RESULTADOS QUANTITATIVOS DO ESTUDO EMPÍRICO

Esse Capítulo descreve os dados obtidos através dos métodos de extração, bem como a interpretação dos mesmos sobre uma análise de seus comportamentos.

A.0.1 Community-based Strategy

Essa categoria agrupa características direcionadas nas falhas das **estratégias adotadas pela plataforma de denúncias**, ou seja, possíveis explorações por parte dos atacantes em eventos relacionados ao controle *anti-phishing* que a plataforma oferece. Ela foi explicada e detalhada no Capítulo 20.

A.0.2 Life-cycle

Essa categoria destaca características direcionadas ao **ciclo de vida do *phishing***, considerando aspectos como o tempo de atividade, brevidade da existência e propagação através de clonagem. Os dados obtidos visam evidenciar o **tempo médio de atividade** do *phishing* em certos comportamentos.

A.0.2.1 C06. Activity time

Essa característica avalia o tempo de atividade de um *phishing*, no intuito de observar padrões temporais. Esse aspecto é um importante indicador para identificar casos específicos de *phishing* com uma vida útil maior que a média. Os dados extraídos seguem na Figura 51 e a análise GQM é descrita na Tabela 14.

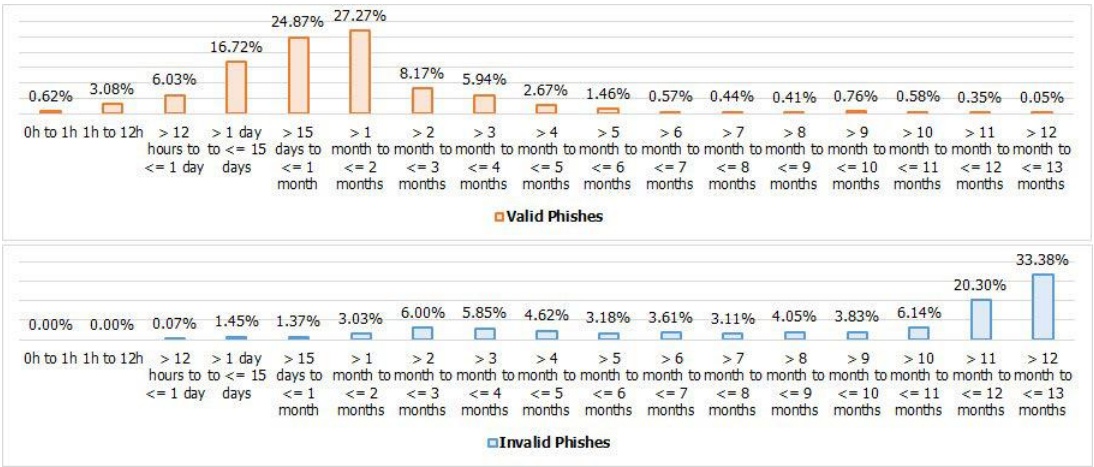

Figura 51 – C06 - ocorrências

Conforme o gráfico de colunas no canto superior da Figura 51, o eixo X discrimina os intervalos de tempo como uma escala, já o eixo Y representa as ocorrências em percentual

Tabela 14 – GQM de C06. Activity time

Objetivo 2	Analisar o ciclo de vida do *phishing* sobre o ponto de vista de sua atividade.				
Questão	Q06. Qual o tempo de vida médio de um *phishing*?				
Métricas	[M11] Tempo entre 01/01/2019 até a data de submissão do *phishing* válido.				
	[M12] Tempo entre 01/01/2019 até a data de submissão do *phishing* inválido.				
Hipóteses	*phishing* tem um tempo de atividade curto.				
Amostras	1 e 2	**Relevance**	STRONG	**Relations**	C07, C10, C14
Extração	Subtrair a quantidade de válidos e inválidos.				
Limitações	-				
Observações	A data de submissão não oferece precisão do tempo real de atividade, apenas registra o momento em que o *phishing* em questão foi denunciado.				
Análise	Quase 80% dos *phishing* tem um período menor que 2 meses de atividade. Estranho o fato de muitos *phishing* com uma idade considerável de atividade, 1.75% com tempo de atividade entre 9 e 13 meses, não serem banidos por seus respectivos servidores de hospedagem.				

considerando a totalidade da amostra. Com base nos dados extraídos, foi possível observar que 78.59% dos *phishing* válidos teve uma vida útil menor que 2 meses. Em contrapartida, 70.81% dos *phishing* inválidos possuem entre 7 e 13 meses de atividade, ou seja, evidenciando que o *phishing* apresenta ciclo de atividade bem inferior das páginas legítimas. Diante disso, a característica foi considerada com relevância *STRONG*.

A.0.2.2 C07. Age of Domain

Essa característica avalia a idade de um determinado domínio. Consequentemente, também possibilita identificar casos específicos de *phishing* registrados com a finalidade de realizar fraudes ou mesmo levantar suspeitas de domínios que possam ter sidos sequestrados. Os dados extraídos seguem na Figura 52 e a análise GQM na Tabela 15.

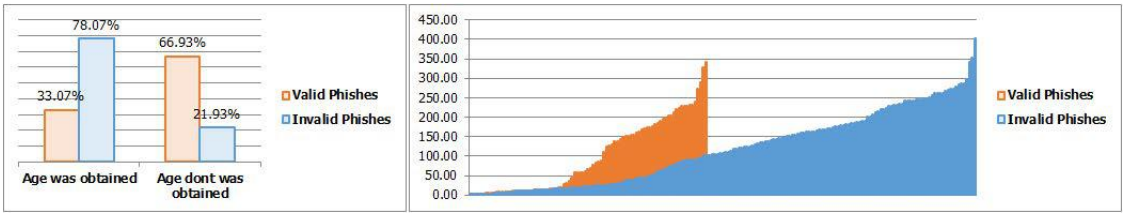

Figura 52 – C07 - ocorrências

Com base na Figura 52, o gráfico de colunas da esquerda descreve que 33.07% dos *phishing* válidos possuíam domínio próprio. Em contrapartida, 78.07% dos *phishing* inválidos possuíam domínio próprio, evidenciando que o inválido comumente possui um endereço com domínio registrado. Para a obtenção dos dados, foi utilizado o protocolo *WHOIS* para observar a idade

Tabela 15 – GQM de C07. Age of Domain

Objetivo 2	Analisar a idade de um domínio sobre o ponto de vista de sua atividade.				
Questão	Q07. Qual a idade média de um domínio registrado para a prática de *phishing*?				
Métricas	[M13] Calcular a idade de um registro utilizado em um *phishing* válido.				
	[M14] Calcular a idade de um registro utilizado em um *phishing* inválido.				
Hipóteses	Domínios para a prática de *phishing* tem uma idade jovem.				
Amostras	1 e 2	**Relevance**	STRONG	**Relations**	C09, C16, C26
Extração	Subtrair a quantidade de válidos e inválidos.				
Limitações	Foi considerado apenas 33.07% dos *phishing* válidos e 78.07% dos inválidos.				
Observações	Importante observar que muitos *phishing* operam em domínios fornecidos por serviços de terceiros, além disso, existem casos em que o domínio é sequestrado pelo atacante, tais comportamentos poderiam enviesar os resultados. Diante disso, foi necessário realizar a observação em amostras pequenas para uma análise subjetiva.				
Análise	A característica C07 pode ser considerada um indicador para levantar suspeitas sobre uma URL. Considerando a probabilidade, 66.93% dos *phishing* válidos não possuíam domínio registrado, sugerindo que a URL em questão é mais propensa a não ser perigosa quando a mesma possui um domínio registrado, contudo, a análise também indica que 1/3 dos mal intencionados investiram na sua fraude. Tais aspectos que reforçam a veracidade são descritos na categoria "User susceptibility".				

do registro do domínio, e o processo de automação foi realizado através da API Domage[1].

Ainda na mesma figura, o gráfico de colunas da direita, o eixo Y mede quantidade de meses de atividade de cada domínio, considerando apenas os registros com domínio registrado. Diante da imagem, foi possível observar que a idade do registro de *phishing* inválidos é bem superior em comparação aos *phishing* válidos. Diante disso, a característica foi considerada com relevância *STRONG*.

A.0.2.3 C08. Clonning strategy

Essa característica avalia a ocorrência de clones propagados na Web, conforme ilustrado na Figura 53. Esse aspecto é importante para ter um parâmetro sobre *phishing*, uma vez que a prática além da propagação, pode ter como objetivo também burlar os mecanismos de lista negra. A análise GQM é descrita na Tabela 16.

Conforme ilustrado na Figura 53, o gráfico de colunas descreve que quase 1/3 de todos os *phishing* da amostra #3 foram clonados, mais precisamente, 32,43% foram replicados. Na mesma figura, o gráfico de barras descreve os 10 *phishing* mais clonados na amostra e diferenciados por cores. O mais clonado teve um total de 525 ocorrências, representando 1.07% de todos os registros da amostra.

Ainda sobre os 10 registros, no gráfico de dispersão é possível observar a contínua ação dos mal intencionados em manter suas fraudes durante os meses de 2018. O eixo X descreve a dispersão das ocorrências durante os meses de 2018, já no eixo Y discrimina unidade de ocorrências no respectivo período. Foi possível observar que as 525 clonagens foram bem

[1] Domage API: https://ipty.de/domage/index.php

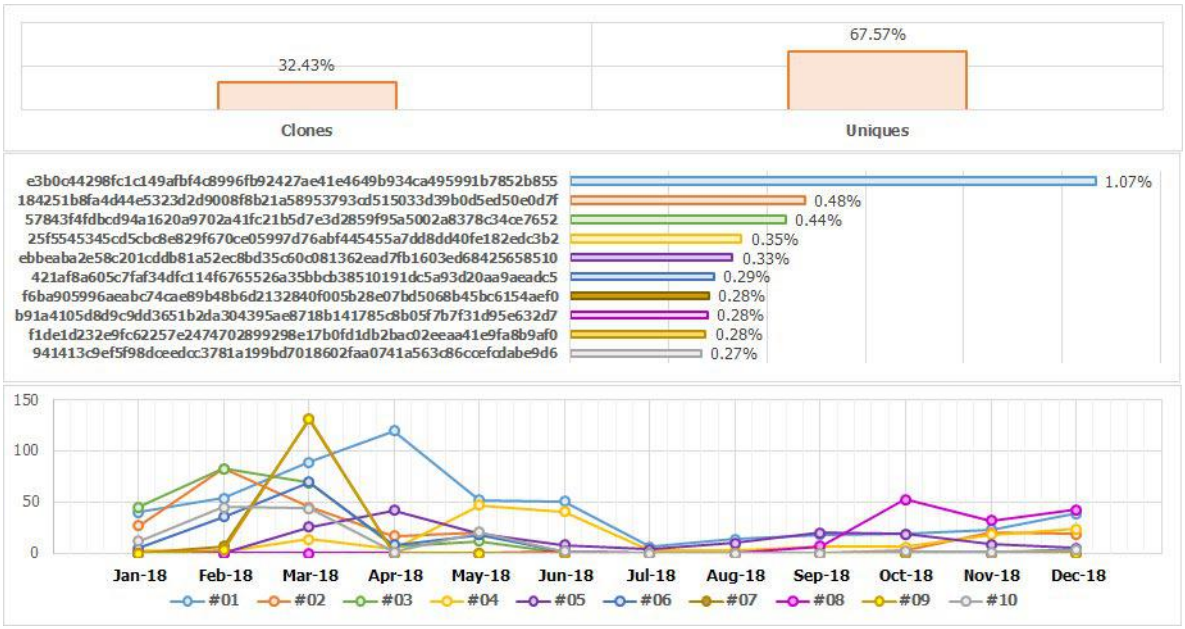

Figura 53 – C08 - ocorrências

Tabela 16 – GQM de C08. Cloning strategy

Objetivo 2	Analisar o ciclo de vida do *phishing* sobre o ponto de vista de sua atividade.				
Questão	Q08. Qual a quantidade de *phishing* clonados?				
Métricas	[M15] Quantidade de *phishing* clonados.				
	[M16] Os 10 *phishing* mais clonados.				
Hipóteses	Devido ao tempo curto e alta transição na atividade, o ataque adota clonagem para maior propagação.				
Amostras	3	**Relevance**	STRONG	**Relations**	C03
Extração	Obter o http response e transformar em hash. Após isso, realizar colisão entre hash para computar.				
Limitações	-				
Observações					
Análise	Com base na amostra, 32.42% dos *phishing* possuem um ou mais clones. Para maior precisão, o processo de análise foi baseado no *hash* do conteúdo da página com considerável resistência a colisão (SHA-256). Um *phishing* em específico foi clonado de tal maneira que sua quantidade de clones representa 1.07% de toda a amostra analisada. Aliada a essa característica, os resultados de C03 e C09 reforçam a volatilidade e brevidade do *phishing*. Uma vez que o *phishing* entra na lista, o fraudador além de criar outros também acaba por usar o mesmo, só que em servidores diferentes, fazendo com que a URL mude, tornando-se 0-day para os mecanismos de lista negra.				

distribuídas durante todo o ano, e seus maiores picos foram presentes no mês de março. Diante disso, a característica foi considerada com relevância *STRONG*.

Tabela 17 – GQM de C09. SEO score

Objetivo 2	Analisar o ciclo de vida do *phishing* sobre o ponto de vista de sua atividade.				
Questão	C09. Quais as pontuações de SEO dos *phishing* válidos e inválidos?				
Métricas	[M17]: Contabilizar o ranking de *phishing* válidos.				
Métricas	[M18]: Contabilizar o ranking de *phishing* inválidos.				
Hipóteses	Devido ao seu propósito, um *phishing* dificilmente terá investimentos sobre marketing. Diante disso, sites suspeitos são excluídos do índice do mecanismo de pesquisa, para que os usuários não sejam colocados em risco.				
Amostras	1.1 e 2.1	**Relevance**	MODERATE	**Relations**	C23, C26, C27
Extração	Informar a URL completa para a página do Alexa [2].				
Limitações	-				
Observações					
Análise	Foi observado que a grande maioria das URL maliciosas possuem um *rank* muito inferior em comparação as páginas legítimas, fazendo o Alexa ser um parâmetro interessante para levantar suspeitas em um determinado site.				

A.0.2.4 C09. SEO score

Essa característica avalia o score SEO de uma determinada página. Esse aspecto remete a reputação de um site durante sua atividade. Por exemplo, situações em que o site são construídos por *Content Management System* (CMS) compartilham vulnerabilidades em comum que podem ser explorados por atacantes, nesse contexto, o fato de utilizar ou não um determinado CMS já pode ser um critério para reduzir o SEO do site. Em suma, um site que já tenha sido violado ou com chances reais para tal, já perde score de SEO. O fato é que geralmente um *phishing* tem um objetivo bem definido, de curto prazo e motivações bem direcionadas, ou seja, dificilmente será um ambiente em que seu proprietário invista em marketing digital, representando assim algo que pode sugerir suspeitas em um determinado site. Os dados estão ilustrados na Figura 54 e a análise GQM na Tabela 17.

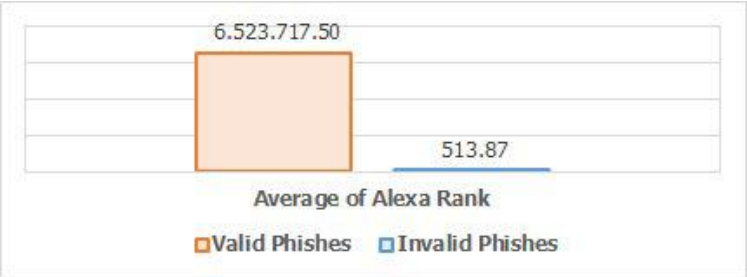

Figura 54 – C09 - ocorrências

Importante salientar que a característica *Google page rank* por ter sido descontinuada, acabou tornando-se obsoleta e não foi tratado pelo estudo, maiores detalhes na Seção 3.5. Portanto, para analisar o SEO score da página foi considerado o Alexa. Conforme descrito no gráfico de colunas da Figura 54, o eixo Y descreve a média dos scores dos *phishing* válidos

e inválidos. Diante disso, foi possível observar que o montante de válidos foi bem maior em comparação ao montante de inválidos, no caso dos válidos a média global foi 6 vezes maior. No Alexa, uma página com certa reputação e investimento em SEO tende a ter um número baixo, por exemplo, o site www.google.com tem score 1, já um site desconhecido e sem estratégias SEO tende a ter números altos, por exemplo, o *phishing* válido https://id-apple-account.usa.cc/ tem um score de 2,366,789.

Diante disso, a média global destaca o contraste entre as páginas válidas e inválidas. Contudo, nem todo site era possível extrair o rank do Alexa, muitos retornavam o erro *We don't have enough data to rank this website.*, diante disso, foi possível evidenciar que o Alexa não tem uma cobertura ampla diante do cenário proposto, fazendo com que a característica tenha uma relevância *MODERATE*. Outro fator é que o tamanho da URL também influência no score, tamanhos menores que 50-60 caracteres influenciam positivamente, já valores acima de 100 influenciam negativamente, fazendo assim relação com a característica C23. Na mesma linha, domínio com reputação e sequestrado (C26), fará a fraude ter SEO alto, e futuramente, o domínio perderá SEO pelo ocorrido. Além disso, o tunelamento também impulsiona o SEO, portanto tem relação com C27.

A.0.2.5 C10. Volatility

Essa característica avalia a probabilidade de mudanças do status de um *phishing*, ou seja, a transição de *online* para *offline*. Para a leitura desses dados, foi criado um gráfico anual e mensal, conforme ilustrados na Figura 55 e a análise GQM é descrita na Tabela 18.

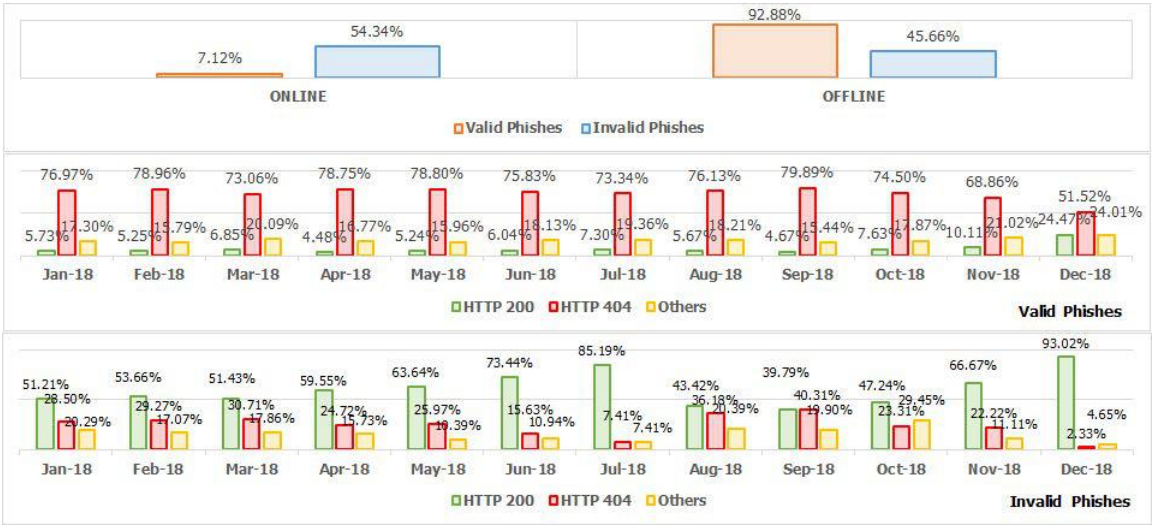

Figura 55 – C10 - ocorrências

Conforme a Figura 55, o primeiro gráfico de colunas descreve o total de incidências de *online* e *offline*, entre os *phishing* válidos e inválidos de 2018. O eixo X agrupa os registros com status *online* e *offline*, já o eixo Y descreve as ocorrências em percentual considerando a totalidade da amostra. Diante disso, ficou notório que os casos de *phishing* válidos, sua

Tabela 18 – GQM de C10. Volatility

Objetivo 2	Analisar o ciclo de vida do *phishing* sobre o ponto de vista de sua atividade.				
Questão	Q10. Qual a variação na frequência de *online* e *offline* durante os meses?				
Métricas	[M19] Diferença entre *phishing* válido *online* e *offline*.				
	[M20] Diferença entre *phishing* inválido *online* e *offline*.				
Hipóteses	Por ter um tempo curto de atividade, o ataque apresenta um grande volume na transição de *online* para *offline*.				
Amostras	1 e 2	**Relevance**	STRONG	**Relations**	C06
Extração	Contabilizar o número de *phishing* com status 200, 404 e outros.				
Limitações	O aspecto temporal é bastante comprometido, *phishing* mais antigos tendem a ter um maior número de *offline*, da mesma forma, *phishing* mais recentes são mais suscetíveis a estarem *online*.				
Observações					
Análise	O criador da fraude usa a estratégia de criar o *phishing* e fazer uso do mesmo em um curto espaço de tempo, visando atingir o maior número de vítimas antes que o mesmo caia na lista negra. A variação de *phishing online* e *offline* durante os meses é pequena. No caso o mês de dezembro não representou bem a variância padrão devido as limitações descritas.				

volatilidade é muito maior em comparação aos inválidos. De todos os *phishing* válidos de 2018, apenas 7.12% terminaram o ano permanecendo *online*, e em sua maioria pertencentes ao últimos meses. Já os *phishing* inválidos terminaram o ano com 54.34% *online*.

Na mesma figura, o segundo e terceiro gráfico de colunas distribuem os registros, respectivamente, entre válidos e inválidos, durante os meses de 2018 considerando o código HTTP 200, 404 ou "Others". No eixo X agrupa os registros por meses durante o respectivo ano, já o eixo Y apresenta a totalidade em percentual considerando a totalidade da amostra. Com base nos dados, foi possível observar um padrão na transição entre *online* e *offline* dos *phishing* válidos durante os meses, com mediana de 75.98%.

Importante frisar que os meses de novembro e dezembro apresentam dados com um certo *outlier*, 68.86% e 51.52% respectivamente, devido o atraso das votações conforme descrito na Seção 3.2.2. Como consequência, os primeiros meses de 2019 quantificam registros para os últimos meses de 2018, situação observada e minimizada com o **intervalo de coleta**, conforme a Seção 3.2.2. Diante disso, a característica foi considerada como *STRONG*.

A.0.3 Target profile

Essa categoria segmenta características direcionadas as **tendências dos incidentes**. A posse desses dados visa evidenciar serviços, idiomas, contexto e demais aspectos mais **visados** pelos mal intencionados.

Tabela 19 – GQM de C11. Content page most exploit

Objetivo 3	Analisar as tendências nas ocorrências de *phishing*.				
Questão	Q11. Qual o tipo de conteúdo mais explorado pelo ataque?				
Métricas	[M21] Contabilizar os tipos de *phishing* válidos por categoria.				
	[M22] Contabilizar os tipos de *phishing* inválidos por categoria.				
Hipóteses	Existe uma tendência referente ao tipo de conteúdo a ser abordado, sendo o mesmo mais suscetível a ataques.				
Amostras	1.1 e 2.1	**Relevance**	MODERATE	**Relations**	C14, C15
Extração	Obter o http response e analisar o tipo de conteúdo em questão com base em categorização.				
Limitações	-				
Observações	O tipo de conteúdo foi agrupado em 7 categorias, com inspiração do relatório da PWG.				
Análise	Em relação aos *phishing* válidos, é notório que as grandes ocorrências dos ataques foram direcionadas para o segmento de internet banking e transações financeiras. Já os falsos relatos remetem ao âmbito das redes sociais.				

A.0.3.1 C11. Content page most exploit

Essa característica avalia o tipo de conteúdo mais visado pelos mal intencionados. Para a leitura desses dados, foram definidas **7 categorias de conteúdo**, e os dados segmentados por mês e respectiva categoria, conforme ilustrados na Figura 56 e o GQM na Tabela 19.

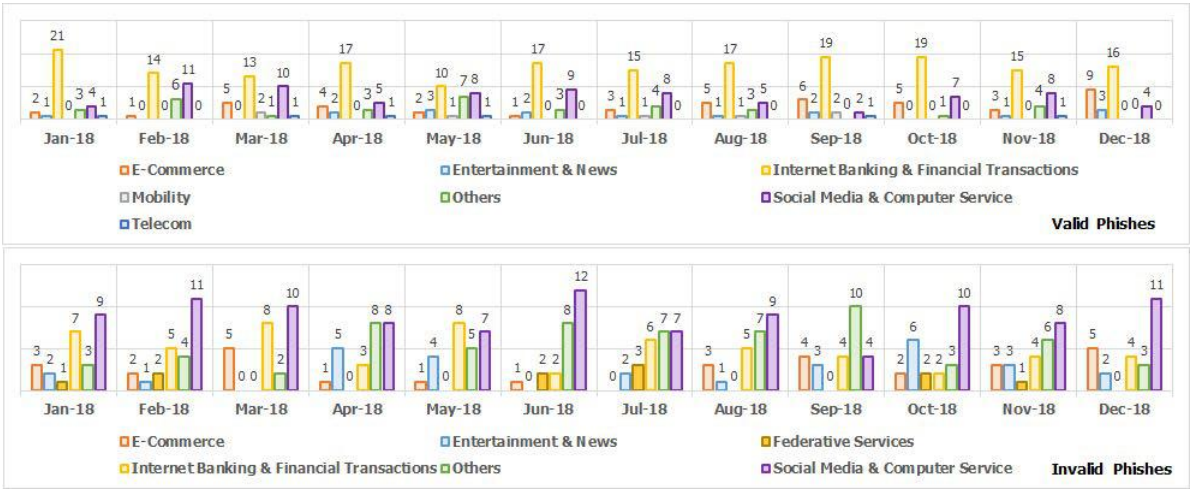

Figura 56 – C11 - ocorrências

Conforme a Figura 56, os gráficos de colunas descrevem o total de registros entre válidos e inválidos segmentado por 7 categorias definidas e que tiveram como base o campo *target* do **PhishTank Info**, conforme anteriormente descrito na Seção 3.3. O eixo X agrupa os dados considerando os meses observados em 2018, já o eixo Y descreve as ocorrências para cada categoria no mês. As categorias são: *E-commerce, Mobility, Telecom, Entertainment & News, Others, Internet Banking & Financial Transactions e Social Media & Computer Services*.

Alguns segmentos tinham muitas ocorrências combinadas, portanto esses foram agrupados em uma única categoria, a exemplo de Social Media & Computer Services. A categoria "Others" representa o agrupamento de página que possuíam um conteúdo muito miscigenado ou mesmo de um segmento com uma ou duas ocorrências isoladas, sendo mais favorável serem representados em uma única categoria.

Diante disso, ficou evidente que *Internet Banking & Financial Transactions* é o segmento mais explorados pelos fraudadores, com uma média de 16.08% de ocorrências a cada mês, o que representa 4.19% de todos os registros da amostra a cada mês. O desvio padrão de ataques nessa categoria durante os meses foi de 2.97%, ou seja, um padrão bem consolidado no decorrer do ano. Diante disso, a característica foi considerada com relevância *MODERATE*.

A.0.3.2 C12. Host most exploit

Essa característica identifica o serviço de hospedagem mais explorado. Diversos aspectos podem tornar um determinado serviço mais ou menos utilizados pelos mal intencionados, seja a localização, planos de hospedagem e a disponibilidade de armazenamento. Os dados extraídos estão ilustrados na Figura 57 e a análise GQM é descrita na Tabela 20.

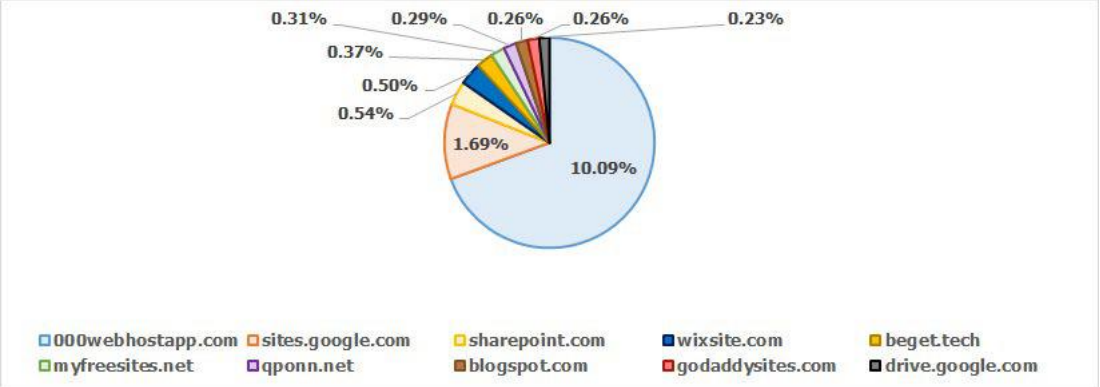

Figura 57 – C12 - ocorrências

Conforme a Figura 57, o serviço oferecido pelo *000webhospapp.com* representou 10.09% de todos os registros da amostra #3, ou seja, é evidente que tal serviço acaba por ser o mais visado para realizar fraudes na Web. Um dos possíveis motivos são os benefícios e praticidades oferecidas, como um generoso armazenamento de 1GB gratuito, um plano de baixo custo de assinatura, fazendo com que o mesmo não tenha *add-ons* e banda dedicada, além de poucos critérios no ato da realização do cadastro. O segundo serviço mais visado foi o do *Google*, que tem o diferencial da praticidade com seu *template* previamente definido, facilitando o processo de criação das páginas. Diante disso, a característica foi considerada com relevância *STRONG*.

A.0.3.3 C13. Language most exploit

Essa característica avalia o idioma mais explorado pelos atacantes. Diversos aspectos precisam ser considerados na definição do idioma, por exemplo, os casos de um mesmo idioma para

Tabela 20 – GQM de C12. Host most exploit

Objetivo 3	Analisar as tendências nas ocorrências de *phishing*.				
Questão	Q12. Qual o serviço de hospedagem mais explorado pelo ataque?				
Métricas	[M23] Contabilizar os serviços de armazenamento mais explorados.				
Hipóteses	Existe uma tendência referente ao serviço de hospedagem, das quais são intrínsecas as políticas de uso, tornando o mesmo mais suscetível a ataques.				
Amostras	3	**Relevance**	STRONG	**Relations**	C15
Extração	Obter a URL e contabilizar a quantidade por domínio, verificar se o mesmo se trata de um serviço de hospedagem.				
Limitações	Certas URL possuíam domínio registrado, nesses casos para saber a hospedagem em questão só foi possível através do *WHOIS*, contudo, nem todas as URL oferecem suporte para obtenção de detalhes através do *WHOIS*. Por exemplo, a API só obtém o *WHOIS* para domínios .com, .net e .edu.				
Observações	Foi utilizada uma API para capturar o resultado do *WHOIS*.				
Análise	10% de todos os *phishing* da amostra estavam hospedados no serviço 000webhostapp. Foi observado que o respectivo serviço tem poucos critérios quanto ao cadastro, baixo custo (até mesmo com plano gratuito de 1GB de armazenamento). Esses aspectos fazem com que o serviço seja muito atraente para os mal-intencionados.				

diversos países distintos ou mesmo páginas com mais de um idioma. Diante disso, foi preciso extrair os dados de forma manual, considerando o idioma do conteúdo predominante e observando alguns elementos HTML que especificam o idioma. Esse nível de refinamento também possibilitou considerar que essa característica, além do idioma, consequentemente representa o país que o ataque é direcionado. Os dados extraídos estão ilustrados na Figura 58 e a análise GQM é descrita na Tabela 21.

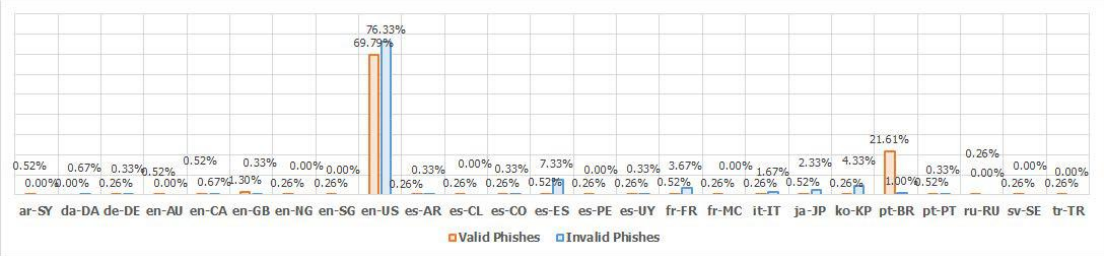

Figura 58 – C13 - ocorrências

Conforme a Figura 58, o gráfico de coluna apresenta os incidentes observados. No eixo X descreve o idioma e no Y o total de ocorrências, em valor percentual com base na totalidade da amostra. Diante disso, fica evidente que o idioma inglês americano é o mais explorado pelos mal intencionados. Em contrapartida, por ser um idioma universal, não pode ser considerado que o Estados Unidos da América é o país mais explorado.

Contudo, o que chama atenção é o resultado dos incidentes direcionados para o Brasil, o segundo colocado, com 21.61%, totalizando 16 vezes o valor do inglês britânico, o terceiro mais explorado. Além isso, a ocorrência de páginas brasileiras nos casos de inválidos é muito baixa em comparação aos válidos. Por anos, o Brasil é um país que se destaca em relação

Tabela 21 – GQM de C13. Language most exploit

Objetivo 3	Analisar as tendências nas ocorrências de *phishing*.				
Questão	Q13. Qual o idioma mais explorado pelo ataque?				
Métricas	[M24] Contabilizar o idioma mais utilizado nos *phishing* válidos, considerando o país de origem.				
	[M25] Contabilizar o idioma mais utilizado nos *phishing* inválidos, considerando o país de origem.				
Hipóteses	Existe uma tendência referente ao idioma, sendo o mesmo mais suscetível a ataques.				
Amostras	1.1 e 2.1	**Relevance**	WEAK	**Relations**	C30
Extração	Obter o http response e analisar o idioma predominante.				
Limitações	Certas URL possuíam domínio registrado, nesses casos para saber a hospedagem em questão só foi possível através do *WHOIS*, contudo, nem todas as URL oferecem suporte para obtenção de detalhes através do *WHOIS*. Por exemplo, a API só obtém o *WHOIS* para domínios .com, .net e .edu.				
Observações					
Análise	A grande maioria dos ataques analisados faziam uso do inglês americano. Destaque mesmo para o idioma pt-BR, presente significativamente em relação aos demais idiomas.				

aos incidentes de *phishing* em uma escala global. Diante disso, a característica foi considerada com relevância *WEAK*.

A.0.3.4 C14. Seasonality most exploit

Essa característica avalia os períodos mais propensos de ocorrência de *phishing*. Diante disso, foi preciso categorizar eventos fixos do calendário, como *natal* e *blackfriday*. Considerando possam existir eventos atípicos, a exemplo dos saques de FGTS realizados no Brasil, como medida de confiabilidade, foi desenvolvido gráficos distintos como medida de observar eventos fixos e atípicos durantes os anos. Na Figura 59, os gráficos apresenta os dados segmentados por meses durante o ano de 2018 no intuito de observar os eventos fixos. Já na Figura 60, os gráficos propostos consideram os últimos 5 e 10 anos visando observar eventos atípicos. Adicionalmente, a análise GQM é descrita na Tabela 22.

Importante frisar que alguns eventos considerados fixos podem não ter a mesma data em certos países, ou mesmo não existirem, o critério para considerar o evento como "fixo" foi considerar a grande maioria dos países do mundo. Já os eventos "atípicos" só puderam ser observados quando representavam algo bastante significativo, ou seja, uma vez comparado os meses entre os últimos 5 anos, havendo, consideravelmente, casos de incidentes com maior ou menor quantidade, esses foram analisados.

Conforme ilustrado na Figura 59, o gráfico de colunas descreve as ocorrências de *phishing* durante os meses de 2018. No eixo X apresenta o agrupamento dos dados considerando os meses de 2018, já no eixo Y o percentual de ocorrências considerando a totalidade da amostra. Diante disso, foi possível observar um padrão na quantidade de submissões de *phishing* válidos,

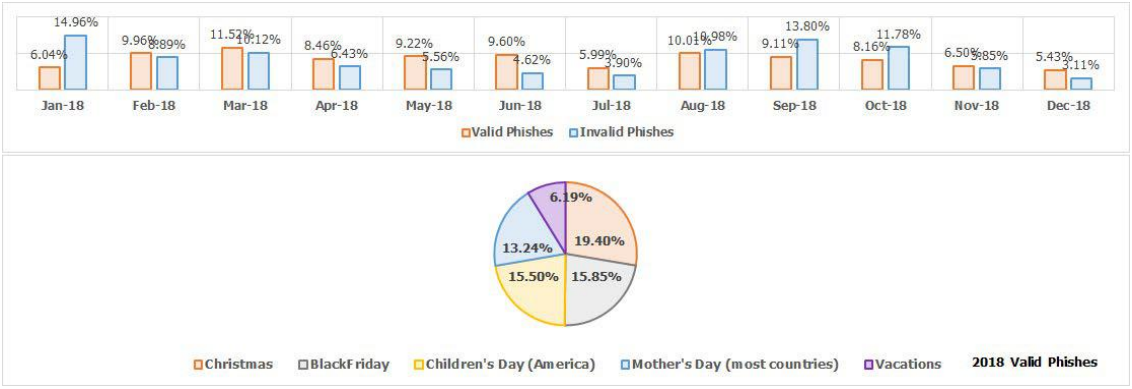

Figura 59 – C14 - ocorrências no ano de 2018

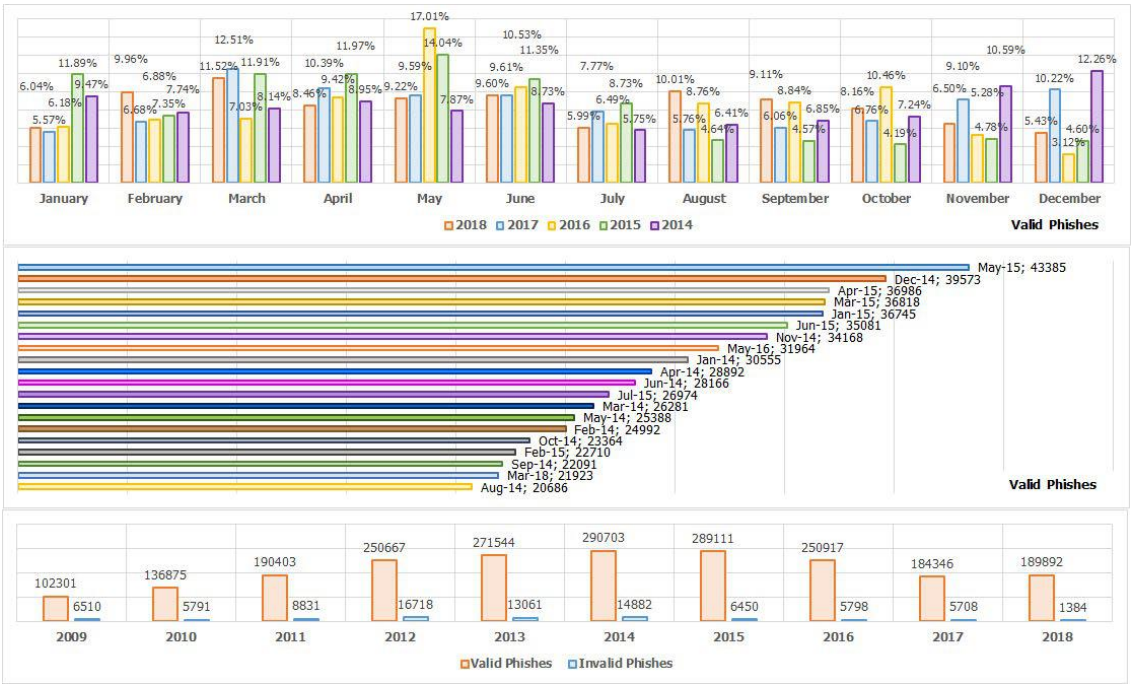

Figura 60 – C14 - ocorrências durante os últimos 10 anos

com uma média mensal de 8.33% registros, com base em toda a amostra #1, e desvio padrão de 0.02. Sobre os eventos fixos, o gráfico pizza ilustra que a maior parte dos incidentes de *phishing* válidos em 2018 ocorreram nos eventos de natalino e *blackfriday*, períodos que apresentam aquecimento no mercado global.

Na Figura 60, o primeiro gráfico de colunas descreve as ocorrências de *phishing* válidos nos últimos 5 anos. O eixo X apresenta o mês correspondente para os 5 anos observados, já o eixo Y o percentual de ocorrências. Foi possível observar casos atípicos que elevaram a quantidade de ataques em determinados meses, resultando em uma assimetria entre as colunas de um mesmo mês. Por exemplo, nos últimos 5 anos, no Brasil ocorreram algumas liberações para saques de FGTS, a exemplo de 2014 e 2015, devido a calamidade em decorrência de enchentes[3,4], por

[3] https://glo.bo/2UunEVt
[4] https://bit.ly/2DAf8yX

Tabela 22 – GQM de C14. Seasonality most exploit

Objetivo 3	Analisar as tendências nas ocorrências de *phishing*.				
Questão	Q14. Qual o período mais explorado pelo ataque?				
Métricas	[M26] Contabilizar os ataques distribuídos por mês dos *phishing* válidos.				
	[M27] Contabilizar os ataques distribuídos por mês dos *phishing* inválidos.				
Hipóteses	Existe uma sazonalidade nos ataques devido a eventos determinados ou não no calendário.				
Amostras	1 e 2	**Relevance**	STRONG	**Relations**	C11
Extração	Contabilizar os períodos mais explorados durante o ano.				
Limitações	-				
Observações	Foram considerados eventos que são programados para um determinado periodo do ano, a exemplo de natal, dia das mães e blackfriday.				
Análise	Eventos como black Friday, dia das mães, natal, ano novo e férias são mais atraentes para os atacantes. Além disso, certos eventos específicos foram identificados devido a relevância nos dados obtidos, como o caso de saques do FGTS no Brasil.				

se tratar de um evento regional, a propagação não foi significativa em termos de incidentes globais.

Contudo, ainda em 2015, houve um decreto que permitiu o saque de FGTS em contas inativas, fazendo com que no mesmo ano houvesse um considerável aumento dos incidentes. Na mesma linha, o evento se repetiu em dezembro de 2016, ou seja, fazendo com que o ano de 2017 tivesse um significativo aumento, padrão do qual pode ser observado durante o primeiro semestre do mesmo ano. Importante salientar que, apesar do mês de maio de 2016 apresentar maior pico de incidência no gráfico, com 17.01%, é preciso entender que esse valor é proporcional a quantidade do respectivo ano, ou seja, os 14.04% do mesmo mês no ano de 2015, em termos quantitativos, é muito maior.

Já no gráfico de barras da Figura 60 descreve os 20 meses mais explorados nos últimos 10 anos, evidenciando o grande número de ataques nos anos de 2014 e 2015. O evento atípico do FGTS contribuiu para que as ocorrências de ataques em 2017 superassem o ano de 2018. Nesse mesmo gráfico, outro fato que chama atenção é a quantidade de *phishing* inválidos no ano de 2018 em comparação aos demais anos, um valor bem inferior, o que poderia enviesar os resultados entre válido e inválidos. Contudo, o estudo se preocupou em realizar as comparações proporcionalmente para minimizar esse problema.

Por fim, conforme ilustrado no último gráfico de colunas da Figura 60, o eixo X segmenta os dados por ano durante os últimos 10 anos, já o eixo Y apresenta o total de ocorrências para o respectivo ano. Diante disso, foi possível observar a projeção das denúncias na plataforma *PhishTank* nos últimos 10 anos, ficando evidente que o ano de 2014 teve o maior número de ataques. Diante os dados, a característica foi considerada com relevância *STRONG*.

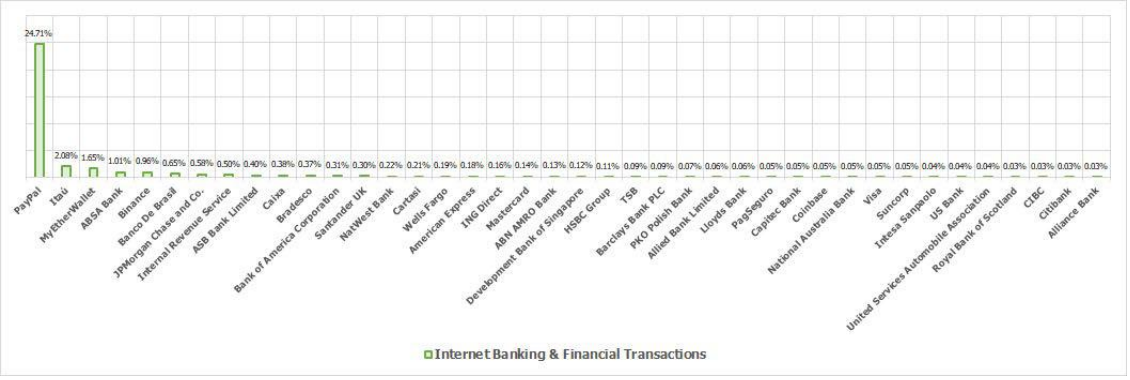

Figura 61 – C15 - ocorrências de Internet Banking & Financial Transactions

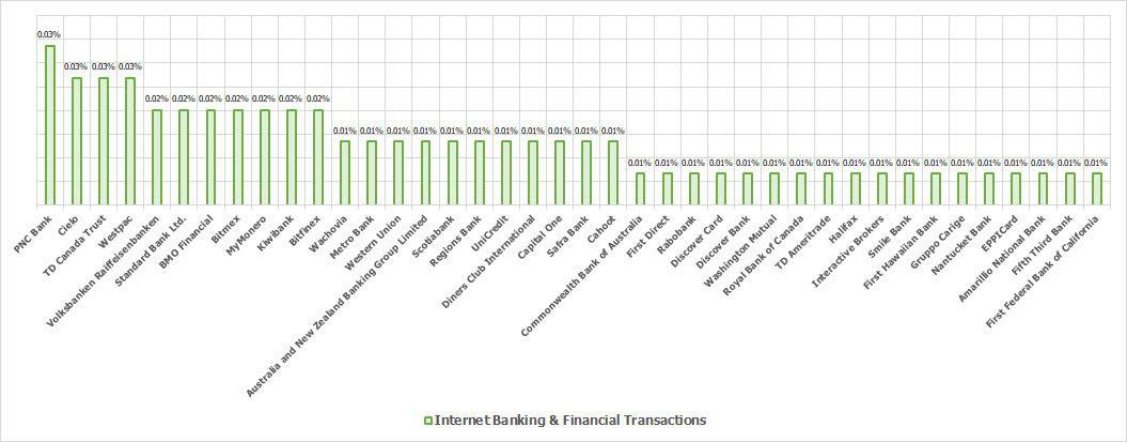

Figura 62 – C15 - ocorrências de Internet Banking & Financial Transactions

A.0.3.5 C15. Service most exploit

Essa característica avalia serviços mais propensos de *phishing*. Diante disso, foi preciso fazer uso das categorias de tipo de conteúdo definidas na C10. Os dados foram segmentados por essas categorias e identificados através de uma extração automatizada com base nos dados informados pelo *PhishTank*. Os dados extraídos estão ilustrados nas Figuras 61, 62, 63 e 64 e a análise GQM é descrita na Tabela 23.

Conforme ilustrado na Figura 61, o eixo X descreve as organizações no segmento de Internet banking & Financial Transactions, já o eixo Y descreve o quantitativo em percentual das ocorrências considerando toda a amostra. Diante disso, o serviço PayPal foi o mais explorado pelos fraudadores, representando quase 1/4, 24.71%, de todos os *phishing* válidos da Amostra #1.1. Além disso, depois do PayPal, os alvos mais explorados são em sua maioria bancos atuantes no Brasil, como *Bradesco, Caixa Econômica Federal, Santander, Itaú, Safra e Banco do Brasil*. Como os dados foram segmentados pelas categorias definidas em *C10. Content page most exploit*, foi possível observar, de forma individualizada, uma considerável exploração em determinadas empresas de segmentos distintos.

Por exemplo, conforme ilustrado na Figura 62, a área de *Telecom* e *Mobility* houveram

Figura 63 – C15 - ocorrências de E-commerce, Telecom and mobility

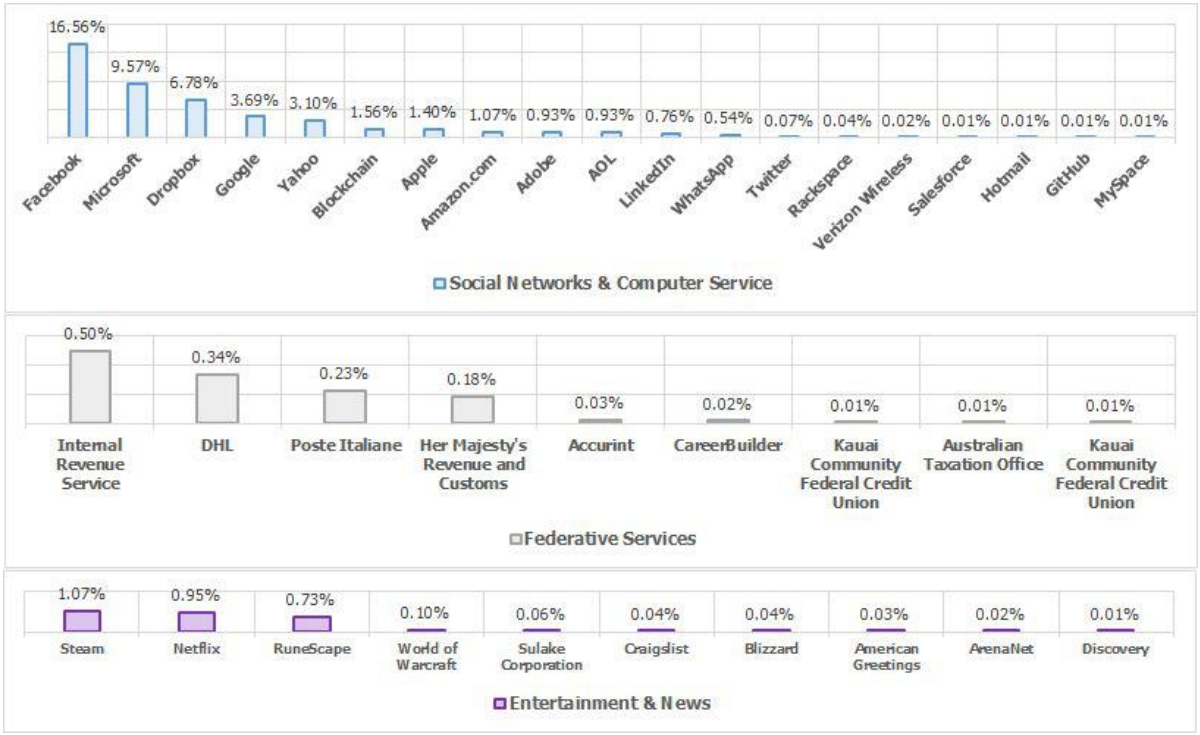

Figura 64 – C15 - ocorrências de Social Networks & Computer Services, Federal Services and Entertainment & News

Tabela 23 – GQM de C15. Service most exploit

Objetivo 3	Analisar as tendências nas ocorrências de *phishing*.				
Questão	Q15. Qual o serviço (marca) mais explorado pelo ataque?				
Métricas	[M28] Contabilizar os serviços mais direcionados nos *phishing* válidos.				
	[M29] Categorizar os serviços com base na categorização definida em C10.				
Hipóteses	Existe uma tendência referente à popularidade de um serviço, sendo o mesmo mais suscetível a ataques.				
Amostras	3	**Relevance**	STRONG	**Relations**	C11, C30
Extração	Contabilizar as ocorrências de marcas atingidas através do campo target de *PhishTank Info*.				
Limitações	A amostra obtida no PhishTank tinha muitas ocorrências de "Other", para minimizar essa situação, foi rodado um algoritmo para identificar a menção de um serviço na URL. Contudo, registros como "Other" sem nenhum tipo de indicio em sua URL precisaram ser destacartados.				
Observações	As marcas foram agrupadas com base na categorização definida no C10.				
Análise	Serviços populares como facebook e google são os alvos que mais merecem destaque. Contudo, o que merece mais destaque é a presença de bancos e lojas brasileiras. Por fim, o paypal apresenta-se como o serviço mais explorado.				

ocorrências com números muito discretos, contudo, foi possível observar que empresas consolidadas como *Orange* e *Delta Air Lines* são alvos constantes dos fraudadores, contudo, outras que ainda estão se consolidando no mercado mundial, a exemplo da Uber, já se tornaram alvos dos fraudadores. Outro fato que chama atenção é a quantidade de companhias áreas exploradas, indicando sazonalidade em períodos do turismo de alta temporada.

Conforme a Figura 63, no *E-commerce* foi possível observar uma motivação dos fraudadores em fazer uso de marcas como Alibaba, eBay e Walmart. Já no segmento de *Social Networks & Computer Services*, marcas como *Facebook, Microsoft, Dropbox e Google* são as mais exploradas. Na mesma linha, o que chamou atenção foram os casos de repartições federais, representada pela categoria *Federative Services*, que são bastante recorrentes e com certa expressão, em sua maioria relacionados a tributos federais e encomendas postais. Por fim, em *Entertainment & News*, empresas que estão emergindo, como *Steam* e *Netflix*, dominam a representatividade de ocorrências de *phishing* em seu segmento. Diante disso, a característica foi considerada com relevância *STRONG*.

A.0.3.6 C16. TLD most exploit

Essa característica avalia os domínios mais propensos de ocorrência de *phishing*. Foi preciso definir categorias para os tipos de domínio, como *Commercial, Government, Not Registered, Organizational and Others*. Importante frisar que "Not Registered" remete a domínios que utilizam registros de segundo ou terceiro nível, ou seja, que podem ser gratuitos e muitas vezes atrelados a um domínio de nível de topo. Devido a isso, foi necessário realizar uma extração manual para a análise dos dados. Também foi observado se havia URL com ou

sem " www" e se o terceiro nível foi utilizado para um determinado propósito, por exemplo *ftp.example.com*, que utiliza o nome do protocolo FTP para indicar que seria uma sessão da página para *download*. Os dados extraídos estão ilustrados nas Figuras 65 e 66 e a análise GQM é descrita na Tabela 24.

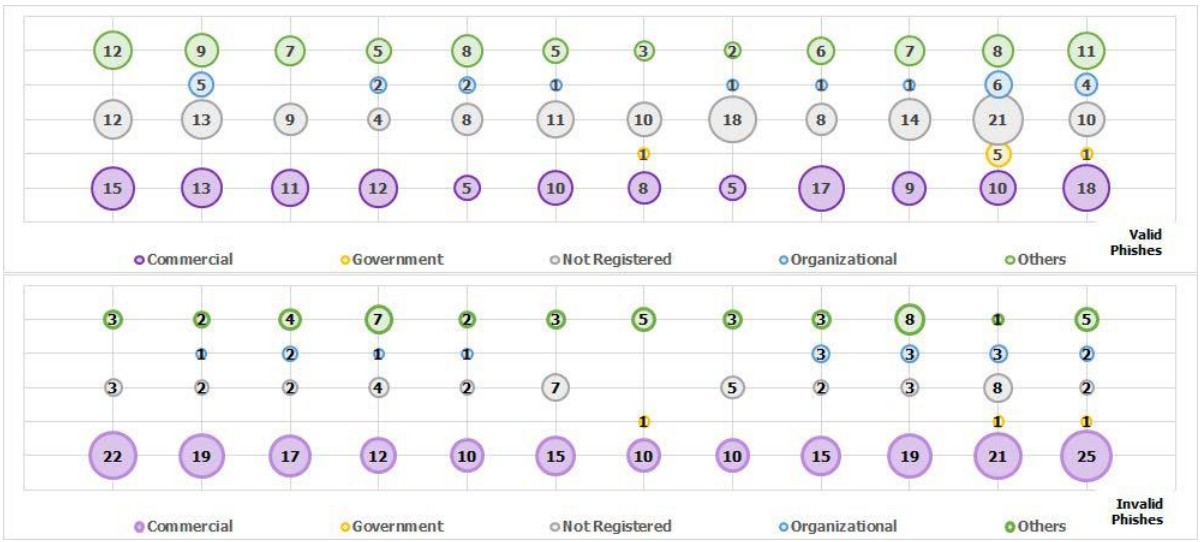

Figura 65 – C16 - ocorrências no ano de 2018 por categorias

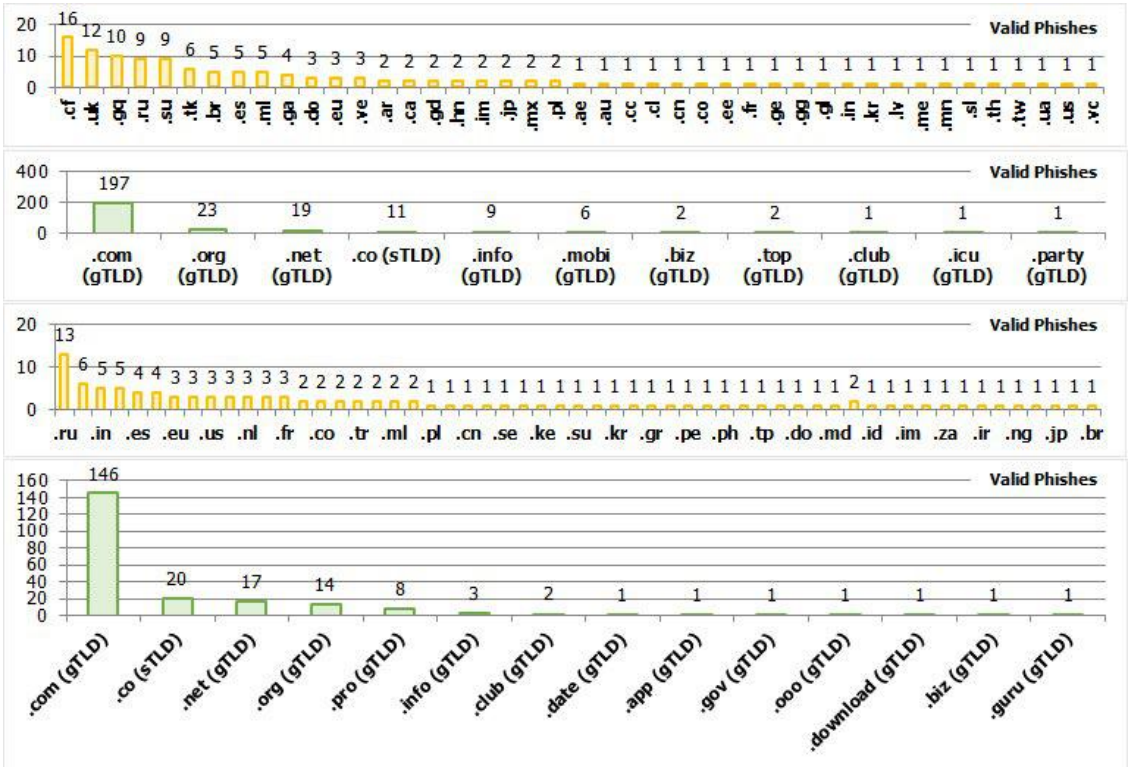

Figura 66 – C16 - ocorrências com detalhes nos casos de gTLD e sTLD

Conforme a Figura 65, as ocorrências foram agrupadas pelos meses de 2018 e segmentadas respeitando as categorias do tipo de domínio, conforme o gráfico de bolhas, tanto para válidos

Tabela 24 – GQM de C16. TLD most exploit

Objetivo 3	Analisar as tendências nas ocorrências de *phishing*.				
Questão	Q16. Qual o topo de domínio mais explorado pelo ataque?				
Métricas	[M30] Contabilizar os ataques distribuídos por mês dos *phishing* válidos, considerando o uso de TLD.				
	[M31] Contabilizar os ataques distribuídos por mês dos *phishing* inválidos, considerando o uso de TLD.				
Hipóteses	Existe uma tendência referente aos níveis de domínio, com o intuito de tornar a fraude mais fidedigna, fazendo com que um determinado nível seja mais suscetível a ataques.				
Amostras	1.1 e 2.1	**Relevance**	STRONG	**Relations**	C23, C25, C26, C30
Extração	Obter a URL e analisar o domínio manualmente.				
Limitações	A necessidade de ter apenas registros *online*, de certa forma, pode inviesar o resultado final.				
Observações	Foi necessário realizar uma análise manual pelo fato de algumas URL terem um TLD, mas os mesmos não terem sido registrados pelo fraudulento (por exemplo, casos em que um "com" da URL ser referente ao serviço de hospedagem. Nesses dados, podem existir casos em que o registro pertença a um TLD de domínio que foi sequestrado pelo atacante, por exemplo, o mesmo ter invadido um servidor legítimo. Tais situações são tratadas na característica C26.				
Análise	Foi possível observar muitos casos de registros de domínio, apresentando uma tendência dos fraudadores em tornar seus ataques mais fidedignos. Além disso, o que também chama atenção são os casos da utilização de restritos como .gov e .org. Além disso, ficou notório que a grande maioria dos falsos positivos foram com domínio .com, destes, poucos são os não registrados.				

como inválidos. Foi possível observar que os domínios comerciais são os mais utilizados pelos mal intencionados, estes inclusive, além de apresentarem maior quantidade em comparação aos "Others", quase superaram as ocorrência de "Not Registered". O fato é que, atualmente, muitos fraudadores registram seus domínios para dar maior veracidade a sua fraude, incluindo combinar técnicas conforme descritas na característica C25 "Concatenate subdomains", necessitando assim de um registro de domínio. Além disso, algumas opções, a exemplo de .tk, .ga, .cf entre outros, são oferecidos com gratuidade no primeiro ano ou eternamente.

Outro comportamento que justifica é que alguns dos domínios registrados são sequestrados e utilizados para realizar crimes, conforme descrito na característica C26. Além disso, o que também chama atenção é a existência de registros "Government" e "Organizational", que na teoria, sua obtenção deveria ser mais criteriosa. Contudo, casos de sequestros também justificam essas ocorrências. Por fim, ficou evidenciado que geralmente um *phishing* inválido tem domínio registrado do tipo "Commercial".

Dando sequencia, conforme consta na Figura 66, o primeiro gráfico descreve a ocorrência de domínios de segundo nível (SLD), muito utilizados para descrever uma seção do site, diferente do TLD que representa a extensão do domínio. Ou seja, existem técnicas para ludibriar o olhar desatento do usuário final, em que, através de um jogo de palavras, faz com que o usuário pense que o ambiente da navegação pertence a uma determinada organização, essa técnica é

descrita nas características C25 e C30.

Na mesma figura, o segundo gráfico descreve os domínios com propósito generalizado, que são atrativos por ser mais baratos ou mesmo gratuitos, no intuito do fraudador reduzir o tamanho da URL, aspecto analisado na característica C23. URL size. Já no terceiro gráfico, é descrito os domínios que representam uma região, ficando evidente que os domínios com extensão ".ru" são os mais explorados. Um dado interessante é que essa extração deixou evidente que o fato da extensão ser de um determinado país, a fraude não será necessariamente direcionada para a respectiva região. Em muitos casos, uma página pode ter uma extensão de um determinado país e ter seu conteúdo um idioma de outro, casos dessa natureza foram observados na característica C13. Language most exploit.

Por fim, na mesma figura, o quarto gráfico discrimina a quantidade de casos considerando toda a amostra #1.1, desconsiderando categorias. Foi possível observar que os domínios ".com" são os mais explorados. Diante disso, a característica foi considerada como *STRONG*.

A.0.4 URL blacklist bypass

Essa categoria descreve características direcionadas a investidas com intuito de **enganar mecanismos baseados em lista negra**. A posse desses dados visa evidenciar padrões aplicados na composição da URL.

A.0.4.1 C17. Encoded exploit

Essa característica avalia as URL que possuem o hostname ou endereço IP codificado em hexadecimal ou base64, ambos suportados pelo navegador. Diferente da exploração *punycode*, a ser descrita na característica C24, essa exploração não visa diretamente o usuário final, mas sim o mecanismo da lista negra, já que ao gerar o *hash* de uma URL com o hostname ou IP codificado, resultará em um *hash* diferente de uma URL sem codificação. Os dados seguem na Figura 67 e a análise GQM na Tabela 25.

Figura 67 – C17 - ocorrências de codificação em hostname e IP

Conforme o gráfico de colunas da esquerda na Figura 67, apenas 0.003% dos *phishing* válidos possuem codificação no *hostname*. Contudo, um dado interessante é que nenhum *phishing* inválido na amostra possui a ocorrência de codificação, levando a crer que uma vez que exista, a referida URL já pode levantar suspeitas. Em contrapartida, na mesma figura,

Tabela 25 – GQM de C17. Encoded exploit

Objetivo 4	Analisar as técnicas de ataque utilizadas para burlar mecanismos de lista negra.				
Questão	Q17. Quais registros são propagados fazendo uso de encode no hostname ou no endereço IP?				
Métricas	[M32] Contabilizar registros que possuem o hostname encoded em *phishing* válidos e inválidos.				
	[M33] Contabilizar registros que possuem o hostname encoded em *phishing* inválidos.				
Métricas	[M34] Contabilizar registros que possuem o IP encoded em *phishing* válidos.				
	[M35] Contabilizar registros que possuem o IP encoded em *phishing* inválidos.				
Hipóteses	Existe um interesse dos mal-intencionados em propagar URL maliciosas fazendo encoded do hostname ou endereço IP, para um possível "by-pass" em casos que a filtragem restringe pela parte 1 da URL.				
Amostras	1 e 2	**Relevance**	STRONG	**Relations**	C18, C23
Extração	Obter a URL com a parte 1 e 2 e analisar a presença de hostname ou endereço IP encoded.				
Limitações	-				
Observações					
Análise	A exposição de encoded no hostname da URL é razoavelmente existente nas URL maliciosas e inexiste em sites legítimos. Na mesma linha, em endereço IP também é razoavelmente existente nas URL maliciosas, contudo, tem uma presença discreta em sites legítimos.				

o gráfico da direita descreve que as ocorrências de codificação no endereço IP foram mais frequentes, havendo uma presença de 0.13% em *phishing* válidos e 0.07% em inválidos. Ainda sim, apesar de pouco aplicados, o comportamento pode ser suspeito, teoricamente com baixa margem de erro. Diante disso, a característica foi considerada com relevância *STRONG*.

A.0.4.2 C18. IP address explosure

Essa característica avalia as URL que, ao invés de expor o DSN, expõe o endereço IP. Configurando-se como uma investida para burlar os mecanismos de lista negra que geraram o hash com base na URL utilizando DNS. Esses casos levantam tantas supeitas que até mesmo, alguns aplicativos, a exemplo do *WhatsApp*, não pemitem o *hyperlink* nas mensagens com URL que utilizem um endereço IP, como forma de atenuar ataques de *phishing*. Os dados extraídos estão ilustrados na Figura 68 e a análise GQM na Tabela 26.

Conforme a Figura 68, o gráfico de pizza descreve os endereço IP que foram mais duplicados em registros de *phishing* válidos, ou seja, uma mesma página maliciosa, em um mesmo servidor, mas com URL distintas. Por exemplo, 406 registros, 0.21% de todos os *phishing* da amostra #1, são referentes a um mesmo endereço IP. Já no gráfico de colunas, foi possível observar que os casos de utilização do endereço IP foi presente tanto em *phishing* válidos como em inválidos. Diante disso, a característica foi considerada com relevância *MODERATE*.

Figura 68 – C18 - ocorrências

Tabela 26 – GQM de C18. IP address explosure

Objetivo 4	Analisar as técnicas de ataque utilizadas para burlar mecanismos de lista negra.				
Questão	Q18. Quais registros são propagados fazendo uso do IP público?				
Métricas	[M36] Contabilizar registros que explicitam o IP público em *phishing* válidos.				
	[M37] Contabilizar registros que explicitam o IP público em *phishing* inválidos.				
Hipóteses	Existe um interesse por parte dos mal-intencionados em propagar as URL maliciosas de duas maneiras, com ou sem seu DNS, no intuito de realizar um possível "by-pass", caso a abordagem por lista negra em questão não preveja a modificação do hash através dessa manobra.				
Amostras	1 e 2	**Relevance**	WEAK	**Relations**	C17
Extração	Obter a URL e analisar a presença de IP público. Também verificar a incidência de páginas com DNS que fazem referência ao mesmo IP, evidenciando assim a propagação por essa estratégia.				
Limitações	-				
Observações	Pode haver casos em que a abordagem despreze a presença de porta padrão e gere o hash sem o mesmo, evitando assim um possível "by-pass" através dessa técnica. Essa característica em questão apenas analisa a presença e não a efetividade dos mecanismos de proteção.				
Análise	Foi possível observar que 3.30% dos *phishing* reportados faziam uso de tal técnica, um número que merece atenção visto que representa mais de 6.000 registros. Além disso, foi possível observar que 406 URL com DNS fazia referência a um mesmo número de IP.				

A.0.4.3 C19. Shortened URL

Essa característica avalia as URL que utilizam serviços de encurtamento para encapsular a URL original. A hipótese em questão remete ao interesse dos mal intencionados em propagar a URL encurtada para que o hash seja gerado. Não obstante, foi identificado a ocorrência de exploração por profundidade, ou seja, diversos encurtamentos de uma mesma URL encurtada. Os dados extraídos estão ilustrados na Figura 69 e a análise GQM é descrita na Tabela 27.

Conforme a Figura 69, o eixo X do gráfico de colunas na parte superior segmenta os registros por serviço de encurtamento utilizado, ou "Genuine URL" para os casos de URL não encurtada, já o eixo Y apresenta o quantitativo de ocorrências em percentual considerando toda a amostra. Diante dos dados, 99.12% dos *phishing* válidos fizeram uso de serviços de encurtamento, ou seja, apenas 0.82% apresentaram esse recurso. Apesar de pouco explorado, ainda sim a tática perdura mesmo com o passar dos anos.

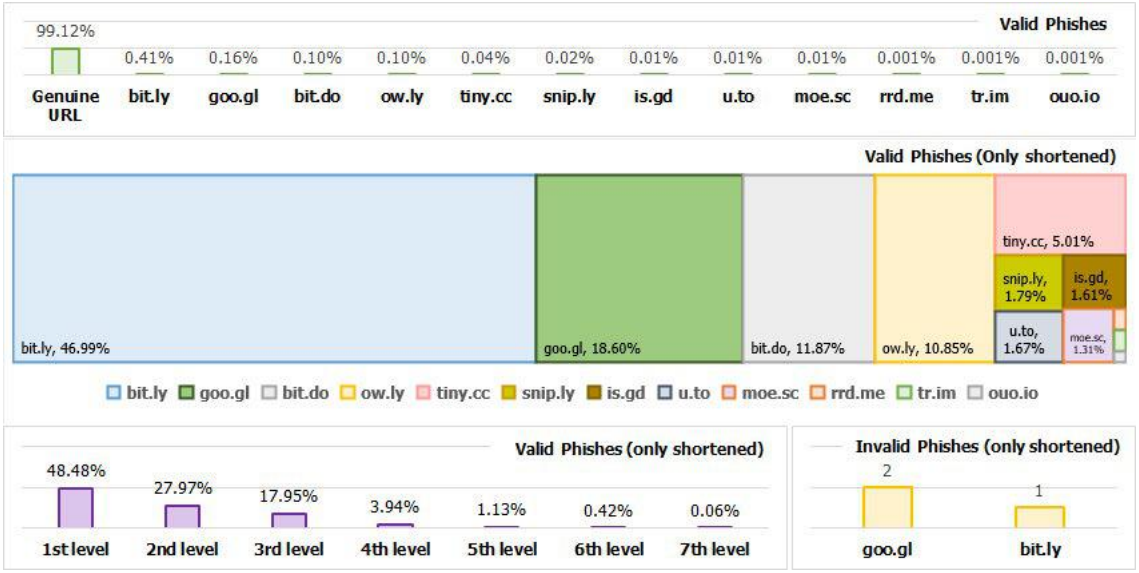

Figura 69 – C19 - ocorrências

Não obstante, alguns serviços, a exemplo do *goo.gl*, descontinuaram o serviço[5], e outros, a exemplo do *ow.ly* tornaram-se mais criteriosos, exigindo cadastro prévio e aplicando políticas de restrições, como proibir encurtar URL com endereço completo para acessar um arquivo. Contudo, o serviço *bit.ly* ainda permite tal comportamento e não adotam muitas políticas de uso, justificando sua popularidade nas fraudes. Aparentemente, a prática de encurtar uma URL de recursos de terceiros vem sendo desencorajada, o proprietário vem se motivando em encurtar a URL de seu próprio recurso, a exemplo do *YouTube* encurtando através do *youtu.be*.

Na mesma figura, o gráfico de mapa de árvore agrupa os registros de URL encurtadas por seus respectivos serviços de encurtamento, a diferença é que o eixo Y considera apenas os registros que foram encurtados. O gráfico deixa evidente que o *bit.ly* é o mais explorado. Por fim, o gráfico de colunas na parte inferior descreve *phishing* válidos que utilizaram técnica de profundidade. Essa técnica ocorre quando o atacante encurta uma URL já encurtada por diversas vezes, fazendo um encapsulamento mais profundo da URL genuína no intuito de burlar as listas negras, fazendo links diversos para uma mesma página. Diante disso, a característica foi considerada com relevância *STRONG*.

Importante salientar que a plataforma proprietária da lista negra deveria fazer um tratamento desses incidentes, o que faria dessa característica pertencer a categoria "Community-based Strategy". Contudo, pelo fato da prática ter complexidade na resolução e demais aspectos a serem observados, a exemplo da profundidade, o estudo considerou a categoria "URL blacklist bypass" por ser o comportamento predominante.

[5] O serviço https://goo.gl/ foi descontinuado em 30 de março de 2018

Tabela 27 – GQM de C19. Shortened URL

Objetivo 4	Analisar as técnicas de ataque utilizadas para burlar mecanismos de lista negra.				
Questão	Q19. Quais os registros que são propagados fazendo uso de serviços de encurtamento de URL?				
Métricas	[M38] Contabilizar os registros de *phishing* válidos que utilizam encurtamento de URL.				
	[M39] Reconhecer níveis de encurtamento em *phishing* válidos.				
	[M40] Contabilizar os registros de *phishing* inválidos que utilizam encurtamento de URL.				
	[M41] Reconhecer níveis de encurtamento em *phishing* inválidos.				
Hipóteses	Existe um interesse por parte dos mal-intencionados em propagar as URL maliciosas fazendo uso de serviços de encurtadores de URL, no intuito de realizar um possível "by-pass", caso a abordagem por lista negra em questão não preveja a modificação do hash através dessa manobra.				
Amostras	1 e 2	**Relevance**	STRONG	**Relations**	C23
Extração	Obter a URL encurtada e analisar a url resultada do desencurtamento.				
Limitações	-				
Observações	Essa análise também observou que certas URL faziam uso de 2 ou mais encurtamentos em serviços distintos para uma mesma URL, ou seja, uma determinada URL possuía níveis de encapsulamento da URL final.				
Análise	Foi observado que serviços como bit.ly e goo.gl foram os mais utilizados em 2018. Contudo, muitos desses serviços, a exemplo de goo.gl, foram descontinuados, já outros, como o ow.ly começou a adotar uma política mais rigorosa para a criação de URL encurtadas, como obrigar o usuário a ter um cadastro. Um fato curioso é que o serviço bit.ly permite encurtamento de URL que direcionam para arquivos, como executáveis ou PDF, outros serviços, como o goo.gl, não permitem esse tipo de situação. Também foi possível observar URL com até 7 níveis de encapsulamento, ou seja, uma mesma URL que fez 7 vezes o encurtamento.				

A.0.4.4 C20. URL with variables

Essa característica avalia as URL em que o atacante manipula os valores do *path* ou *querystring* da mesma para que no final o *hash* resultante seja diferente. Os dados extraídos estão ilustrados na Figura 70 e a análise GQM é descrita na Tabela 28.

Conforme ilustrado na Figura 70, os gráfico de linha fazem um comparativo da ocorrência de exploração por *path* ou *querystring* entre *phishing* válidos inválidos, deixando evidente que o comportamento é muito explorado em *phishing válidos*, com milhares de registros, diferente dos *phishing* inválidos que nem chegam a representar dezenas. Dando sequência, os primeiros gráficos de coluna fazem um comparativo entre válidos e inválidos, quantificando a exploração por *path* ou *querystring* em uma mesma URL, sendo possível identificar que uma única URL possuía 48 *path* e 8 *querystring*, algo bastante incomum. Já nos casos de *phishing* inválidos, as ocorrências são bem mais modestas, evidenciando que a ocorrência dessa exploração justifica suspeitas sobre a URL.

Na mesma figura, o penúltimo gráfico de colunas descreve as URL que realizam *hash* nos valores do *path* ou *querystring*, modificando assim a URL e burlando os mecanismos de lista negra. Através do tamanho do *hash*, foi possível identificar o algoritmo utilizado, como o MD5,

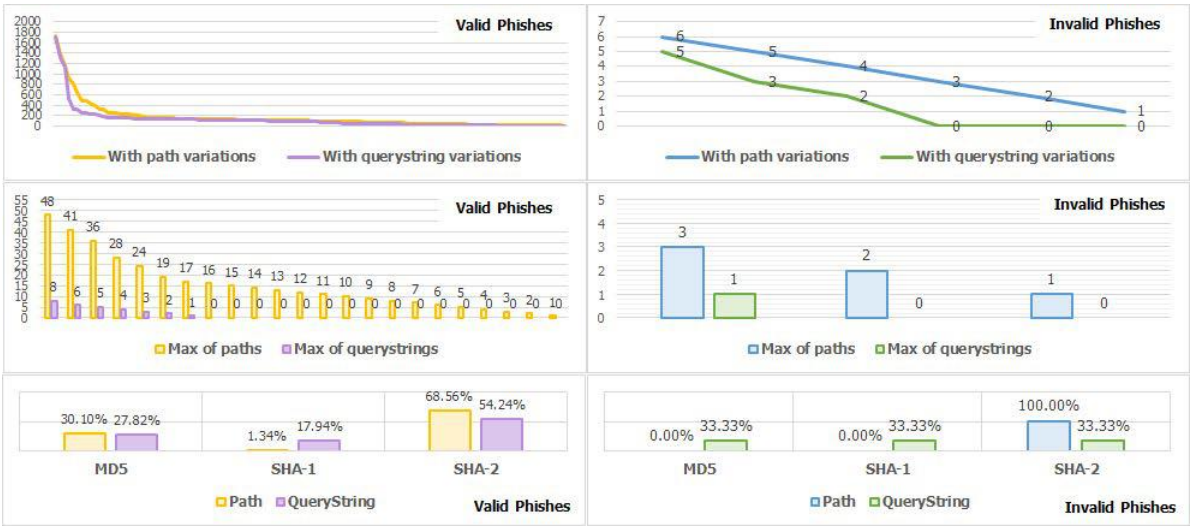

Figura 70 – C20 - ocorrências

SHA-1 e SHA-2. Diante disso, a característica foi considerada com relevância *MODERATE*.

A.0.5 URL morphology

Essa categoria descreve características que não, necessariamente, tem o intuito de realizar algum tipo de ataque, mas sim analisar **aspectos morfológicos de uma URL maliciosa**, como padrões de caracteres e tamanho da URL, resultantes de investidas.

A.0.5.1 C21. Amount of separators

Essa característica avalia as URL em que o atacante explora com uma significativa quantidade de caracteres separadores, resultando em uma URL com um determinado padrão. Diferentemente das explorações que visam prover veracidade, que são abordados na categoria "User susceptibility", os aspectos abordados nessa seção remete aos padrões resultantes da exploração que tornam-se aparentes na morfologia da URL. Os dados extraídos estão ilustrados na Figura 71 e a análise GQM é descrita na Tabela 29.

Importante frisar que a quantidade de slash "/" não foi considerada nessa característica. Na parte 1 da URL o carácter é utilizado apenas na especificação do protocolo (com double slash), já na parte 2 é utilizado para especificar um *path*, caso já analisando anteriormente na característica C20.

Conforme ilustrado na Figura 71, a identificação dos separadores foi separada pelas partes da URL. Na parte 1, foram identificados 3 separadores comumente utilizados, o "-", "_" e o "@". Foi notório que nos registros de *phishing* válidos a incidência é de fato muito maior, por exemplo, no caso do uso do *hífen*, existiu uma média de 8.33 caracteres por URL, um valor superior a quantidade máxima de ocorrências nas URL não maliciosas. Na parte 2, além dos caracteres apresentados, também foi observado o uso do ".". O fato de não ter sido abordado

Tabela 28 – GQM de C20. URL with variables

Objetivo 4	Analisar as técnicas de ataque utilizadas para burlar mecanismos de lista negra.				
Questão	Q20. Quais registros são propagados com variáveis na composição da URL?				
Métricas	[M42] Contabilizar os tipos de valores dinâmicos no *querystring* da URL em *phishing* válidos.				
	[M43] Contabilizar os tipos de valores dinâmicos no path da URL em *phishing* válidos.				
	[M44] Contabilizar os registros que utilizam valores dinâmicos no *querystring* da URL em *phishing* inválidos.				
	[M45] Contabilizar os registros que utilizam valores dinâmicos no path da URL em *phishing* inválidos.				
	[M46] Contabilizar os tipos de valores dinâmicos no *querystring* da URL em *phishing* inválidos.				
	[M47] Contabilizar os tipos de valores dinâmicos no path da URL em *phishing* inválidos.				
	[M48] Contabilizar os tipos de valores dinâmicos no *querystring* da URL em *phishing* inválidos.				
	[M49] Contabilizar os tipos de valores dinâmicos no path da URL em *phishing* inválidos.				
Hipóteses	Existe um interesse por parte dos mal-intencionados em propagar as URL maliciosas fazendo uso de valores dinâmicos no *querystring* ou *path*, no intuito de realizar um possível "by-pass", caso a abordagem por lista negra em questão não preveja a modificação do hash através dessa manobra.				
Amostras	1 e 2	**Relevance**	MODERATE	**Relations**	C03, C23, C30
Extração	Obter a URL e analisar sua parte 2.				
Limitações	-				
Observações					
Análise	Certas abordagens através de lista negra não criam o *hash* com base apenas da primeira parte da URL, mas sim com ambas as partes. Nesses casos, os valores dinâmicos propiciam uma considerável variação de URL para a mesma página fraudulenta. Foi possível observar um considerável número de variações fazendo uso dessa técnica. Um fato que chama atenção é aproximadamente 50 URL maliciosas possuiam 48 *paths* em sua composição, diferente das URL legítimas que apresentaram um máximo de 3 *paths* em sua composição.				

na primeira parte é devido sua função de delimitar subdomínios nessa parte da URL, caso que será analisado com outra ótica na característica C25.

Nessa segunda parte da URL a ocorrência de utilização dos separadores é bastante superior a primeira, a exemplo do uso do *hífen*, com média de 15.37 caracteres utilizados em uma URL maliciosa, contra 7.60 caracteres nas URL não maliciosas. Diante disso, a característica foi considerada com relevância *MODERATE*.

A.0.5.2 C22. HTTP with specification port

Essa característica avalia as URL em que o atacante utiliza uma porta diferente da porta padrão. De fato, um dos motivos de utilizar uma porta específica seja para burlar a lista negra, já que o atacante poderia periodicamente ir modificando a mesma para fazer "bypass'

Tabela 29 – GQM de C21. Amount of separators

Objetivo 5	Analisar padrões morfológicos de uma URL maliciosa.				
Questão	Q21. Quais registros possuem um maior número de caracteres separados na composição da URL?				
Métricas	[M50]: Contabilizar registros que utilizam o caractere "-" na primeira parte dos *phishing* válidos.				
	[M51]: Contabilizar registros que utilizam o caractere "_" na primeira parte dos *phishing* válidos.				
	[M52]: Contabilizar registros que utilizam o caractere "@" na primeira parte dos *phishing* válidos.				
	[M53]: Contabilizar registros que utilizam o caractere "-" na primeira parte dos *phishing* inválidos.				
	[M54]: Contabilizar registros que utilizam o caractere "_" na primeira parte dos *phishing* inválidos.				
	[M55]: Contabilizar registros que utilizam o caractere "@" na primeira parte dos *phishing* inválidos.				
	[M56]: Contabilizar registros que utilizam o caractere "-" na segunda parte dos *phishing* válidos.				
	[M57]: Contabilizar registros que utilizam o caractere "." na segunda parte dos *phishing* válidos.				
	[M58]: Contabilizar registros que utilizam o caractere "_" na segunda parte dos *phishing* válidos.				
	[M59]: Contabilizar registros que utilizam o caractere "@" na segunda parte dos *phishing* válidos.				
	[M60]: Contabilizar registros que utilizam o caractere "-" na segunda parte dos *phishing* inválidos.				
	[M61]: Contabilizar registros que utilizam o caractere "." na segunda parte dos *phishing* inválidos.				
	[M62]: Contabilizar registros que utilizam o caractere "_" na segunda parte dos *phishing* inválidos.				
	[M63]: Contabilizar registros que utilizam o caractere "@" na segunda parte dos *phishing* inválidos.				
Hipóteses	Tipicamente, uma URL maliciosa adota estratégias de utilização de separadores em sua URL.				
Amostras	1 e 2	**Relevance**	MODERATE	**Relations**	C23, C25
Extração	Obter a URL e analisar suas partes 1 e 2.				
Limitações	-				
Observações	A url foi dividida em duas partes, a primeira compõe seu protocolo, subdomínios, domínio e porta padrão. Já a segunda parte contem *paths*, *querystrings* e nomenclatura de arquivos. Na parte 1 o caractere ponto foi descartado, pois representa a separação de subdomínios, analisado na C26.				
Análise	Foi possível observar um grande número de separadores sendo utilizado na prática das fraudes. Aparentemente, domínios sem tantos caracteres podem levantar maiores suspeitas para o usuário final. Essa característica reflete na C23, resultando em URL com tamanho considerável.				

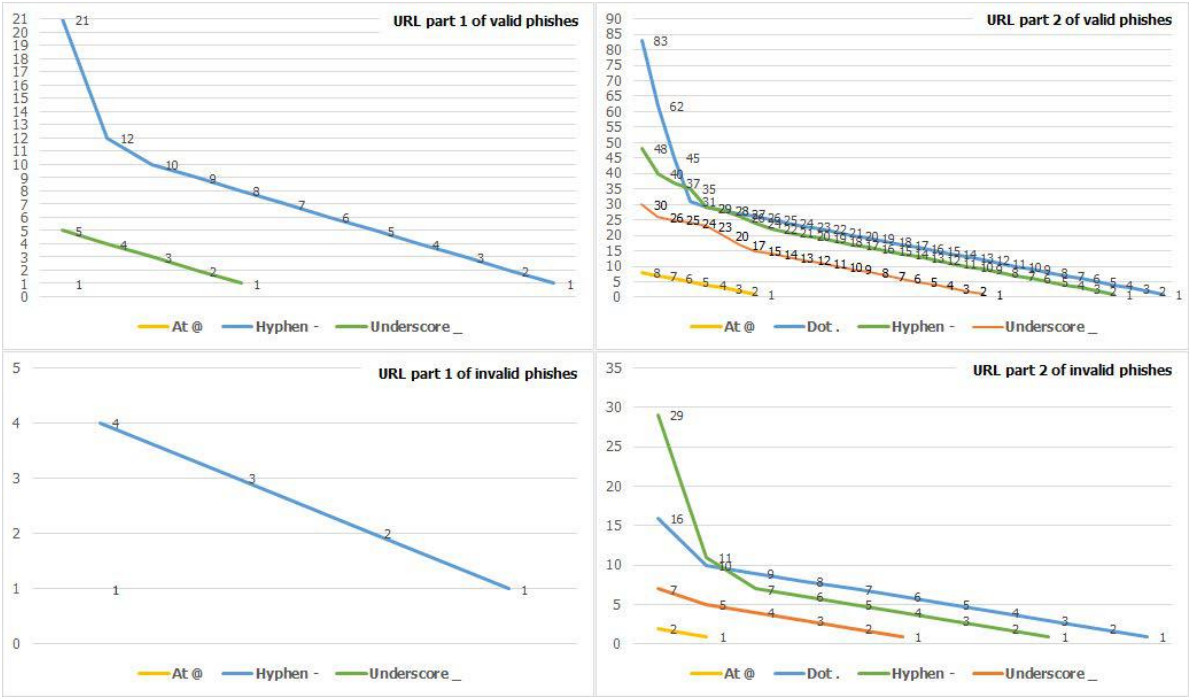

Figura 71 – C21 - ocorrências

na URl, contudo, a hipótese em questão analisa se o uso de porta específica é comum em incidências de *phishing*, tornando-se assim um padrão recorrente a morfologia da URL. Em suma, a característica em questão visa observar se o fato do site utilizar uma porta diferente da padrão fortalece suspeitas sobre uma possível fraude. Os dados extraídos estão ilustrados na Figura 72 e a análise GQM é descrita na Tabela 30.

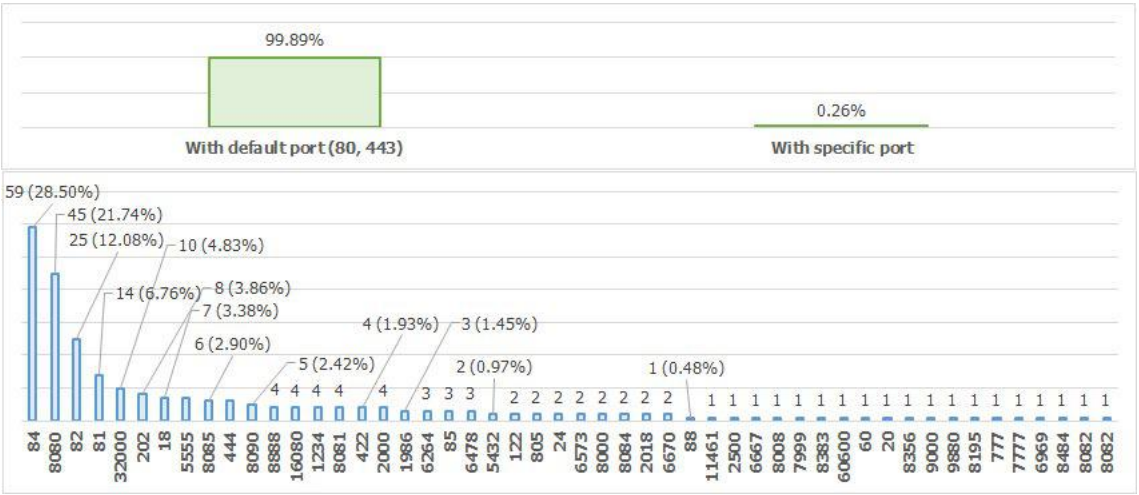

Figura 72 – C22 - ocorrências

Conforme a Figura 72, foi possível observar um discreto número de portas sendo utilizadas nos registros de *phishing* válidos, apenas 0.26%, conforme descrito no primeiro gráfico de colunas. Contudo, o que chama atenção é que não ocorreram casos de uso de porta específica nos registros de *phishing* válidos, fazendo considerar a característica como um indicador

Tabela 30 – GQM de C22. HTTP with specification port

Objetivo 5	Analisar padrões morfológicos de uma URL maliciosa.				
Questão	Q22. Quais registros fazem uso de porta específica para rodar o serviço HTTP?				
Métricas	[M64]: Contabilizar registros de *phishing* válidos que utilizam portas diferentes de 80 ou 443.				
	[M65]: Contabilizar registros de *phishing* inválidos que utilizam portas diferentes de 80 ou 443.				
Hipóteses	Tipicamente, uma URL maliciosa adota estratégias de utilização de separadores em sua URL.				
Amostras	1 e 2	**Relevance**	WEAK	**Relations**	C01
Extração	Obter a URL e analisar sua parte 2 e observar se ocorre menção a portas diferentes de 80 e 443.				
Limitações	Não houveram registros de *phishing* inválidos com uso de porta específica.				
Observações					
Análise	Essa característica apresenta-se presente nas URL maliciosas, porém, sem valores tão significativos.				

para uma URL suspeita. Além disso, existe uma considerável diversificação do uso da porta quando a mesma é utilizada, conforme descreve o segundo gráfico de colunas. Diante disso, a característica foi considerada com relevância *WEAK*.

Não existe uma maneira precisa para descobrir a tecnologia utilizada pelo mal intencionado para desenvolver a fraude, em contrapartida, a porta poderia ser um possível indicador para tentar responder esse questionamento, por exemplo, muitos servidores de aplicação adotam uma determinada porta como padrão, conforme é especificado no site do IANA [6], a porta 8080 é comumente utilizada para rodar serviços como o *tomcat*, ou seja, possivelmente as fraudes são desenvolvidas, em sua maioria, na tecnologia Java.

A.0.5.3 C23. URL size

Essa característica avalia o tamanho da URL e visa investigar se o padrão do tamanho da URL pode levantar suspeitas da mesma ser maliciosa, uma vez que demais estratégias, a exemplo das características C17, C19 e C25, que como consequência, aumentam o tamanho de caracteres da URL. Os dados extraídos estão ilustrados na Figura 73 e a análise GQM é descrita na Tabela 31. Adicionalmente, na Figura 74 ilustra que a característica também avaliou o tamanho dos domínios e subdomínios das respectivas URL.

Conforme a Figura 73, o gráfico de colunas descreve no eixo X a quantidade de caracteres, agrupado por margens como uma escala. O mesmo gráfico apresenta os resultados em dados categóricos, ou seja, propõe uma escala de intervalos entre a quantidade, conforme apresentado no eixo Y. Diante disso, foi possível observar que a maioria das URL possuem entre 25 e 50 caracteres. Um comportamento observado é que boa parte dos *phishing* válidos concentram-se

[6] https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml

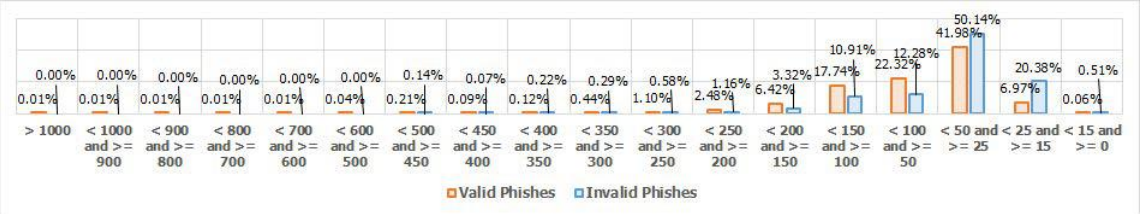

Figura 73 – C23 - ocorrências

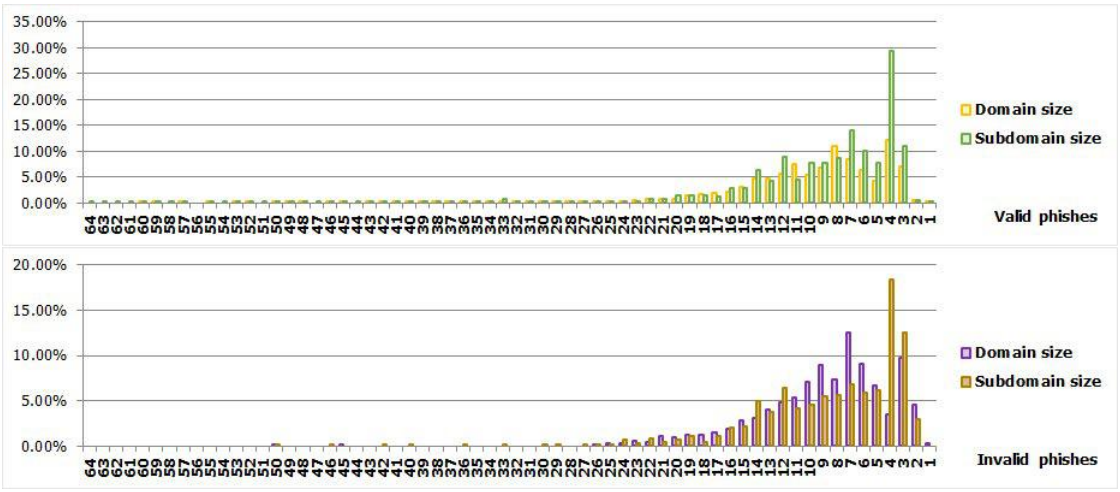

Figura 74 – C23 - análise do tamanho de domínios e subdomínios

nas ocorrências com maior número de caracteres, já os inválidos são mais presentes a medida que a quantidade de caracteres é reduzida, conforme o gráfico de colunas.

Um fato que chama atenção é que a partir de 500 caracteres foi encontrado apenas URL maliciosas, traduzindo como um possível indicador para levantar suspeitas sobre uma determinada URL. Diante disso, a característica foi considerada com relevância *MODERATE*. A Figura 74 descreve no eixo X a quantidade de caracteres utilizada no domínio e subdomínio, que vai de 1 até 64 caracteres, conforme o mínimo e máximo definido no protocolo DNS. Como uma URL pode ter 1 ou mais subdomínios, cada subdomínio (delimitado por ponto) foi considerado individualmente. Foi possível observar que tanto válidos ou inválidos, os domínios tem em sua maioria 7 caracteres e os subdomínios 4.

A.0.6 User susceptibility

Essa categoria relaciona características direcionadas a investidas com intuito de prover **maior veracidade** no conteúdo a fim de **persuadir o usuário final**. A posse desses dados visa evidenciar padrões aplicados na composição da URL.

A.0.6.1 C24. Browser punycode exploit

Essa característica avalia as explorações de recursos do browser que podem ser feitas através da URL da página com o ataque denominado *punycode*. Conforme o RFC 3492 (COSTELLO, 2003),

Tabela 31 – GQM de C23. URL size

Objetivo 5	Analisar padrões morfológicos de uma URL maliciosa.				
Questão	Q23. Quais registros de URL possuem grande número de caracteres?				
Métricas	[M66]: Contabilizar os caracteres na URL de um *phishing* válido.				
	[M67]: Contabilizar os caracteres na URL de um *phishing* inválido.				
	[M68]: Contabilizar os caracteres do domínio na URL de um *phishing* válido.				
	[M69]: Contabilizar os caracteres do domínio na URL de um *phishing* inválido.				
	[M70]: Contabilizar os caracteres do subdomínio na URL de um *phishing* válido.				
	[M71]: Contabilizar os caracteres do subdomínio na URL de um *phishing* inválido.				
Hipóteses	Tipicamente, muitas das URL maliciosa apresentam um grande número de caracteres.				
Amostras	1 e 2	**Relevance**	MODERATE	**Relations**	C07, C09, C16, C17, C19, C20, C21, C23
Extração	Obter a parte 1 e 2 da URL e contabilizar seus caracteres.				
Limitações	-				
Observações					
Análise	URL maliciosas possuem um grande número de caracteres quando há ausência de encurtamento devido as técnicas de variáveis, separadores e subdomínios aplicados sobre a mesma. Foi observado que mais de 1% das URL possuem mais de 250 caracteres em sua composição. Já as URL legítimas, mais de 50% das mesmas possuem menos de 50 caracteres.				

o *punycode* é um protocolo que permite a conversão de caracteres com *unicode* específico, como o chinês ou russo, em uma versão compatível para nomes de domínios DNS. Essa cadeia de caracteres é sempre precedida do prefixo *"xn–"* e a responsabilidade da conversão é atribuída ao navegador *Web*.

Na prática, domínios com caracteres acentuados, por exemplo "Netflíx.com", resultam em "http://xn–netflx-7va.com". Contudo, as opções não se restringem ao uso do *unicode* latino. Ao utilizar o alfabeto cirílico (línguas eslavas), a palavra *apple* em *unicode* cirílico (JOVANOVIC, 2009) convertida em *punycode* resulta em *xn–80ak6aa92e*, ou seja, uma versão ASCII válida para um domínio. Outra maneira de sofisticar o golpe é a possível combinação de *unicode* distintos para resultar em uma cadeia de caracteres homógrafos, ou seja, idêntica ao site genuíno. O fato é que alguns idiomas, a exemplo do cirílico, não possuem caracteres como *g* ou *l* minúsculos, o que dificulta o infrator forjar um conjunto de caracteres que se assemelhem a um domínio como o *google.com*. O mais aproximado seria algo como *GooGle.com*, porém, suscetível à suspeitas.

Contudo, nada impede que o infrator use combinações de *unicode* diferentes, como por exemplo usar o *g* e *l* minúsculos do alfabeto latim básico e os demais caracteres do alfabeto cirílico, resultando nos caracteres *xn–ggl-tdd6ba.com*, que convertidos pelo *punycode*,

Tabela 32 – GQM de C24. Browser punycode exploit

Objetivo 6	Analisar padrões adotados pelo ataque para minimizar suspeitas do usuário final.				
Questão	Q24. Quais registros utilizam o recurso de punycode?				
Métricas	[M72]: Contabilizar os registros de *phishing* válidos que utilizam punycode.				
	[M73]: Contabilizar os registros de *phishing* inválidos que utilizam punycode.				
Hipóteses	Eventualmente, para simular maior fidedignidade, uma URL maliciosa utiliza punycode, resultando em maior veracidade.				
Amostras	1 e 2	**Relevance**	MODERATE	**Relations**	C28
Extração	Obter a parte 1 da URL e contabilizar e analisar os caso de utilização de punycode.				
Limitações	-				
Observações	A análise por punycode foi realizada direto na URL.				
Análise	Apesar dos números modestos, ficou evidenciado que ataques de punycode tornaram-se presentes durante o ano de 2018, sendo uma investida explorada pelos atacantes.				

resultam em *google.com*[7]. Essa "funcionalidade" do navegador foi proposta como uma representação mais amigável para o usuário final, mas acabou se tornando uma oportunidade para os mal intencionados. O atacante muitas vezes adotar medidas para deixar um site forjado ainda mais verossímil, como, por exemplo, registrar o domínio *xn–80ak6aa92e.com* e oferecer tunelamento. Os dados extraídos estão na Figura 75 e a análise GQM é descrita na Tabela 32.

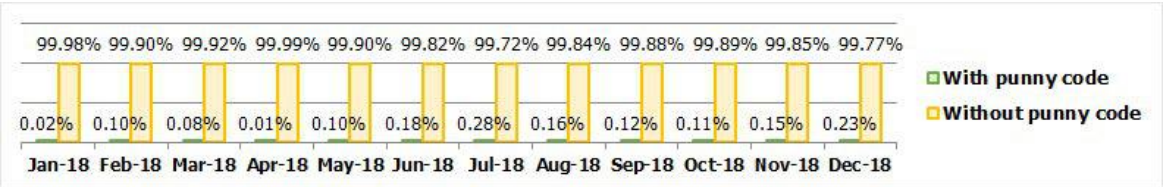

Figura 75 – C24 - ocorrências

Atualmente os navegadores minimizaram tais problemas, uma vez que no ato da conversão com *punycode* deixou de realizar a troca "amigável" na barra de endereço, permanecendo na exibição dos caracteres originais. Outro critério foi considerar como defensivo em casos em que os caracteres fazem uso de *unicode* distintos. Contudo, ainda é possível fazer uso dessas técnicas em elementos HTML como páginas *Web* ou corpo de *e-mails*, a exemplo do atributo *href* em elementos $< a >$ e o uso de *unicode HTML*, exigindo assim maior acuidade por parte dos usuários-finais em observar tal exploração. Por fim, também foi observado páginas que solicitavam a instalação de *plug-in* ou extensões no navegador, evidenciando a utilização de *malware*. Diante disso, a característica foi considerada com relevância *MODERATE*.

[7] Conversor *online* de *punycode* para ASCII: http://idna-converter.com/

Tabela 33 – GQM de C25. Concatenate subdomains

Objetivo 6	Analisar padrões adotados pelo ataque para minimizar suspeitas do usuário final.				
Questão	C25. Quais registros que exploram o uso de subdomínios?				
Métricas	[M74]: Contabilizar os subdomínios em registros de *phishing* válidos.				
	[M75]: Contabilizar os subdomínios em registros de *phishing* inválidos.				
Hipóteses	Eventualmente, uma URL maliciosa tem inúmeros subdomínios para ocultar do usuário final o real domínio a ser trafegado.				
Amostras	1 e 2	**Relevance**	MODERATE	**Relations**	C16, C21 e C23
Extração	Obter a parte 1 da URL e contabilizar os subdomínios utilizados.				
Limitações	-				
Observações					
Análise	Para simular maior fidedignidade, uma URL maliciosa pode persuadir um usuário para que pense que esteja navegando em domínio legítimo, mas que na verdade é um jogo de palavras através de subdomínios. Essa técnica pode ser combinada com C20 e C30 para maior elaboração na simulação de fidelidade.				

A.0.6.2 C25. Concatenate subdomains

Essa característica avalia padrões em URL que apresentam diversas concatenações de subdomínios para levar a crer, através de uma observação pouco apurada do usuário final, que a URL apresentada no browser induz um ambiente de execução com domínio legítimo, por exemplo, *facebook.edit.youraccount.com*, quando na verdade o domínio em questão trata-se de um registro com nome *youraccount.com* que foi criado pelo mal intencionado, e o subdomínio edit e facebook foi o efeito visual manipulado pelo fraudador. Os dados extraídos estão ilustrados na Figura 76 e a análise GQM é descrita na Tabela 33.

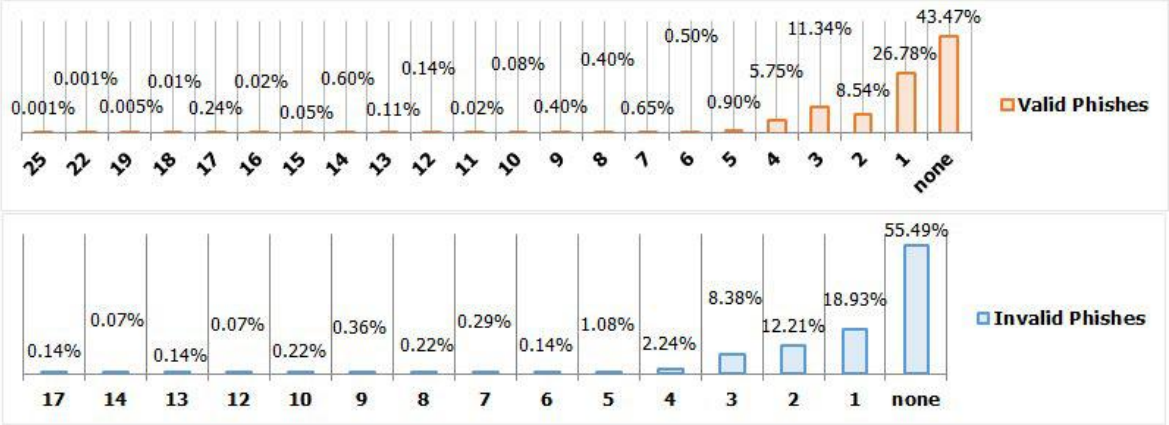

Figura 76 – C25 - ocorrências

Conforme ilustrado na Figura 76, foi possível observar que o uso de muitos subdomínios é bastante comum em ataques de *phishing*, além disso, comparando entre válidos e inválidos, a incidência é bem maior em casos de *phishing* válidos, justificando a característica com relevância *STRONG*. Conforme descrito na Figura 76, foi possível detectar casos de 25 subdo-

mínios em uma única URL. As palavras-chave mais comum na composição dessa concatenação era fazendo alusão a serviços consolidados, como facebook ou dropbox, em combinação com termos como security, login ou autenticação.

A.0.6.3 C26. Domain with reputation

Essa característica avalia casos em que o fraudador conseguiu obter controle de um determinado registro fidedigno e usa a reputação do mesmo para persuadir o usuário final, ou seja, um domínio sequestrado. Os casos de sequestros ocorrem comumente por falha de sanitização das entradas das aplicações, por exemplo, sessões de *upload* que permitem injetar página maliciosa. Além desses, casos em que a fraude possui um domínio com registro de acesso mais restrito, a exemplo de domínios ".gov" e ".org", também se enquadram na característica. Não é incomum encontrar domínios governamentais e organizacionais sendo utilizados para o crime, sejam sequestrados ou com registro oficialmente concedido. Os dados extraídos estão ilustrados na Figura 77 e a análise GQM é descrita na Tabela 34.

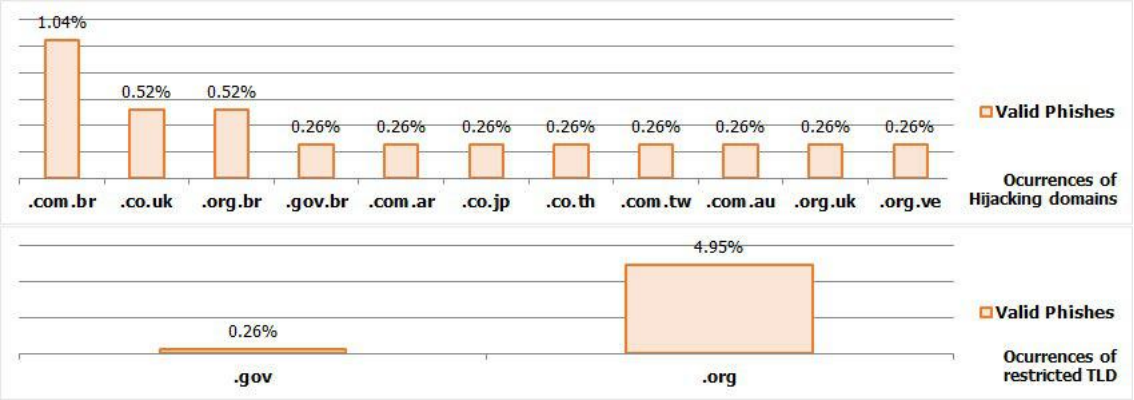

Figura 77 – C26 - ocorrências

Conforme a Figura 77, esse tipo de incidente ocorre com uma quantidade considerável e bem disseminado pelo mundo todo. Contudo, o que chama atenção é a quantidade de domínios brasileiros (.com.br) sequestrados, que apresenta-se como o mais explorado e com o dobro de ocorrências em comparação ao segundo colocado. Outro ponto que merece destaque são os domínios .org, que por serem mais restritos, trazem ainda mais veracidade, sendo assim, mais atrativos para os mal intencionados, justificando a característica com relevância *STRONG*.

A.0.6.4 C27. HTTP tunneling

Essa característica avalia os casos em que o fraudador não mede esforços quanto ao investimento em garantir maior fidedignidade em sua fraude, contratando domínios com recursos de tunelamento e inclusive registrando os mesmos em certificadoras digitais. Os dados extraídos estão ilustrados na Figura 78 e a análise GQM é descrita na Tabela 35.

Tabela 34 – GQM de C26. Domain with reputation

Objetivo 6	Analisar padrões adotados pelo ataque para minimizar suspeitas do usuário final.				
Questão	C26. Quais *phishing* que se apropriaram de domínios com reputação?				
Métricas	[M76]: Contabilizar a quantidade de *phishing* válidos em domínios que foram sequestrados.				
	[M77]: Contabilizar a quantidade de *phishing* válidos em domínios que possuem registro restrito para organizações.				
Hipóteses	Eventualmente, para simular maior fidedignidade, uma URL maliciosa pode fazer uso de um domínio com reputação que foi apropriado pelo mal-intencionado, seja através de sequestro ou por falta de critério na contratação do mesmo.				
Amostras	1.1	**Relevance**	STRONG	**Relations**	C09, C16, C28, C29
Extração	Obter a parte 1 da URL e contabilizar manualmente os *phishing* que pertenciam a um domínio sequestrado e casos de registro em domínios organizacionais.				
Limitações	Não foi necessário realizar análise comparativa entre *phishing* válidos e inválidos. Para a obtenção dos dados, foi realizado através de processo manual. Inicialmente foi pesando utilizar a API Domage para obter os dados através do *WHOIS*, contudo, ficou inviável confirmar se o domínio era sequestrado ou não. Além disso, foi preciso analisar casos em que um determinado domínio havia sido sequestrado e pertencia a um domínio restrito. Nesses casos, os registros eram considerados apenas na métrica de domínios sequestrados.				
Observações	O roubo de domínio pode ocorrer de diversas formas, seja explorando a aplicação, o servidor de hospedagem ou o serviço mantenedor do DNS. Conforme descrito anteriormente na característica C16, o mesmo não fez restrição em casos de domínios roubados ou não, pelo fato de ter como objetivo apresentar os dados no contexto de tendência, já a característica C26 tem o objetivo de apresentar os dados considerando o contexto da persuasão.				
Análise	Existe um número considerável de casos de roubo, chamando atenção aos domínios brasileiros.				

Tabela 35 – GQM de C27. HTTP tunneling

Objetivo 6	Analisar padrões adotados pelo ataque para minimizar suspeitas do usuário final.				
Questão	C27. Quais registros que fazem uso de HTTPS?				
Métricas	[M78]: Contabilizar os registros de *phishing* válidos que utilizam HTTPS.				
Métricas	[M79]: Contabilizar os registros de *phishing* inválidos que utilizam HTTPS.				
Hipóteses	Eventualmente, para simular maior fidedignidade, uma URL maliciosa pode fazer uso de HTTPS.				
Amostras	1.1	**Relevance**	MODERATE	**Relations**	C09
Extração	Obter a parte 1 da URL e analisar o protocolo utilizado.				
Limitações	-				
Observações	-				
Análise	Foi observado que poucas URL faziam uso de HTTPS. Mas ainda sim, uma parte de mal intencionados não exitam em investir para que sua fraude apresente-se mais fidedigna ao usuário final.				

Figura 78 – C27 - ocorrências

Conforme ilustrado na Figura 78, foi possível observar que 88.85% dos *phishing* válidos não utilizam o protocolo HTTPS, já no caso dos válidos, os casos que não possuem o cadeado foi apenas 27.53%. Diante disso, era possível considerar a característica com alta relevância, contudo, no ano de 2017 o quantitativo de páginas com cadeado era de apenas 4.92%, ou seja, menos da metade das ocorrências em 2018, com 11.15%. Esse comportamento evidencia que a característica vem perdendo força com o passar do tempo. É fato que recursos gratuítos, a exemplo do *Let's Encrypt*[8], aumentam a ocorrência de páginas com cadeado, fazendo o mesmo ser questionável de ser considerado como critério para aumentar ou diminuir a fidedignidade, portanto, a relevância da característica ficou definida como *MODERATE*.

A.0.6.5 C28. Malicious browser-based code

Essa característica avalia as explorações através do código-fonte da página no intuito de trazer maior fidedignidade para o usuário final, persuadindo o mesmo a acreditar que a fraude em questão trata-se de um site genuíno. Apesar do código-fonte fazer parte do conteúdo da página, a característica é considerada estática porque descreve uma análise léxica do código-fonte da página, ou seja, o conteúdo analisado não é o mesmo a ser apresentado ao usuário final, portanto, aspectos subjetivos não são considerados.

Padrões dessa natureza são recorrentes através da manipulação de cabeçalhos HTTP como User-Agent e Refere, tipicamente utilizados para ataques de *SMiShing*, ou seja, direcionados para um dispositivo móvel, ou para um determinado contexto, a exemplo dos ataques de *Spear Phishing*. Não obstante, casos de persuadir o usuário a instalar *plug-ins* ou extensões maliciosas no browser também foram investigadas. Os dados extraídos estão ilustrados na Figura 79 e a análise GQM é descrita na Tabela 36. Como o comportamento dessa característica é totalmente ofensivo, não havia sentido realizar comparação entre amostras de válidos e inválidos.

Conforme a Figura 79, foi possível identificar que boa parte das páginas analisadas realizavam combinação de "User-Agent" com "Force 404 by IP", resultando que uma determinada página era aberta no navegador do smartphone, mas ao tentar abrir no navegador de um PC era apresentado um erro forjado de 404, deixando evidente que tratava-se de um *phishing* do tipo *SMiShing*. Diante disso, a característica foi considerada com relevância *STRONG*.

[8] https://letsencrypt.org/

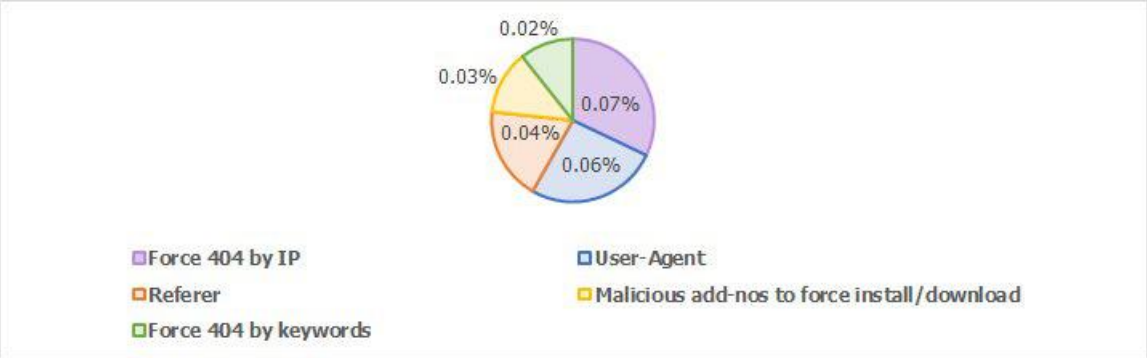

Figura 79 – C28 - ocorrências

Tabela 36 – GQM de C28. Malicious browser-based code

Objetivo 6	Analisar padrões adotados pelo ataque para minimizar suspeitas do usuário final.				
Questão	Q23. Quais registros exploram recursos do navegador?				
Métricas	[M80]: Contabilizar os registros que utilizam código javascript para manipular o resultado do carregamento da página no cliente.				
	[M81]: Contabilizar os registros que utilizam o HTTP header para manipular o resultado do carregamento da página no cliente.				
	[M82]: Contabilizar os registros que sugerem ao usuário final o download de algum arquivo.				
Hipóteses	Eventualmente, para simular maior fidedignidade, uma URL maliciosa pode necessitar ao usuário final a instalação de um add-ons no browser para maior experiência de uso.				
Amostras	1.1	**Relevance**	STRONG	**Relations**	C24, C26
Extração	Obter o conteúdo e cabeçalho da página e analisar os recursos do browser que são explorados.				
Limitações	-				
Observações	A análise das métricas foram observadas através do conteúdo da página, como elementos e códigos javascript.				
Análise	Foi observado que certos registros utilizam recursos do cabeçalho HTTP para intervir no resultado carregamento da página. Os cabeçalhos como User-Agent e Referer foram bastante utilizados, alguns destes forjava um erro 404 para o caso de abrir a página em um dispositivo que não fosse móvel, como um smartphone, representando assim um tipo de ataque de SMiShing. Além disso, também foi identificado código javascript para forçar erro 404 para um determinada região (através do endereço IP), representando assim um tipo de ataque de Spear *phishing*. Por fim, os ataques de punycode tornaram-se presentes durante o ano de 2018.				

A.0.6.6 C29. URL redirection

Essa característica avalia os casos em que o fraudador explora URL de sites legítimos que possibilitam um redirecionamento através da manipulação de *path* ou *querystring*. Na prática, o protocolo HTTP permite que os valores desses parâmetros sejam modificados arbitrariamente durante as requisições GET. Em muitos casos, a aplicação portadora da URL não realiza um tratamento dessas entradas, possibilitando que um mal intencionado informe uma URL maliciosa dentro dessa URL legítima. Por ser legítima, visualmente o usuário pode acabar

Tabela 37 – GQM de C29. URL redirection

Objetivo 6	Analisar padrões adotados pelo ataque para minimizar suspeitas do usuário final.				
Questão	C29. Quais registros que são propagados fazendo uso de redirecionamento na URL?				
Métricas	[M83]: Contabilizar os registros de *phishing* válidos que possuem redirecionamento em sua URL.				
Métricas	[M84]: Contabilizar os registros de *phishing* inválidos que possuem redirecionamento em sua URL.				
Hipóteses	Existe um interesse por parte dos mal-intencionados em informar suas URL maliciosas em parâmetros de uma URL legitima, fazendo parecer que a URL maliciosa seja confiável devido a URL principal proporcionar essa confiança.				
Amostras	1.1 e 2.1	**Relevance**	WEAK	**Relations**	C26
Extração	Obter a URL com outra URL em seu path ou parâmetros *querystring*.				
Limitações	-				
Observações	Em certos casos, a aplicação da URL principal não faz tratamentos, portanto, a mesma irá redirecionar o usuário-final para a URL maliciosa informada para redirecionamento, seja como *querystring* ou *path*.				
Análise	Foi identificado um considerável número de URL dessa natureza, quase 5% .				

confiando. Contudo, a mesma irá redirecionar o usuário para a página que o fraudador informou nos parâmetros GET, representando assim um perigo ao usuário final. Os dados extraídos desse tipo de ataque estão ilustrados na Figura 80 e a análise GQM é descrita na Tabela 37.

Figura 80 – C29 - ocorrências

Conforme ilustrado na Figura 80, tanto para válidos como inválidos, existe um discreto número de *phishing* com redirecionamento, evidenciando que as páginas legitimas comumente utilizam, quase que na mesma proporção das páginas fraudes, a prática de redirecionar através de um parâmetro da URL, seja armazenado no *path* ou *querystring*. Diante disso, a característica ficou avaliada com relevância *WEAK*.

A.0.6.7 C30. URL spoofing

Essa característica avalia os casos em que o fraudador faz uso de trocadilhos, que podem ser palavras com erros na grafia, parônimos, homógrafos, homônimos, entre outros, e muitas vezes associados a jogos de palavras para um olhar desatento do usuário final. Importante salientar que essa categoria considera o conteúdo da página e todos elementos da URL, contudo, ataques de *punycode* foram isolados na categoria C24 por serem muito específicos, estas possuem o

Tabela 38 – GQM de C30. URL spoofing

Objetivo 6	Analisar padrões adotados pelo ataque para minimizar suspeitas do usuário final.				
Questão	C30. Quais registros fazem jogos de palavras para aparentar o serviço mais fidedigno?				
Métricas	[M85]: Contabilizar os registros de *phishing* válidos que utilizam trocadilhos de serviços em C15.				
Hipóteses	Eventualmente, para simular maior fidedignidade, uma URL maliciosa pode fazer uso de palavras combinadas para que a mesma apresente-se confiável para um usuário desatento.				
Amostras	1.1	**Relevance**	STRONG	**Relations**	C13, C15, C16, C20
Extração	Obter a parte 1 e 2 da URL e analisar suas partes.				
Limitações	Não foi necessário realizar análise comparativa entre *phishing* válidos e inválidos.				
Observações	Palavras que trocam a letra "O" por "0" (faceb00k) ou demais caracteres que são similares.				
Análise	Foram encontrados alguns registros que descrevem o uso do jogo de palavras. Os serviços de redes sociais são os mais explorados nesse cerne.				

escopo limitado ao domínio da URL e ao comportamento do navegador. Os dados extraídos estão na Figura 81 e a análise GQM na Tabela 38.

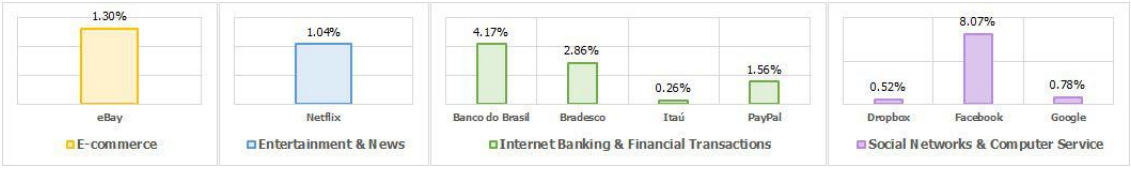

Figura 81 – C30 - ocorrências

A Figura 81 descreve marcas exploradas com spoofing, ou seja, ocorrências na URL de palavras "faceb00k", "Netfliiix" ou "dr0pbox", prática conhecida como *typosquatting* (STOUT; MCDOWELL, 2012). No caso do Facebook, foram detectados 6 trocadilhos, totalizando 8.07% da Amostra # 1.1. Não obstante, destaque para os bancos brasileiros, em especial ao Banco do Brasil com 4.17% de todo o montante mencionado. Diante disso, a característica foi considerada com relevância *ALTA*. Não obstante, outro comportamento interessante é a considerável quantidade de domínios que são registrados propositalmente similares a uma marca famosa, porém, criados por terceiros, prática denominada *cybersquatting* (STOUT; MCDOWELL, 2012).

A ação nem sempre é criminosa, já que não é de hoje que costumam realizar registros de domínios com alusão a marcas famosas para posteriormente revender ao respectivo grupo representante, contudo, não deixa de ser uma oportunidade de exploração para os mal intencionados. Um postura interessante adotada por algumas empresas, como Facebook e Netflix, foi a apropriação de domínios com *typosquatting* com propósito de redirecionar para a página correta, a exemplo de "facbook.com", "fcebook.com", "Netfiix.com", "Netflix.com" ou "Netfliix.com", no intuito de proteger seus usuários.

Outra consideração importante de ser mencionada é a escolha da amostra. Considerando que a amostra #3 já possui o campo "target_brand", seria mais conveniente se basear nesse campo para realizar a análise, contudo, a característica em questão observa casos de *typosquatting* na URL e também casos de *cybersquatting* em domínios registrados. Sendo assim, foi necessária uma análise subjetiva e manual através da amostra #1.1. Analisar a "marca mais explorada" foi um objetivo apresentado anteriormente na característica C15.

APÊNDICE B – ARTEFATOS

Tabela 39 – Fontes do projeto e experimento

Descrição	**Origem**	**Link**
Sistema especialista	Github	https://github.com/cmrevoredo/piracema.io
Serviço de registros de entradas	Github	https://github.com/cmrevoredo/api.piracema.io
Portal piracema.io	Github	https://github.com/cmrevoredo/portal.piracema.io
Extensão para Chrome	Dropbox	https://www.dropbox.com/sh/jxi439vsfzvoyi0/AACdf8_M3qLekkDE-TIixg2Xa?dl=0
Amostras extraídas do *PhishTank* e *OpenPhish*	Dropbox	https://www.dropbox.com/s/b3qd1kus21fhh1c/samples.7z?dl=0
Instâncias para o treino da aprendizagem de máquina	Dropbox	https://www.dropbox.com/s/7hguc3ybpdwiv50/sample-fit.csv?dl=0

Tabela 40 – Hosts e TLDs considerados na Greylist

HOST	**TLD**
000webhostapp.com	.tk
webcindario.com	.ml
sharepoint.com	.ga
wixsite.com	.cf
myfreesites.net	.gq
blogspot.com	.nom.za
beget.tech	.tt
qponn.net	.2ya.com
hol.es	.vze.com
drive.google.com	1sta.com
vineyard-garden.com	24ex.com
godaddysites.com	
oreo-e-assistance.com	
1drv.com	
umbler.com	
igotrip.info	
sym-global.com	
stcroixlofts.com	
day-giftcard.com	
wefbee.com	
greendatainfo.com	
kylelierman.com	
dermrefresh.com	
bayandtools.com	
munozbr.com	
avaksystems.com	

Tabela 41 – Exemplos de palavras-chave na indexação por *Cybersquatting* e *Typosquatting*

Paypal	**Facebook**	**Google**	**Apple**	**Bradesco**	**Banco do Brasil**	**Dropbox**	**Microsoft**	**Caixa**	**Twitter**
p4ypal	faceb00k	g00gle	icloud	brades	bb	dr0pb0x	azure	fgts	tweet
p4yp4l	facebo0k	go0gle	4pple	br4desco	bancobrasil	dropb0x	office365	previdencia	tw33t
payp4l	faceb0ok	g0ogle	4ppl3	bradesc0	pbb	dr0pbox	msdn	-	-
-	f4ceb00k	g00gl3	-	br4des	Gerenciador Financeiro	-	skydrive	-	-
-	f4cebo0k	go0gl3	-	brad3s	b4nc0br4sil	-	-	-	-
-	f4ceb0ok	g0ogl3	-	br4d3s	banc0br4sil	-	-	-	-
-	f4c3b00k	drive	-	bradesfacil	b4nc0brasil	-	-	-	-
-	f4c3bo0k	gmail	-	-	obb	-	-	-	-
-	f4c3b0ok	-	-	-	-	-	-	-	-

Tabela 42 – Exemplo de entradas registradas no serviço

Listagem no formato JSON

[“asset”:“id”: “5ec3f6153afff8b8fb8ab744fc87251f99c508fe5c8542b0e99768f5acd01c3a”, “owner”:“id”: “aca7f9b31bbf2bed5010b53c8e07b7a5ceacf58bac0725a9b90ef73fd9b7465a”, “name”: “bradesco”,“token”: “banco.bradesco”, “type”: “host”, “user”:“id”: “3b320d6529c71be3b81c3e431ad82d779f48d1a48dd41fd519d11f60c5e5794c”, “login”: “revoredo”,“id”: “3bb486e1b2b593a24b1f3817942d4f11d26828dab96d2e0f6e1838075e7fb2c7”, “asset”:“id”: “f506a02e41f222495fbe330d6656466763433ea075fe23badb9485a59c3d6e4d”, “owner”:“id”: “aca7f9b31bbf2bed5010b53c8e07b7a5ceacf58bac0725a9b90ef73fd9b7465a”, “name”: “bradesco”,“token”: “237923720592005595985190033232017000000000000000”, “type”: “billet”, “user”:“id”: “3b320d6529c71be3b81c3e431ad82d779f48d1a48dd41fd519d11f60c5e5794c”, “login”: “revoredo”,“id”: “b1033b891cac6931dcfe2ead214e8843559e572665da4e488a283800ee8a1796”, “asset”:“id”: “0d0b02e178dd369b66e179abef9ecf797ffa22173e7d1c84e77f2b8df3e97a2f”, “owner”:“id”: “814deae4996089a90e3035636fee0ec93de47ccfa3f9f8d11c747bbffa4e3cb0”, “name”: “google”,“token”: “google.com”, “type”: “host”, “user”:“id”: “3b320d6529c71be3b81c3e431ad82d779f48d1a48dd41fd519d11f60c5e5794c”, “login”: “revoredo”,“id”: “b23a2a485e734a30d5d794e0ba0dd39cb8d361497c3d2100041be7531921c983”, “asset”:“id”: “aea5e28418e1e0e3417836d1c7e5dad95a16f519051583b85db141d63a4c40fe”, “owner”:“id”: “298c2d2231d1c8e93afcfd0765b2dbfa2313ee52ea856b757d243b9df60c513e”, “name”: “bancodobrasil”,“token”: “www2.bancobrasil.com.br”, “type”: “host”, “user”:“id”: “3b320d6529c71be3b81c3e431ad82d779f48d1a48dd41fd519d11f60c5e5794c”, “login”: “revoredo”,“id”: “39bcbe3bac3b98b79455a3fb926ea72ce99fb46c6c3e0aaa2e7df0f575d57e55”, “asset”:“id”: “65a4c6a544d4f0d0780410cd7f6b561957a6402f2b5bf14f9192883c9d1d2331”, “owner”:“id”: “814deae4996089a90e3035636fee0ec93de47ccfa3f9f8d11c747bbffa4e3cb0”, “name”: “google”,“token”: “accounts.google.com”, “type”: “host”, “user”:“id”: “3b320d6529c71be3b81c3e431ad82d779f48d1a48dd41fd519d11f60c5e5794c”, “login”: “revoredo”,“id”: “ad16ec3ab7c290565b569a9d7948a83abcb562a24fca779e3b6423f2cd110010”]

Tabela 43 – Exemplo de requisições para validar as entradas

Objetivo	**Requisição**	**Resposta**
Requisição para validar a entrada com host “banco.bradesco” e marca “bradesco”.	http://200.196.181.164/api.piracema.io/entries/check/bradesco/banco.bradesco	“owner”:“name”: “bradesco”, “type”: “host”, “uri”: “banco.bradesco” (entrada válida)
Requisição para validar a entrada com host “santander.com.br” e marca “santander”.	http://200.196.181.164/api.piracema.io/entries/check/santander/satander.com.br	"message": “io.piracema.api.msg.error.entry.not.found”, “statusCode”: 403 (entrada inválida)