UNIVERSIDADE FEDERAL DO AMAZONAS
INSTITUTO DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

Detecção Automática de *Phishing* em Páginas Web

Janainny Sena Carvalho

Manaus – Amazonas
Outubro de 2013
Janainny Sena Carvalho

Detecção Automática de *Phishing* em Páginas Web

Proposta de mestrado apresentada ao Programa de Pós-Graduação em Informática da Universidade Federal do Amazonas, como requisito parcial para obtenção do título de Mestre em Informática.
Área de concentração: Redes de Computadores.

Orientador: Prof. Dr. Eduardo James Pereira Souto
Co-Orientadora: Profª. Dra. Eulanda Miranda dos Santos
Janainny Sena Carvalho

Detecção Automática de *Phishing* em Páginas Web

Proposta de Mestrado apresentada ao Programa de Pós-Graduação em Informática da Universidade Federal do Amazonas, como requisito parcial para obtenção do título de Mestre em Informática.
Área de concentração: Redes de Computadores.

Banca Examinadora

_____

Prof. Dr.

_____

Profª.

_____

Prof.

Manaus – Amazonas
Outubro de 2013

Agradecimentos

“Para tudo o que almejamos na vida sempre será exigido,
no mínimo, um sacrifício”.

Resumo

Abstract

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

A Internet vem servindo como infraestrutura para a disponibilização uma ampla variedade de recursos e serviços, incluindo suporte a correio eletrônico, comércio eletrônico, Internet banking, mídias sociais, entre outros. Toda essa popularização tem tornado a rede mundial cada vez mais relevante na vida diária das pessoas e das organizações, crescendo sua importância na atividade social e econômica global.

Nesse contexto, vem crescendo a cada momento o número de usuários conectados a Internet. Em 2007, aproximadamente 28,7% da população mundial tinha acesso a Internet, em torno de 1,96 milhões de usuários. Atualmente esses números são superiores a 77,8 milhões de usuários acessando a rede mundial de computadores [Zhang *et al.,* 2007].

Entretanto, essa massiva utilização de serviços *online*, passou a ser alvo frequente de diversos agentes que exploram recursos computacionais de forma não autorizada. Nesse sentido, serviços oferecidos pela Internet podem trazer consigo as fraudes eletrônicas como *spam*, códigos maliciosos *(malwares*), ataques de negação de serviços (DDoS - *Distributed Denial of Service*), vírus, *worms* e *phishing*.

Infelizmente, a demanda cada vez maior por novos recursos, a fim de prover mais serviços e funcionalidades para os usuários da Internet, principalmente os serviços de comércio eletrônico e Internet banking tem promovido ainda mais o crescimento de ataques *phishing*. Pesquisas recentes indicam que os ataques *phishing* se mantém no topo das listas das maiores vulnerabilidades em aplicações web nos últimos anos, conforme as estatísticas divulgadas em [APWG, 2010] [Zhang, Jianyi et al,. 2011] .

Phishing é uma forma de fraude que combina a engenharia social e técnicas de falsificação de páginas *web* com intuído de roubar informações confidencias como senhas, *logins*, número de cartões de crédito entre outras informações que são consideráveis críticas e sensíveis [Basnet, 2008]. Na Internet, o *phishing* pode atingir ao usuário (vítima) de várias maneiras, através de uma

janela *pop-up* no navegador (*browser*), uma URL maliciosa, de mensagens instantâneas ou através de mails recebidos. Geralmente, a vítima é convencida a executar uma ação (por exemplo, clicar num link), que poderá levar a instalação de algum *malware* ou ao redirecionamento do usuário a um site malicioso.

Em decorrência da grande quantidade de ataques *phishing*, diferentes abordagens e técnicas para solucionar ou mitigar o problema têm sido propostas nos últimos anos. Dentre elas, técnicas baseadas no emprego de listas negras (blacklists) [Ma, Justin et al., 2009] [Ye, Cao et al., 2009], de técnicas de aprendizagem e mineração de dados [Whittaker et al., 2009] [Miayamoto et al., 2009] [Fette et al., 2009] e na análise da estrutura estática e dinâmica dos elementos do aplicação web [Miayamoto et al., 2009] [Basnet ae al., 2008].

Entretanto, apesar das contribuições desses trabalhos, o problema ainda figura no topo das principais listas de vulnerabilidade, fato que motiva a pesquisa em busca de novas técnicas que possam contribuir com soluções de prevenção, detecção ou contenção de ataques *phishing*.

Este trabalho apresenta e avalia um método que emprega técnicas supervisionadas de aprendizagem de máquina, que têm como vantagem, a descoberta de padrões a partir de um conjunto de fatos ou observações rotuladas, induzindo a máquina a um processo de aprendizagem.

O método apresenta foco especializado na detecção de ataques *phishing*, permitindo uma exploração mais detalhada do problema e a análise de um conjunto de características relevantes que são extraídas de exemplos obtidos de bases reais da Internet e submetidos a métodos de aprendizagem estáveis e amplamente usados em problemas de classificação, tais como o *Naive Bayes*, SVM e Árvore de Decisão.

A estratégia proposta é estruturada em fases que tem o objetivo de viabilizar a extração de um conjunto de características que são relevantes para detecção de *phishing* em páginas *web,* essa extração ocorreu com a análise do conteúdo estático da URL, ou seja, características extraídas da URL, do documento que compõe a *web* e das extraídas a partir de informações contidas na base de dados *online*.

1.1 Motivação

A preocupação com a segurança da informação tem sido cada vez mais destacada pelos desenvolvedores de aplicações *web*, principalmente devido aos inúmeros serviços que são disponibilizados por estes na Internet.

Essa preocupação é justificada principalmente pela sofisticação e aumento crescente das

técnicas utilizadas para roubar dados na web, em especial dados de transações comerciais.

A fraude *phishing* se tornou a partir de 2009 responsável por 66% dos ataques realizados em páginas *web,* foi identificado neste ano que apenas um grupo de golpista foi responsável por dois terços de todos ataques *phishing* lançados na Internet e que também em 2009, *phishing* foi considerado uma avalanche e cada vez mais grupos fraudulentos vem ser aperfeiçoando para realizar novas tentativas de ataques *phishing*. [APWG], [Aaron Greg, 2010].

Nesse contexto, apesar de existirem várias soluções propostas para detecção de ataques *phishing*, assim como implementações para prevenção desse tipo de ameaça, *Anti-PhishingWork Group* [APWG, 2010] identificou cerca de 20 mil sites novos de *phishing* nos meses de julho a dezembro de 2008.

Ainda vale ressaltar que fatos que comprovam a "eficiência" do *phishing* podem ser observados diariamente. Assim se destaca a empresa de consultoria Gartner Inc. [2009] ao divulgar que, no ano de 2008, os criminosos *phishers* causaram um prejuízo de mais de 1,7 bilhões de dólares nos Estados Unidos. Ainda, a China Daily [2011], jornal *on-line* da empresa de segurança Beijing Rising Information Technology Co denuncia que páginas *phishing* roubaram cerca de 3 bilhões de dólares na China, em 2010. No Brasil, foram contabilizadas 31.008 tentativas de frades *phishing* em 2010 [CERT.Br,2011].

Nesse sentido, a implementação de técnicas para identificar automaticamente sites *phishing,* por exemplo, é algo que devemos dar a devida atenção. Inclusive algumas técnicas já são utilizadas por grandes sites, tais como: Google, eBay, Paypal. Uma das técnicas que esses sites utilizam é a classificação automática observando as características que um site *phishing* apresenta. Assim, um site que apresentar certas características pode ser classificado como um site *phishing* e, a utilização de classificadores de aprendizagem de máquina nessa análise é fundamental, uma vez que é possível ensinar padrões para que esses sejam classificados automaticamente.

Contudo, diante das ameaça que a fraude *phishing* vem impactando, diversas abordagens técnica tem sido proposta para que de algum modo seja realizada a detecção de *phishing* em páginas *web*. Porém, sempre existirá a necessidade de se obter novas ferramentas ou mecanismos para que esses possam ajudar na identificação de *phishing,* uma vez que a cada dia surgem novas formas de realizar fraudes em páginas *web*.

1.2 Objetivos do Trabalho

O objetivo deste trabalho é desenvolver um mecanismo de detecção de *Sites Phishing* em páginas *web* que utilize técnicas de aprendizagem de máquina e mineração de dados. Primeiramente, serão construídas as características e também a base de dados. Posteriormente, a extração das características mais relevantes serão selecionadas e extraídas através de testes em diversos classificadores e avaliadores. Por fim, serão aplicados algoritmos de aprendizagem de máquina, tais como *Naive Bayes, SVM* e Árvore de Decisão, para a classificação e análise das taxas obtidas.

Os objetivos específicos são os seguintes:

1. Desenvolver técnicas para detecção de *phishing* em páginas *web*.
2. Identificar base de dados com sites *phishing* e legítimos de diversos repositórios;
3. Definir principais características de sites *phishing* que sejam relevantes no processo de detecção de sites *phishing* propostas na literatura;
4. Analisar as características selecionadas, utilizando algoritmos de aprendizagem de máquina, tais como: *Naive Bayes*, SVM e Àrvore de Decisão;
5. Avaliar a relevância das características, dentro da amostra geral, com objetivo de melhorar a desempenho dos algoritmos de aprendizagem e reduzir o custo computacional.

1.3 Contribuições

Conforme os objetivos e metodologia utilizada neste trabalho foi possível produzir as seguinte contribuições:

1. Demonstrar um método para detectar *phishing* em páginas *web*, a partir da utilização de técnicas de aprendizagem de máquina e com essas identificar padrões de sites *phishing*.
2. Estabelecer e analisar um conjunto de características que identifiquem sites *phishing* em páginas *web*, e que ainda possam serem aplicadas em diversos contextos que envolvam outras soluções complementares;
3. Expor uma análise comparativas dos resultados obtidos com os métodos de classificação *Naive Bayes*, SVM e Árvore de Decisão, como intuito de apresentar as taxas obtidas nesse

processo de classificação, e a partir desse resultado demonstrar o desempenho obtido de *Phishing* em páginas *web*;

4. Disponibilizar a base de dados utilizada para emprego em futuros trabalhos de pesquisa na área de aprendizagem de máquina.

1.4 Organização do Documento

O desenvolvimento desta dissertação, a partir do Capítulo 2, está organizado como segue:

- O Capítulo 2 apresenta informações fundamentais para a compreensão do tema deste trabalho. Serão detalhados os conceitos básicos de detecção de *phishing*, formas de propagação e detecção, aprendizagem de máquinas e, principalmente, o funcionamento básico do classificador *Naive Bayes* SVM e Árvore de Decisão.
- O Capítulo 3 apresenta os trabalhos relacionados ao tema proposto, abordando sobre: Listas, Filtros, Heurísticas e trabalhos que utilizaram aprendizagem de máquina para detecção de sites *phishing* em páginas *web*.
- O Capítulo 4 descreve as etapas e o método proposto, também será detalhada a grupo de características selecionada para detecção de *phishing* em páginas *web*, a base de dados utilizada ;
- O Capítulo 5 descreve todos experimentos realizados com o conjunto de características e classificadores de aprendizagem de máquina utilizado neste trabalho, e com resultados obtidos, é demonstrado uma análise e comparação com outros desfechos mencionados em outras literaturas. E por fim, um resumo dos resultados e conclusões obtidas tomando por base a literatura consultada e a pesquisa no que diz respeito às direções futuras.

Capítulo 2

Conceitos Básicos

Este Capítulo apresenta uma breve descrição de ataques *phishing,* suas categorias e técnicas empregadas no combate dessas fraudes. Além disso, o capítulo também define os conceitos sobre aprendizagem de máquina e descreve os classificadores usados nos experimentos deste trabalho.

2.1 Phishing

De acordo com Richard [2005], o termo *phishing* foi sugerido para descrever o roubo de senha e contas da American On Line (AOL) em 1996. Desde então, os ataques *phishing* tem evoluído, sendo empregados, principalmente, para obter nomes de usuários e senhas de contas bancárias e de outros sistemas de autenticação online. Mais recentemente, os ataques *phishing* têm sido empregados em redes sociais (por exemplo, Facebook e Myspace) e websites de jogos (por exemplo, World of Warcraft and Steam) [Cluley, 2011].

Phishing é uma forma de roubo de informações confidenciais como nome, senha, CPF, número do cartão de crédito ou outras informações que podem levar ao roubo de identidade. Esse tipo de fraude cibernética utiliza técnicas da engenharia social e vetores de ataque sofisticados para colher informações de usuários que navegam em páginas *web*. [Garera et al., 2007] [Rosiello et al,. 2007][Blum et al,. 2010].

Nesse sentido, ataques *phishing* podem ser realizados por meio do envio de uma mensagem eletrônica (*mail*) que contém *links* para um *site* falso com semelhanças ao *site* original. Em geral, esses *mails* costumam incluir mensagens com um sentido de urgência, apontando a necessidade da operação para os destinatários fornecerem informações confidenciais ou clicar num *link* que o redireciona para uma página falsa semelhante a página legítima. A Figura 2.1 mostra um site legítimo e um site *phishing*, para destacar a semelhança entre ambos.

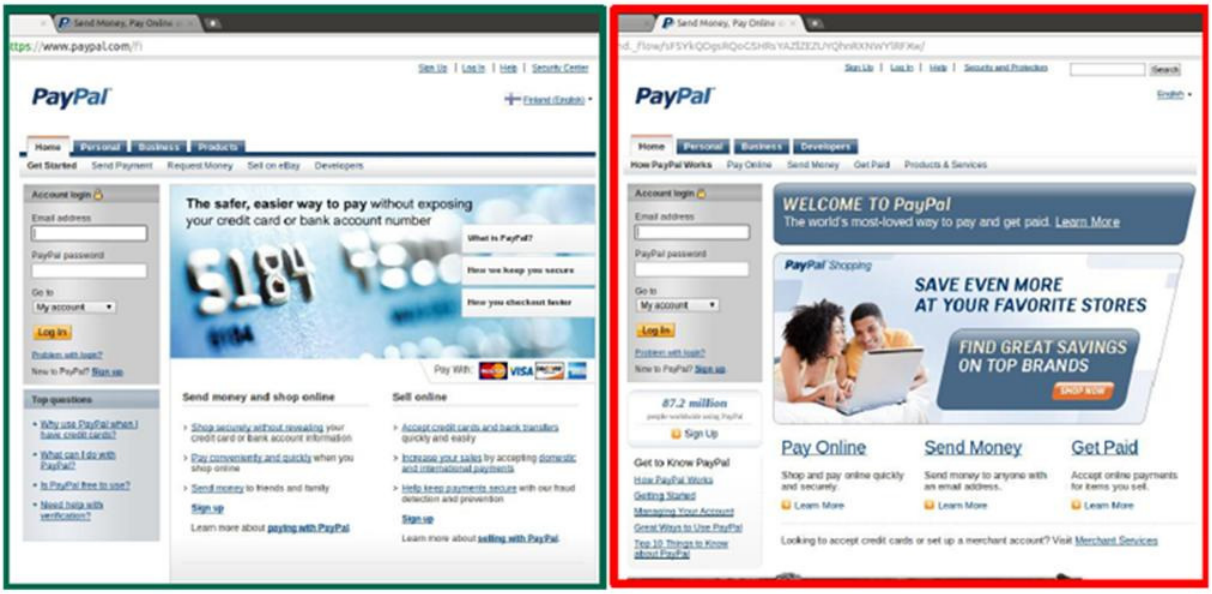

Figura 2.1: (a) página legítima, (b) página *phishing*.

O usuário normalmente não consegue identificar um site *phishing*, pois o construtor de uma página *phishing*, intitulado de *phisher,* elabora o site de forma que este seja o mais idêntico possível ao site original, dificultando a percepção dos usuários em distinguir páginas originais e fraudulentas.

O *phisher* tenta enganar o usuário disfarçando a URL, na tentativa de parecer ao máximo com o real, por exemplo: www.paypal.com (endereço legítimo) e www.paypal1.com (endereço falso). Ao acessar o endereço www.paypal1.com, o usuário poderia achar que está acessando o endereço do paypal legítimo, mas na realidade o mesmo estaria acessando o site falso. Existem muitas formas de burlar a atenção dos usuários para dificultar a identificação de sites fraudulentos. O objetivo dos golpistas é construir sites cada vez mais parecidos com os originais.

Outros autores [Olivo, 2012] [Ma, Justin et al., 2009] [Sheng, et al., 2009] apontam *phishing* como uma forma de estelionato, que usa engenharia social para fazer vítimas, as quais são enganadas geralmente com o objetivo de obter suas informações pessoais.

Esse aspecto pode ser interpretado de acordo com o Código Penal Brasileiro, considerando que estelionato é "obter, para si ou para outrem, vantagem ilícita, em prejuízo alheio, induzindo ou mantendo alguém em erro, mediante artifício, ardil, ou qualquer outro meio fraudulento" [Código Penal Brasileiro, Título II, Cap. VI, Art. 171].

2.2 Formas de Propagação

Conforme Downs *et al*. [2006], os ataques *phishing* são mais bem sucedidos quando o *phisher* é capaz de manipular os usuários. Portanto, é importante compreender tipos de modelos mentais que as pessoas utilizam para ler um *e-mail* ou páginas *Web.* Neste caso, o desenvolvimento de uma melhor compreensão dos motivos que levam as pessoas a caírem em sites fraudulentos é imprescindível. *Phishers* exploram a diferença entre o modelo do sistema e o modelo mental dos usuários com o intuito de melhor ludibriar a vítima [Wu, 2006].

A maior parte dos usuários desconhece a segurança fornecida pelo navegador *Web* e os canais de obtenção de informações de segurança. Um ataque *phishing* é bem sucedido porque os usuários tomam decisões imprudentes ao navegar num site sem as devidas precauções de segurança.

Um estudo realizado por Dhanija *et al*. [2006] sustenta que muitos usuários têm problemas na detecção de ataque *phishing*, pois não conseguem distinguir um site legítimo de um site fraudulento. Esse estudo apresentou sites *phishing* que conseguiram enganar até 90% dos usuários participantes.

Os *Phishers* adotam diversos vetores para distribuir seus ataques, incluindo distribuição massiva de mensagens conhecida como *spam*, via website e por propagação de malware.

2.2.1 Spam

A principal propagação de *phishing* ocorre através da divulgação de mensagens eletrônicas não solicitadas (*spam*). Essas mensagens são camufladas para serem confundidas com mensagens legítimas, de tal forma que a vítima que as recebe pode ser facilmente convencida de que estas foram enviadas por pessoas ou organizações legítimas [Rosiello, 2007]. Por exemplo, é comum mensagens fraudulentas de agências bancárias solicitando informações sigilosas ou atualização de cadastros. Segundo Fette *et al*. [2007], muitos usuários caem na armadilha da retenção de dados sigilosos, em diversas operações, fornecendo informação vulnerável às ações fraudulentas.

A Figura 2.2 exibe a sequência de passos de um ataque *phishing* usando o mail com vetor de propagação.

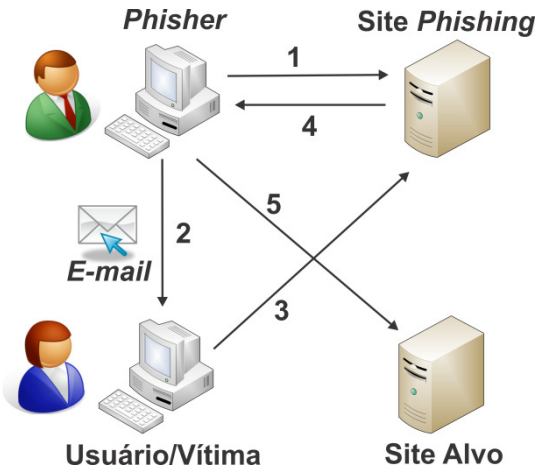

Figura 2.2: Etapas de um ataque *phishing*.

No primeiro passo, o atacante (*phisher*) cria um site falso em um servidor Web. Em geral, a página web é mantida por um serviço *web hosting* gratuito. Em seguida (passo 2), o *phisher* envia mails contendo *link* para URL fraudulenta para suas vítimas. O usuário ao clicar neste link é direcionado para site malicioso (passo 3). Em seguida, é solicitada a vítima o envio de informações pessoais, por exemplo, através do preenchimento dos campos de um formulário (passo 4). Por último, passo 5, os golpistas usam as informações para entrar em contas bancárias ou vender as informações para outros criminosos.

2.2.2 Redirecionamento da URL

Outra forma de propagação de *phishing* compreende no emprego de um método malicioso que desvia o acesso de um site de seu respectivo servidor legítimo, tal abordagem também é conhecida como *pharming* [Abu-Nimeh *et al*. 2007]. Neste tipo de fraude, ao tentar se conectar a um *site,* por exemplo*, Internet Banking,* o usuário é redirecionado para a página falsificada, normalmente idêntica ao *site* legítimo. Em geral, esse tipo de ataque ocorre devido ao sequestro ou envenenamento do DNS (*Domain Name Server*) pelo atacante, que por sua vez realiza o redirecionamento a sites falsos [Ye *et al.* 2008].

A Figura 2.3 mostra um exemplo de redirecionamento de URL.

Figura 2.3: Exemplo de Redirecionamento de URL

Na Figura 2.3, é possível visualizar que esta página contém *link* que redireciona o usuário para uma URL fraudulenta (http://receita-federal.mail333.su/Declaracao 2011.rar). O usuário ao clicar nesse link é convidado a realizar o *download* de um arquivo RAR, que corresponde a um executável "Declaração.exe".

O redirecionamento de URL é bastante empregado em ataques *phishing* pelo fato de ser de simples implementação. A Figura 2.4 exibe um trecho de código HTML que emprega redirecionamento de URL usando a tag <meta>.

Redirecionamento via TAG META

```
1 <html>
2 <title>Titulo do site</title>
3 <head>
4 <meta http-equiv="refresh" content="1;url=http://www.site_a_ser_redirecionado.com">
5 </head>
6 <body>
7 </body>
8 </html>
```

Figura 2.4. Exemplo de um redirecionamento usando *script* HTML.

Este tipo de redirecionamento é o mais simples, contudo, o mesmo irá modificar o endereço que aparece na barra de endereços pelo outro que está na tag <meta>.

2.2.3 Software Maliciosos

Códigos maliciosos são softwares especificamente desenvolvidos para executar ações danosas e atividades maliciosas em um computador. Existem diversas formas como códigos

maliciosos podem ser usados para comprometer um computador, dentre as quais se destacam [Cert, 2012]:

- Exploração de vulnerabilidades existentes nos programas instalados.
- Execução de mídias removíveis infectadas, como pen-drives.
- Acesso às páginas *Web* maliciosas utilizando navegadores vulneráveis.
- Ação direta de atacantes que, após invadirem o computador, incluem arquivos contendo códigos maliciosos;
- Execução de arquivos previamente infectados, obtidos em anexos de mensagens eletrônicas, via mídias removíveis, em páginas Web ou diretamente de outros computadores (através do compartilhamento de recursos).

Uma vez instalados, os códigos maliciosos passam a ter acesso aos dados armazenados no computador e podem executar ações em nome dos usuários, de acordo com as permissões de cada usuário [Rosiello, 2007].

A Figura 2.4 ilustra um vetor usado em um ataque *phishing* solicitando a instalação de um código malicioso.

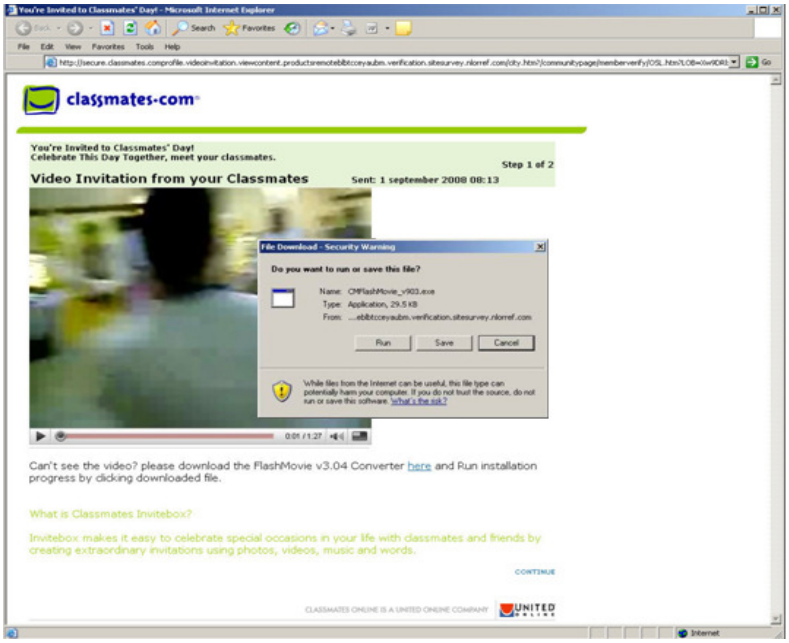

Figura 2.4. Exemplo da instalação de um software malicioso [Mills, 2008]

No exemplo da Figura 2.4, o usuário ao acessar determinado vídeo é solicitado a instalação de um software (neste caso, um software malicioso). Ao clicar em *RUN,* o software

malicioso será instalado no computador, permitindo ao atacante a coleta de informações confidenciais.

2.3 Técnicas de Detecção e Prevenção

É criado a todo instante: *mails*, sites e janelas *popup* falsas, com objetivo de ludibriar usuários que navegam na Internet. Em contrapartida, diversos métodos têm sido empregados para detecção e prevenção de *phishing*, por exemplo: Listas Negras, Baseada no Conteúdo e na Educação do Usuário. Esses métodos podem ser utilizados juntos ou separados e serão descritos a seguir:

2.3.1 Listas Negras

As listas negras são as técnicas mais utilizadas para a prevenção de ataques *phishing*. Trata-se de uma lista de *mails*, domínios ou endereços Internet Protocol (IP), reconhecidamente fontes de *spam*. Geralmente, utiliza-se este recurso (*blacklist*) para bloquear os *mails* suspeitos de serem *spam* no servidor de *mails*. Em alguns casos, os filtros configurados no programa leitor de *mails* também podem utilizar *blacklists*. [Antispam, 2013] e [Huang, et al., 2009].

Muitos navegadores (Internet Explorer, Mozilla Firefox, Google Chrome e Opera) utilizam soluções de *anti-phishing* baseadas em *blacklist* para notificar usuários quando esses estiverem navegando em páginas suspeitas [Whittaker et al,. 2010] e [Sharifi & Siadat 2008]. Por exemplo, o Filtro de *Phishing* da *Microsoft* é um recurso do Internet Explorer (IE) que ajuda a detectar sites de *phishing*. O Filtro de *Phishing* é executado em segundo plano enquanto o usuário navega pela *Web* e utiliza três métodos de proteção contra golpes de *phishing* [Microsoft, 2013].

- Primeiro, é procurado os endereços dos sites visitados em uma lista de sites registrados na *Microsoft* como legítimos. Essa lista está armazenada no computador que está realizando a consulta.
- Depois, é analisado os sites visitados para verificar se têm características comuns a um site de *phishing*.
- Por último, o Filtro de *Phishing* já com o consentimento do usuário, envia alguns endereços de site à *Microsoft* para serem verificados em uma lista atualizada frequentemente de sites de *phishing* relatados.

Muitas soluções de *blacklists* são integradas as barras de ferramentas dos navegadores, também conhecidas como *toolbars.* Tais ferramentas são destinadas à detecção de *phishing* e são normalmente utilizadas por usuários sem muitas experiências. NetCraft [Netcraft, 2013], Cloudmark [Cloudmark,2013], McAfee SiteAdvisor [Siteadvisor, 2013] e Norton 360[Norton, 2013] são exemplos de empresas que fornecem *toolbars*.

Uma desvantagem de utilizar listas negras é que estas estão limitadas a reconhecer apenas sites que já foram denunciados como *phishing.* Estas listas são alimentadas, em sua maioria, por usuários que serão vítimas ou por usuários já experientes que suspeitam de páginas ilegítimas e as denunciam de modo a serem avaliadas [Felegyhazi *et al*., 2010].

2.3.2 Baseada na Análise de Conteúdo

Outras abordagens empregam técnicas que analisam o conteúdo da página acessada. O objetivo é analisar os sites visitados para verificar se têm características comuns a um site de *phishing*. Por exemplo, Chandrasekaran *et al*. [2006] classificam páginas *phishing* com base nas propriedades estruturais de *mails phishing*. Características extraídas da estrutura da linha do assunto e da estrutura da saudação do corpo do *mail* são submetidas a um classificador que sinaliza a presença ou não de um ataque *phishing.*

Outro exemplo é a análise de recursos extraídos da URL tais como: Quantidade de número de pontos contido na URL, Tamanho da URL, Presença de caracteres hexadecimais ou Especiais, esses recursos, serão detalhados no Capítulo 4, e serão descritos suas características e funcionalidades [Ma et al., 2009].

Os métodos citados têm vantagens e desvantagens: *blacklist* tem um alto nível de precisão, no entanto, não consegue detectar novos ataques *phishing* imediatamente, requisitando antes a atualização. Ainda, a *blacklist* normalmente requer intervenção e verificação, o que pode acarretar um grande consumo de recursos material e humano. O emprego de na analise de conteúdo, por outro lado, consegue detectar *phishing* quando são disseminados, embora possam produzir falsos positivos, ou seja, páginas legítimas incorretamente classificadas como *phishing*. . [Whittaker *et al,*. 2010], . [Ma *et al,*. 2009] e [Wardman & Warner 2008].

2.3.3 Educação do usuário

Outros pesquisadores têm concentrado na educação dos usuários da Internet para prevenir futuros ataques. Tais estudos têm utilizado uma série de materiais educativos *anti-phishing* como "Momentos de ensino". Por exemplo, o sistema *Anti-Phishing Phil* é um jogo *on-line* [projetado para ensinar os usuários não a ser vítima de ataques de *phishing* (Sheng *et al*., 2007].

Nesse sentido, a educação dos usuários é frequentemente recomendada e amplamente utilizada na detecção de *phishing* [Timko, 2008], mas poucos estudos têm avaliado a eficácia das abordagens no mundo real [Kumaraguru *et al.* 2008].

Algumas abordagens têm sido concentradas, e a primeira pode ser relatada na oferta de informações *online* contra os riscos e ataques de *phishing* e como evitá-los, esses materiais são normalmente fornecidos pelos governos ou organizações sem fins lucrativos. Em outra abordagem são enfatizados testes *online* de habilidades do usuário que atendam a certas pontuações de acertos às páginas ilegítimas [Zhang *et al.,* 2011].

O jogo *PhishGuru* por exemplo*,* motiva os usuários a prestar atenção nos sites que estão acessando ou que ainda irão acessar. Aos usuários do *PhishGuru* são enviados simulação de ataques *phishing* via e-mail e são apresentados informações de treinamento identificando formas para que esses usuários não caiam nas armadilhas da *Internet*.

Existem outras abordagens para treinamento de usuários sobre prevenção de *sites phishing*, incluindo: artigos sobre *phishing* em *sites*, *cartoons online* sobre segurança, avisos de segurança por e-mail e treinamento em sala de aula [Kumaraguru *et al.* 2008].

A análise realizada no trabalho desenvolvido por [Kumaraguru *et al.* 2008] mostra que estas abordagens tem um custo variado e também sua eficácia. Por exemplo: o treinamento em sala de aula pode ser mais eficaz que os outros métodos de formação devidos os usuários serem obrigados há passar um tempo dedicado ao treinamento. Contudo, é um processo que pode ser demorado e ter um custo elevado para a empresa. Materiais *online* podem ter baixo custo, mas usuários dificilmente se interessariam em ler um material extenso, portanto, nem sempre são eficazes.

Os autores Miller e Wu [2005] chegaram a afirmar que os *phishers* exploraram a diferença entre o modelo do sistema e o modelo mental dos usuários com finalidade de enganá-los. Psicólogos e pesquisadores de comunicação têm estudado formas preventivas para evitar que os usuários sejam enganados. Nesse contexto, uma das metas de *anti-phishing*

é desenvolver ferramentas de formação dos usuários de modo que eles sejam capazes de gerar e testar hipóteses de ameaças adequadas e protegerem-se de possíveis armadilhas virtuais.

2.4 Aprendizagem de Máquina

2.4.1 Definição

Aprendizagem de máquina é uma área de inteligência artificial (IA) cujo objetivo é o desenvolvimento de técnicas computacionais sobre o aprendizado, bem como a construção de sistemas capazes de adquirir conhecimento de forma automática [Rezende, 2005].

Nesse contexto, é caracterizado como um sistema de aprendizagem de máquina aquele que é capaz de adquirir conhecimento de forma automática, assim um computador que toma decisões baseado em experiência acumuladas por meio de soluções bem sucedidas de problemas anteriores é sem dúvida uma forma de descrever o conceito de aprendizagem de máquina.

Técnicas de aprendizagem de máquina podem ser utilizadas para o aperfeiçoamento de determinadas atividades computacionais. Seu uso vai desde o auxílio em diagnósticos médicos até o reconhecimento de escrita, robótica, segurança da informação, etc. [Alpaydin, 2010].

Aprendizagem de máquina, também chamada de *machine learning,* é um termo que engloba um conjunto de metodologias e comportamentos em dados que representam exemplos de acontecimentos do mundo real ou experiências passadas. Dessa forma, os dois objetivos de qualquer projeto de aprendizagem são: induzir o modelo processando uma grande quantidade de dados e realizar inferências a partir dele. Dentre esses objetivos, processar essa grande quantidade de dados é a que exige maior tempo e esforço computacional. [ ].

2.4.2 O Processo Indutivo de Aprendizagem

O aprendizado indutivo é efetuado a partir do raciocínio sobre exemplos fornecidos por um processo externo ao sistema de aprendizado. O aprendizado indutivo pode ser dividido em dois tipos principais: supervisionado e não-supervisionado [ ].

No paradigma supervisionado o conhecimento prévio do ambiente é exigido, nesse tipo de paradigma, o objetivo relevante é entender o mapeamento principal da entrada para saídas de informações, nesse projeto, esse será paradigma utilizado.

No aprendizado não-supervisionado, para cada exemplo apenas os atributos de entrada estão disponíveis. O indutor analisa os exemplos fornecidos e tenta determinar se alguns deles podem ser agrupados de alguma maneira, formando os *clusters* [Cheeseman e Stustz, 1990]. Na Figura 2.4 é mostrada a hierarquia do aprendizado descrito acima:

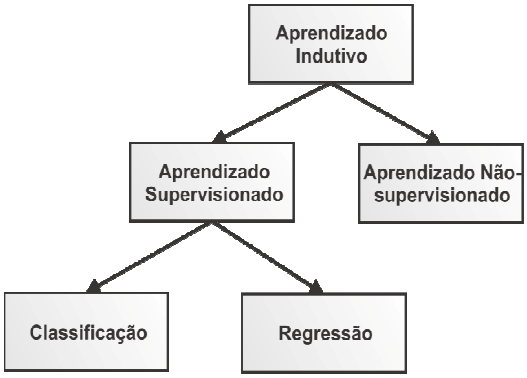

Figura 2.3: Hierarquia do aprendizado [Rezende, 2005].

2.4.3 Terminologia

Para uma compreensão mais abrangente dos algoritmos de aprendizagem de máquina, é necessário que alguns dos termos e conceitos relevantes sejam definidos. As definições apresentadas a seguir, foram extraídas de [Kuncheva, 2004]:

1. **Classes**

   Uma classe possui objetos similares, enquanto objetos de classes diferentes são dissimilares. Por exemplo, imagens pornográficas podem ser categorizadas como objetos da classe nudez, enquanto imagens de nudez são objetos da classe não-nudez. Algumas classes são facilmente definidas, por exemplo, um e-mail deverá obrigatoriamente ser *spam* ou não *spam.* Outras classes, porém, são de difícil definição, como por exemplo, a classe das pessoas destras e das pessoas não destras.

   Resumidamente, pode-se definir que um problema de classificação de dados possui *c* classes, rotuladas de ω_1 a ω_c, organizadas em um conjunto de rótulos $\Omega = \{\omega_1,..,\omega_c\}$e que cada objeto pertence a apenas uma classe.

2. **Características ou atributos**

   Os objetos que compõem as classes são descritos através de características ou atributos. As características podem ser nominais, tais como endereço e profissão, e podem ser numéricas, como tamanho da url e quantidade de palavras-chaves. Os

valores de características de um objeto *x* são organizados em um vetor *n*-dimensional $x = [x_1,...,x_n] \in \Re^n$. O espaço $\Re^n$ é chamado espaço de características, sendo que cada eixo corresponde a uma característica do objeto.

2. Base de Dados

A informação fundamental para algoritmos de aprendizagem de máquina, independentemente do tipo de aprendizagem, é proveniente dos dados disponíveis do problema. Essa informação normalmente está organizada na forma de um conjunto de dados $Z = \{z_1,...,z_N\}, z_j \in \Re^n$. Em problemas de aprendizagem supervisionada, o rótulo da classe de z_j é definido por $l(z_j) \in \Omega, j = 1,...,N$. Para construir uma base de dados, as instâncias do problema devem ser transformadas em vetores de características. Nem sempre essa tarefa é simples. Por exemplo, dadas diversas instâncias de sites com código *script*, não é uma tarefa trivial representar as páginas através de vetores de características como: tamanho de URL, quantidade de pontos da URL, etc. Entretanto, o uso de características relevantes para representar as instâncias do problema a ser tratado é fundamental para o sucesso do processo de classificação automática.

2.4.4 Métodos e Algoritmos de Classificação:

Os algoritmos de aprendizagem de máquina que serão discutidos nas próximas sessões são supervisionados, isto são algoritmos (classificador) que tentam mapear entradas desejando uma saída com uma função específica. No caso de classificação de sites *phishing*, um classificador irá tentar classificar um site como *phishing* ou legítimo, por aprender certas características dos sites.

Nesse sentido, o conceito *aprendizagem de máquina*, em particular classificação automática, tornou-se popular envolvendo trabalhos relacionados à detecção de *e-mail*, s*pam* e detecção de *sites phishing* [Bergholz et. al, 2008].

Os autores afirmam, contudo, que em comparação com os filtros construídos manualmente as regras automáticas avaliam a relevância da entrada de características $x\ (x_1, \dots x_m)$ (e.g, características de páginas *phishing*) e a estabilidade como função para determinar a classificação desejada de y (e.g, *phishing* ou não *phishing*)

$$y = f(x, \gamma) \tag{2.1}$$

O vetor dos valores de parâmetros desconhecidos é determinado na fase de treinamento, de tal maneira que a relação entre x e y nos dados observados $(x_1,...y_1),...,(x_D,...y_D)$ é reproduzida de acordo com algum critério de otimização. Na fase da aplicação as características iguais são extraídas a partir de uma entrada de novo site. Baseado nessas características o modelo do classificador produz uma classificação de *phishing* e não *phishing*. Para melhor entendimento, a abordagem de aprendizagem de máquina será sumarizada na Figura 2.4.

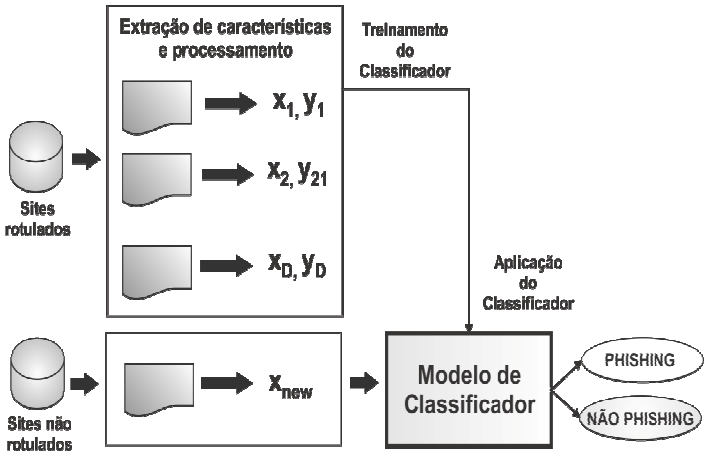

Figura 2.4: Abordagem sobre aprendizagem de máquina [...]

2.4.4.1 *Support Vector Machines* (SVM)

É uma técnica de classificação amplamente aplicada em problemas de segurança de redes tais como detecção de *phishing* [Miyamoto et al., 2009] , [Nimed et al., 2009] e Pan & Ding 2011].

Supoort Vector Machine (SVM) é um dos classificadores mais populares hoje em dia. A ideia geral desse classificador é definida ao em encontrar a separação ótima do hiperplano entre duas classes por maximização da margem entre as classes com pontos mais próximos [Abu-Nime et al, 2007].

Basicamente, o funcionamento de SVM pode ser descrito da seguinte forma: dadas duas classes e um conjunto de instâncias de treinamento cujas amostras pertencem a essas classes, SVM constrói um hiperplano que divide o espaço de características em duas regiões, maximizando a margem de separação entre as mesmas. Esse hiperplano é conhecido como hiperplano de separação ótima. As amostras desconhecidas (exemplos de teste) são então mapeadas para esse mesmo espaço, e atribuídas a uma das classes [Alpaydim, 2010].

Nesse contexto, supondo que temos uma função discriminante linear e duas classes linearmente separáveis com valores $+1$ e -1, um hiperplano discriminante pode ser definido como:

$$w'x_i + w_0 \geq 0 \; i + \; t_i = \; +1 \tag{2.4}$$

$$w'x_i + w_0 < 0 \; i + \; t_i = \; -1 \tag{2.5}$$

A distância de qualquer ponto x para o hiperplano e $|w'x_i + w_0 \,|/ \, \|w\| \; e$ a distância para a origem é $|w_0 \,|/\|w\|$. A Figura 2 mostra que os pontos situados sobre o limites são chamados de vetores de suporte, e no meio da margem é o hiperplanos de separação ótima que maximiza a margem de separação [apud Abu-Nime et al, 2007].

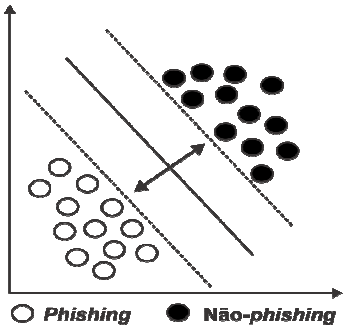

Figura 2.5: Separação linear com SVM. [Olivo, 2010].

O problema de detecção de *phishing* pode ser visto como um problema de classificação de duas classes (*phishing* e não *phishing*). Assim, a utilização do algoritmo de aprendizagem de máquina SVM (é adequado, pois foi desenvolvido originalmente para resolver esse tipo de problema [Vapnik 1995]. A literatura técnica mostra que o SVM tem sido aplicado com bastante sucesso em diversos domínios de aplicação, inclusive na detecção de *phishing* [Basnet, Mukkamala e Sung 2008] e [Chandrasekaran, Narayanan e Upadhyaya 2006].

2.4.4.2 Naive Baye

O classificador Naive Baye e tratado como um dos algoritmos mais simples, porem eficaz e tem sido utilizado em diversas aplicações, tais como detecção de filtragem de *mail spam phishing.* [Miyamoto *et al*., 2009].

Ainda o autor Ye *et al.* [2008] afirma que esse classificar possui uma abordagem mais eficaz tratando-se de classificação de documento de texto. E que dada uma quantidade de amostras de treinamento, uma aplicação pode aprender a partir destas amostras, de maneira

que venha a prever a classe da amostra. Ainda, pode-se destacar que um classificador Naive é usado na filtragem de spam, como já mencionado acima e cada e-mail pode ser representado por um vetor com as características : $x = (x_1, x_2, x_3, \ldots x_n)$, onde cada propriedade, $x = (x_1, x_2, x_3, \ldots, x_n)$ é independente.

A baixa complexidade na fase de treinamento é uma das principais vantagens de *Naive Bayes*, uma vez que essa fase envolve apenas o cálculo de frequências para que as probabilidades sejam obtidas. Essa característica torna *Naive Bayes* indicado para aplicações *online*, tais como problemas envolvendo segurança de redes cujo treinamento precisa ocorrer de forma *online* e com frequência regular. Outra característica que torna *Naive Bayes* atrativo para a área de segurança *web* é o fato de possibilitar a manipulação de atributos nominais e numéricos. Atributos nominais são frequentes em detecção de *spam*, *XSS*, páginas *phishing*, dentre outros problemas de classificação de documentos textuais.

2.4.4.3 Árvore de Decisão

Algoritmos que induzem Árvores de Decisão pertencem á família de algoritmos Top Down *Inductions nos Decision Trees (TDIDT).* Uma arvore de decisão (AD) é uma estrutura de dados definida recursivamente como [Rezende, 2005]:

- Nó raiz: corresponde ao nó de decisão inicial que normalmente é gerado utilizando-se o atributo mais discriminante entre as classes envolvidas no problema;
- Arestas: correspondem aos diferentes valores possíveis das características;
- Nó folha: corresponde a um nó de resposta, contendo a classe a qual pertence o objeto a ser classificado.

Em árvores de decisão, duas grandes fases devem ser asseguradas:

- Construção da árvore: uma árvore de decisão é construída com base no conjunto de dados de treinamento, sendo dependente da complexidade dos dados. Uma vez construída, regras podem ser extraídas através dos diversos caminhos providos pela árvore para que sejam geradas informações sobre o processo de aprendizagem.
- Classificação: para classificar uma nova instância, os atributos da amostra são testados pelo nó raiz e pelos nós subsequentes, caso seja necessário. O resultado

deste teste permite que os valores dos atributos da instância dada sejam propagados do nó raiz até um dos nós folhas. Ou seja, até que uma classe seja atribuída à amostra.

Alguns algoritmos foram desenvolvidos a fim de assegurar a construção de árvores de decisão e seu uso para a tarefa de classificação. O ID3 e C4.5, algoritmos desenvolvidos por Quinlan [1993], são provavelmente os mais populares.

2.5 Comentários Finais

Este Capítulo, apresentou abordagens de propagação, detecção e prevenção de *phishing*. Identificou-se que existem diferentes formas e mecanismos de detecção que estão sendo adotados na tentativa de combater fraudes disseminadas na Internet. Dentre as técnicas propostas para detecção de *phishing*, destaca-se a utilização de aprendizagem de máquina. Diante disso, foi apresentada uma breve descrição sobre tal técnica, bem como os algoritmos utilizados nos experimentos deste trabalho. Para um melhor aprofundamento dos assuntos relacionados à AM, os seguintes autores são recomendados: Alpaydim, E. (2004) e Witten, Ian; Eibe, Frank; Hall, Mark A. (2001).

No Capítulo a seguir são apresentados trabalhos relacionados às ferramentas para detecção de *phishing* que estão sendo empregadas e os resultados obtidos pelos autores.

Capítulo 3

Trabalhos relacionados

Este Capítulo realiza uma breve análise de alguns trabalhos relacionados à detecção de *phishing*. Os trabalhos são organizados de acordo com o tipo de análise empregada nas soluções propostas. Ao final do capítulo é apresentado um quadro com a sumarização dos mesmos e seus resultados.

A prevenção e a detecção de *phishing* podem ser feitas em dois níveis de abordagens: filtragem de *mails* e de páginas *Web* [Sheng *et al*., 2009]. Nesse contexto, as abordagens mencionadas podem utilizar métodos estatísticos de aprendizagem de máquina para classificar a reputação da base local, que nesse caso analisará as características baseadas em conteúdo e URL de um site *phishing* para então classifica-la ou método dinâmico que provem da utilização de diversos algoritmos para classificação de sites falsos.

Nesse sentido, as ferramentas *anti-phishing* baseadas em listas ou baseada no conteúdo do *site*, adotam ambos os métodos, que inclusive podem estarem juntas ou separada em uma mesma ferramenta, isso pode ser visualizado nos trabalhos relacionado dos autores: [Whittaker et al., 2010], [Ma et al., 2009], [Likarish et al., 2008],[Zang et al., 2007] entre outros. Este trabalho utiliza o método estatístico.

3.1 Abordagem baseada em Listas

Likarish *et al*. [2008] destacam que muitos *anti-phishing* dependem da utilização de listas (*blacklist* e *whitelist)* para classificar páginas web como verdadeiras ou falsas. Os autores apresentam uma abordagem baseada em filtros bayesianos, nomeada B-APT (*Bayesian Barra Anti-Phishing*), para ajudar usuários a identificar *sites phishing.*

A ferramenta B-APT consiste de duas partes: o módulo B-APT e a interface do usuário. O módulo B-APT, consiste em 03 componentes: um analisador JavaScript DOM (*Document Objeto Module*), um módulo de pontuação, e uma *whitelist,* conforme ilustrado na

Figura 3.1. A interface do usuário é pode ser instalada no navegador *Mozila Firefox*.

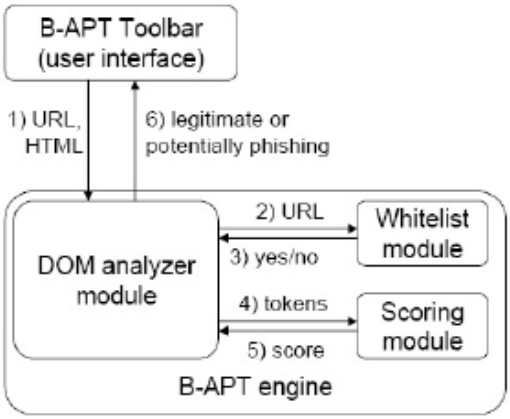

Figura 3.1: Arquitetura empregada na ferramenta B-APT *toolbar* Likarish et al. [2008].

A verificação se um site é *phishing* é realizada da seguinte forma: a ferramenta B-APT envia a URL e o HTML da página para o analisador DOM, que por sua vez, verifica primeiro se a URL consta ou não na lista de sites legítimos (*whitelist)*. Se a resposta for negativa, ou seja, a página consultada não está na *whitelist,* o analisador DOM faz uma nova requisição ao módulo de pontuação, que aplica o filtro *Bayesian* proposto e gera uma pontuação. Este valor, denominado de escore, é enviado para o analisador DOM. Caso a pontuação exceda um limite pré-definido, o analisador de DOM notifica a barra de ferramentas B-APT que o site é potencialmente *phishing*.

Os resultados obtidos neste trabalho mostram que a ferramenta B-APT obtém uma maior precisão na detecção de ataques de sites *phishing* recém-gerado em comparação com outras abordagens baseadas em lista negra, como IE e *Netcraft.*

Sheng *et al*. [2009] estudaram a efetividade de oito ferramentas de *anti-phishing* para *blacklist,* utilizando um conjunto de 191 páginas *phishing* recém-geradas. Os autores identificaram que as *blacklists* são atualizadas em velocidades diferentes, onde 47% a 83% das páginas *phishing* demoraram em torno de 12 horas para iniciar sua classificação. Em relação à eficiência na detecção de sites *phishing* através de *blacklist*, destaca-se que as duas ferramentas que utilizam heurísticas nas *blackilisst* obtiveram um melhor desempenho.

Os autores mencionam que um dos problemas principais com *blacklist* é que elas não conseguem identificar URLs de *phishing* nas primeiras horas de um ataque de *phishing*

devido ser um processo de atualização demorada. Normalmente os sites *phishing* duram em torno de duas horas, portanto é possível que um site *phishing* entre na *blacklist*, após ter sido tirado do ar.

Ye *et al.* [2008] apontaram que a maioria das detecções de *phishing* ocorre baseada em *blacklist,* devido essas listas serem responsáveis por acionar alerta do ataque quando um usuário visita um site que esteja em seu banco de dados. Portanto, manter a *blacklist* atualizada exige grande dose de esforço, considerando que a todo o momento surgem novos *sites phishing.*

No trabalho realizado pelos autores, é apresentada uma ferramenta *anti-phishing* intitulada *Automated Individual White-List* (AIWL). Tal ferramenta, arquiva o *Login User Interfaces* (LUIs) em uma *whitelist* utilizando o classificador *Naive Bayes*, o qual tem o papel de identificar se o *login* do usuário de fato foi realizado com sucesso. Neste caso, as vantagens apontadas nesta proposta é que AIWL reconhece sites *pharming* verificando o endereço IP do site, principalmente levando em conta que os sites populares possuem endereço estável, de modo que AIWL pode detectar *pharming.*

Com a utilização do classificador *Naive Bayes,* a AWIL teve uma taxa de falso negativo de 100% e, falso positivo de 0% na identificação do processo de *login* do usuário, resultado satisfatório que se baseia no comportamento do *login* atual do usuário em um determinado *site* [Ye *et al*., 2008].

A eficiência da *whitelist* é devido o conteúdo inserido nela ser estável. Em contrapartida, caso AIWL seja instalado numa máquina local é dificil controlar ataques do tipo cavalo de troia e vírus de computador, sendo necessário armazenar essa *whitelist* em um telefone inteligente.

Segundo Ma *et al*. [2009] informar aos usuários sobre sites fraudulentos significa que parte do problema referente aos ataques *phishing* seria atenuada. Para isso, as *blacklist* servem para fornecer aos usuários a notificação de *sites* fraudulentos e estão embutidas na *toolbar* do navegador.

Os autores indicam que a criação de *blacklist* é realizada por uma série de técnicas, como: relatório manual, *honeypots*[1] e coletor *web (crawlers)*[2], combinando com site de

[1] Software, cuja função é detectar ou de impedir a ação de um cracker, de um spammer, ou de qualquer agente externo estranho ao sistema, enganando-o, fazendo-o pensar que esteja de fato explorando uma vulnerabilidade daquele sistema.

análise heurística. A utilização de aprendizagem de máquina (AM) faz-se necessário nesse processo realizando a classificação da reputação dos *sites* entre URLs e as características lexicais dos *hosts.*

O trabalho realizado pelos autores contou com a confecção de conjunto de dados com 20.000 *sites*, sendo 15.000 do tipo legítimo, 5.500 do tipo *phishing* e 15.000 *spams*. Diversos experimentos foram realizados com combinações entre sites do repositório *PhishTank* e Dmoz. Nessa proposta, a utilização de classificadores de aprendizagem de máquina tais como: *Naives Bayes*, *Suporte Vetor Machine* (SVM) e Regressão Logística (LR) se fizerem presente.

Os resultados contaram com a análise do impacto de cada uma das características empregada e os autores afirmam que o resultado é muito satisfatório. Nesse sentido, os experimentos contaram com a utilização de um conjunto de dados do *Yahoo* e *PhishTank*, as taxas atingidas para falso positivo e falso negativo foram respectivamente: 0,1% e 7,6%, usando Regressão Logística.

No trabalho realizado por Prakash et al., [2010] os autores propõem a ferramenta *PhishNet* que aprimora *blacklist* existentes descobrindo novas *URLs* maliciosas relacionadas. A Figura 3.2 realiza o calculo e atribui uma pontuação a *URLs*.

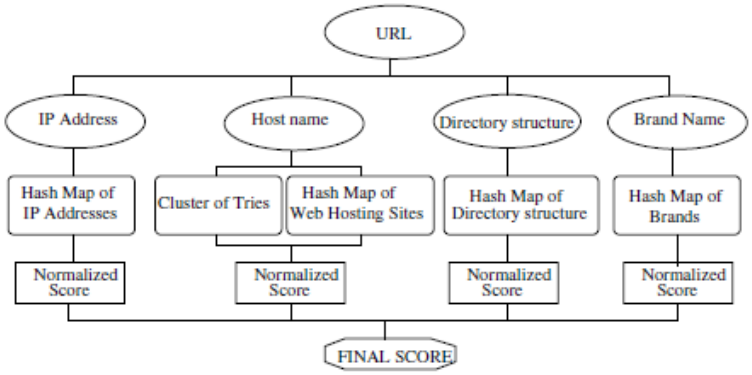

Figura 3.2: Arquitetura empregada na ferramenta *PhishNet* Prakash et al. [2010].

Na ferramenta *PhishiNet* quando é realizado o calculo e o resultado for acima de um limite, a *URL* é marcada como um site de *phishing*. Os autores mencionam que as *URLs* possuem a seguinte estrutura: http://domain.TLD/directory/filename?query_string. O diretório especifica o caminho em que o arquivo é passado com uma *string* de busca, formando junto o caminho da *URL*. A heurística empregada nesse trabalho, envolve trocar o nome desses campos entre *URLs* lexicamente agrupadas ou ao longo de alguma outra dimensão.

---

[2] Programa de computador que navega pela *World Wide Web* (WWW) de uma forma metódica e automatizada.

A ferramenta empregada neste trabalho utiliza cinco heurística para classificar um site *phishing,* a seguir, é descrito funcionamento.

URL: A primeira heurística de previsão é baseada em encontrar as variantes de entradas originárias de *blacklists* obtidas pela troca de TLDs. No trabalho, foi usado 3. 210 domínios efetivos de alto nível (TLDs). Assim, para cada nova *URL* que entra com uma determinada *blacklist*, é substituído o TLD eficazmente da *URL* com 3.209 outras TLDs que formam a URLs candidatas que precisam ser validadas.

IP Address: Na segunda heurística é mantida classes de equivalência onde *URLS phishing* que possuem o mesmo endereço IP são agrupados em *clusters*. Em seguida, é criada novas *URLs*, considerando todas as combinações de servidores e caminhos.

Host name: Nessa heurística a intuição básica é que, há uma boa chance de que dois *URLs* compartilharem uma estrutura de diretório em comum que pode incorporar um conjunto de nomes semelhantes de arquivos. Por exemplo, se www.abc.com/online/signin/paypal.htm e www.xyz.com/online/signin/ebay.htm são duas *URLs* de *phishing* conhecidas, então a heurística em questão prevê a existência de www.abc.com/online/signin/ebay.htm e www.xyz.com/online/signin/paypal.htm. Assim, é mantido um caminho de classe de equivalência no qual as *URLs* com estrutura de diretórios similares são agrupadas. É construídas novas *URLs* trocando os nomes dos arquivos entre *URLs* pertencentes ao mesmo grupo.

Directory structure: Na análise do banco de dados do repositório do *PhishTank*, foi observado várias *URLs* com exatamente a mesma estrutura de diretórios diferem apenas uma parte da *string* de busca parte da *URL*, Por exemplo, se existem duas URLS: www.abc.com/online/signin/ebay?XYZ e www.xyz.com/online/signin/paypal?ABC é criado duas novas URLS, www.abc.com/online/signin/ebay?ABC e www.xyz.com/online/signin/paupal?XYZ. Nessa heurística, o objetivo é assegurar combinações finitas, podendo considerar apenas as combinações existentes na *blacklist*.

Brand Name: Primeiro passo nesse processo é realizar uma pesquisa de DNS para filtrar os sites que não podem ser resolvidos. Para cada uma das URLs resolvidas, e procurado estabelecer uma conexão com o servidor correspondente. Para cada conexão bem-sucedida, é iniciada uma solicitação HTTP GET para obter conteúdo do servidor. Se o cabeçalho HTTP do servidor tem o código de status 200/202 (pedido de sucesso), realizamos uma semelhança de conteúdo entre as URLs pai e filho usando uma ferramenta pública de

detecção de similaridade. Se o conteúdo da URL tem semelhança acentuada (acima de 90%) com a URL pai, é concluído que a URL filha é um site ruim legítimo que precisa ser adicionado à *blacklist*.

Nas considerações relacionadas pelos autores e dito que a analise foi realizada em tempo real, foi descoberto em torno de 18 mil novas URLs de *sites phishing* a partir de um conjunto de 6.000 novas entradas de *blacklist.* É identificado também que o algoritmo proposto obteve muito poucos falsos positivos (3%) e negativos (5%).

A seguir serão descritas ferramentas *anti-phishing* que utilizam abordagem baseadas no conteúdo.

3.3 Baseada no Conteúdo da URL

As heurísticas procuram verificar se a página supostamente legítima possui características de um ataque *phishing*. Conforme Sheng *et al*. [2009], a maioria das heurísticas para detectar sites *phishing* usam as seguintes características: conteúdo da HTML, conteúdo dos sites ou assinaturas URL para identificar *phishing*. Os algoritmos de aprendizagem de máquina são normalmente aplicados para construir modelos de classificação sobre as heurísticas, de maneira a classificar novas páginas *web*.

No trabalho realizado por Abu-Nimeh *et al*. [2007], a detecção de *e-mails phishing* se processa por meio das características extraídas de corpo do *e-mail,* apresentando as mais recentes tendências de *e-mail phishing*. O conjunto de dados das páginas analisadas pelo autor consistiu na identificação de 43 características, as quais apresentam resposta binária como *phishing* =1 e, legítimos = 0. Além disso, o autor utilizou TF-IDF (Termo freqüência-inversa do original do termo), que é o algoritmo usado em recuperação da informação desenvolvido por [Phelps *et al*., 2000].

No contexto tratado acima foram aplicados os classificadores: Regressão Logística (RL), *Classification and Regressions Trees* (CART), *Bayesian Additive Regression Tress* (BART), SVM, Floresta Aleatória (FA) e *Neural Network* (NNET), considerando que todos esses classificadores realizaram a estratégia de validação cruzada em dez partes.

Os resultados obtidos no experimento de Abu-Nimeh *et al.* [2007] apresenta detecção de sites *phishing* e legítimo com pesos iguais, e o classificador FA tem a melhor taxa de erro, equivalente a 7,72%. Assim sendo, o classificador mencionado alcança o pior índice de falso positivo, o equivalente a 8,29%. Quando aplicado o custo de erro, penalizando falsos

positivos por nove vezes a mais que falsos negativos, a taxa do classificador RL supera a taxa de todos os classificadores, alcançando taxa de erro de 3,82%. A pior taxa quando aplicada, a penalidade é de FA com 5,79%, assim, para detecção de email, seja *phishing* ou *spam*, requer a aplicação de penalidade em falsos positivos, estes possuem um custo maior no mundo real.

A utilização de aprendizagem de máquina aplicando um classificador automático para detecção de páginas *phishing* alimenta automaticamente a *blacklist,* o que é proposto por [Whittaker *et al*. 2010]. A idéia consiste em realizar a análise de milhões de páginas diariamente por esse projeto, encontrando entre as características analisadas: URL e o conteúdo de uma página para determinar *phishing*. A diferença dos demais projetos desse trabalho é que a data set é processada contendo ruídos.

Ainda assim, para a montagem da base de dados o sistema classifica páginas *web* submetidas pelos utilizadores finais e recolhidas de filtros de *Spam* e do *Gmail*, com essas páginas é extraída uma série de características, estas descrevem a composição de URL da página, hospedagem da própria página e da HTML do conteúdo.

A classificação com Regressão Logística se faz presente nesse processo contando com a vantagem que menos de 1% da entrada é realizada manualmente. O algoritmo de recuperação de informação TF-IDF também compõe esse sistema que mantém uma taxa de falsos positivos abaixo de 0,1%.

Esse projeto utiliza a combinação de vários métodos para detecção de *phishing* que apresenta taxas baixas na utilização de heurísticas, e no momento de classificar a inclusão da página na *blacklist* medições são realizadas com o objetivo de classificar corretamente.

A proposta lançada por Zhang, Yue *et al*. [2007] é a implementação de Cantina (*Carnegie Mellon Anti-phishing and Network Analysis Tool*), um novo contexto baseado em aproximação para detecção de páginas *phishing.* A Cantina examina o conteúdo da página, por exemplo, a URL, e o nome do domínio utilizando o algoritmo TF-IDF.

A ferramenta Cantina pode detectar de 94 a 97% de *sites phishing*, mostrando que é possível utilizar um conjunto de heurísticas com outras ferramentas para reduzir o falso positivo. Com a utilização do algoritmo TF-IDF é possível detectar cerca de 97% sites *phishing* com apenas 6% de falso positivos e, combinando algumas heurísticas a taxa é de 90% de sites *phishing*, com apenas 1% de falsos positivos.

Apesar da utilização de técnicas mescladas no emprego de aprendizagem de máquina

e algoritmo de recuperação de informação, a taxa de falsos positivos se equipara [Whittaker, 2010]. Porém, a taxa de detecção entre 94 e 97% deixa em torno de 6% os sites classificados indevidamente como legítimos.

Como salientado por Miayamoto *et al*. [2009], a detecção da precisão em soluções baseadas em heurísticas estão longe de serem ideais, pois para melhorar a precisão utilizaram heurísticas com aprimoramento do cálculo da probabilidade. Nesse sentido, para melhorar a detecção da previsão, os autores empregaram *AdaBoost*, uma técnica de AM, como método para calcular a probabilidade da previsão, e assim apresentar resultados melhores.

Os autores utilizaram técnicas de aprendizagem de máquina, tais como: *AdaBoost*, *Bagging*, SVM, CART, RL, Floresta Aleatória, Redes Neurais (NN), *Naive Bayes* (NB) e BART. A base de dados foi composta baseada no critério estabelecido por CANTINA [Zhang, *et al*. 2007].

Nesta base de dados deve haver a mesma quantidade de *sites phishing* e legítimo, respectivamente. Nesse sentido, a base de dados foi composta por 1.500 de *sites phishing* e 1.500 de *sites* legítimos, obtendo taxa de falso positivo de 13,64%, com NB, sendo a menor taxa de falso negativo 13,54 com NN.

Nessa direção, Fette *et al*. [2007] propõem um método intitulado PILFER (do Inglês *Phishing Identification by Learning on Features of Email Received*) que é um algoritmo que identifica *phishing* por meio de AM sobre as características de e-mail recebido.

Um conjunto de dados foi utilizado para treinar e testar a base de dados usando validação cruzada com dez (10) partições para melhor obter a média do resultado, tendo sido utilizado o classificador SVM. Para montagem do conjunto de dados, dois conjuntos de dados disponíveis forma utilizados: *corpora ham (SpamAssassin*) - 6.950 amostras com *e-mails* legítimos e, 860 *e-mails phishing* retirados do *phishincorpus*.

O conjunto de dado PILFER conseguiu um total de 99% de precisão, uma taxa de falso positivo de menos de 1%. Por outro lado, uma taxa de falso negativo de 7 a 8%, o que equivale à metade do *SpamAssassin*.

Nesse sentido, Fette *et al.* [2007] afirmam que é possível detectar *e-mail phishing* com alta precisão usando filtro especializado, assim como a utilização de recursos que são diretamente aplicados nos e-mails *phishing,* diferentemente dos empregados por filtros *spam* de propósito geral. A utilização do algoritmo PILFER remove a interação do usuário,

deixando este sem chance de dispensar diálogos de alertas, além de apresentar taxas de precisão de falso positivo baixo.

No trabalho realizado por Blum et al., [2010] foi explorado uma abordagem em tempo real para detecção de URL de *phishing*. E para isso, foi utilizado recursos de nível da superfície de URLs para treinar algoritmos de aprendizagem com confiança ponderada - Message-Digest algorithm 5 (MD5).

Nesta proposta, cada URL é representada como um vetor de características binária que são alimentadas para o algoritmo de linha , ou seja, a confiança ponderada e realizada durante o treinamento. No momento do teste, as URLs inéditas são mapeadas a este vetor de característica binária. Nesse processo surge um novo vetor que o classifica como (*phishing* ,não- *phishing*).

O sistema adotado neste trabalho é capaz de detectar ameaças emergentes quando aparecem e, posteriormente, podem fornecer maior proteção contra ameaças de zero hora ao contrário tradicional lista negra técnicas que funcionam de forma reativa.

No trabalho realizado por Justin Ma et al., [2009] os autores trabalharam com vários recursos lexicais e características baseada em *host* da URL para a classificação de sites *phishing*, contudo os autores excluíram as características com o conteúdo da página *web* pelas seguintes razões: evitar *download* da página e seguro e o sistema fica mais leva quando somente e analisado a URL.

Neste trabalho, os autores apontam como maior contribuição a utilização de algoritmos *on-line,* que segundo os autores, estes podem processar um grande número de exemplo e são bem mais eficientes que o método baseado em algoritmos em lote, também é salientado que com algoritmos *on-line* é possível adaptar as alterações em URL maciliosas e nas características ao longo do tempo, o que é de extrema importância, pois a cada minutos as características podem mudar.

Os autores utilizaram 12 características, sendo 6 com recursos léxicos e 6 baseada na em URL, estas serão sumarizadas na tabela 4,1 do capítulo 4.

A Figura 3.3 ilustra a arquitetura de coleta de dados, que começa com dois modelos de *URLs* maliciosas e legítimas.

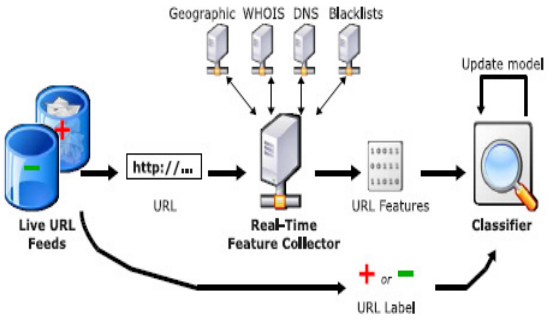

Figura 3.3: Arquitetura empregada na ferramenta utilizada por Ma, Justin et al. [2009].

A figura 2 ilustra a arquitetura de coleta de dados, que começa com duas alimentações de URLs maliciosas e benignas. Foi obtido exemplos de URLs maliciosas a partir de um grande provedor *Web* de *e-mail*, e com alimentação em tempo real pode fornece entre 6.000 - 7.500 exemplos de *spam* e URls *phishing* por dia. As URLs maliciosas são extraídas a partir de mensagens de *e-mail* que são rotuladas pelos usuários manualmente como *spam*, e executado através de pré-filtros para extrair facilmente detectando falsos positivo, e em seguida, verificado manualmente como malicioso.

Como análise a construção de um sistema de detecção de URL maliciosa em tempo real, foi avaliada algoritmos de aprendizagem em lote e *on-line* tendo intuito de verificar benefícios e compensações. Experimentos mais recentes de alimentação de URLs *on-line* revelou as limitações de algoritmos de lote neste cenário, onde foram obrigados a fazer uma troca entre precisão e lidar com recursos limitações (por exemplo, a falta de memória) . Algoritmos on-line são capazes de alcançar classificação precisões de até 99%.

3.4 Discussão

Soluções apresentadas nesta seção para detecção de *phishing* são destacadas na Tabela 3.1 a seguir:

Tabela 3.1 Soluções para detecção de *phishing.*

Publicações	Métodos e Técnicas	Resultados em AM
Fett, Ian *et al*., (2007)	Floresta Aleatória	Precisão: 99%, FN: 7% e FP: 1%
Likarish,Peter et al., (2008)	Filtro *Bayesian*	100% de acertos em sites *phishing*
Ye, Cao et al., (2009)	*Whitelist, Naive Bayes* (Processo de *Login*)	FN: 100%, FP:0% (no processo de *login*)
Ma, Justin et al., (2009)	*Blacklist* e Regressão Logística	FP: 0,1 e FN: 7,6%
Sheng, Steve, et al. (2009)	*Blacklist* e Heurística	FP: 0% (*nas blacklists*)
Abu-Nimeh et al., (2007)	TF-IDF e Regressão Logística	FP: 3,82%
Whittaker, Colin et al., (2010)	*Blacklits*, Regressão Logística e TF-IDF	FP: 0,1%
Zhang, Yue et al., (2007)	AM, TF-IDF	Acerto: 97% e FP: 6%
Miayamoto et al,. (2009)	*Naive Bayes* e Redes Neurais	Precisão: 99%, FN: 7% e FP: 1%

Legenda: FN (Falso Negativo), FP (Falso Positivo) e AM (Aprendizagem de Máquina).

Capítulo 4

Detecção de *Phishing* em Páginas *Web*

Neste capitulo será apresentado a metodologia utilizada e a descrição das etapas compostas, de forma a explicitar cada passo realizado no experimento.

4.1 Modelagem da base de dados

A metodologia adotada neste trabalho está dividida em cinco etapas: download da página, normalização, extração das características, classificação e análise das características.

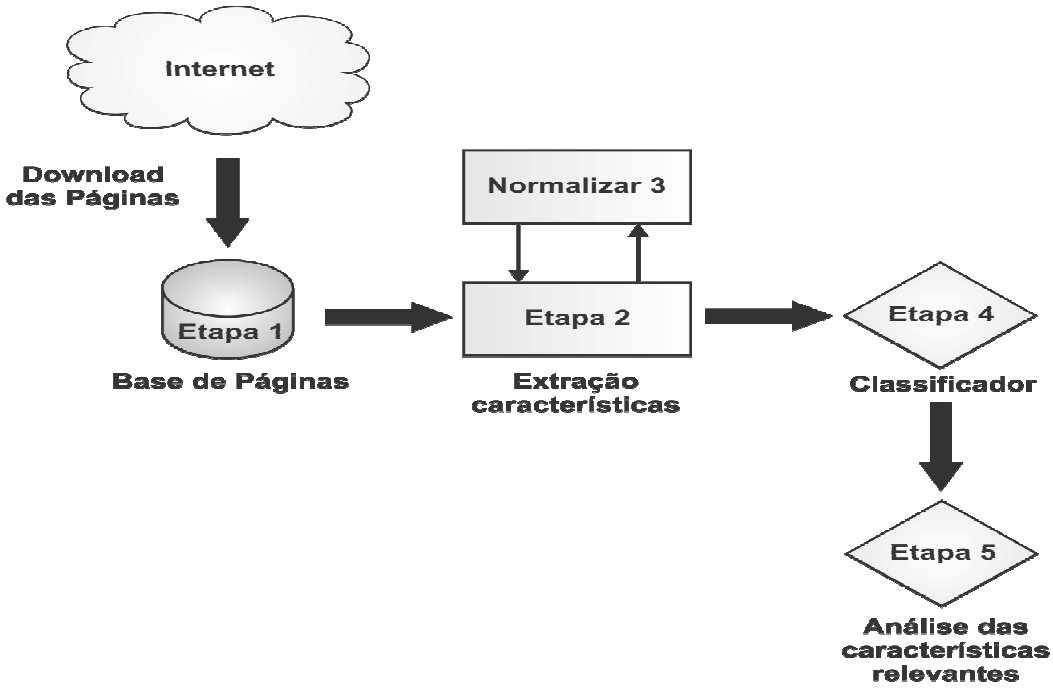

Figura 4.1: Etapas da metodologia adotada.

4.2 Download da página

A análise das páginas pode ser feita tanto em tempo real (*online*), como a partir de arquivos salvos (*off-line*). No entanto, como páginas *phishing* ficam *online* por um curto período de tempo, em média 72 horas, o *download* é necessário. Isto permite que análises posteriores

sejam feitas na página, quando esta já não estiver disponível na Internet.

4.3 Extração das características

A partir dos trabalhos relacionados e da análise das páginas na base de dados, foram selecionadas dezeito características que evidenciam a diferença entre páginas *phishing* e páginas legítimas: URL baseada em IP, quantidade de pontos na URL, tamanho da URL, caracteres suspeitos na URL, domínio de topo, palavras-chave na URL, links para outros domínios, objetos para outros domínios, presença de formulário, palavras-chave no título da página, geolocalização do servidor de hospedagem, *pagerank* da página no Google, caracteres hexadecimais, idade do domínio, anonimizador, algoritmo TF-IFD e velocidade da conexão.

4.3.1 Características extraídas da URL

C1: URL baseada em IP

Algumas páginas *phishing* são hospedadas em máquinas que não têm registro no DNS, a saída para o *phisher* é referenciar pelo IP; isso também previne que o site seja tirado do ar pelo desativamento do domínio. Além disso, entidades legítimas raramente usam IPs em suas URLs.

C2: Quantidade de pontos na URL

Uma das formas de obscurecer a URL e tentar fazer com que os usuários acreditem que a URL seja verdadeira é usar subdomínios, como em http://www.bank.com.br.badsite.com/ - ou utilizar o domínio alvo de *phishing* no caminho, como em http://badsite.com/www.bank.com.br/. Em ambos os casos existe uma grande quantidade de pontos.

C3:Tamanho da URL

Através da análise da base foi verificado que as URLs de páginas *phishing* costumam ter uma quantidade muito maior de caracteres do que as páginas legítimas, numa tentativa de desviar a atenção do usuário.

C4: Palavras-chave na URL

Existem determinadas *strings* que são comuns em URLs de *phishing*, usadas na tentativa de dar segurança ao usuário: account, update, confirm, verify, secur, notif, log, click, inconveninen, ebay, paypal.

C5: Caracteres suspeitos

A ferramenta SpoofGuard identifica dois caracteres encontrados na URL como suspeitos: o "@" (arroba) e, o "-" (underline). Destes, foi utilizado apenas o arroba, o qual é o mais perigoso uma vez que, quando presente no domínio, tudo o que vem antes dele é considerado nome de usuário e repassado ao endereço da página. Assim, ao acessar http://www.bank.com.br@badsite.com/, o usuário pode pensar que está acessando o site http://www.bank.com.br, mas na realidade está navegando em http://badsite.com, sendo o nome de usuário www.bank.com.br.

C6:Domínio de topo (TLD)

Um nome de domínio é composto por uma série de nomes separados por pontos. O último desses nomes é o chamado domínio de topo que pode ser organizado em dois grupos: TLD - com duas letras que representam países (como ".br" que representa o Brasil), e TLD - com mais de duas letras usados para propósitos genéricos (como ".gov" que é de uso restrito para entidades do governo). Domínios podem ser registrados em alguns desses TLD sem restrições enquanto outros devem seguir alguns pré-requisitos, assim determinados TLD podem oferecer maior facilidade de serem usados por *phishers.*

C7: Anonimizador

Alguns *phishers* adicionam URL idênticas do site original, completando com site onde se encontra a página hospedada, normalmente esse site de hospedagem é grátis como no exemplo: http://www.bb.com.br. v10.com.br. Em casos assim, o *phisher* espera que o usuário não perceba o final do site "v10.com.br" e que "bb.com.br" chame mais atenção do usuário;

C8: Quantidade de subdomínios na URL

Alguns *phishers* adicionam subdomínios para dar uma aparência mais confiável à URL utilizando nomes de entidades autênticas e bem conhecidas como no exemplo abaixo:

http://recadastro.receitafederaldobrasil.badsite.com

Em casos assim, o *phisher* espera que os subdomínios "recadastro" e "receitafederaldobrasil" chamem mais atenção do usuário do que o próprio domínio badsite.com.

C9: Caracteres hexadecimais

O caractere "%", quando lido pelo navegador web, indica que os próximos dois caracteres

lidos são hexadecimais. Com este artifício é possível "disfarçar" alguns caracteres. Contudo, conforme Pan & Ding [2006] muitos sites legítimos utilizam notação hexadecimal para representar símbolos de pontuação como aspas, caractere de espaço, ponto de interrogação, etc., assim essa característica só faz sentido quando o valor hexadecimal é uma letra ou número inválido.

4.3.2 Características extraídas a partir de informações em bases de dados *online*

Informações sobre sites são armazenadas por diversas companhias em bases de dados com finalidades específicas, como prover informações de rastreamento e indexação. Essas informações por si não podem comprovar que uma página é *phishing*, mas podem reforçar evidências encontradas em outras características. Duas características desse tipo foram extraídas:

C10: Geolocalização da página

A hospedagem de páginas *phishing* pode se concentrar em determinadas regiões do planeta. Essa característica foi implementada a partir da base de dados fornecida pela empresa MaxMind (2011), onde blocos de IPs estão relacionados aos países.

C11: Google PageRank™

O pagerank é um valor numérico que representa a importância de uma página num conjunto de páginas *web*. Quanto maior o *pagerank* de uma página, mais importante ela é. Mas como páginas *phishings* têm um curto período de vida, seu *pagerank* é muito baixo ou inexistente.

C12: Idade do domínio

Páginas *phishing* normalmente têm um curto período de vida. Os domínios são registrados poucos dias antes dos *e-mails phishing* serem enviados. Nós sinalizamos páginas que foram registradas a menos de 60 dias da data da coleta ou páginas em que essa informação está indisponível. Foi feita uma busca WHOIS para implementar essa característica.

4.3.3 Características extraídas a partir do conteúdo da página

Mesmo que a URL pareça legítima, é possível determinar se uma página é *phishing* analisando o conteúdo da mesma. Para isso, foram extraídas quatro características a partir do HTML da página:

C13: Presença de formulário

Através de um formulário com campos para entradas de texto o *phisher* consegue obter os dados pessoais das vítimas. Caso a tag <INPUT> esteja presente no HTML da página e for do tipo "textfield" ou "password" a página é sinalizada. Páginas legítimas que têm essa característica são facilmente distinguíveis de *phishing* através das outras características.

C14: Razão de *links* para outros domínios

Normalmente os *links* em uma página apontam para o mesmo domínio. Em páginas *phishing*, na tentativa de ficar o mais parecido possível com a página real, os *links* normalmente apontam para o domínio real. Nesta característica é calculada a razão R_1 conforme equação abaixo:

$$R_1 = \frac{L_O}{L} \tag{4.1}$$

onde L_O é a quantidade de *links* para outros domínios e L a quantidade total de *links* na página.

C15: Razão de objetos carregados a partir de outros domínios

Uma página *Web* é composta de diversos objetos incluindo imagens, *css, iframes, scripts*, etc. Em uma página comum, a grande maioria desses objetos é carregada a partir do próprio domínio. Em páginas *phishing* é comum os objetos serem carregados a partir do site real. A razão R_2 entre a quantidade de objetos carregados a partir de outros domínios é calculada pela equação abaixo.

$$R_2 = \frac{O_O}{O} \tag{4.2}$$

Onde "O_O" é a quantidade de objetos para outros domínios e "*O"* a quantidade total de objetos na página.

C16: Palavras-chave no título do site

Assim, como nas URLs, algumas *strings* são comuns no título de páginas *phishing.* Foi utilizado o mesmo conjunto de *strings* aplicado à extração dessa característica na URL para a extração no título da página.

C17: Algoritmo TF-IDF

È um algoritmo que calcula o quão é importante um termo para um documento. Esse valor é obtido contando com o número de vezes que o termo aparece no documento, dividido pelo log

da frequência desse termo em todos os documentos. Páginas *phishing* usam termos comuns nas páginas de seus alvos e isso reflete num valor alto de TF-IDF para esses termos. Já em páginas normais esses termos não terão um valor alto, pois não são relevantes.

C18: Velocidade da conexão

Se alguns sites maliciosos tende a residir em máquinas comprometidas, tais como as residenciais (conectada via cabo ADSL), então e adequado registra a velocidade de conexão do host.

4.4 Normalização

É o processo formal passo a passo que examina os atributos de uma entidade, com o objetivo de evitar anomalias observadas na inclusão, exclusão e alteração de registros.

Objetivos

- Minimização de redundâncias e inconsistências;
- Facilidade de manipulações do banco de dados;
- Facilidade de manutenção do sistema de Informação.

4.5 Classificação

Como citado na Seção 2.3, um classificador depois de treinado consegue predizer a qual classe uma amostra não rotulada pertence a partir da leitura do seu vetor de características. O uso desses algoritmos torna a detecção de *phishing* muito mais eficiente, onde ao invés de se criar e atualizar manualmente as regras de filtragem de dados, eles o fazem automaticamente. Neste trabalho foram utilizados os classificadores SVM, *Naive Bayes* e Árvore de Decisão.

4.6 Análise da relevância de cada característica

Os algoritmos de aprendizagem de máquina ajudam a identificar quais são os atributos mais adequados a serem utilizados para tomar decisões. Os objetivos com a eliminação dos atributos irrelevantes são:

- Melhorar o desempenho dos algoritmos de AM;
- Simplificar o modelo de predição e reduzir o custo computacional;
- Fornecer um melhor entendimento sobre os resultados.

Tabela 4.1: Sumarização dos modelos de detecção de *phishing*.

Características	Autores Relacionados								
	Fett et al., (2007)	Abu-Nimeh et al., (2007)	Zhang et al., (2007)	Likarish et al., (2008)	Ye et al., (2009)	Ma et al., (2009)	Sheng et al. (2009)	Miayamoto et al,. (2009)	Whittaker et al., (2010)
C1:URL baseada em IP	X		X			X		X	X
C2: Quantidade de pontos na URL	X		X					X	X
C3:Tamanho da URL	X		X			X			X
C4: Palavras-chave na URL	X		X			X			X
C5: Caracteres suspeitos			X					X	X
C6:Domínio de topo (TLD)						X			
C7: Anonimizador						X			X
C8: Quantidade de subdomínios na URL									
C9: Caracteres hexadecimais									X
C10: Geolocalização da página						X			X
C11: Google PageRank™			X						X
C12: Idade do domínio	X		X		X	X		X	X
C13: Presença de formulário			X		X			X	X
C14: Razão de links para outros domínios									
C15: Razão de objetos carregados a partir de outros domínios								X	
C16: Palavras-chave no título do site	X							X	
C17: Algoritmo TF-IDF		X	X					X	X
C18: Velocidade da conexão						X			
Não Documentadas		X		X		4	X(5)		
Total de Caracteristicas	6	2	9	1	2	12	4	8	12
Tamanho da Base de dados									
Phishing	860	1718	100	não mencionada	18	6000-7500	191	1500	não mencionada
Não Phishing	6950	1171	100	não mencionada	16	2 milhões	não mencionada	1500	não mencionada
Base de Dados (Phishing e Não phishing respectivamente)	SpamAssassin e Phishingcorpus	Spambase	PhishTank e 3Sharp's	Phishtrack e Alexa	PhishTank e coletada manualmente	Não mencionada e Yahoo!	Universidade de Alabama	PhishTank, 3Sharp, Alexa Web Search e Yahoo!	Google

Capítulo 5

Experimentos e Análise dos Resultados

Nesta seção são apresentados detalhes sobre o ambiente utilizado para a realização dos experimentos, incluindo os parâmetros e as configurações do classificador SVM, Naive Bayes e Árvore de Decisão e filtros *Wrapper*, InfoGain e *CfsSubSet* na análise da relevância das características implementadas. Todos os experimentos foram desenvolvidos em uma estação de trabalho Core i7, 2,67 MHz, com 4GB de memória e um disco SATA com 500 GB de espaço de armazenamento. Os algoritmos para aprendizagem de máquina, filtros de análise das características foram executados através da ferramenta *Weka*.

5.1 Experimentos

5.2 Base de Dados

Como não existe uma base de dados de *phishing* normalizada, ou seja, rotulada e com as características extraídas, foi necessário preparar uma base de cunho próprio. Esta base contém 20.000 páginas, sendo que 10.000 amostras de *phishing* e 10.000 amostras de páginas legítimas. As amostras de *phishing* foram coletadas a partir do repositório *PhishTank* [2011] entre os dias 01/12/2012 a 01/02/2013 amostras de páginas legítimas foram sorteadas da lista composta por mais de 4 milhões de páginas fornecida pelo *Open Directory Project* [2010] e *Clueweb* 2009.

A construção da base de dado adotou critérios para escolher URLs, com base nos critério da CANTINA, foi coletadas URLs com o mesmo número de* phishing *e sites legítimos, conforme ilustrado na figura 5.1.

5.3 Métricas

5.3.1 Desempenho geral

Quatro métricas foram utilizadas para avaliar o desempenho da classificação:

a) Precisão: definida como a razão entre quantidade de páginas *phishing* corretamente classificadas e a quantidade total de páginas classificadas como *phishing*.
b) Taxa de verdadeiros positivos: corresponde à razão entre a quantidade de páginas *phishing* corretamente classificadas e quantidade total de páginas *phishing*.
c) Taxa de falsos positivos: calculada pela razão entre o número de páginas legítimas classificadas como *phishing* e a quantidade total de páginas legítimas.
d) Taxa de acertos: é a razão entre a quantidade de páginas corretamente classificadas e o número total de páginas.

5.4 Resultados

Capítulo 6

Conclusões e Trabalhos Futuros

Este trabalho apresentou um método para classificação e detecção de *Phishing* em páginas *web* baseado na extração de características relevantes do conteúdo estático do documento *web* e da URL, empregando uma abordagem apoiada em técnicas de aprendizagem de máquina.

Referências

Anti-Phishing Working Group. (2010) “Phishing Activity Trends ReportQ1 2010”, Disponível em: http://www.antiphishing.org/reports/apwg_report_Q1_2010.pdf.

Abu-Nimeh, S., Nappa, D.; Wang, X.; and Nair, S. (2007) “A Comparison of Machine Learning Techniques for Phishing detection”, In: Proceedings of the Anti-phishing Working Groups 2nd annual eCrime Researchers Summit (eCrime '07), pp. 60-69.

Alpaydim, E. (2004) “Introduction to Machine Learning” The MIT Press. Cambridge, Massachusetts, EUA. 415 pp.

Basnet, Ram ; Mukkamala, Srinivas; Sung, Andrew H.(2008) “Detection of Phishing Attacks: A Machine Learning Approach” , SPRINGER, Verlag Berlin Heidelberg ,pp. 373–383.

Changxin, Gao. Phishing websites rake in $3 billion. China Daily, Shanghai, 15 janeiro 2011. Disponível em:

<http://www.chinadaily.com.cn/china/2011-01/15/content_11859319.htm>. Acesso em: 25 Novembro 2011.

CERT.br – Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil. (2011) *Estatísticas do CERT.br*. Disponível em: <http://www.cert.br/stats/incidentes>. Acesso em: 16 Novembro de 2011.

Código Penal Brasileiro, Título II, Cap. VI, Art. 171, Disponível em: <http://www.planalto.gov.br/ccivil_03/Decreto-Lei/Del2848.htm>. Acesso em 05 Jan. 2012.

Kuo, Cynthia; PARNO, Bryan; PERRIG, Adrian. Browser Enhancements for Preventing Phishing Attacks. Disponível em: <http://research.microsoft.com/pubs/138297/browser.pdf> Acesso em: 12 outubro 2011.

Kumaraguru, Ponnurangam., Sheng, Steve., Acquisti, Alessandro., Cranor Lorrie F., Hong, Jason (2008), “Lessons From a Real World Evaluation of Anti-Phishing Training” In: *IEEE International Conference System* , eCrime Researchers Summit, pp. 1-12.

Downs, Julie S.; Holbook, Mandy B.; Cranor, Lorrie. (2006) “Decision Strategies and Susceptibility to Phishing”. In: *Proceedings of the second symposium on Usable privacy and security* (SOUPS '06). ACM, New York, pp. 79-90.

Fette, Ian; Sadeh, Norman; Tomasic, Anthony. (2007) “Learning to Detect Phishing Emails”. In: International Conference On World Wide Web, ACM, New York, 2007, 16th, n. 1357, pp. 649-656.

Garera, Sujata et al. (2006) “A Framework for Detection and Measurement of Phishing Attacks” In: International Conference On World Wide Web, ACM, New York, 16, pp.1-8.

Loftesness, S. (2004) *Responding to "Phishing" Attacks*.Glenbrook Partners.

Litan, A. (2004) *Phishing Attack Victims Likely Targets for Identity Theft*. Gartner Research.

Likarish, Peter et al. (2008) “B-APT: Bayesian Anti-Phishing Toolbar” In: IEEE International Conference on Communications, Beijing, pp.1745 – 1749.

Chandrasekaran , M.; Narayanan, K. ; Upadhyayas. (2006) “Phishing email detection based on structural properties” In: NYS Cyber Security Conference.

Miyamoto, D.; Hazeyama, H.; Kadobayashi, Y. (2009) “An Evaluation of Machine Learning-Based Methods for Detection of Phishing Sites” In: Proceedings of the 15th International Conference on Advances in Neuro-Information Processing, vol. 1, pp. 539-546.

Ma, Justin et al. (2009) “Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs” In: *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining* (KDD '09). ACM, New York, pp.1245-1254.

McMillan, R., Gartner: Consumers to lose $2.8 billion to phishers in 2006, *etworkWorld*, 2006.Disponível em: http://www.networkworld.com/news/2006/110906-gartnerconsumers-to-lose-28b.html. Acesso em Setembro 2010.

N. Chou, R. Ledesma, Y. Teraguchi, and J. C. Mitchell. Client-side defense against web-based identity theft.” in *NDSS*, 2004. [Online]. Disponível em: http://www.isoc.org/isoc/conferences/ndss/04/proceedings/Papers/Chou.pdf. Acesso em Outubro 2011.

Netcraft toolbar, 2006. [Online]. Disponível em: <http://toolbar.netcraft.com/>

Opendns. PhishTank. Disponível em: <http://www.phishtank.com/>. Acesso em 20 out 2011.

SpoofGuard. Disponível em: <http://crypto.stanford.edu/SpoofGuard/>. Acesso em: 30 outubro 2011.

Phelps, T.A.; R. Wilensky. Robust Hyperlinks and Locations, *D-Lib Magazine*, vol. 6(7/8), 2000.Disponível em: http://www.dlib.org/dlib/july00/wilensky/07wilensky.html

Quinlan, J. R. (1993). “C4.5, Programs for machine learning”. Morgan Kaufmann, San Mateo, Ca.

Richard, Clayton. Insecure real world authentication protocols (or why is phishing so profitable), 2005. Disponível em: http: //www.cl.cam.ac.uk/users/rnc1/phishproto.pdf. Acesso em Dezembro 2011.

Rosiello, Angelo P. E., et al. A Layout-Similarity-Based Approach for Detecting Phishing Pages. In: International Conference ON Security And Privacy Communication Networks, Nice, 3, 2007, p. 454 – 463.

R. Dhamija, J. D. Tygar ; M. Hearst. (2006) ” Why phishing works” In: Proceedings of the SIGCHI conference on Human Factors in Computing Systems, pp. 581–590.

Sheng, Steve, et al. (2009) “An Empirical Analysis of Phishing Blacklists” In: Conference On Email and Anti-Spam,6, Mountain View.

Timko, D. (2008). “The social engineering threat” In: Information Systems Security Association Journal.

Whittaker, Colin; Ryner, Brian; Nazif, Marria. (2010) “Large-Scale Automatic Classification of Phishing Pages” In: Network and Distributed System Security Symposium ,17, San Diego. Proceedings.

Wu, Min.(2006) “Fighting Phishing at the User Interface”. Ph.D. Dissertation. Massachusetts Institute of Technology, Cambridge, MA, USA.

Ye, Cao; Weili, Han; Yueran, Le.(2008) “Anti-phishing Based on Automated Individual White-

List" In: *Proceedings of the 4th ACM workshop on Digital identity management (DIM '08),* ACM, New York, pp.51-60.

Zhang, Yue; Hong, Jason; Cranor, Lorrie. (2007) "CANTINA: A Content-Based Approach to Detecting Phishing Web Sites" In: International Conference On World Wide Web, ACM, New York, n.1357, pp. 639-648.

Zhang, Y., Egelman, S., Cranor, L., Hong, J. Phinding Phish (2007) "Evaluating Anti-Phishing Tools" In: Proceesings of the 14th Annual Network and Distributed System Security Symposium *(NDSS'07).*

Zhang, Jianyi, Luo Shoushan, Gong, Zhe, Ouyang, Xi, Wu, Chaichua, Xin, Yan. (2011)."Protection Against Phishing Attacks: A Survey" …

A Phishing Sites Blacklist Generator

Mohsen Sharifi and Seyed Hossein Siadati

2008

On the Potential of Proactive Domain Blacklisting

Mark Felegyhazi
mark@icsi.berkeley.edu
Christian Kreibich
christian@icir.org
Vern Paxson
vern@icir.org

http://www.antispam.br/faq/#10

http://windows.microsoft.com/pt-br/windows-vista/phishing-filter-frequently-asked-questions

http://news.netcraft.com/

http://www.cloudmark.com/

http://www.siteadvisor.com/

http://www.symantec-norton.com/Norton_360_p128.aspx?lang=pt-BR&par=goo_br_norton_360&par1=pt_br_norton_360_promotion&gclid=CJWzz5CB2LgCFenm7

AodfCYA0A