Análise de categorizações de URL's com o algoritmo de árvore de decisões a partir de uma base de dados predefinida

Cleiton Ricardo Copceski, Ricardo Augusto Manfredini

[1] Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul (IFRS)

Campus Farroupilha - Farroupilha, RS - Brasil

cleiton.copceski@hotmail.com

Resumo

Este projeto tem como intuito utilizar o algoritmo de árvore de decisões para predições com URL's que acessamos no navegador todos os dias para uma possível integração com ferramentas de segurança, como por exemplo: Firewalls e antivírus. Aplicando a linguagem Python, foi desenvolvido o algoritmo para o carregamento de uma base de dados com as URLs e as respectivas classificações, e com estes dados já carregados e tratados foi possível utilizar o algorítmo estudado para aplicar as predições desejadas, onde podemos, por exemplo, realizar uma avaliação dos dados retornados para futuras aplicações Web e sabermos as melhores práticas para não sermos confundidos com URL's classificadas como Phishing pelas ferramentas de proteção disponíveis no mercado. As ferramentas utilizadas para chegar neste resultado foram, o Google Colaboratory para realizar a execução do algoritmo que necessita de demasiado processamento e a linguagem Python para desenvolvimento do algoritmo em questão.

Abstract

This project aims to use the decision tree algorithm for predictions with URLs that we access in the browser every day for a possible integration with security tools, such as: Firewalls and antivirus. Applying the Python language, an algorithm was developed to load a database with the URL and its respective classification, and with these data already loaded and treated, it was possible to use the studied algorithm to apply the desired predictions, where we can, for example, carrying out an evaluation of the data returned for future Web applications and knowing the best practices so as not to be confused with URLs classified as Phishing by the protection tools available on the market. The tools used to reach this result were Google Collaboratory to carry out the execution of the algorithm that requires too much processing and the Python language to develop the algorithm in question.

1. Introdução

Este projeto de conclusão de curso tem como objetivo principal desenvolver uma ferramenta de aprendizado de máquina, onde realiza-se a predição da classificação de URL's carregadas para o algoritmo. É de suma importância abordar o tema segurança da informação, este que é cada vez mais importante e que está aninhado ao objetivo deste trabalho, já que nos dias de hoje nossos dados acabam sendo uma forma rentável de utilização para marketing em redes sociais e em outras plataformas.

O desenvolvimento do trabalho tem como premissa a possível aplicação em outras ferramentas de segurança, como Endpoints ou Firewalls para prevenção de intrusões ou melhorias em utilização de URL's em nossos redirecionamentos em softwares Web.

Através deste projeto o usuário consegue treinar o algoritmo para que possa importar as próprias URL's e verificar a sua confiabilidade comparando-a a outras semelhantes, para assim entender como tomar a ação de prevenção seguinte. O arquivo CSV com as URL's e a suas respectivas classificações está disponível na plataforma *Kaggle*[1] para estudos e o mesmo é atualizado anualmente. Com esta base de dados e o algoritmo de árvore de decisões foi possível realizar a abordagem pretendida, visto que a confiabilidade dos dados e da predição realizada está de acordo com o que procurávamos, com uma confiabilidade próxima dos 97%[2].

As ferramentas utilizadas no desenvolvimento deste trabalho foram o *Python* e o *Google Colaboratory,* e os temas abordados foram escolhidos por conta da área de atuação que o autor se encontra nos dias de hoje, a segurança da informação de pequenas e médias empresas. O intuito é realmente entendermos os riscos e a alta probabilidade de cairmos em fraudes apenas clicando em links que se passam por uma outra URL mudando alguns caracteres.

Hoje no mercado, alguns equipamentos de proteção já fornecem ferramentas parecidas. Com um grande banco de dados, os Firewall's da marca *Sophos*, por exemplo, trazem a ferramenta de categorização de URL's embarcada no próprio *Firmware* do equipamento, sendo um facilitador para o administrador de redes na hora de realizar algum

[1] https://www.kaggle.com/datasets/sid321axn/malicious-urls-dataset
[2] Retirado do algoritmo por meio do método logit.score( )

tipo de filtragem ou prevenção. Isso acaba fazendo parte do dia a dia de um analista de segurança da informação dentro de um ambiente corporativo, e entender como a predição de dados pode nos auxiliar neste passo decisivo na hora da prevenção, é o essencial para caminharmos lado a lado com a tecnologia.

2. Referencial Teórico

Este trabalho abordará os conceitos de *Cyber Security, Lei Geral de Proteção de Dados, Firewall, IPS, URL, Decision Tree, Python* e *Machine Learning*.

2.1. Cyber Security

A cibersegurança, ou *Cyber Security*, é uma área fundamental no mundo digital atual, uma vez que a rápida evolução tecnológica e a grande conectividade entre aparelhos apresentam novos desafios e riscos à segurança da informação na grande maioria de acessos que realizamos diariamente.

A crescente sofisticação dos ataques cibernéticos também acaba sendo um desafio constante para os profissionais de segurança da informação. Criminosos utilizam uma grande variedade de técnicas, como *phishing*, *ransomware, malware* e ataques de engenharia social, para explorar vulnerabilidades em sistemas e obter acesso não autorizado a informações confidenciais. Segundo o Relatório de Ameaças Cibernéticas da *Symantec* de 2022, foram identificados mais de 5,6 bilhões de ataques cibernéticos em 2021, um aumento significativo em relação aos anos anteriores (TREND MICRO, 2022). Além disso, a expansão da Internet das Coisas (IoT) apresenta novos desafios em termos de segurança. Visto que agora temos muitos outros dispositivos conectados à rede que antes não estavam integrados, como câmeras de segurança, dispositivos domésticos inteligentes e até mesmo alarmes de segurança residencial, que são alvos frágeis e que possuem uma grande quantidade de informações sensíveis, acaba comprometendo assim, a privacidade e a segurança dos usuários.

Para enfrentar os desafios da cibersegurança, precisa-se adotar uma abordagem variada e complexa, envolvendo medidas técnicas, educacionais e regulatórias. Em termos técnicos, a

implementação de firewalls, sistemas de detecção e prevenção de intrusões (IDS – *Intrusion Detection System* e IPS – *Intrusion Prevention System*) e criptografia de dados são fundamentais para proteger sistemas e redes contra ataques.

> O IPS geralmente fica diretamente atrás do firewall e fornece uma camada complementar de análise que seleciona negativamente conteúdo perigoso. Ao contrário de seu antecessor, o Sistema de Detecção de Intrusão (IDS) — que é um sistema passivo que verifica o tráfego e informa sobre ameaças — o IPS é colocado em linha (no caminho de comunicação direta entre fonte e destino), analisando ativamente e tomando ações automatizadas em todos os fluxos de tráfego que entram na rede *(RANGEL, Rafael).*

Ademais, para identificarmos comportamentos maliciosos, precisamos cada vez mais utilizarmos soluções de segurança baseadas em inteligência artificial e aprendizado de máquina, pois só assim teremos rapidamente os padrões de comportamento maliciosos e as respostas a ameaças em tempo real. Segundo a Trend Micro, marca renomada no mercado de segurança da informação, que divulgou o relatório *Fast Facts* com análise do panorama mundial de ameaças cibernéticas de janeiro de 2022, o Brasil é o quarto país no mundo com mais ataques disseminados por e-mail (SECURITY REPORT, 2022).

A conscientização e a educação dos usuários também são fundamentais para fortalecer a segurança cibernética. Treinamentos e campanhas de conscientização podem ajudar a promover boas práticas de segurança, como o uso de senhas fortes, a não abertura de anexos ou links suspeitos e a atualização regular de software e sistemas operacionais. Bruce Schneier, especialista em segurança cibernética já havia dito: "A segurança é um processo, não um produto" (SCHNEIER, 2000)[3].

Os governos também possuem um papel importantíssimo no processo de segurança cibernética, pois é de suma importância que se desenvolvam leis e regulamentos que incentivem a adoção de boas práticas e responsabilizem empresas e organizações por violações. A implementação de normas de conformidade, como a LGPD (Lei Geral de Proteção de Dados), é um passo importante para garantir a privacidade e a proteção dos dados pessoais dos indivíduos.

[3] Security is a process, not a product.

2.2. Lei Geral de Proteção de Dados

A Lei Geral de Proteção de Dados (LGPD) é uma legislação brasileira que tem como objetivo garantir a privacidade e proteção dos dados pessoais dos brasileiros. Foi aprovada em 2018 e entrou em vigor em setembro de 2020. Ela estabelece diretrizes para a coleta e armazenamento, processamento e compartilhamento de dados pessoais, sempre por parte de empresas e organizações, que são detentoras da maior quantidade de informações dos cidadãos.

Foi inspirada em outras leis internacionais de proteção de dados, como por exemplo, o Regulamento Geral de Proteção de Dados (GDPR) da União Europeia. A sua criação se deve à crescente preocupação com a privacidade e segurança dos dados pessoais, especialmente com o crescimento da utilização no contexto digital.

Uma das principais bases da LGPD é o consentimento do indivíduo utilizador. Isso significa que as empresas devem obter o consentimento explícito antes de coletar, usar ou compartilhar seus dados pessoais. Além disso, a lei estabelece que os titulares dos dados têm o direito de acessar suas informações, corrigi-las, solicitar a exclusão, dentre outros direitos. Também impõe obrigações às empresas e organizações que coletam e processam estes dados pessoais. Elas devem adotar medidas de segurança adequadas para proteger essas informações contra acesso não autorizado, vazamento, perda ou qualquer forma de tratamento não esperado.

As empresas são obrigadas a ter um encarregado de proteção de dados (DPO – *Data Protection Officer*), este é responsável por garantir o cumprimento da LGPD dentro da organização. Suponhamos que ocorra algum incidente de segurança que possa resultar em risco ou danos aos titulares dos dados, a empresa fica responsável por comunicar a Autoridade Nacional de Proteção de Dados (ANPD) e os indivíduos afetados. A ANPD é o órgão responsável pela fiscalização e aplicação das penalidades previstas, segundo Alexandre Almeida da Silva, estas podem variar desde advertências até multas, que podem chegar a 2% do faturamento da empresa, mas está limitada a R$ 50 milhões por infração (SILVA, 2023). Cabe salientar que a LGPD se aplica tanto a empresas privadas quanto a órgãos públicos. Resumindo, a qualquer organização que colete, armazene, processe ou compartilhe dados pessoais de residentes

brasileiros.

No geral, a Lei Geral de Proteção de Dados busca estabelecer a necessária proteção de dados no Brasil, assim como o balanceamento entre empresa e indivíduo, não removendo a necessidade das empresas de utilizarem informações pessoais para fins legítimos, e garantindo a privacidade e a segurança dos indivíduos que fornecem os dados a elas.

2.3. Firewall

Firewall é um componente fundamental da segurança de rede que desempenha um papel importante na proteção de sistemas contra ameaças. Ele atua como uma barreira entre a rede interna e a Internet. O objetivo principal de um firewall é controlar o tráfego de rede com base em um conjunto de regras pré-definidas. Essas regras determinam quais tipos de comunicação são permitidos ou bloqueados, com base em critérios como endereço IP, porta de rede ou protocolo.

Existem diferentes tipos de firewalls, incluindo firewalls de rede, firewalls de host e firewalls de aplicativos. O firewall de rede (figura 1) é geralmente implementado em um dispositivo dedicado e monitora e gerencia o tráfego de entrada e saída da rede interna. Ele pode examinar os pacotes de dados e aplicar regras de filtragem para decidir se os pacotes devem ser permitidos ou negados.

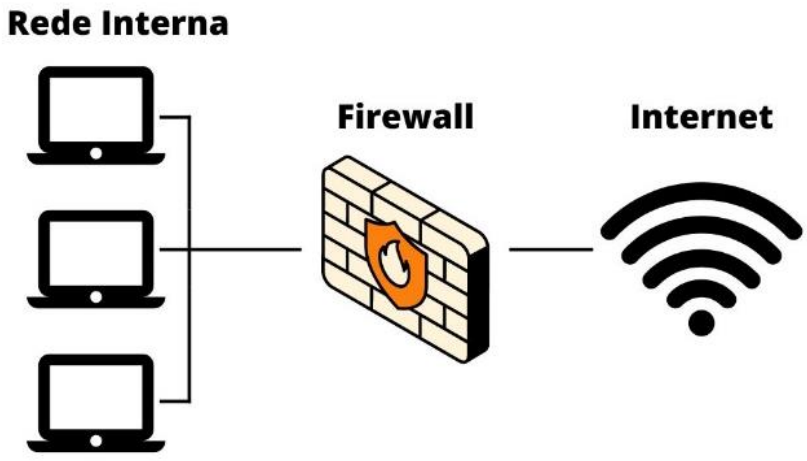

Figura 1 - Firewall de Rede
Fonte: Próprio Autor

O firewall de host (figura 2), por outro lado, é executado diretamente em um computador e controla o tráfego de rede específico desse sistema. Ele pode ser um software instalado no sistema operacional ou até mesmo estar integrado ao próprio sistema operacional, como o Firewall do Windows. É extremamente eficaz para proteger um sistema individual contra ataques externos caso o administrador de rede não possa implementar um Firewall de rede.

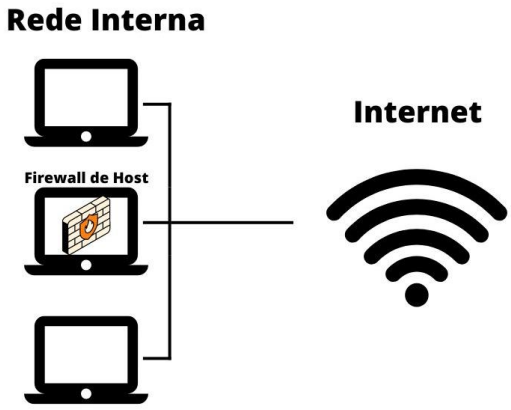

Figura 2 - Firewall de Host
Fonte: Próprio Autor

Os firewalls de aplicativos são projetados para proteger aplicativos específicos, como servidores web ou bancos de dados, alguns firewalls de rede de próxima geração oferecem esta funcionalidade para proteção contra ataques externos à serviços específicos, como o HTTPS e o HTTP.

Além disso, os firewalls podem utilizar diferentes técnicas para filtrar o tráfego de rede, como firewalls de estado, firewalls de inspeção de pacotes e firewalls de próxima geração. Os firewalls de estado mantêm registros do estado das conexões de rede e permitem que o tráfego associado a conexões já estabelecidas seja permitido. Os firewalls de inspeção de pacotes examinam o conteúdo dos pacotes de dados, permitindo uma análise mais profunda do tráfego. Os firewalls de próxima geração combinam várias técnicas de segurança, como inspeção de pacotes, filtragem de aplicativos e prevenção de intrusões.

Em resumo, um firewall desempenha um papel essencial na proteção de redes e sistemas contra ameaças. Ele ajuda a evitar o acesso não autorizado, filtrar o tráfego malicioso e proteger a integridade dos dados.

2.4. IPS – Intrusion Prevention System

O Intrusion Prevention System (IPS), ou Sistema de Prevenção de Intrusões, é uma tecnologia de segurança que atua como uma camada adicional de defesa para proteger redes e sistemas contra ameaças cibernéticas, como ataques de hackers, malwares e intrusões indesejadas.

Ao contrário de um firewall, que controla o tráfego de rede com base em regras predefinidas, um IPS (figura 3) é capaz de analisar o tráfego em tempo real em busca de atividades suspeitas ou maliciosas. Ele monitora o tráfego de rede e examina os pacotes de dados em busca de padrões e comportamentos anormais. Quando o IPS detecta uma atividade que corresponde a um padrão conhecido de ataque ou violação de segurança, ele pode tomar medidas imediatas para bloquear ou prevenir a intrusão. Isso pode incluir o descarte de pacotes de dados maliciosos, a interrupção de conexões suspeitas ou o envio de alertas para os administradores de segurança tomarem ações apropriadas.

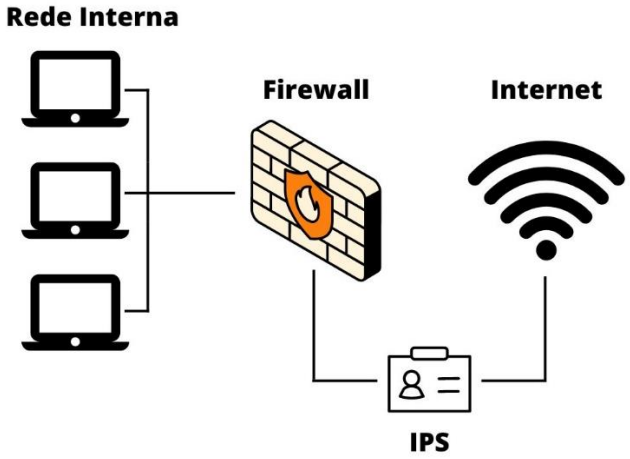

Figura 3 - IPS
Fonte: Próprio Autor

Um IPS pode ser implantado em diferentes pontos da rede, como em roteadores, switches, firewalls, mais comumente encontrado em firewalls de próxima geração. Ele também pode ser implementado em modo de detecção, onde ficará apenas observando o tráfego e fornecendo alertas para o administrador sem bloquear o tráfego, ou em modo de prevenção, onde atua proativamente para bloquear ou mitigar as ameaças.

Além de identificar e bloquear atividades maliciosas, um IPS também pode fornecer recursos de detecção de anomalias, que buscam comportamentos incomuns na rede. Isso ajuda a identificar ameaças desconhecidas ou ataques sofisticados que possam não ser detectados por assinaturas de ameaças conhecidas.

Os IPSs evoluíram muito ao longo do tempo por consequência do aumento significativo de tentativas de ataques, e os sistemas de prevenção de intrusões de próxima geração (NGIPS) incorporam recursos adicionais, como análise comportamental, inteligência artificial e aprendizado de máquina para melhorar a precisão na detecção de ameaças e reduzir os falsos positivos.

2.5. URL

Uma URL (*Uniform Resource Locator*) é uma sequência de caracteres que identifica e localiza um recurso na internet. Ela é usada para especificar o endereço de um recurso específico, como um site, uma página da web, um documento, uma imagem ou um arquivo. As URLs são a base do sistema de endereçamento da web e são utilizadas para acessar informações na internet.

Uma URL consiste em vários componentes principais, o primeiro que iremos abordar é o protocolo. Ele é o esquema usado para acessar o recurso, como HTTP (*Hypertext Transfer Protocol*), HTTPS (*HTTP Secure*), FTP (*File Transfer Protocol*), entre outros. O protocolo define as regras para a comunicação entre o cliente e o servidor.

Depois temos o domínio, que é o nome exclusivo que identifica um site na internet. Por exemplo, na URL "https://www.site.com", o domínio é "site.com". O domínio é registrado e controlado por uma autoridade de registro de domínio, como por exemplo o Registro BR no

Brasil.

O terceiro componente é o subdomínio que pode ser usado ou não, pois uma URL pode conter um subdomínio que precede o domínio principal. Por exemplo, em "https://blog.site.com", "blog" é o subdomínio. Geralmente o subdomínio do site é o próprio ”www”, muito utilizado no início dos anos 2000 para indicar que o site era acessível pela internet.

A extensão de domínio, também chamado de domínio de topo (TLD - *Top-Level Domain*), é o que vem após o domínio. Por exemplo, em "https://blog.site.com”, “com” é a extensão de domínio, que é utilizado por cerca de 47% de todos os sites hoje em dia (W3TECHS, 2023).

O caminho, é outro componente principal, ele é a sequência hierárquica de diretórios ou pastas que leva ao recurso específico. Por exemplo, em "https://www.site.com/ diretorio_raiz/diretorio/index.html ", o caminho seria "/diretorio_raiz/diretorio/index.html".

Já os parâmetros, são informações adicionais fornecidas na URL que podem ser usadas pelo servidor para personalizar a resposta ou o comportamento do recurso solicitado. Os parâmetros são separados do caminho por um ponto de interrogação (?) e podem ser especificados no formato chave=valor. Por exemplo, em "https://www.site.com/index.html?param1=valor1&param2=valor2", os parâmetros são "param1=valor1" e "param2=valor2".

E por fim, a âncora, que acaba sendo um identificador dentro de uma página da web que permite navegar para uma seção específica do documento. A âncora é indicada pelo símbolo "#" seguido pelo nome da âncora. Por exemplo, em "https://www.site.com/index.html#secao", a âncora é "secao".

Visualmente podemos exemplificar conforme abaixo (figura 4), onde lê-se *Scheme* como esquema ou protocolo, *Domain Name* como domínio, *Port* como porta (caso haja), *Path* como caminho, *Parameters* como parâmetros e *Anchor* como âncora.

Figura 4 - URL
Fonte: MDN Web Docs[4]

As URLs são os caminhos e os locais que desejamos acessar, e elas nos dizem muito sobre o conteúdo que estamos acessando através dos componentes principais que abordamos. Dependendo da variação da URL conseguimos, até mesmo visualmente, distinguir uma URL falsa de uma URL original. Mas cabe as tecnologias disponíveis nos auxiliarem na filtragem destes acessos e destas mudanças sutis nas URLs, ou como comumente chamamos: Links.

3. Metodologia

3.1. Base de Dados

Como base de dados para o desenvolvimento do trabalho, foi utilizada uma encontrada na plataforma *Kaggle*[5], e a escolha se deu através da área em que estou situado no mercado de trabalho, segurança da informação. Nesta plataforma podemos ter acesso a muitas opções para a realizaçao de tratamento de dados e sua futura predição. A base de dados em questão é de domínio público e pode ser utilizada por qualquer pessoa para fins de pesquisa, já a plataforma *Kaggle* foi apresentada a mim na disciplina de Tópicos Avançados em Programação ministrada pelo Prof. Dr. Ricardo Augusto Manfredini. A base de dados foi escolhida por conta da área de atuação em que me encontro, tendo em vista a quantidade de demandas que tenho no trabalho relacionadas a este tipo de prevenção e também para filtragem de endereços em clientes que atendo diariamente.

O arquivo que é baixado a partir da plataforma *Kaggle* possui formato CSV, ou seja, possui o endereço URL em uma coluna e logo após a vírgula possui a classificação desta referida URL, conforme a figura 5 abaixo.

[4] https://developer.mozilla.org/en-US/docs/Learn/Common_questions/Web_mechanics/What_is_a_URL
[5] Disponível em: https://www.kaggle.com/datasets/sid321axn/malicious-urls-dataset

	A	B	C	D	E
1	url,label				
2	br-icloud.com.br,phishing				
3	mp3raid.com/music/krizz_kaliko.html,benign				
4	bopsecrets.org/rexroth/cr/1.htm,benign				

Figura 5 - Base de Dados
Fonte: Próprio Autor

O arquivo possui 651.191 URL's e suas respectivas classificações. A classificação se divide em *Benign*, *Defacement*, *Phishing* e *Malware*, onde 428.103 são *Benign*, 96.457 são *Defacement*, 94.111 são *Phishing* e 32.520 são *Malware*. *Benign* podemos traduzir como URL's favoráveis, ou seja, que não possuem risco, já as URL's classificadas como *Defacement* podemos chamá-las de desfiguradas, são URL's que foram alteradas e podem apresentar riscos, pois redirecionam para outro destino. As URL's *Phishing* uma vez acessadas podem capturar dados do indivíduo pois possuem a ideia de fisgar o usuário que acessá-la.

> *O phishing é um exemplo de engenharia social: uma coleção de técnicas que os golpistas usam para manipular a psicologia humana. As técnicas de engenharia social incluem falsificação, desorientação e mentira – todas as quais podem desempenhar um papel em ataques de phishing. Em um nível básico, os e-mails de phishing usam engenharia social para encorajar os usuários a agir sem pensar nas coisas. (PROOFPOINT).*

As outras restantes são classificadas como *Malware* e podemos entender como a URL que poderá trazer algo nocivo para o equipamento ou sistema, como download de arquivos indesejados.

A partir da classificação das URL's podemos montar as classes para predição dos dados que usaremos na parte de importação e tratamento, para isso utilizaremos ferramentas extremamente necessárias para auxilar no decorrer do trabalho.

3.2. Bibliotecas

Todas elas foram utilizadas para a preparação do que será a futura predição dos dados. A importação das bibliotecas é realizada na primeira parte do código conforme abaixo

retratado, sendo mais fácil de visualizar e entender o que será utilizado no decorrer do código.

```
[ ]  import pandas as pd

     from sklearn.preprocessing import LabelEncoder

     from sklearn.linear_model import LogisticRegression

     from sklearn.tree import DecisionTreeClassifier

     from sklearn.metrics import accuracy_score
     from sklearn.model_selection import train_test_split
     from sklearn.feature_extraction.text import TfidfVectorizer
```

Figura 6 - Importação das Bibliotecas
Fonte: Próprio Autor

Algumas das bibliotecas utilizadas foram as reproduzidas na disciplina de Tópicos Avançados em Programação, como *Pandas*, *LabelEncoder* e *accuracy_score.* Outras já foram necessárias um estudo maior para encontrarmos a forma correta de tratar os dados e como utilizá-los. Por exemplo o *TfidfVectorizer*, que foi utilizado e que não foi visto em sala de aula.

A biblioteca Pandas é utilizada para o carregamento do arquivo CSV, com ela passamos por parâmetro o diretório do arquivo e ela o carrega para o ambiente de desenvolvimento, conforme a figura 7 abaixo.

```
[ ]  urls = pd.read_csv("/content/drive/MyDrive/Colab Notebooks/end_malicious_phish_final.csv")
```

Figura 7 - Importação do arquivo CSV com a ferramenta Pandas
Fonte: Próprio Autor

Após a importação do arquivo que já está carregado no Google Drive, estamos prontos para utilizar os dados nos próximos passos do algoritmo. A próxima etapa escolhida no algoritmo foi a impressão dos dados para verificarmos como estão alinhados após a importação do arquivo. Para isso usamos o comando *urls.head( )*, ele nos traz as colunas e as primeiras cinco linhas do arquivo carregado conforme a figura 8.

```
urls.head()
```

	url	label
0	br-icloud.com.br	phishing
1	mp3raid.com/music/krizz_kaliko.html	benign
2	bopsecrets.org/rexroth/cr/1.htm	benign
3	http://www.garage-pirenne.be/index.php?option=...	defacement
4	http://adventure-nicaragua.net/index.php?optio...	defacement

Figura 8 - Dados importados das primeiras 5 linhas do arquivo CSV
Fonte: Próprio Autor

Com os dados corretos podemos aplicar a função para separarmos as URL's em partes, podendo assim utilizá-las na predição dos dados na etapa que vem a seguir. A função escolhida pode ser encontrada na internet no site medium.com[6]. A quebra das URL's se faz a partir das barras, dos hífens e dos pontos, remove-se as redundâncias e as extensões de domínio (com).

A função usa o método *.split( )* para a quebra das strings de URL e após a quebra por barras, percorre-se o vetor de todas as URL's e vai se repetindo o processo de quebra e gravando estes dados em um vetor final *total_tokens*, conforme mostrado na figura 9.

[6] https://medium.com/@ismaelbouarfa/malicious-url-detection-with-machine-learning-d57890443dec

```
[5]  def makeTokens(f):
         tkns_BySlash = str(f.encode('utf-8')).split('/')  #cria tokens quebrando por barra
         total_Tokens = []
         for i in tkns_BySlash:
             tokens = str(i).split('-')  #cria tokens quebrando por traço
             tkns_ByDot = []
             for j in range(0,len(tokens)):
                 temp_Tokens = str(tokens[j]).split('.') #cria tokens quebrando por ponto
                 tkns_ByDot = tkns_ByDot + temp_Tokens
             total_Tokens = total_Tokens + tokens + tkns_ByDot
         total_Tokens = list(set(total_Tokens))  #remove redundâncias
         if 'com' in total_Tokens:
             total_Tokens.remove('com')  #remove os com
         return total_Tokens
```

Figura 9 - Função de quebra de Strings
Fonte: Próprio Autor

Após este vetor estar pronto para utilização, separamos as *urls* das *labels* que estão contidas na variável urls conforme abaixo demonstrado na figura 10. *Urls* são os links da base de dados e *labels* são as suas respectivas classificações.

```
[6]  lista_urls = urls["url"]
     y = urls["label"]
```

Figura 10 – Separação das urls
Fonte: Próprio Autor

Com a separação de *urls* e sua classificação realizada, podemos então aplicar o método de separação por tokens descrito na Figura 9, passando-o por parâmetro no comando *TfidfVectorizer*, iremos então atribuir as variáveis strings já quebradas em tokens para a variável *vectorizer* conforme visto na Figura 11. Após esta etapa, podemos aplicar o *fit_transform* na variável *lista_urls* e atribuírmos para a nova variável *X,* nela iremos transformar as URL's já quebradas em dados numéricos, pois os mesmos estavam separados como dados categóricos e desta forma não é possível fazer a predição e aplicar o método, portanto, é preciso realizar

esta etapa de conversão.

```
[7]  vectorizer = TfidfVectorizer(tokenizer=makeTokens)
```

Figura 11 - Método de separação por tokens
Fonte: Próprio Autor

Abaixo (figura 12) podemos ver como os dados ficaram após realizar o processo de conversão de dados categóricos para numéricos.

```
[18] print(X)

  (0, 512402)   0.3577105549067412
  (0, 233965)   0.5377021363050032
  (0, 670923)   0.46137214873035587
  (0, 670929)   0.608321717430244
  (1, 356899)   0.3806695741082364
  (1, 748542)   0.5683612945629615
  (1, 356900)   0.3806695741082364
  (1, 748540)   0.5464815233588478
  (1, 660828)   0.1241453205702225
  (1, 802023)   0.27035656742001335
  (2, 559546)   0.35587144279212335
  (2, 659178)   0.14078123623513628
  (2, 828883)   0.12201325032303711
  (2, 19191)    0.19487614007043041
```

Figura 12 – Dados após conversão
Fonte: Próprio Autor

Depois de termos preparado os dados para definitivamente a sua utilização, precisaremos aplicar um método para que não façamos o uso de todos estes dados do arquivo como base de treino para o algoritmo. Definimos, no momento da aplicação do comando de treino, que o ideal para treino do algoritmo é 70% da base, e os outros 30% são usados para

teste.

3.3. Treino

Agora que definimos a parte que será usada para treino e a outra que será usada para teste da máquina, podemos realmente aplicar o comando que irá fazer a utilização das variáveis que já possuem todos os dados convertidos e prontos para uso. O comando é separado em 4 partes e o método utilizado é o *train_test_split( ).* Nele aplicamos a parcela de Treino da parte 1 dos valores, a de Testes da parte 1 e repete-se com a parte 2. Então o comando apresenta-se da seguinte forma (figura 13).

```
[10] X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

Figura 13 - Comando de teste
Fonte: Próprio Autor

Passamos as duas partes separadas por X e Y e definimos a parcela de testes, que neste caso ficou como 30% *(test_size=0.3),* o *random_state=42* acaba não sendo importante, utilizamos um valor inteiro maior que zero. Com os valores de treino definidos podemos passar para a próxima etapa que é definir o algoritmo, neste caso é o Algoritmo de Árvore de Decisões, e aplicá-lo.

3.4. Aplicação do Algoritmo de Árvore de Decisões

Agora podemos partir para a próxima etapa do código, pois já temos o algoritmo decidido e podemos então passar como parâmetros os dados que já preparamos e especificamos as porcentagens de treino e teste anteriormente. Para isso, usamos o comando abaixo (demonstrado na figura 14) ditando os dados necessários.

```
[11] logit = DecisionTreeClassifier(criterion = 'entropy')
```

Figura 14 - Algoritmo de Árvore de decisões
Fonte: Próprio Autor

O algoritmo de Árvore de Decisões foi escolhido pois dentre as opções estudadas foi o que teve maior acurácia e um tempo aceitável de execução, chegando nos 97% de acurácia e próximo dos trinta minutos de execução. Os outros algoritmos que foram utilizados para teste foram: Regressão Linear, Floresta Randômica e Naive Bayes, todos trazendo uma acurácia abaixo de 95%, com exceção da Floresta Randômica que não executou no computador e nem mesmo no *Google Colaboratory*, o algoritmo ficou mais de 12 horas executando no computador e não finalizou, já no *Google Colaboratory* a unidade computacional era encerrada por excesso de processamento na conta gratuita.

Depois de todos estes passos tomados, podemos calcular a acurácia com o comando descrito na figura 15 e descobrir o quanto nosso algoritmo se faz utilizável.

```
[13] print("Acurácia: ",logit.score(X_test, y_test))

     Acurácia:  0.9746071141793458
```

Figura 15 - Cálculo da acurácia
Fonte: Próprio Autor

Sabendo que podemos confiar na predição dos dados que iremos realizar a entrada por termos um valor alto e positivo na acurácia, podemos passar como parâmetro agora as URL's que gostaríamos de realizar a predição e obtermos o retorno do algoritmo construído.

No retorno obtivemos a confiabilidade esperada considerando as URL's que passamos como parâmetro para o código já construído e treinado com o algoritmo de Árvore de Decisões. A aplicação deste código pode ser realizada em equipamentos de segurança ou até mesmo para

futuras consultas quando o analista de segurança da informação necessitar de um apoio da inteligência artificial para busca de informações.

4. Caso de uso

Nos dias atuais, onde a segurança dos dados dos indivíduos está cada vez mais em risco, nós, como desenvolvedores e analistas de sistemas, devemos construir e lapidar cada vez mais as ferramentas disponíveis, tornando a internet cada vez mais um local que transmita segurança ao invés de receio e confusão nos momentos de pesquisa e busca de informações. A ferramenta elaborada com a junção de informações encontradas na internet e conhecimentos recebidos durante o curso, faz-se necessária a partir do momento em que nos encontramos com dúvidas referentes a uma URL, seja ela recebida por e-mail, ou por algum redirecionamento estranho que reparamos em nosso navegador em um determinado momento.

Integrando-a com recursos de segurança da informação, sejam eles equipamentos físicos como Firewalls ou aplicativos Antivírus, poderemos obter maior certidão na classificação das URL's que acessamos, onde então temos a oportunidade de popular um banco de dados com sua respectiva classificação e num futuro próximo teremos acesso à esta informação sem a necessidade da predição.

Como podemos constatar na figura 16, o resultado é confirmado quando aplicamos o algoritmo para a predição, espera-se assim a futura utilização das URL's carregadas e que foram devidamente classificadas.

```
[16] X_predict2 = ["www.raidbr.com.br",
     "38zu.cn",
     "www.atlanticooceano.com",
     "xilften.club" ]

[17] X_predict2 = vectorizer.transform(X_predict2)
     Nova_predicao2 = logit.predict(X_predict2)
     print(Nova_predicao2)

     ['benign' 'phishing' 'benign' 'phishing']
```

Figura 16 - Retorno do algoritmo com as URL's de entrada personalizadas
Fonte: Próprio Autor

5. Considerações

Este trabalho teve como objetivo a junção de conhecimentos do autor sobre segurança da informação e a utilização das ferramentas e métodos desenvolvidos no curso, sendo duas delas imprescindíveis para o desenvolvimento de seres humanos cada vez mais racionais, a capacidade de pesquisa e desenvolvimento, habilidades estas que cada vez mais foram sucateadas nos últimos quatro anos[7], onde tivemos cada vez menos o incentivo devido e o fornecimento de recursos destinados para tal fim.

A qualidade dos recursos e dos temas abordados nesta pesquisa foram disponibilizados pelos professores do campus, onde tivemos acesso à uma educação de qualidade durante os sete semestres desenvolvidos durante o curso.

A utilização das ferramentas e da linguagem de programação escolhidas foram com base nas disciplinas de desenvolvimento de software, visto que foram específicas para a aplicação deste tipo de algoritmo, que utiliza a importação de dados para a manipulação, a aprendizagem de máquina e o possível retorno baseado no algoritmo de predição estudado, Árvore de Decisões.

[7] Durante o governo federal de 2019 a 2022

Sobre os resultados esperados, podemos concluir que foram satisfatoriamente alcançados, visto que o tempo de execução e a confiabilidade dos dados obtidos pelo algoritmo foram bons. O tempo de execução para este tipo de predição poderá ser diminuído utilizando um hardware melhor, e a acurácia próxima de 97% é muito assertiva quando pensamos em integrar a ferramenta com algum equipamento de segurança. Com pesquisas avançando rapidamente na área de aprendizado de máquina, poderemos melhorar o algoritmo para que o seu uso tenha um tempo de resposta menor, e claro, um consumo de hardware proporcionalmente menor também, ponto que acaba pesando muito no momento da consulta da informação retornada pelo algoritmo.

A aplicação desta pesquisa e do desenvolvimento também poderá ser alavancada a partir do momento que se integrar a um software para auxilio em pesquisas e melhoramentos da segurança voltada a URL's. Para este passo seguinte, será necessária maior dedicação com ferramentas e linguagens de front-end, onde teremos as integrações visuais voltadas à usabilidade, além do código desenvolvido nesta pesquisa.

A melhoria constante dos recursos e do desenvolvimento da aplicação se fará necessária, assim como foi citado Bruce Schneier anteriormente no texto "A segurança é um processo, não um produto", portanto, tudo precisa passar por um processo para se tornar cada vez mais confiável, e o desenvolvimento de software e a segurança da informação não são diferentes.

6. Referências

RANGEL, Rafael. **O que é um IPS(Intrusion Prevention System)?** Disponível em: <https://xtech.com.br/Blog/O-Que-E-Um-Ipsintrusion-Prevention-System/b/51/>. Acesso em: 29 mai. 2023.

SCHNEIER, Bruce. **The Process of Security.** Disponível em: <https://www.schneier.com/essays/archives/2000/04/the_process_of_secur>. Acesso em: 29 mai. 2023.

TREND MICRO. **Email Threat Landscape Report:**
Cybercriminal Tactics, Techniques That Organizations Need to Know. Disponível em: <https://www.trendmicro.com/vinfo/us/security/research-and-analysis/threat-reports/roundup/annual-trend-micro-email-threats-report>. Acesso em: 29 mai. 2023.

TREND MICRO. **Trend Micro alerta: Brasil é o segundo país que mais sofre com ameaças ransomware, atrás apenas dos EUA.** Disponível em: <https://www.trendmicro.com/pt_br/about/newsroom/press-releases/2019/fast-facts-may-2019.html>. Acesso em 30 mai. 2023.

SECURITY REPORT. **Brasil é o quarto país no mundo com mais ataques disseminados por e-mail.** Disponível em: <https://www.securityreport.com.br/overview/brasil-e-o-quarto-pais-no-mundo-com-mais-ataques-disseminados-por-e-mail/>. Acesso em 30 mai. 2023.

BRASIL. **Lei Geral de Proteção de Dados Pessoais (LGPD). Brasília, DF: Presidência da República, [2018].** Disponível em: <https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13709.htm>. Acesso em 30 mai. 2023.

GOV.BR. **Lei Geral de Proteção de Dados Pessoais (LGPD).** Disponível em: <https://www.gov.br/esporte/pt-br/acesso-a-informacao/lgpd>. Acesso em 30 mai. 2023.

MINISTÉRIO PÚBLICO FEDERAL. **Lei Geral de Proteção de Dados.** Disponível em: <https://www.mpf.mp.br/servicos/lgpd/o-que-e-a-lgpd>. Acesso em 30 mai. 2023.

CISCO. **O que é um Firewall?.** Disponível em: <https://www.cisco.com/c/pt_br/products/security/firewalls/what-is-a-firewall.html>. Acesso em: 30 mai. 2023.

KASPERSKY. **O que é um firewall? Definição e explicação.** Disponível em:

<https://www.kaspersky.com.br/resource-center/definitions/firewall>. Acesso em: 30 mai. 2023.

FORTINET. **Next-Generation Firewall (NGFW).** Disponível em: <https://www.fortinet.com/br/products/next-generation-firewall>. Acesso em 30 mai. 2023.

GOGONI, Ronaldo. **O que é URL?.** Disponível em: <https://tecnoblog.net/responde/o-que-e-url/>. Acesso em 31 mai. 2023.

GONÇALVES, Ariane. **O Que é URL: Exemplos, Estrutura e Muito Mais.** Disponível em: <https://www.hostinger.com.br/tutoriais/url>. Acesso em 31 mai. 2023.

ESTRELLA, Carlos. **O Que é TLD (Top-Level Domain)?** Disponível em: <https://www.hostinger.com.br/tutoriais/o-que-e-tld>. Acesso em 31 mai. 2023.

W3TECHS. **Usage statistics of top level domains for websites.** Disponível em: <https://w3techs.com/technologies/overview/top_level_domain>. Acesso em 02 jun. 2023.

PROOFPOINT. **What Is Phishing?** Disponível em: <https://www.proofpoint.com/us/threat-reference/phishing>. Acesso em 09 jun. 2023.

STOJILJKOVIĆ, **Mirko. Split Your Dataset With scikit-learn's train_test_split().** Disponível em: <https://realpython.com/train-test-split-python-data/>. Acesso em 09 jun. 2023.

SILVA, Alexandre Almeida. **Multas por descumprimento da LGPD podem chegar a 50 milhões de reais.** Disponível em: <https://www.migalhas.com.br/depeso/383661/multas-por-descumprimento-da-lgpd-podem-chegar-a-50-milhoes-de-reais>. Acesso em 19 jun. 2023.