

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE



SBSeg 2019

XIX Simpósio Brasileiro de Segurança da
Informação e de Sistemas Computacionais



LaTARC
Research Lab

PhishKiller

Uma Ferramenta para Detecção e Mitigação de Ataques de Phishing
Através de Técnicas de Deep Learning

Cristian Souza (IFRN)

Marcilio Lemos (UFRN)

Felipe Dantas (IFRN)

Robinson Alves (IFRN)

Agenda

1. Phishing: Conceitos básicos;
2. Proposta: Apresentação do PhishKiller e trabalhos relacionados;
3. Avaliação e resultados: Descrição dos testes para validação da proposta;
4. Conclusão: Considerações finais e trabalhos futuros.

1. Phishing

- Tentativa de obter credenciais através da suplantação de identidade por parte dos criminosos.



Etapas de um ataque de phishing



O atacante clona a página original e envia um e-mail com o endereço malicioso para a vítima.



O usuário acessa o endereço.



A página exige o log in do usuário para prosseguir.



O atacante faz uso dos dados da vítima para benefício próprio.



O usuário é redirecionado para o serviço original.



Os dados são enviados para o atacante.

Estatísticas

1. 23% dos internautas brasileiros sofreram ao menos uma tentativa de phishing em 2018;
2. 60% dos ataques simulam instituições financeiras;
3. 58% dos sites maliciosos já utilizam HTTPS;
4. 76% das organizações já presenciaram uma tentativa de phishing;
5. Tentativas de phishing aumentaram 40.9% em 2018.

Características de um URL

`http://sbc.org.br/eventos/calendario-de-eventos`

Protocolo Domínio Top-level domain Caminho



The diagram illustrates the components of the URL `http://sbc.org.br/eventos/calendario-de-eventos`. Brackets and labels identify the following parts: `http:` is labeled as 'Protocolo'; `://` is part of the protocol; `sbc` is labeled as 'Domínio'; `.org.br` is labeled as 'Top-level domain'; and `/eventos/calendario-de-eventos` is labeled as 'Caminho'.

Características de um URL malicioso

- Na tentativa de enganar os usuários mais desatentos, o atacante adiciona elementos do URL original ao endereço malicioso.
- Há um padrão na maioria dos endereços maliciosos:
 - Grande quantidade de dígitos;
 - Muitos caracteres repetidos sequencialmente ou muito aleatórios;
 - Muitos subdomínios.
- Exemplos (fonte: PhishTank):
 - `http://appleid-appleupdatesec.com-appidkey4834668.com/manage/`
 - `http://mhammadrizwan1600.000webhostapp.com/paypal.php`
 - `https://paypal.co.uk.os3z.icu/n/`

2. PhishKiller

- O presente trabalho tem como objetivo prover uma solução para detecção e mitigação de ataques de phishing.

Para isso, o mecanismo deve:

- Detectar os ataques no lado do cliente (*client-side*);
 - Alcançar um alto grau de generalização;
- Impedir o acesso ao site malicioso por parte do usuário;
- Não exigir esforço por parte do usuário para que o mesmo seja protegido.

Trabalhos relacionados

Proposta	#1	#2	#3
[Sahingo et al. 2019]	✓		97.98%
[Jain and Gupta 2018]	✓		99.09%
[Tyagi et al. 2018]			98.40%
[Niakanlahiji et al. 2018]	✓		95.40%
[Desai et al. 2017]		✓	96.11%
[Marchal et al. 2014]			94.91%
[Belabed et al. 2012]			98.40%
PhishKiller	✓	✓	98.30%

Legenda:

#1: a solução não faz consultas a servidores WHOIS ou indexadores de sites;

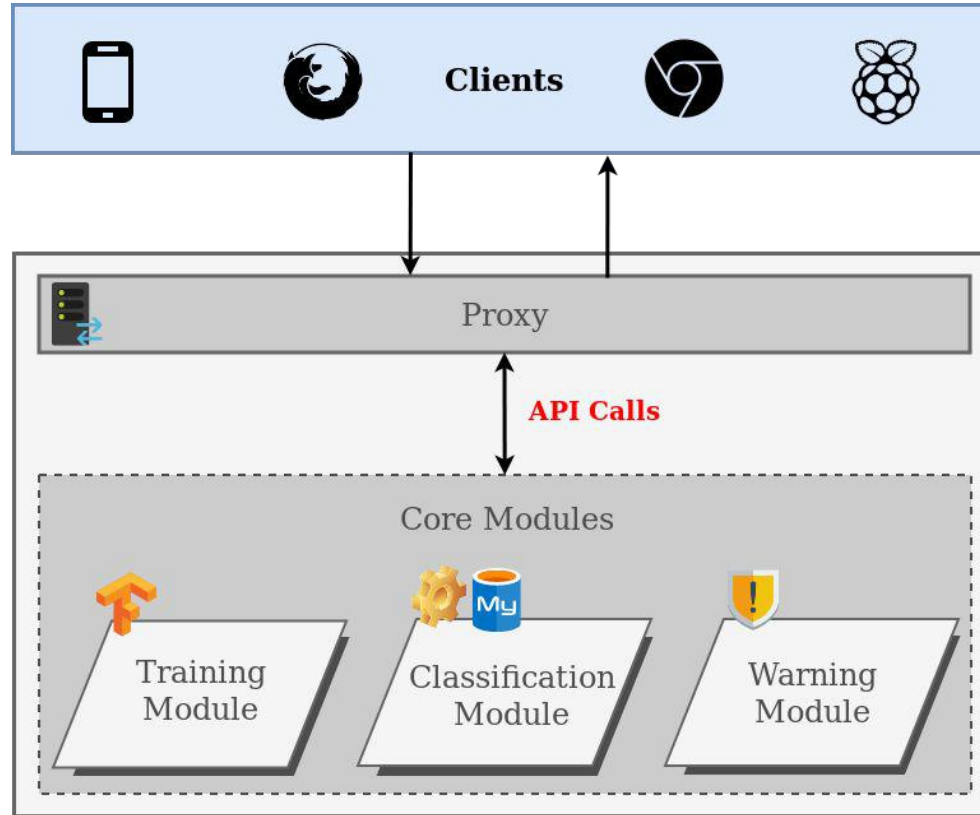
#2: propõe alguma ferramenta para efetivamente mitigar o ataque;

#3: porcentagem de acertos do mecanismo avaliado.

Trabalhos relacionados: Classificação

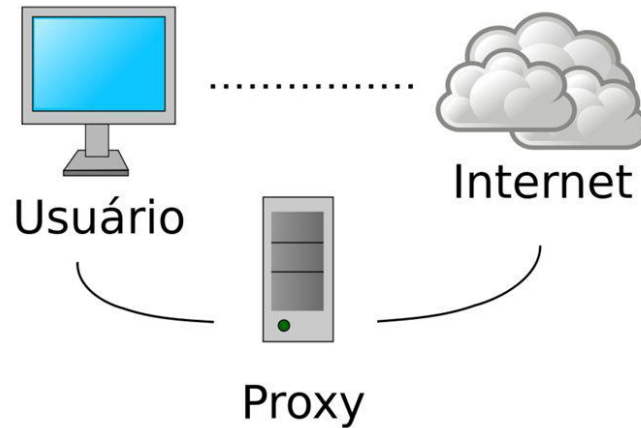
1. Sahingoz et al. (2019): Processamento de linguagem natural / *random forest*;
2. Jain and Gupta (2018): *Random Forest*;
3. Tyagi et al. (2018): *Random Forest e Principal Component Analysis*;
4. Niakanlahiji et al. (2018): *Random Forest*;
5. Desai et al. (2017): *Random Forest*;
6. Marchal et al. (2014): Regressão linear;
7. Belabed et al. (2012): *Support Vector Machine*;
8. **PhishKiller (2019)**: *Long short-term memory (LSTM)*.

Arquitetura do PhishKiller



Proxy

- Configurado no navegador do cliente.
- É responsável por interceptar as requisições para envio do URL ao classificador.
- Suporte ao HTTPS (com auxílio do *framework* Tornado).



Training Module

- Responsável por realizar o treinamento da rede neural.
- Utiliza o TensorFlow e Keras como *back-end*.
- Pré-processamento com Word2Vec (CBOW).
- Técnicas empregadas: Modelo de convolução 1D e LSTM.
- O *dataset* utilizado tem 452.835 endereços, dos quais 108.013 são de phishing e 344.829 são autênticos.
- 80% do *dataset* foi utilizado para treinamento e 20% para testes.



Classification Module

- É requisitado sempre que um URL é detectado pelo proxy.
- Consulta se o URL já existe no banco de dados.
- Em caso negativo, faz a predição do endereço (maligno ou benigno).
- Ao encontrar um endereço malicioso, salva o mesmo no banco de dados para evitar futuras classificações desnecessárias.

Warning Module

- Responsável por exibir uma página com informações ao usuário.



Phishing attack detected!

We have detected that the site you are trying to access is possibly a fraud attempt.

URL: [bb.tk/login.php](#)

Probability of a phishing attack: **98.77%**

3. Avaliação e resultados

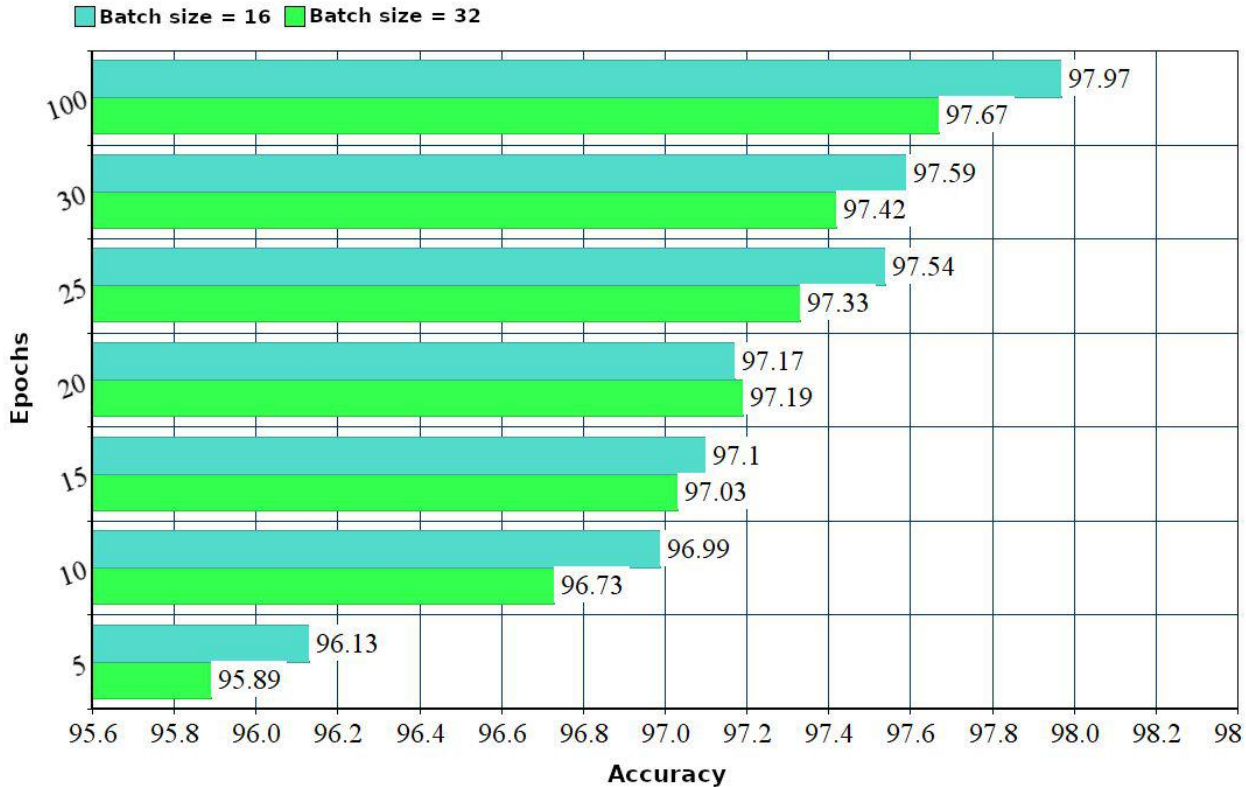
→ Ambiente para testes:

- ◆ Dell Inspiron I14-5457-A30
- ◆ 2.40 GHz Intel Core i7-6500U
- ◆ 8GB RAM
- ◆ Ubuntu 18.04.02 LTS
- ◆ 30Mbps Internet link
- ◆ curl v7.64.0

→ Ambiente de treinamento:

- ◆ VM com Intel Xeon E5-2620 2.40 GHz (8 núcleos)
- ◆ 16 GB RAM
- ◆ Ubuntu 18.04.02 LTS

3. Avaliação e resultados



3. Avaliação e resultados

Épocas	Tempo <i>batch size</i> = 16	Tempo <i>batch size</i> = 32
5	38.96	23.58
10	69.13	46.70
15	103.53	69.51
20	139.96	95.12
25	175.83	116.86
30	205.54	142.44
100	884.32	440.57

3. Avaliação e resultados

- Acurácia nos 20% do *dataset*: **97.59%**.
 - 30 épocas e *batch size* de 16.
- Acurácia em testes automatizados com o *curl*: **98.3%**.
 - URLs: 1000 malignas e 1000 benignas.
- Tempo médio para classificar um endereço: **81.68ms**.

Demonstração

4. Conclusão

- Neste trabalho foi proposto o PhishKiller, uma ferramenta para detecção e mitigação de ataques de phishing.
- O mecanismo faz uso de um proxy, que captura os sites acessados e envia seus endereços para classificação baseada em deep learning.
- A solução apresenta uma boa generalização, com acurácia de 98.3% e *delay* de 81.68ms.

4. Conclusão

- Alguns problemas:
 - As técnicas empregadas exigem uma grande base de dados;
 - A acurácia pode cair significativamente no caso de um *dataset* pouco criterioso;
 - Tempo de treinamento elevado.

- Trabalhos futuros:
 - Adaptação do sistema para o contexto de Software-defined networking (SDN).



SBSEg 2019

XIX Simpósio Brasileiro de Segurança da
Informação e de Sistemas Computacionais



LaTARC
Research Lab

Agradecimentos

- Os autores agradecem ao CNPq e ao IFRN pelo fomento ao desenvolvimento do presente trabalho.
- Agradecemos também à Comissão Especial em Segurança da Informação e de Sistemas Computacionais (CESeg) pelo subsídio da inscrição para participação no SBSEg 2019.