Esteno[*]: Uma Abordagem para Detecção Visual de *Bankers*

Victor F. Martins[1], André R. A. Grégio[1,2], Vitor M. Afonso[1], Paulo Lício de Geus[1]

[1]Universidade Estadual de Campinas (Unicamp) – Campinas – SP – Brasil

[2]Centro de Tecnologia da Informação Renato Archer (CTI)– Campinas – SP – Brasil

{furuse, vitor, paulo}@las.ic.unicamp.br, andre.gregio@cti.gov.br

Abstract. *Bankers—Internet Banking information stealer programs—usually present windows that mimic legitimate bank sites to lure users into providing sensitive information. In addition, bankers may run on the target operating system in a non-intrusive mode, making the detection and analysis provided by unsupervised, automated analysis systems difficult. In this paper, we propose a solution for the identification of Brazilian bankers. To this end, we leverage three visual analyzers (based on color properties, known logotype presence and textual patterns) that are tuned using a supervised machine learning technique (Random Forest). We tested our approach on over 1,100 unknown binaries´ images, yielding 92.1% of correctly classified samples.*

Resumo. *Bankers—programas maliciosos para roubo de informações bancárias—geralmente usam janelas que imitam sites dos bancos reais para ludibriar os usuários. Eles podem atuar de maneira não intrusiva no sistema alvo, o que dificulta a detecção e análise por sistemas automáticos não supervisionados. Neste artigo, apresenta-se uma proposta de solução para a identificação de bankers brasileiros. Para tanto, aplica-se três analisadores visuais (cores, presença de logotipos de banco e conteúdo de textos) refinados usando aprendizado de máquina supervisionado (Random Forest). Testes com mais de 1.100 imagens extraídas de binários desconhecidos resultaram em 92,1% de exemplares corretamente classificados.*

1. Introdução

A sociedade atual tem migrado para o espaço virtual, do comércio aos relacionamentos interpessoais. Consequentemente, tem sido crescente a utilização de *Internet Banking* para a realização de transações financeiras, como pagamentos e transferências. Com isso, aumentou também a motivação dos atacantes para roubar credenciais bancárias utilizadas pelos clientes no acesso via Internet. Logo, surgiram programas maliciosos cujo principal objetivo é ludibriar os usuários visando obter suas credenciais. Este tipo de *malware* ficou conhecido como *crimeware*, *information stealer*, *phishing Trojan*, *banking Trojan* ou simplesmente *banker* [Corporation 2007], no caso mais específico que é o foco deste artigo.

Bankers geralmente aplicam técnicas de engenharia social para levar um usuário a fornecer os dados de acesso a sua conta bancária na Internet, tais como a agência, conta corrente, nome de *login* e senha, número do cartão de crédito ou débito e valores de tabelas de senhas ou *tokens* de segurança. As informações coletadas são então enviadas ao atacante, podendo ser vendidas ou utilizadas para o pagamento de contas e compras não autorizadas ou, ainda, podendo ocorrer a transferência de dinheiro da vítima para contas de terceiros. Existem *bankers* em todos os países que adotam

[*]*Esteno* é uma Górgona da mitologia grega, irmã de Medusa.

plataformas de *Internet Banking*, porém, no Brasil, os ataques e prejuízos atingem cifras alarmantes, devido ao avanço deste tipo de tecnologia no país em decorrência das peculiaridades da situação financeira do passado (por exemplo, a inflação desenfreada).

Embora existam diversos *bankers* internacionalmente disseminados, como *Zeus* [Binsalleehet al. 2010] e *SpyEye* [Coogan 2010] – cujo roubo de credenciais bancárias de usuários está associado à modificação de arquivos e bibliotecas do sistema ou a injeção de código malicioso em processos sem apresentar referências aos bancos – os *bankers* que atingem o Brasil operam de maneira diferente. No Brasil, *bankers* costumam infectar o usuário através de *links* ou anexos em mensagens de e-mail forjadas com o remetente do banco (*phishing*), solicitando a atualização de mecanismos de segurança ou de cadastro. Para isto, os desenvolvedores de *bankers* precisam fazer o usuário crer que está realmente atendendo a um pedido de seu banco, utilizando para tanto imagens, textos, *layout* e logotipos muito próximos ou retirados do *site* original. Este tipo de *banker* tem sido visto frequentemente no ciberespaço brasileiro. Em 2006, um estudo da empresa F-Secure estimou em 30.7% a quantidade de *bankers* observados tendo por alvo bancos brasileiros [Corporation 2007]. Neste estudo, dividiu-se os *bankers* em "brasileiros" e "demais", nos quais os demais referiam-se principalmente a bancos na América do Norte, Austrália e Europa. Em 2012, um estudo da empresa Kaspersky apontou o Brasil como o país mais afetado por *bankers* [Kaspersky 2012].

Os *bankers* brasileiros atuam principalmente enganando o usuário por meio da apresentação de telas do banco alvo, aguardando assim a entrada de informações sensíveis. Isto dificulta a detecção desses *bankers*, dado que eles nada mais são do que programas com interfaces gráficas que não se baseiam em técnicas intrusivas, mas trazem embutidas em seu binário (ou realizam o *download* de) imagens obtidas dos bancos e as carregam em memória sem efetuar ações comprometedoras no registro ou no sistema de arquivos. Além disso, o fato de haver vários bancos com padrões diferentes de autenticação, necessidades de interações diversas e mecanismos de segurança distintos, faz com que a análise dinâmica em *sandboxes* (ambiente controlado passível de restauração) também seja dificultada.

Portanto, faz-se necessário o desenvolvimento de técnicas que auxiliem a análise automática e não supervisionada de *bankers*, visando aumentar sua taxa de detecção. Para isso, propõe-se a ferramenta *Esteno*, que lança mão de métodos de aprendizagem de máquina e de reconhecimento visual e textual a fim de identificar logotipos e padrões de texto relacionados a bancos brasileiros em imagens obtidas durante a execução de códigos maliciosos. Vale ressaltar que o escopo da ferramenta está restrito a análise dinâmica usando *sandbox*, logo a mesma não tem como finalidade ser usada no computador de um usuário final, como no caso dos programas antivírus, mas sim por grupos de resposta a incidentes.

Como contribuições principais deste artigo pode-se citar (i) a discussão do modo de operação dos *bankers* brasileiros, (ii) a proposta de aplicação de técnicas de análise visual e reconhecimento de textos para identificação de *bankers* e (iii) o desenvolvimento de um protótipo para testar e validar as técnicas propostas em exemplares reais vistos em atividade, mostrando resultados promissores com base em uma alta taxa de detecção. O restante do artigo está organizado como segue. Na Seção 2, o conceito de *banker* é brevemente explicado. Na Seção 3, apresenta-se a solução proposta (*Esteno*). Na Seção 4, as etapas e métodos de análise de imagens para identificar *bankers* são detalhadas. Na Seção 5, apresentam-se os testes e resultados obtidos com a solução proposta. A Seção 6 contém uma revisão da literatura associada à detecção de *bankers*. Na Seção 7, são feitas as considerações finais sobre o trabalho.

2. *Bankers Brasileiros: Operação e Apresentação*

Um *banker* pode atuar de diversas maneiras, desde solicitar as informações por *e-mail* ou site clonado do banco alvo, até o *download* e execução de um programa malicioso na máquina do usuário. Este último vetor de atuação visa fazer o usuário acreditar que está acessando o ambiente real de *Internet Banking* por meio da simulação desses próprios sistemas. Assim, tais *bankers* são de difícil identificação automática por sistemas de análise dinâmica, pois além de sua principal estratégia ser a de enganar o usuário por meio de imagens e telas idênticas as dos bancos-alvo, eles possuem rotinas de execução curtas e apresentam baixa interação com o sistema operacional. Isto ocorre porque esses *bankers* consistem de apenas algumas telas com formulários a serem preenchidos com os dados bancários (agência, conta, senha de *Internet Banking*, valores da tabela de senhas etc.), os quais são enviados para o atacante via conexão com a rede.

2.1 Rotina Típica de um *Banker* Brasileiro

No Brasil, as soluções de segurança para *Internet Banking* incluem alguns fatores adicionais de autenticação, além de uma senha própria para o acesso via Internet, diferente da senha do cartão do cliente. Os fatores adicionais de autenticação são requeridos pelo banco para autorizar uma transação via Internet, como uma transferência ou um pagamento a ser efetuado.

Portanto, uma das motivações dos atacantes é capturar informações acerca dos fatores de autenticação para efetuar transações com a conta da vítima. Exemplos de fatores adicionais de autenticação são os *tokens* físicos – dispositivos de *hardware* que armazenam um certificado digital ou proveem senhas para uso único (OTP – *One-Time Password*) trocadas a cada minuto – e as tabelas de senhas – cartões com posições indexadas contendo valores que servem de senha para uma dada sessão, mas que podem ser reutilizadas em uma próxima transação caso seu índice seja requisitado. Para ilustrar o modo de atuação geral de um *banker* brasileiro, mostra-se o fluxo comum de uma rotina de execução possível na Figura 1.

No primeiro estágio, "Início", o *malware* é executado como um programa qualquer do sistema operacional, em geral sem consumir recursos excessivos. No segundo estágio é mostrada a tela inicial do banco e é apresentado algum motivo para o usuário informar seus dados. O exemplo comum é alertar o usuário sobre a necessidade urgente de atualização do mecanismo de segurança do banco. Solicita-se então a agência e a conta corrente. Esta talvez seja a parte mais importante do golpe, portanto não deve despertar suspeita, pois é a partir desta tela que o usuário será enganado ou perceberá a fraude. Devido a isto, os desenvolvedores de *bankers* em geral se utilizam de imagens originais obtidas diretamente de *sites* de *Internet Banking*.

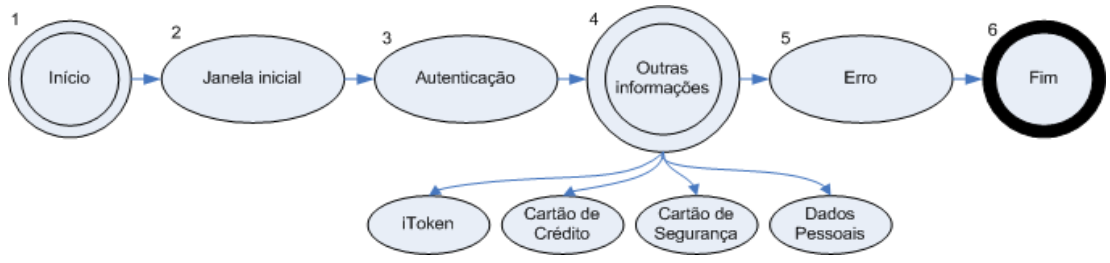

Figura 1: Rotina típica de um *banker*.

Nas janelas seguintes, as demais informações são solicitadas, sendo em geral enviadas (via métodos HTTP POST ou *e-mail*) para o atacante a cada mudança de estágio. Podem ser pedidas informações pessoais, como RG e CPF sob a alegação de atualização cadastral. Finalmente, é simulado o processamento das informações ou atualização dos mecanismos de segurança, o qual culmina em uma mensagem de erro.

Todas as janelas apresentadas durante a execução buscam possuir aparência igual ao dos sistemas legítimos dos bancos alvos, ou seja, utilizam o mesmo *brand* – logo, cores, textura, fontes e outras características que diferenciem uma marca – de forma a remeter à marca do banco, pois este é o principal fator de sucesso para conquistar a confiança do usuário sobre a legitimidade do que está sendo pedido.

Além disso, a interação do *banker* com o sistema operacional é muito semelhante a de um programa não malicioso, dado que são abertas apenas algumas janelas, o usuário preenche alguns formulários e, no final, as informações são enviadas por HTTP. Desta forma, não há nenhuma ação suspeita que possa ser utilizada facilmente para caracterizá-lo como um *malware*, somente o conteúdo das janelas e as informações pedidas. Isto é o fator de maior entrave para que a identificação dos *bankers* seja feita de modo automático por sistemas de análise dinâmica não-supervisionados e até mesmo por mecanismos antivírus.

Por fim, os textos mostrados nas telas sempre buscam usar as mesmas imagens, posicionamento de logotipos e produtos dos bancos alvos, além de termos que remetem a segurança e proteção, para dar maior credibilidade ao golpe.

3. Solução Proposta

Dadas as características levantadas sobre o comportamento apresentado por *bankers* brasileiros, fica claro que o fator mais relevante de seu sucesso são as imagens e telas mostradas às vítimas, pois são elas que persuadem o usuário a crer que está no sistema real do banco. Desta forma, a melhor maneira de identificar este tipo de *malware*, em sistemas automatizados e não-supervisionados, é analisando justamente essas imagens. Para tanto, propõe-se o *Esteno*, uma solução para a classificação automatizada de imagens e telas de *bankers*. Essas imagens são obtidas durante a execução automatizada de códigos maliciosos em um sistema de análise dinâmica de *malware* (*sandbox*), ambiente almejado para a aplicação da ferramenta.

3.1 Análise das Imagens: Logotipo, Conteúdo dos Textos e Cores

As janelas apresentadas por programas em execução nada mais são do que uma composição de vários elementos, os quais são apresentados visualmente para o usuário. No caso das janelas de *bankers*, os elementos que mais se destacam são: o logotipo do banco, o conteúdo dos textos e as cores usadas, principalmente na parte superior das imagens, que é onde se localiza o *banner* ou o cabeçalho do *site* de *Internet Banking*. Destes elementos, o logotipo é extremamente relevante, pois é o símbolo que representa o banco do cliente. Devido a isso, bem como para passar uma sensação de legitimidade e fazer com que o usuário se identifique, raramente o logotipo não está presente em uma janela apresentada por um *banker*.

Já os textos contidos nas janelas dos *bankers* costumam possuir conteúdo que remeta a segurança e proteção, como mencionado anteriormente, além de usar termos próprios e familiares de cada banco tido como alvo. Exemplos para ilustrar a afirmação anterior são os termos: "Superlinha" (Santander), "*iToken*" e o *slogan* "30 Horas" (Itaú). Com isso em mente, é possível passar as imagens por uma ferramenta de OCR (*Optical Character Recognition*) – mecanismo para converter textos presentes em imagens novamente em texto editável – para extrair partes dos textos e, posteriormente, buscar-se por termos e padrões referentes a bancos.

Quanto às cores, elas são uma das principais características representativas de uma marca (*brand*). No caso dos bancos, percebe-se que há simplicidade, pois apenas uma cor majoritária é utilizada, sendo que outra cor secundária é escolhida para a composição das bordas e *banner* das telas. Dessa forma, a identificação de uma marca

pode ser alcançada por meio do estudo da distribuição estatística das cores vermelho, verde e azul que compõem o espaço de cores RGB.

3.2 Arquitetura da Solução

A ferramenta *Esteno* foi desenvolvida na linguagem Java, por haver bibliotecas prontas capazes de codificar os requisitos delineados. Houve também o uso de ferramentas externas presentes em distribuições Linux, para facilitar alguns trabalhos, como o processamento das cores e sua distribuição.

Na Figura 2, mostra-se o esquema da arquitetura proposto para a ferramenta *Esteno*, na qual a entrada é uma imagem e a saída é a sua classificação. Internamente, existe a divisão em duas partes, a primeira abrangendo os três analisadores que extraem as características relacionadas à presença do logotipo, ao conteúdo dos textos e à estatística das cores, enquanto que a segunda é a aplicação da aprendizagem de máquina, com o algoritmo *Random Forest (RF)* (maiores detalhes na seção 5).

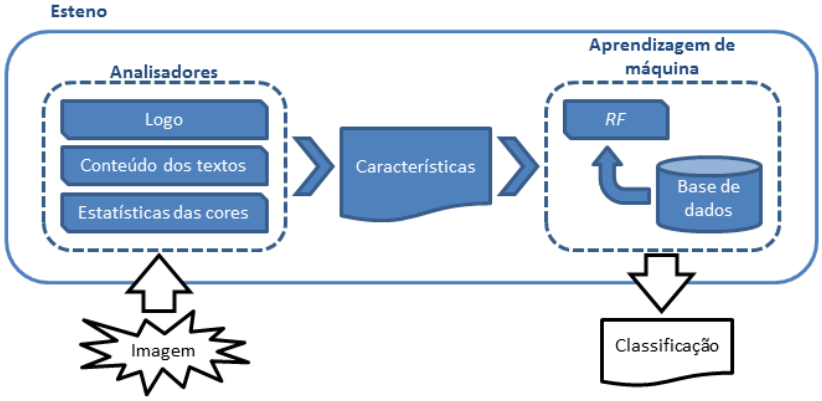

Figura 2: Arquitetura proposta para o *Esteno*.

3.3. Obtenção das Imagens

As imagens usadas no *Esteno* foram obtidas de duas maneiras: primeiramente, executou-se diversos exemplares, tirando *screenshots* das janelas ativas e associando-os ao exemplar executado. A segunda maneira deu-se pela extração de figuras encontradas embutidas dentro dos executáveis maliciosos. Para extrair as imagens presentes dentro da estrutura de dados de um arquivo binário não é necessário executá-lo, bastando-se inspecionar seu código com uma ferramenta de análise forense, como *Foremost* [for 2013]. As imagens pequenas, menores que os logotipos usados como referência, foram redimensionadas devido a limitações do algoritmo para analisa-los. Dessa forma, obteve-se uma ampla base de imagens que foram usadas no *Esteno*.

4. Classificação Visual: Técnicas Utilizadas e Detalhes de Implementação

Nesta seção serão apresentadas com maiores detalhes as técnicas utilizadas para classificação de *bankers* e como elas foram aplicadas na ferramenta *Esteno*, assim como alguns resultados individuais de cada um dos analisadores visuais implementados.

4.1 Preparação dos Analisadores

O *brand* de uma empresa, por definição, deve ser único e marcante, portanto é necessário avaliar individualmente cada banco, a fim de preparar adequadamente cada analisador, em específico, para os bancos esperados. Foi identificado que os exemplares coletados tinham por alvo os bancos: Banco do Brasil, Bradesco, Caixa Econômica Federal, Itaú e Santander; inclusive estes são os cinco maiores bancos comerciais brasileiros segundo o Banco Central do Brasil (BCB). Assim, os analisadores foram preparados com os logotipos e termos bancários utilizados por eles.

4.2 Analisador de Logotipos

Para procurar pela presença de logotipos conhecidos nas imagens, foi utilizada a biblioteca *JavaCV* [jav 2013], que encapsula uma série de bibliotecas comumente usadas em problemas de visão computacional, como é o caso da biblioteca *OpenCV*.

Na biblioteca *JavaCV* existe a classe `ObjectFinder`, uma adaptação da classe `find_obj` presente na biblioteca *OpenCV*, a qual exemplifica como buscar um objeto em uma imagem utilizando o algoritmo *Speed-Up Robust Features*, também conhecido como SURF [Bay al. 2006]. Este algoritmo é ágil e robusto para detectar características e pontos de interesse em uma imagem, independente de rotação ou escala. A classe `ObjectFinder` tem como parâmetros de entrada duas imagens, o logotipo e a imagem em que este será procurado. Inicialmente, o logotipo é processado a fim de se definir seus pontos característicos, como pode-se observar na Figura 3, representados pelos círculos vermelhos. Posteriormente, estes pontos são procurados na imagem e, se forem encontrados em uma região que condiz com o logotipo (em termos de distância e relação na posição entre os pontos), a região é delimitada por uma linha, indicando o local em que foi encontrado o objeto (logotipo).

Figura 3: Pontos característicos de três logotipos de bancos distintos.

Na Figura 4, pode-se ver três casos de funcionamento do `ObjectFinder`. No primeiro (a), foram localizados inúmeros pontos e a região do logotipo foi marcada corretamente. Cada risco indica um ponto característico encontrado e onde está presente na imagem. No segundo caso (b), foram encontrados 6 pontos e considerou-se equivocadamente que o logo está inclinado, como observamos pela delimitação, todavia a região foi marcada corretamente. Por fim, no último caso (c), foram encontrados apenas três pontos corretos, mas insuficientes para localizar o logotipo na imagem.

O fato mais importante que a Figura 4 nos permite concluir é que, independente da região delimitada, a quantidade de pontos característicos encontrados na imagem está diretamente relacionado à chance de se encontrar o logotipo na imagem. Assim, a classe `ObjectFinder` foi modificada para indicar apenas a quantidade de pontos encontrados. Cabe ressaltar que os logotipos utilizados como referência na ferramenta *Esteno* foram retirados dos *sites* originais dos bancos.

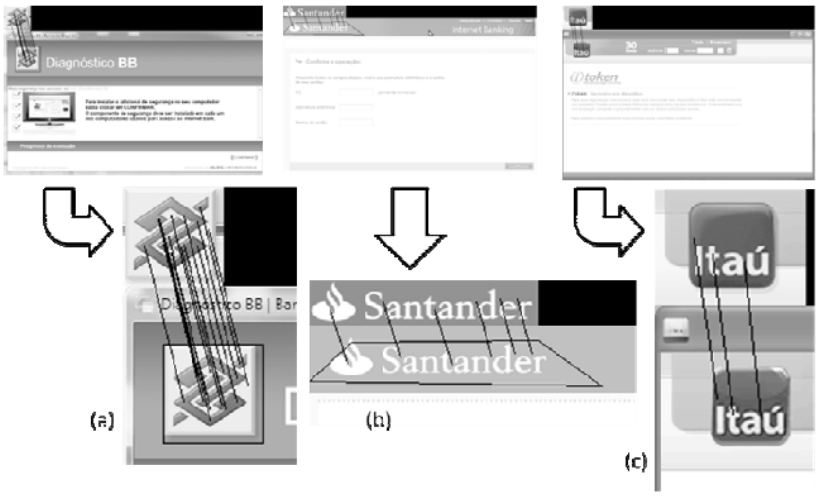

Figura 4: Três exemplos do funcionamento do `ObjectFinder`.

4.3 Analisador de Conteúdo dos Textos

Os textos presentes nas imagens foram convertidos em arquivos editáveis através da ferramenta de OCR *Tesseract* [Holley 2009] e, posteriormente, analisados quanto aos seus conteúdos. A ferramenta *Tesseract* nem sempre consegue mapear corretamente os *pixels* de uma imagem, podendo traduzir erroneamente as palavras encontradas, por não identificar ou misturar as letras. Porém, em geral observou-se que existe uma semelhança da palavra produzida com a palavra correta. Assim, para analisar o conteúdo de forma mais robusta, a qual seja capaz de tratar algumas das falhas mencionadas da *Tesseract*, foram criados padrões de expressões regulares, responsáveis por verificar a existência de nomes de bancos, produtos e termos bancários. Na Figura 5 pode-se ver alguns exemplos de expressões criadas para este fim.

```
Itaú:[EiIÍn@l][ÏtTricl][Aax][l]*[Úùúu\]]
Agência:[aA][gGql3]+[èEêÊée][nNmrv]*[cCxuUnmzl(][iIxzl]*[mnAaãà]
Banco:[BbE]an[ct][ou]
Proteção:[pP][mr][ocu]*[tîl]e[çcgq][aãäāléi][uno]
Seguro:[Ssß][eêæa][gqu]u[rn,v]*[nàäaãõowzum]
```

Figura 5: Exemplos de expressões regulares utilizadas no *Esteno*.

As expressões regulares foram divididas em três grupos, sendo que cada um recebeu uma pontuação definida empiricamente após sucessivos testes. O primeiro grupo é composto pelos nomes de bancos e recebeu a pontuação "3", o segundo é formado pelos nomes de produtos e termos bancários, recebendo "2" pontos e, por último, o grupo das palavras relevantes e frequentes em *bankers*, mas que podem ser encontradas em outros programas, recebeu a pontuação "1". Na Tabela 1, apresenta-se a lista dos grupos e das palavras utilizadas no *Esteno*.

Tabela 1: Lista de grupos e palavras utilizadas como expressões regulares no *Esteno*.

Nome dos bancos (3 pontos)	Nome de produtos e termos bancários (2 pontos)	Termos relevantes e freqüentes (1 ponto)
Banco do Brasil, Bradesco, Caixa, Itaú e Santander	Superlinha, 30 Horas, iToken, Banking, Bankline, Teclado Virtual, Banco, Agência, Cartão de Débito, Cartão de Crédito e Conta Corrente	Segurança, Seguro, Proteção, Senha, Validação, Validando, Proteger, Conta e Chave

4.4 Analisador da Distribuição das Cores

No *banner* de uma tela (porção superior) há muitos elementos do *brand* representado, por exemplo, as cores, que no caso dos bancos são formadas por uma majoritária e outra secundária, conforme mencionado anteriormente. Assim, ao se analisar a distribuição das cores no espaço RGB, a cor dominante fica em destaque e contribui para a identificação de um *banker*.

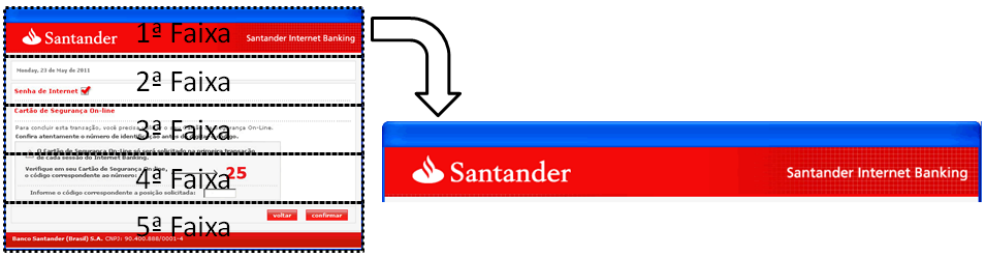

Figura 6: Divisão de faixas nas imagens para separação do *banner*.

No *Esteno*, as imagens de entrada são divididas igualmente em cinco faixas horizontais, conforme (Figura 6). A faixa mais superior, onde se localiza o *banner*, é

separada à parte para uma análise mais aprofundada. Nesta faixa, calcula-se a distribuição das cores com o pacote de ferramentas *ImageMagick* [ima 2013].

4.5 Aprendizagem de Máquina

No *Esteno* foram definidas 27 características a serem utilizadas pela aprendizagem de máquina: seis para lidar com a presença dos logotipos (uma para cada banco e uma média), três para o conteúdo dos textos e seis para cada cor do espaço de cores RGB (somando 18). Na Tabela 2 podem-se ver todas as características adotadas.

Tabela 2: Características adotadas como atributos no *Esteno*.

Presença de logo	Conteúdo dos textos	Características estatísticas da distribuição das cores (RGB)
- Quantidade de pontos característicos: Banco do Brasil, Bradesco, Caixa Econômica Federal, Itaú e Santander. - Média dos pontos encontrados	- Quantidade de expressões regulares encontradas. - Soma da pontuação de todas as expressões encontradas. - Média de pontuação por expressão regular encontrada.	- *Red*, *Green* e *Blue*: mínimo, máximo, média, desvio padrão, curtose e assimetria.

Assim, os algoritmos *k*-NN, SVM (utilizando *kernel* Linear e RBF) e *Random Forest (RF)* do *framework* Weka [Hall et al. 2009] foram aplicados ao conjunto de testes. Estes algoritmos tiveram seus hiper parâmetros ajustados segundo a Tabela 3, com o auxílio da plataforma R Project (`www.r-project.org/`).

Tabela 3: Ajuste dos hiper parâmetros

Algoritmo	Hiper parâmetros
k-NN	k = {1, 3, 5, 11, 21, 31}
SVM Linear	C de 1e-3 a 1e4, em múltiplos de 10
SVM RBF	C e gamma de 1e-3 a 1e4, em múltiplos de 10
Random Forest (RF)	mtry = {2, 3, 5, 10, 20, 40, 60}

O *k*-NN realiza a classificação de objetos com base nos elementos de treinamentos com menor distância (calculada por uma função, por exemplo, a distância Euclidiana) no espaço de atributos [Cover and Hart 1967]. Em comparação, o SVM busca encontrar, estatisticamente, o hiperplano que melhor separe geometricamente os conjuntos [Platt 1988]. Por fim, a RF utiliza a técnica de *bagging* para combinar árvores preditoras [Liaw and Wiener 2002].

5. Testes e Resultados

Para validação da ferramenta *Esteno*, utilizou-se uma base com 1.394 exemplares† extraídos de *links* e anexos provenientes de mensagens de *phishing*, coletados entre os anos de 2010 e 2013. Os exemplares passaram pelo processo de extração de imagem explicado anteriormente, resultando em um total de 1.122 imagens.

As imagens desta base foram classificadas em dois grupos: *bankers* e *others*. O agrupamento se deu de acordo com a proposta da ferramenta, isto é, distinguir as imagens que são de *bankers* das restantes. A classe dos *bankers* teve 572 instâncias, enquanto que a classe *others*, 550. A partir destes dados, foram feitos testes no Weka, utilizando validação-cruzada (*cross-validation*) de 10-*folds* [Bengio and Grandvalet 2004] [Markatouet al. 2005], tanto para o cálculo da acurácia média, como para a escolha dos hiper parâmetros. Os melhores resultados, variando os hiper parâmetros listados na Tabela 3, dos algoritmos de aprendizagem de máquina supervisionada testados são apresentados na Tabela 4. O tempo gasto no treinamento dos algoritmos e

† Para a lista dos MD5 dos exemplares utilizados, favor contactar os autores.

na classificação de cada instância levou em média 4 minutos e menos de 1 segundo, respectivamente.

Tabela 4: Resultado dos algoritmos de aprendizagem de máquina

	k-NN	SVM Linear	SVM RBF	RF
Taxa de acerto (%)	91,5%	73,4%	91,5%	92,1%
Melhores hiper parâmetros	k = 5	C = 1	C = 1 e gamma = 10	mtry = 3

Com base na Tabela 4, o algoritmo *Random Forest* obteve o melhor resultado, gerando a classificação correta de 92,1% das instâncias, portanto foi o escolhido para compor o *Esteno*. O detalhamento do resultado obtido pela ferramenta está apresentado na matriz de confusão, típica para problemas de aprendizado de máquina, na Tabela 5.

Tabela 5: Matriz de confusão do resultado obtido pelo *Esteno*.

Classificado pelo *Esteno*			
Bankers	***Others***		
a. 537	b. 35	***Bankers***	**Classificado manualmente**
c. 54	d. 496	***Others***	

A Tabela 5 apresenta a classificação da base de imagens sobre duas perspectivas de classificação, a automatizada pela ferramenta *Esteno* (colunas) e a manual por um profissional (linhas) – revisadas por outras ferramentas, como VirusTotal (`www.virustotal.com`) e Anubis (`http://anubis.iseclab.org`). Para medir a qualidade dos resultados, observa-se na Tabela 6 as taxas de verdadeiro-positivo (VP), falso-positivo (FP), precisão, *recall* e média harmônica (*F-Measure*) – baseada na precisão e no *recall* – de cada classe definida (*Bankers* e *Others*).

Tabela 6: Valores que caracterizam a qualidade dos resultados obtidos no teste, por classe.

		VP	**FP**	**Precisão**	***Recall***	***F-Measure***
Classe	***Bankers***	93,9%	9,8%	90,5%	93,9%	92,2%
	Others	90,2%	6,1%	93,7%	90,2%	91,9%

6. Soluções existentes

Em [Buescheret al. 2011] os autores apresentam uma ferramenta de detecção de *bankers* que analisa *rootkits* de nível de usuário e detecta a instalação de *hooks* no Internet Explorer. Esses *hooks* são redirecionamentos no código que modificam o fluxo de execução do navegador para roubar informações do usuário. A ferramenta, chamada BankSafe, executa o *malware* em um ambiente controlado e utiliza assinaturas para verificar se foram feitas modificações na API usada pelo Internet Explorer. Os autores afirmam que a ferramenta possui uma taxa de detecção muito boa, mas está limitada à detecção de *malware* que utilizam *hooks*, o que não é comum no *banker* brasileiro.

Uma abordagem para detectar páginas de *phishing* na Internet é apresentada em [Medvetet al. 2008], onde os autores propõem um método de detecção visual que permite ao usuário saber de antemão se seus dados estão sendo interceptados por um atacante. A detecção se baseia em três atributos extraídos das páginas analisadas: a parte textual, as imagens e a aparência da página quando renderizada pelo navegador. Além disso, os autores propõem o uso de seu método em conjunto com outras ferramentas de detecção de *phishing* (AntiPhishand DOM AntiPhish). A sua limitação é detectar apenas tentativas de *phishing* baseadas no uso de páginas falsas carregadas pelo navegador, deixando de detectar aqueles que não utilizam o navegador para apresentar o ambiente bancário forjado. Este último caso é tratado pelo Esteno.

Outro método de detecção de *bankers* se baseia no tráfego de rede gerado pelo *malware*, ao contrário do *Esteno* que é visual. Em [Riecket al. 2010] os autores

apresentam Botzilla, uma ferramenta que utiliza assinaturas para detectar tráfego de rede característico de *malware*. O tráfego detectado está relacionado ao envio de informações para páginas controladas pelo atacante.

7. Considerações Finais

Ataques por *bankers* trazem muitos prejuízos aos usuários e instituições financeiras e, por isso, é importante a criação de medidas de proteção. Devido a natureza e modo de operação dos *bankers*, sistemas de análise dinâmica de *malware* utilizados por grupos de resposta a incidentes, podem ter dificuldades em identificá-los. Para auxiliar na detecção de *bankers* de maneira automatizada, propôs-se o *Esteno*, uma ferramenta que se utiliza de técnicas de detecção visual, classificação e identificação de padrões de texto em imagens extraídas de *malware*. Os resultados obtidos foram promissores, alcançando taxas de acerto de 92,1% na detecção de exemplares de *malware* com características de *bankers* brasileiros.

Referências

(2013). Foremost. http://foremost.sourceforge.net/.

(2013). Imagemagick.http://www.imagemagick.org/script/index.php.

(2013). Javacv.https://code.google.com/p/javacv/.

Bay, H., Tuytelaars, T., and Gool, L. V. (2006). Surf: Speeded up robust features. In *ECCV*.

Bengio, Y. and Grandvalet, Y. (2004). No unbiased estimator of the variance of K-fold cross-validation. In *J. Mach. Learn. Res.*, v. 5, pages 1089–1105. JMLR.org

Binsalleeh, H., Ormerod, T., Boukhtouta, A., Sinha, P., Youssef, A., Debbabi, M., and Wang, L. (2010).On the Analysis of the Zeus Botnet Crimeware Toolkit. In *Privacy Security and Trust (PST), 2010 8th Annual International Conference* on, pages 31–38.

Buescher, A., Leder, F., and Siebert, T. (2011).Banksafe information stealer detection inside the web browser. In *Proceedings of the 14th international conference on RecentAdvances in Intrusion Detection, RAID'11*, pages 262–280. Springer-Verlag.

Coogan, P. (2010). Spyeye bot versus zeus bot. http://www.symantec.com/connect/ blogs/spyeye-bot-versus-zeus-bot.

Corporation, F.-S. (2007). Thetrojan money spinner. Available at http://www.f-secure.com/weblog/archives/VB2007_TheTrojanMoneySpinner.pdf.

Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification.*Information Theory, IEEE Transactions* on, 13(1):21–27.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The weka data mining software: an update. *SIGKDD Explor.Newsl.*, 11(1):10–18.

Holley, R. (2009). How good can it get? Analysing and improving OCR accuracy in large scale historic newspaper digitisation programs. *D-Lib Magazine*, 15(3/4).

Kaspersky (2012). Number of the week: 780 new malicious programs designed to steal users' online banking data detected every day. http://www.kaspersky.com/about/news/virus/2012/Number_of _the_week_780_new_malicious_programs.

Liaw A. and Wiener M. (2002).*Classification and Regression by Random Forest.* In R News 2.

Markatou, M., Tian, H., Biswas, S., and Hripcsak, G. (2005). Analysis of variance of cross-validation estimators of the generalization error. *Journal of Machine LearningResearch*.

Medvet, E., Kirda, E., and Kruegel, C. (2008).Visual-similarity-based phishing detection. In *Proceedings of the 4th international conference on Security and privacy in communication netowrks*, SecureComm '08, pages 22:1–22:6, New York, NY, USA.

Platt, J. (1998). *Fast Training of Support Vector Machines using Sequential Minimal Optimization*. Advances in Kernel Methods - Support Vector Learning, B. Schoelkopf, C. Burges, and A. Smola, eds., MIT Press.

Rieck, K., Schwenk, G., Limmer, T., Holz, T., and Laskov, P. (2010).Botzilla: detecting the "phoning home"of malicious software. In *Proceedings of the 2010 ACM Symposium on Applied Computing*, SAC '10, pages 1978–1984, New York, NY, USA. ACM.