

UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"

CIÊNCIA DA COMPUTAÇÃO

Matheus Bordino Moriel

Filtragem de Tráfego DNS para Detecção de Fraudes Relacionadas a Temas de Alta Visibilidade Nacional

São José Do Rio Preto

2024

Matheus Bordino Moriel

Filtragem de Tráfego DNS para Detecção de Fraudes Relacionadas a Temas de Alta Visibilidade Nacional

Trabalho de Conclusão de Curso apresentado como parte dos requisitos para obtenção do título de Bacharel em Ciência da Computação, junto ao Departamento de Ciências de Computação e Estatística, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista "Júlio de Mesquita Filho", Câmpus de São José do Rio Preto.

Financiador: NIC.br - Proc. 3467/2023

Orientador: Prof. Dr. Adriano Mauro Cansian

São José Do Rio Preto

2024

M855f

Moriel, Matheus Bordino
Filtragem de tráfego DNS para detecção de fraudes relacionadas a temas de alta visibilidade nacional / Matheus Bordino Moriel. -- São José do Rio Preto, 2024
36 p. : il., tabs.

Trabalho de conclusão de curso (Bacharelado - Ciência da Computação) - Universidade Estadual Paulista (UNESP), Instituto de Biociências Letras e Ciências Exatas, São José do Rio Preto
Orientador: Adriano Mauro Cansian

1. Redes de computadores. 2. Nomes de domínio na Internet. 3. Fraude na Internet. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Dados fornecidos pelo autor(a).

Matheus Bordino Moriel

Filtragem de Tráfego DNS para Detecção de Fraudes Relacionadas a Temas de Alta Visibilidade Nacional

Trabalho de Conclusão de Curso apresentado como parte dos requisitos para obtenção do título de Bacharel em Ciência da Computação, junto ao Departamento de Ciências de Computação e Estatística, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista "Júlio de Mesquita Filho", Câmpus de São José do Rio Preto.

Banca Examinadora

Prof. Dr. Adriano Mauro Cansian
UNESP - São José do Rio Preto
Orientador

Prof. Dr. Arnaldo Cândido Júnior
UNESP - São José do Rio Preto

Profª. Drª. Renata Spolon Lobato
UNESP - São José do Rio Preto

São José Do Rio Preto

6 de Novembro de 2024

Este trabalho é dedicado à minha família e todos meus queridos amigos que me apoiaram durante os anos desta graduação.

Agradecimentos

A meus pais, Karina Fernanda Castilho Bordino Moriel e Paulo Cesar Moriel, por sempre me apoiarem incondicionalmente em todos os momentos da vida, possibilitando minha formação acadêmica e uma vida saudável.

A meu querido irmão, Gabriel Bordino Moriel, por sempre estar ao meu lado como maior apoiador e amigo eterno.

A cada um de meus queridos amigos, por todo o apoio, aprendizado, carinho e gentileza que me proporcionaram ao longo dos anos desta graduação.

A meu orientador, Adriano Mauro Cansian, por todas as oportunidades oferecidas e todos os aprendizados consequentes.

À equipe do Laboratório ACME!, por todos os momentos de aprendizado, diversão e pela imensa ajuda na elaboração desta monografia.

Ao Coral da Unesp FEIS e IBILCE e seu regente, Marcio Guirado Zuanazzi, por toda sua influência em minha formação pessoal e toda a amizade que proporcionaram ao longo destes anos.

Aos docentes do curso de Ciência da Computação do IBILCE, por todo o aprendizado e conhecimento concedidos.

Ao milagre da vida, por ter me permitido experienciar todos estes momentos, conhecer todas estas pessoas e tudo o que ainda há por vir.

Este trabalho foi financiado por intermédio do Convênio de Pesquisa e Inovação com o NIC.BR - Núcleo de Informações e Coordenação do Ponto BR, Processo Fundunesp número 3467/2023 - NIC. BR.

'Though life is merely a journey to the grave, it must not be undertaken without hope. Only then will a traveler's story live on, treasured by those who bid him farewell.'

Yuichiro Tanaka (Persona 3, 2006)

Resumo

O Sistema de Nomes de Domínio (DNS) é de grande importância para o funcionamento da rede mundial de computadores. Entretanto, este sistema é, muitas vezes, abusado por agentes maliciosos. Dado que estes agentes almejam fazer de suas fraudes o mais eficientes e furtivas possível, pode-se presumir que farão uso de domínios relacionados a tópicos atualmente relevantes para enganar suas vítimas de forma mais eficaz. Este documento, portanto, propõe o desenvolvimento de um sistema para identificar domínios pertinentes a tópicos atualmente relevantes para que estes possam ser apropriadamente etiquetados conforme sua relevância e os dados resultantes utilizados no processo de classificação de domínios como maliciosos ou benignos. Para tanto, são utilizados dados obtidos a partir da Wikpedia e seus serviços de métricas, em conjunto com dados de registros DNS disponibilizados por *blocklists*. Por fim, o sistema desenvolvido foi utilizado para identificar os tópicos relevantes ao longo do período de 6 meses e gerou palavras-chave relacionadas a estes, permitindo, dentre outros possíveis usos, a busca de nomes de domínio maliciosos que as contêm e verificou-se, assim, o potencial de uma real correlação entre os tópicos de alta visibilidade, tanto nacional quanto internacional, e o aumento de registro de domínios maliciosos pertinentes a estes, em especial para tópicos relacionados a tecnologia.

Abstract

The Domain Name System (DNS) is of great importance for the functioning of the world-wide web. However, this system is often abused by malicious actors. Given that said actors aim to make their frauds as efficient and stealth as possible, it can be presumed they will make use of domains related to currently relevant topics in order to more effectively trick their victims. This document, thus, proposes the development of a system for identifying domains that pertain to currently relevant topics so that they can be properly labeled based on their relevance and the resulting data used when classifying domains as either malicious or benign. To that end, data from Wikipedia and its analytics services are utilized in conjunction with DNS data supplied by blocklists. Finally, the developed system was used to identify relevant topics over a period of 6 months and generated keywords related to them, enabling, among other possible uses, the search for malicious domain names that contain them and, with that, the potential of a real correlation between high visibility topics, be it of national or international relevancy, and an increase in registration for domains pertaining to these topics was verified, especially for topics related to technology.

Lista de Figuras

1 Hierarquia DNS 13

2 Processo de resolução de nome para *www.example.com* 14

3 Estatísticas de visualização da página “Rio Grande do Sul” no período entre 01/04/2024 e 30/06/2024. 18

4 Processo de selecionar artigos relevantes para um dos artigos na categoria de eventos de Julho de 2024. 25

Lista de Listagens

1 Exemplo de requisição whois para o domínio acmesecurity.org. 15
2 Exemplo de requisição `GET`. 15
3 Exemplo de requisição `POST`. 16
4 Requisição em forma de API para estatísticas da página "Rio Grande do Sul" entre 01/04/2024 e 30/06/2024. 17
5 Configurando *User-Agent* e linguagem. 19
6 Imprimindo o resumo do artigo "Drakengard". 19
7 *Links* presentes no artigo "Drakengard". 19
8 Versões em outras linguagens do artigo "Drakengard". 19
9 Inicialização da classe WMstats 21
10 Obtendo artigos em português a partir de seus correspondentes em inglês. 22
11 Obtendo títulos de artigos sobre eventos notórios ocorridos nos últimos 6 meses em escopo global e regional do Brasil. 22
12 Obtendo o dia com maior número de visualizações para cada um dos artigos. 22
13 Obtendo links presentes no resumo de um artigo. 23
14 Verificando picos de interesse em um artigo. 24
15 Extraíndo artigo com maior contagem de visualizações 25
16 Partículas de linguagem removidas do título. 26
17 Vetor de termos candidatos para consulta, a partir do título "2024 CrowdStrike-related IT outages". 26
18 Normalizando frequências de termos. 27
19 Exemplos de domínios detectados se aproveitando do incidente da CrowdStrike. 32

Lista de Tabelas

1 Eventos com maior número de visualização em um único dia. 29

2 Eventos com maior número de visualização em um único dia (a partir do conjunto completo). 30

3 Artigos e os eventos aos quais se referem. 30

4 Comparação entre datas de ocorrência de um evento e a de pico de visualizações em sua página correspondente. 31

5 Artigos e suas respectivas palavras-chave. 31

6 Palavras-chave e seus respectivos resultados da pesquisa no banco de dados do Nevermore. 32

7 Tópicos e suas respectivas categorias. 33

Lista de Acrônimos

API *Application Programming Interface.*

ccTLD *Country Code Top-Level Domain.*

DNS *Domain Name System.*

FQDN *Fully Qualified Domain Name.*

FTP *File Transfer Protocol.*

HTTP *Hypertext Transfer Protocol.*

IP *Internet Protocol.*

pDNS *Passive DNS.*

RFC *Request for Comment.*

TLD *Top-Level Domain.*

URL *Uniform Resource Locator.*

Sumário

1 Introdução **11**

2 Fundamentação Teórica **12**

2.1 Sistema de Nomes de Domínio 12

2.1.1 DNS Passivo 14

2.1.2 WHOIS 15

2.2 *Application Programming Interface* (API) 15

2.3 Wikipedia 16

2.3.1 *Wikimedia Pageviews Analysis* 17

2.3.2 Wikipedia-API 18

2.4 Trabalhos correlatos 20

3 Metodologia **21**

3.1 Determinando tópicos relevantes 21

3.2 Geração de palavras-chave 26

3.3 Verificação de correlação 27

4 Testes e resultados **29**

5 Conclusão **34**

1 Introdução

O Brasil é o quarto país com maior número de usuários de Internet, totalizando, aproximadamente, 187.9 milhões de usuários no início do ano de 2024, correspondente a 86.6% da população (KEMP, 2024). Em meio de tantos usuários, diversos são vítimas de golpes e *websites* maliciosos diariamente (CERT.BR, 2024b). No Brasil, segundo o registro.br (REGISTRO.BR, 2024), foram registrados, durante o mês de Junho de 2024, aproximadamente 130 mil novos domínios, sendo que, no total, existem, aproximadamente, 5.350.000 domínios .br. Destes, uma quantidade considerável é utilizada para atividades maliciosas e uso fraudulento (CERT.BR, 2024a). Estes domínios, em diversos casos, se aproveitam para "surfar na onda" de tópicos que possuem, no momento, alta visibilidade nacional. Isto se dá pelo fato de que, dada a relevância atual do tópico, a disseminação de domínios maliciosos relacionados a este é camuflada em meio ao grande fluxo de informações legítimas sobre o tema, permitindo que estes se disfarcem de maneira muito mais eficaz.

O sistema responsável pelo registro e resolução destes domínios é o Sistema de Nomes de Domínio (*Domain Name System* (DNS)), descrito nos RFCs 1034 (MOCKAPETRIS, 1987a) e 1035 (MOCKAPETRIS, 1987b). Seu propósito primário é a tradução de nomes de domínio – de mais fácil memorização para os humanos – para endereços IP e vice-versa.

Diversos trabalhos já foram publicados visando detectar domínios maliciosos a partir de características intrínsecas dos domínios, como características textuais do nome de domínio (ZHAO et al., 2019) e métricas do registro deste nos servidores DNS (BILGE et al., 2014). Estes se baseiam, primariamente, no uso de modelos de Aprendizado de Máquina e Inteligência Artificial, treinados com listas de domínios maliciosos e benignos previamente conhecidos, para a detecção preventiva de abusos do sistema, permitindo que tais domínios sejam filtrados antes mesmo que algum usuário seja por eles lesado.

Objetivo

A fim de desenvolver medidas de detecção preventiva de domínios maliciosos, este presente trabalho visa verificar a correlação entre a relevância corrente de um determinado tópico na Internet, reflexo parcial do imaginário coletivo, e o número de domínios maliciosos relacionados a este tópico — por meio do desenvolvimento de um sistema que determina tópicos relevantes e gera palavras-chave relacionados a eles —, além de etiquetá-los de acordo. Estas métricas podem, então, ser utilizadas durante o processo de classificação de um domínio como malicioso ou benigno, tendo um peso maior os domínios relacionados ao tópico em questão. Para tal, será feito o uso de métricas disponibilizadas publicamente por ferramentas como o Wikimedia Pageviews.

2 Fundamentação Teórica

Nesta seção, são apresentados alguns conceitos importantes para a compreensão do trabalho desenvolvido, sendo eles os conceitos de: DNS, DNS Passivo, API, Wikipedia, Pageviews Analysis, Wikipedia-API, além de trabalhos relacionados ao desenvolvimento deste, como o DNS Framework e seus diferentes módulos.

2.1 Sistema de Nomes de Domínio

O sistema de nomes de domínio (*Domain Name System* (DNS)) é o sistema responsável, na Internet, por traduzir nomes mais facilmente memorizáveis por humanos, como *example.com*, para seus correspondentes endereços IP, os quais são de fato utilizados pelos próprios computadores para se identificarem entre si.

Originalmente, esta resolução de nomes de domínios para endereços IP era realizada por meio de um arquivo único e monolítico chamado `HOSTS.txt`, com uma cópia armazenada em cada máquina da rede, a qual era obtida via FTP a partir de sua versão de referência atualizada e mantida pelo Instituto de Pesquisa de Stanford. Esta solução centralizada funcionava bem para redes pequenas, porém era pouco escalável, tornando-se um grande problema com o crescimento da Internet. Visando solucionar este problema, o DNS – originalmente proposto nos RFCs 882 (MOCKAPETRIS, 1983a) e 883 (MOCKAPETRIS, 1983b), sucedidos pelos RFCs 1034 (MOCKAPETRIS, 1987a) e 1035 (MOCKAPETRIS, 1987b) – é um sistema distribuído e descentralizado no modelo cliente-servidor, operando, assim como os endereços IP, em uma estrutura hierárquica, a qual permite mais fácil administração local de redes, mas garantindo ainda o acesso global a estes dados (ALBITZ; LIU, 1993).

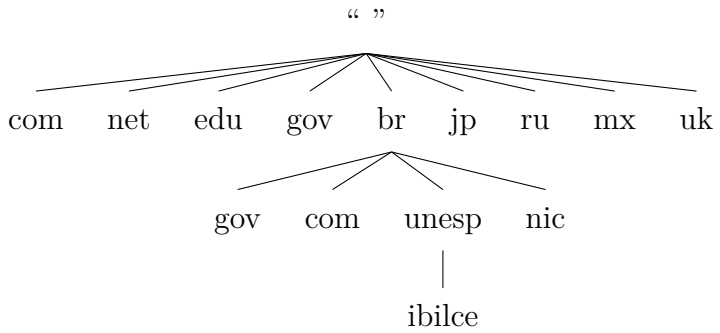
Segundo Albitz e Liu (1993), o modelo cliente-servidor do DNS pode ser dividido, de forma geral, em duas metades:

- ***Name servers:*** Estes servidores contêm informações sobre algum segmento da hierarquia de DNS e fornecem estas informações para todos os clientes que venham a requisitá-la.
- ***Resolvers:*** Clientes que recebem um nome de domínio da aplicação sobrejacente e realizam uma requisição DNS para um *name server* a fim de realizar a tradução de nome de domínio para endereço IP. São responsáveis pelas seguintes funções:
  - Envio de requisição para um *name server*.
  - Interpretação da(s) resposta(s).
  - Retorno das informações adquiridas para a aplicação sobrejacente que as requisitou.

Ademais, segundo Kurose e Ross (2012), os servidores DNS podem ser classificados, de forma geral, em 3 categorias:

- ***Root server***: Servidor raiz, dos quais há 13 distribuídos ao redor do mundo e com diversos espelhos para balanceamento de carga. Estes constituem o início da hierarquia DNS e possuem informações apenas sobre a localização dos servidores de Domínios de Alto Nível (*Top-Level Domain* (TLD)).
- **Servidor de *Top-Level Domain***: Servidor responsável por um determinado TLD, como *com*, *net*, *edu*, *gov*, além dos domínios da subcategoria de *Country Code Top-Level Domain* (ccTLD), isto é, Domínios de Alto Nível de países, como *br*, *jp*, *ru*, *mx*, *uk*, dentre diversos outros. Possuem informações sobre a localização dos servidores autoritativos de seus respectivos subdomínios.
- **Servidor Autoritativo**: Abriga registros de DNS que mapeiam todos os hospedeiros de uma determinada organização que possam ser acessados publicamente na Internet aos seus respectivos endereços IP. Estes servidores geralmente são gerenciados pela própria organização em questão (embora esta função possa também ser delegada a um terceiro) e são responsáveis por manter as informações "oficiais" sobre os domínios que administra.

Figura 1: Hierarquia DNS



Fonte: Autor.

O processo de uma consulta DNS realizada por um navegador *web* pelo site *www.example.com* pode, então, ser resumido aos seguintes passos:

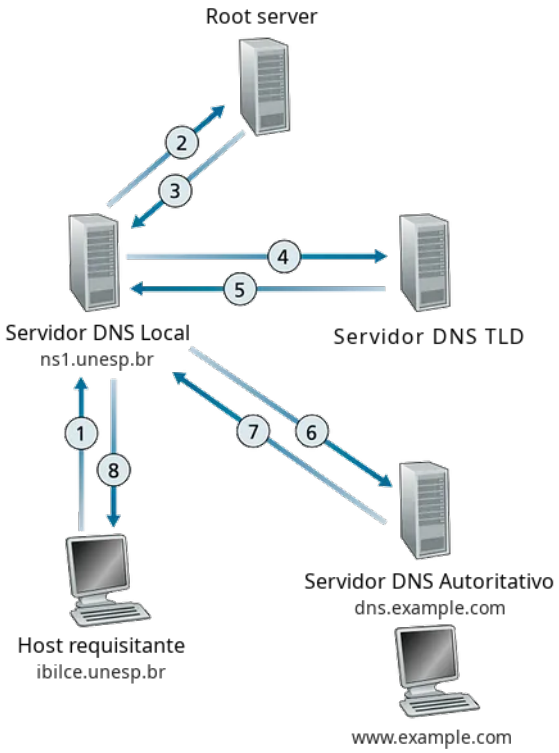
1. Requisição ao servidor DNS: O navegador extrai da URL solicitada o nome de domínio absoluto (*Fully Qualified Domain Name* (FQDN)) e envia uma solicitação de resolução ao servidor de DNS recursivo configurado na máquina.
2. Consulta inicial: O servidor DNS recursivo verifica se o registro solicitado está disponível em seu cache. Caso esteja, retorna imediatamente o resultado para a

máquina solicitante. Caso contrário, inicia sua busca a partir da raiz da hierarquia de DNS, consultando um dos *Root servers*.

3. Consulta iterativa: A partir da primeira consulta aos *Root servers*, o servidor consulta iterativamente outros servidores DNS, como ilustrado na Figura 2, seguindo a hierarquia demonstrada na Figura 1 até conseguir resolver o nome de domínio solicitado ou determinar que este não existe.

4. Resposta: Após finalmente encontrar o servidor que responde autoritativamente pelo registro requisitado, o servidor DNS recursivo que foi consultado pela máquina do usuário retorna para esta a resposta com endereço IP recebida.

Figura 2: Processo de resolução de nome para *www.example.com*



Fonte: Adaptado de Kurose e Ross (2012).

2.1.1 DNS Passivo

DNS Passivo, ou *Passive DNS* (pDNS), se refere à pratica de armazenar as respostas (e, em alguns casos, as próprias consultas – em qual caso podem se ocasionar problemas de privacidade) de DNS que trafegaram em uma determinada rede para análise futura (HOFFMAN; SULLIVAN; FUJIWARA, 2019).

2.1.2 WHOIS

Especificado no RFC 3912 (DAIGLE, 2004), WHOIS é um protocolo de Internet destinado a fornecer informações sobre o dono e o registro de um determinado domínio. Os dados sobre o dono de domínio, geralmente, são mantidos pelo administrador do TLD ao qual o domínio pertence e podem ser requisitados através de diversas aplicações que implementam o protocolo, como a aplicação homônima disponível para GNU/Linux ou a implementação em Python, o `python-whois`, por exemplo.

A Listagem 1 mostra um exemplo de requisição e resposta utilizando a utilidade `whois` para GNU/Linux.

```
$ whois acmesecurity.org
Domain Name: acmesecurity.org
Registry Domain ID: 1573abd537474d2b842533b04ed696dc-LROR
Registrar WHOIS Server: whois.namecheap.com
Registrar URL: http://www.namecheap.com
Updated Date: 2023-11-07T05:34:22Z
Creation Date: 2002-12-02T01:35:58Z
Registry Expiry Date: 2024-12-02T01:35:57Z
...
```

Listagem 1: Exemplo de requisição whois para o domínio acmesecurity.org.

2.2 *Application Programming Interface* (API)

Uma API, ou Interface de Programação de Aplicação, é um conjunto de regras ou protocolos que permite que diferentes aplicações de software se comuniquem para trocar dados, recursos e funcionalidades, permitindo que sejam compartilhadas apenas as informações necessárias, mantendo os detalhes do funcionamento interno do sistema escondidos (GOODWIN, 2024).

O tipo de API mais comum é o de API *Web*, utilizada para habilitar transferência de dados e funcionalidades através da Internet utilizando o protocolo HTTP. Para tal, são definidos *endpoints*, locais onde as solicitações para a API são atendidas, geralmente denotados por um URL (CLOUDFLARE, 2024).

Pode-se realizar requisições em APIs de diversas formas, as principais sendo com os métodos HTTP `GET`, a qual recupera dados de um *endpoint*, e `POST`, que submete dados para um *endpoint* especificado, exemplificados em Listagem 2 e Listagem 3, respectivamente, com a API do *website the visual novel database*.

```
>>> from requests import get
>>> get('https://api.vndb.org/kana/stats').text
```

```
'{"staff":36656,"tags":2930,"producers":21338,"chars":135042,"traits":3238,"
   vn":49676,"releases":118433}'
```

Listagem 2: Exemplo de requisição `GET`.

```
>>> from requests import post
>>> data = {'filters': ['id','=','v2002'], 'fields': 'title, released'}
>>> post('https://api.vndb.org/kana/vn', json=data).text
'{"results":[{"id":"v2002","released":"2009-10-15","title":"STEINS;GATE"}],"
   more":false}'
```

Listagem 3: Exemplo de requisição `POST`.

2.3 Wikipedia

Wikipedia é uma enciclopédia virtual de cunho colaborativo, de forma que todo o conteúdo nela presente é gerado por contribuição voluntária de seus usuários.

Fundada em 2001, a plataforma é gerenciada, desde 2003, pela Wikimedia Foundation, organização sem fins lucrativos, a qual é mantida, financeiramente, primariamente, por doações advindas de seus próprios usuários. A plataforma conta com, aproximadamente, 275000 editores ativos, os quais, juntamente com os inativos, já produziram, no total, mais de 63 milhões de artigos em mais de 300 linguagens diferentes (WIKIPEDIA, 2024b).

Artigos na Wikipedia podem abranger diversos microcosmos e categorias, com informações sobre objetos, conceitos e eventos, dentre uma infinidade de outros tópicos. Embora não possuam uma estrutura padrão definida, estes artigos visam incluir o máximo de informação relevante possível sobre determinado assunto, devendo esta ser verificável, contendo citações para as fontes de todas as informações expostas (WIKIPEDIA, 2024c). Estes artigos também devem aderir aos princípios fundamentais da Wikipedia, resumidos nos “Cinco Pilares da Wikipedia” (WIKIPEDIA, 2024d):

- Wikipedia é uma enciclopédia, combinando características de enciclopédias generalizadas, especializadas e almanaques.
- Wikipedia é escrita a partir de um ponto de vista neutro.
- Wikipedia é conteúdo livre que todos podem usar, editar e distribuir.
- Editores da Wikipedia devem tratar uns aos outros com respeito e civilidade.
- Wikipedia não tem regras fixas.

Artigos que possuem assuntos em comum podem ser agrupados em categorias e, para cada categoria, é gerada, automaticamente, uma página que lista todos os artigos pertencentes a esta (WIKIPEDIA, 2024a).

Além disso, para merecer seu próprio artigo, um tópico deve ter recebido cobertura significante em meios de comunicação em massa ou publicações acadêmicas (WIKIPEDIA, 2024e). Dessa forma, presume-se que todo evento reportado com um artigo próprio na Wikipedia possui algum nível de relevância para o público geral.

2.3.1 *Wikimedia Pageviews Analysis*

Para o intuito de visualização e analise de dados sobre seus projetos (incluindo, mas não limitado à, Wikipedia), a Wikimedia Foundation fornece um conjunto de ferramentas chamado *Pageviews Analysis*. No total, são disponibilizadas as seguintes oito ferramentas (WIKIMEDIA, 2024):

- *Pageviews*: Estatísticas de visualização de uma ou mais páginas dentro de um determinado período de tempo, sendo disponibilizados, também, diversos outros filtros.
- *Langviews*: *Pageviews* para um determinado artigo entre todas as linguagens nas quais este se encontra disponível.
- *Topviews*: Mostra as páginas com mais visualizações de um determinado projeto, podendo ser filtradas por um período de tempo ou plataforma de visualização (*desktop* ou *mobile*).
- *Siteviews*: *Pageviews* de todas as páginas de um determinado projeto da Wikimedia.
- *Massviews*: Permite importar uma lista de páginas e analisar suas estatísticas de visualização.
- *Redirect Views*: *Pageviews* de uma página e todos seus redirecionamentos.
- *Userviews*: *Pageviews* de todos os artigos criados por um determinado usuário.
- *Mediaviews*: Comparar estatísticas de visualização de arquivos de áudio e vídeo presentes nos projetos da Wikipedia.

Estas ferramentas podem ser utilizadas na forma de um *website*, no qual é possível visualizar as estatísticas de forma gráfica, exemplificado, com a ferramenta *Pageviews*, na Figura 3, e também em forma de API, como exemplificado na Listagem 4.

```
1 >>> from requests import get
2 >>> URL = "https://wikimedia.org/api/rest_v1/metrics/pageviews/per-article/
     pt.wikipedia.org/all-access/user/Rio_Grande_do_Sul/daily
     /2024040100/2024063000"
3 >>> headers = {"User-Agent": "mycoolproject-bot/1.0 (bob@example.com)"}
4 >>> get(URL, headers=headers).text
```

```
'{"items":[{"project":"pt.wikipedia","article":"Rio_Grande_do_Sul","
   granularity":"daily","timestamp":"2024040100","access":"all-access","
   agent":"user","views":1084}, ...
```

Listagem 4: Requisição em forma de API para estatísticas da página “Rio Grande do Sul” entre 01/04/2024 e 30/06/2024.

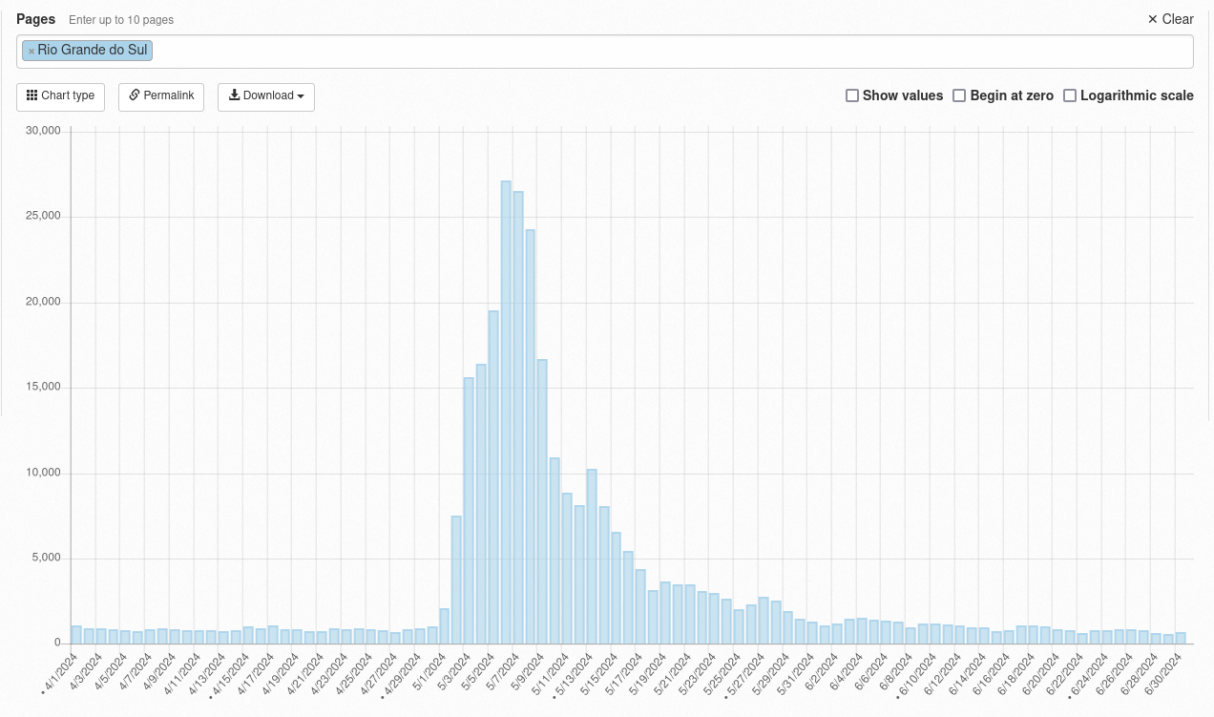


Figura 3: Estatísticas de visualização da página “Rio Grande do Sul” no período entre 01/04/2024 e 30/06/2024.

2.3.2 Wikipedia-API

Wikipedia-API (MAJLIS, 2017) é um *wrapper*[1] em Python para a API aberta da Wikipedia, visando a fácil integração e uso desta em programas dessa linguagem. De forma geral, esta API fornece as seguintes funcionalidades:

- Acessar e manipular os conteúdos de uma página da Wikipedia.
- Checar se uma dada página existe.
- Obter o URL de uma dada página.
- Obter versões de uma dada página em outras linguagens.
- Obter todas as páginas em uma determinada categoria.

[1]Biblioteca que traduz recursos pré-existentes em outro formato para uma forma compatível com a aplicação desejada.

Para utilizá-la, deve-se, primeiramente, definir um *User-Agent*, que identificará, com um nome de projeto, a máquina que está realizando a conexão com a API. Este é utilizado pela Wikipedia para gerenciamento de tráfego e, portanto, de importante configuração para evitar bloqueios por parte do serviço. Além disso, configura-se também a linguagem na qual as páginas serão acessadas.

A Listagem 5 mostra a configuração para o projeto "mycoolproject-bot", utilizando da língua inglesa.

```
>>> import wikipediaapi
>>> wp = wikipediaapi.Wikipedia('mycoolproject-bot (bob@example.org)', 'en')
```

Listagem 5: Configurando *User-Agent* e linguagem.

Pode-se, então, acessar uma página com o método `page` e mostrar seu resumo com o atributo `summary`, por exemplo. A Listagem 6 mostra este procedimento para a página sobre a série de videogames "Drakengard".

```
>>> page = wp.page("Drakengard")
>>> page.summary[0:175]
'Drakengard, known in Japan as Drag-On Dragoon, is a series of action role-
   playing video games created by Yoko Taro, originally developed by Cavia
   and published by Square Enix.'
```

Listagem 6: Imprimindo o resumo do artigo "Drakengard".

É possível, também, obter *links* presentes na página do artigo e as versões deste artigo em outras linguagens, com os atributos `links` e `langlinks`, como demonstrado em Listagem 7 e Listagem 8, respectivamente.

```
>>> for key in page.links.keys():
...     v = page.links[key]
...     print(f"{key}: {v.fullurl}")
1UP.com: https://en.wikipedia.org/wiki/1Up_Network
1Up.com: https://en.wikipedia.org/wiki/1Up_Network
2B (Nier: Automata): https://en.wikipedia.org/wiki/2B_(Nier:_Automata)
9S (Nier: Automata): https://en.wikipedia.org/wiki/9S_(Nier:_Automata)
...
```

Listagem 7: *Links* presentes no artigo "Drakengard".

```
>>> for key in page.langlinks.keys()
...     v = page.langlinks[key]
...     print(f"{key}: {v.fullurl}")
ar: https://ar.wikipedia.org/wiki/%D8%AF%D8%B1%D8%A7%D9%83%D9%86%D8%AC%D8%A7
   %D8%B1%D8%AF
el: https://el.wikipedia.org/wiki/Drakengard
...
```

Listagem 8: Versões em outras linguagens do artigo "Drakengard".

2.4 Trabalhos correlatos

Nesta seção, são apresentados trabalhos relacionados, diretamente ou não, à elaboração deste projeto, conhecimento dos quais auxilia no seu entendimento.

DNS Framework

O DNS Framework (BATISTA, 2020) é um projeto do Laboratório ACME! que visa integrar diversos modelos de classificação de domínios em uma plataforma unificada. A interface *web* deste *framework*, denominada DNSCheck, disponibiliza, atualmente, três módulos funcionais, sendo eles:

- Nevermore: módulo para verificação e monitoramento da presença de um determinado domínio em *blocklists* conhecidas, além do gerenciamento tanto destas como também de *allowlists*.
- Sentinel: módulo que realiza consultas DNS de forma ativa e aplica as métricas obtidas na resposta em um modelo de Aprendizado de Máquina para obter a probabilidade de o domínio fornecido ser malicioso.
- Dashboards: módulo que permite buscar dados de DNS Passivo previamente coletados e armazenados em um banco de dados e exibir diversas estatísticas sobre estes dados de forma gráfica.

3 Metodologia

Para o desenvolvimento deste trabalho, serão analisados tópicos de alta relevância que ocorreram durante os últimos 6 meses (de Abril de 2024 até Setembro de 2024) e, com o auxílio de um programa, escrito em Python, será verificada a quantidade de domínios maliciosos que foram registrados a fim de se aproveitarem da popularidade de tais tópicos.

3.1 Determinando tópicos relevantes

Para aferir quais tópicos tiveram alta relevância ao longo dos 6 meses que serão analisados, será utilizada a ferramenta *Massviews* do Wikimedia Pageviews Analysis. Tendo em vista que a Wikipedia possui categorias para eventos que ocorreram separados por meses do ano (tanto em escopo global quanto regional), utilizaremos a API do *Massviews* para obter os URLs de cada um dos artigos pertencentes a uma determinada categoria.

Visto que não há bibliotecas em Python que implementem uma interface nativa para as ferramentas do Wikimedia Pageviews Analysis, cria-se inicialmente um *wrapper* para as ferramentas que serão utilizadas no projeto, sendo elas: *Massviews* e *Pageviews*, além da função complementar `creation_date`, que obtém a data de criação de um determinado artigo a partir da API da Wikipedia. Estas são implementadas na classe `WMstats`, a qual é inicializada com seus *endpoints* como mostrada na Listagem 9.

```
1 class WMstats:
2     def __init__(self, project="en"):
3         self.wmcloud_base_url = "https://pageviews.wmcloud.org"
4         self.api_base_url = "https://wikimedia.org/api/rest_v1/metrics"
5         self.massviews_url = f"{self.wmcloud_base_url}/massviews/api.php?
    project={project}.wikipedia.org&limit=20000"
6         self.pageviews_url = f"{self.api_base_url}/pageviews/per-article"
```

Listagem 9: Inicialização da classe WMstats

Para os eventos considerados neste projeto, é utilizado o escopo tanto global quanto o específico ao Brasil. As páginas de categorias para ambos os escopos encontram-se disponíveis exclusivamente em inglês e, portanto, os *links* obtidos a partir delas referem-se apenas aos artigos em inglês sobre os tópicos. Para obter uma análise de relevância justa — pois presume-se que artigos sobre tópicos referentes ao Brasil serão acessados mais frequentemente em português — o *wrapper* para a ferramenta *Massviews* extrai tanto as páginas em inglês quanto seus equivalentes em português, quando disponíveis, sendo ambas utilizadas para análise. O processo de extrair as páginas equivalentes em português, dada uma lista de artigos em inglês, é realizada com a Wikipedia-API, como demonstrado na Listagem 10.

```
if region == "Brazil":
        localised_articles = list()
        for article in articles:
            page = wp.page(article["title"])
            if "pt" in page.langlinks:
                localised_articles.append({"lang": "pt", "title": page.
   langlinks["pt"].title.replace(" ","_")})
        return articles + localised_articles
```

Listagem 10: Obtendo artigos em português a partir de seus correspondentes em inglês.

Coletam-se, então, os artigos das categorias relativas a ambos os escopos para cada um dos últimos 6 meses e removemos quaisquer duplicatas, visto que, a depender da duração do evento, este pode aparecer em mais que uma categoria. Este processo é demonstrado na Listagem 11.

```
wmst = WMstats(project="en")
articles = list()
for m,y in dates: # m: mês, y: ano
    articles += wmst.massviews(month=m, year=y, region="Brazil")
    articles += wmst.massviews(month=m, year=y, region="global")

# gera um set de artigos únicos ao longo do período de tempo
collected_articles = {v['title']:v for v in articles}.values()
```

Listagem 11: Obtendo títulos de artigos sobre eventos notórios ocorridos nos últimos 6 meses em escopo global e regional do Brasil.

Desta forma, é obtida uma lista de todos os eventos considerados notórios pelos padrões da Wikipedia, tanto em português quanto em inglês, ocorridos nos últimos 6 meses. A partir disto, utilizando do recurso *Pageviews*, obtém-se estatísticas de visualização para cada um dos artigos na lista, em granularidade diária, desde exatamente 6 meses atrás (relativo à execução do programa) ou desde a data de criação do artigo, caso esta seja mais recente, até a data atual. Com esses dados, pode-se aferir em qual dia o artigo teve maior notoriedade, o que pode-se presumir que tenha sido o auge de interesse no assunto. Portanto, para cada artigo, retêm-se o dia com maior número de visualizações de seu ciclo de vida. Este processo é ilustrado na Listagem 12

```
# verificar período de maior relevância do artigo em questão
today = date.today()
initial_date = today - timedelta(days=180)
end = today.strftime('%Y%m%d00')

for article in collected_articles:
    creation_date = wmst.creation_date(article)

    if creation_date > initial_date:
```

```
        start = creation_date.strftime('%Y%m%d00')
    else:
        start = initial_date.strftime('%Y%m%d00')

    stats = wmst.pageviews(article, start=start, end=end, frequency="daily")
    max_views = max(stats["items"], key=lambda d: d['views'])
    print(max_views["article"].replace("_"," ") + ": " + str(max_views["
   views"]) + " views (" + max_views["timestamp"] + ")")

Resultados:
2024 Porto Alegre fire: 95 views (2024042700)
2024 Rio Grande do Sul floods: 27674 views (2024050900)
Uncle Paulo incident: 242 views (2024042100)
Incêndio em pousada de Porto Alegre: 558 views (2024042700)
Enchentes no Rio Grande do Sul em 2024: 7899 views (2024050900)
...
```

Listagem 12: Obtendo o dia com maior número de visualizações para cada um dos artigos.

Em seguida, verifica-se a presença de outros artigos relacionados a cada um destes artigos obtidos, com o intuito de descobrir outras possíveis palavras-chave que podem ser utilizadas nos domínios maliciosos. Para tal, realiza-se o seguinte procedimento, demonstrado na Listagem 13: para cada artigo que foi recuperado a partir das categorias de eventos mensais, verifica-se a presença de *links* para outros artigos em seu resumo, visto que, em geral, as informações mais relevantes para determinado assunto estarão contidas em seu resumo.

Além disso, considerando que, em geral, golpes tendem a utilizar nomes próprios/conhecidos para poder enganar mais eficientemente suas vítimas, são obtidos apenas os *links* cujo nome começa com uma letra maiúscula. Uma vez que os nomes dos *links* fornecidos pela biblioteca Wikipedia-API estão todos iniciados com letra maiúscula, independente de como apareceram no artigo, ao verificar a presença destes no resumo do artigo, já serão, naturalmente, extraídos apenas *links* que aparecem com letra inicial maiúscula no resumo.

```
wp_localized = wp if article['lang'] == 'en' else wp_pt
relevant_articles = list()
article_page = wp_localized.page(article['title'])
if article_page.exists():
    if article_page.links:
        for link in article_page.links.keys():
            if link in article_page.summary:
                if article_page.links[link].exists():
                    relevant_articles.append(unquote(article_page.links[link
   ].fullurl.split("/")[-1]))
```

Listagem 13: Obtendo links presentes no resumo de um artigo.

Para cada um destes artigos *linkados*, então, é verificado se ocorreu um pico[2] de visualizações em duas datas distintas: a data em que o artigo original — artigo a partir do qual se obteve o *link* para este — teve mais acessos e a data em que o artigo original foi criado (presumidamente, próximo à data de ocorrência do evento em si). Caso haja, de fato, um pico de interesse nesses artigos secundários em uma dessas datas, pode-se conjecturar uma forte correlação entre estes e o artigo original e, portanto, o artigo secundário em questão é adicionado como candidato para análise.

Para classificar um pico, verifica-se, por meio de uma consulta com o *Pageviews*, o histórico de acessos do artigo na semana anterior à data fornecida e calcula-se sua média de acessos. Analisa-se, então, o aumento de visualizações na data fornecida relativo à média da semana anterior. Caso o número de visualizações na data fornecida esteja acima de um limiar (definido, neste caso, por $média/0.5$, ou seja, o dobro da média), considera-se que este artigo teve um pico nesta data. Este processo de verificação de picos é demonstrado na Listagem 14.

```
relevant_articles_info = list()
for title in relevant_articles:
    results_to_compare = list()
    for timestamp in [original_article_peakdate,
   original_article_creation_date]:
        start = (timestamp - timedelta(days=7)).strftime("%Y%m%d00")
        end = timestamp.strftime("%Y%m%d00")

        this_article = {'title': title, 'lang': article['lang']}
        pageviews_result = self.pageviews(this_article, start=start, end=end
   , frequency="daily")['items']j

        # calcular média de acessos na semana prévia ao acontecimento
        viewcounts = list()
        for i, item in enumerate(pageviews_result):
            if item['timestamp'] == end:
               views_in_original_peakdate = pageviews_result.pop(i)
            else:
                viewcounts.append(item['views'])j
        if viewcounts:
            mean = np.mean(viewcounts)j

            # se houve um pico
            if views_in_original_peakdate and views_in_original_peakdate['
   views'] >= mean/THRESHOLD: # THRESHOLD: 0.5
                results_to_compare.append(views_in_original_peakdate)
```

Listagem 14: Verificando picos de interesse em um artigo.

[2]Aqui, define-se "pico" como um rápido aumento considerável em visualizações.

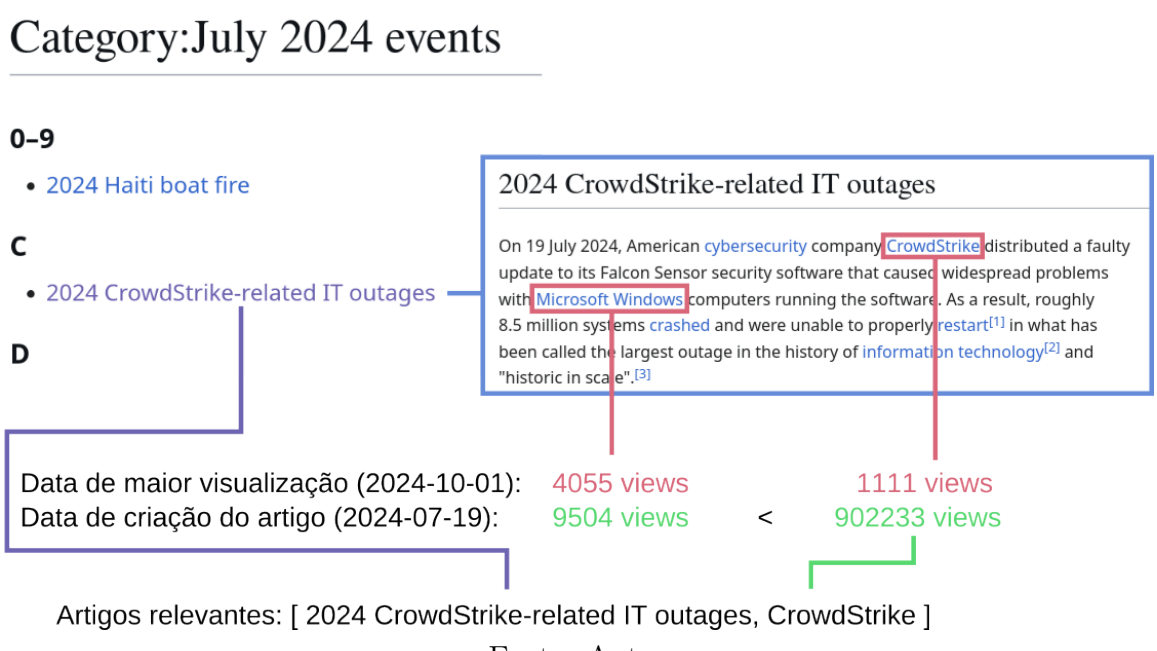
Se houve um pico no artigo em questão em ambas as datas — de maior acesso e de criação do artigo original —, a data em que houve mais acessos entre as duas será a considerada para análise. No código da Listagem 14, `results_to_compare` é a variável que armazena os números de visualizações, caso houve um pico, de um determinado artigo para ambas as datas testadas. Na Listagem 15, verifica-se qual das datas em `results_to_compare` teve mais visualizações, tendo assim uma única estatística de visualização para cada artigo relacionado que foi obtido a partir dos *links* no corpo do artigo original. Por fim, é extraído e retornado, unicamente, o artigo com maior contagem de visualizações entre todos os *linkados* no artigo original.

```
    if results_to_compare:
        relevant_articles_info.append(max(results_to_compare, key=lambda x: x['views']))
if relevant_articles_info:
    most_viewed = max(relevant_articles_info, key=lambda x: x['views'])
    return {'article': {'title': most_viewed['article'], 'lang': article['lang']}, 'views': most_viewed['views'], 'timestamp': most_viewed['timestamp']}
```

Listagem 15: Extraíndo artigo com maior contagem de visualizações

O processo completo de determinar tópicos relevantes é ilustrado graficamente na Figura 4, sendo os números em verde contagens de visualização que configuraram um pico e, em vermelho, os que não configuraram um pico.

Figura 4: Processo de selecionar artigos relevantes para um dos artigos na categoria de eventos de Julho de 2024.



Fonte: Autor.

3.2 Geração de palavras-chave

Após obtida a lista de artigos sobre eventos relevantes, com os quais será analisada a coocorrência de registro de domínios maliciosos, cria-se, então, uma lista de palavras-chave que serão verificadas no banco de dados de domínios maliciosos disponibilizado pelo Nevermore.

Para obter as palavras-chave mais relevantes, realiza-se, inicialmente, alguns passos de pré-processamento: remove-se do título do artigo tanto nomes de meses quanto números relacionados a anos; caso o artigo possua em seu título qualquer número entre 2001 e 2024 (anos em que a Wikipedia esteve ativa), este será removido. Por fim, removem-se quaisquer partículas de linguagem que possam ter sobrado no final do nome do artigo após a remoção de datas, além de outros sinais de pontuação, como '/', '-', ',' e '.'. As partículas de linguagem, tanto em português quanto em inglês, que são consideradas para remoção estão listadas, juntamente com o processo de limpeza do título, na Listagem 16.

```
lang = article["lang"]
title = article["title"]
particles = dict()
particles["pt"]  = ["de", "o", "a", "as", "em", "no", "na", "nos", "nas", "
    que", "do", "da", "um", "uma", "uns", "umas"]
particles["en"] = ["of", "on", "in", "at","from"]
particles["months"] = list(calendar.month_name)[1:]
particles["years"] = [str(x) for x in range(2001,date.today().year+1)]

# sanitizando o titulo
title = title.replace('/','').replace('-','_').replace(',','').replace('.','
    ')
tokens = title.split("_") # espaços no título estão representados por '_'
tokens = [x for x in tokens if x not in particles["years"] and x not in
    particles['months']]

# remove particulas restantes no final do título
if tokens[-1] in particles[lang]:
    tokens.pop(-1)
```

Listagem 16: Partículas de linguagem removidas do título.

A partir desses títulos pré-processados, então, gera-se todas as possíveis combinações das palavras neles contidas. Desta forma, o artigo de título "2024 CrowdStrike-related IT outages", por exemplo, gera o vetor de termos mostrado na Listagem 17.

```
['CrowdStrike', 'CrowdStrike related', 'CrowdStrike related IT', '
    CrowdStrike related IT outages', 'related', 'related IT', 'related IT
    outages', 'IT', 'IT outages', 'outages']
```

Listagem 17: Vetor de termos candidatos para consulta, a partir do título "2024 CrowdStrike-related IT outages".

A partir deste vetor de termos, verifica-se o número de ocorrências de cada um deles ao longo do corpo do artigo, com critério de equivalência exata. Desta forma, ocorrências de "IT outages" não contribuem para a contabilização de ocorrências do termo "outages", por exemplo. Para tal, após obtido o número de ocorrências para cada termo, ordena-se a lista que armazena todos os termos com base nos maiores termos. Começando pelo maior termo, denominado x, de tamanho $len(x)$, verifica-se quais dos termos menores, de tamanho $len(y) < len(x)$, são *substrings*[3] deste e subtrai-se o número de ocorrências de x do número de ocorrências de y. Este processo se repete, com x variando de forma decrescente de tamanho, até que todos os termos sejam contemplados. Isto garante que *substrings* de outro termo candidato sejam contabilizadas apenas quando aparecem de forma solitária. O termo com maior número de ocorrências é, portanto, tomado como o termo do artigo com maior relevância para pesquisa. Este processo é demonstrado na Listagem 18, dada a lista `frequencies`, com as frequências dos termos.

```
frequencies = sorted(frequencies, key=lambda x: len(x['keyword']), reverse=
    True)

# garante que as frequências contabilizadas se refiram a matches exatos
for i, f in enumerate(frequencies):
    string = f['keyword']
    for kw in frequencies[i+1:]:
        if kw['keyword'] in string:
            kw['count'] -= f['count'j
return sorted(frequencies,key=lambda x: x['count'])[-1]
```

Listagem 18: Normalizando frequências de termos.

3.3 Verificação de correlação

Por fim, verifica-se a ocorrência do registro de domínios maliciosos relacionados aos termos obtidos durante o período de até duas semanas após a ocorrência do evento ao qual o termo faz referência. Para isso, utiliza-se da API do Nevermore para encontrar todos os domínios maliciosos em seu banco de dados que contêm cada uma das palavras-chave, em conjunto com a implementação em Python da ferramenta WHOIS, o `python-whois`, para verificar a data de registro destes domínios.

Para cada uma das palavras-chave relacionadas a um acontecimento geradas, realiza-se uma consulta com a função `search` do Nevermore, cuja resposta consiste em todos os domínios maliciosos que contém a palavra-chave em seu nome. Para cada um desses domínios, então, consulta-se sua data de registro a partir de uma requisição WHOIS. Caso o domínio tenha sido registrado dentro do período de duas semanas desde o evento

[3]Cadeia de caracteres contida em uma cadeia maior.

ao qual a palavra-chave se refere, este é contabilizado como tendo a possibilidade de estar se aproveitando da relevância do tópico.

4 Testes e resultados

Para análise, foram utilizados os artigos pertencentes às categorias de eventos de Abril de 2024 até Setembro de 2024, tanto no escopo brasileiro quanto global. Na Tabela 1, são mostrados todos os artigos extraídos e que foram utilizados para a análise, juntamente com a data em que teve maior número de visualizações em um único dia.

Título	Visualizações	Data
September 2024 lunar eclipse	140762	18/09/2024
May 2024 solar storms	49333	11/05/2024
2024 Rio Grande do Sul floods	27674	09/05/2024
Voo Voepass 2283	10677	10/08/2024
Enchentes no Rio Grande do Sul em 2024	7899	09/05/2024
Expedition 71	5108	01/06/2024
International Criminal Court investigation in Palestine	4702	20/05/2024
United Nations Security Council Resolution 2735	2245	11/06/2024
List of pro-Palestinian protests on university campuses in 2024	2183	06/05/2024
Blocking of Twitter in Brazil	2152	09/10/2024
2024 Brazil wildfires	1439	14/09/2024
Voepass Flight 2283	1333	28/09/2024
Bloqueio do X no Brasil	1237	03/09/2024
2024 CrowdStrike-related IT outages	1075	01/10/2024
Expedition 72	992	30/09/2024
Caso Tio Paulo	794	22/04/2024
United Nations Security Council Resolution 2736	761	14/06/2024
Steadfast Defender 2024	711	04/05/2024
Incêndio em pousada de Porto Alegre	558	27/04/2024
Incêndios florestais no Brasil em 2024	303	26/08/2024
Expedition 70	243	06/06/2024
Uncle Paulo incident	242	21/04/2024
2024 South American wildfires	238	25/09/2024
United Nations Security Council Resolution 2749	151	11/10/2024
2024 Porto Alegre fire	95	27/04/2024
United Nations General Assembly Resolution 78/282	81	10/06/2024
Wildcat Fire	49	02/08/2024
Prêmio da Música Brasileira 2024	42	14/06/2024
Global Alliance Against Hunger and Poverty	38	25/07/2024
2024 Haiti boat fire	33	01/09/2024
Varahi declaration	19	07/10/2024
Framework Convention on Artificial Intelligence	15	25/09/2024
Prêmio da Música Brasileira de 2024	7	21/09/2024

Tabela 1: Eventos com maior número de visualização em um único dia.

A partir destes artigos, são extraídos artigos relacionados e estes são concatenados ao conjunto inicial. Por fim, ordena-se o conjunto com todos os artigos — tanto originais quanto os relacionados — por número de visualizações em um único dia e, para esta

análise, são selecionados os 10 primeiros colocados, mostrados na Tabela 2.

Título	Visualizações	Data
CrowdStrike	902233	19/07/2024
ATR 72	148758	09/08/2024
September 2024 lunar eclipse	140762	18/09/2024
Karim Ahmad Khan	139613	20/05/2024
ATR-72	56672	09/08/2024
May 2024 solar storms	49333	11/05/2024
2024 Rio Grande do Sul floods	27674	09/05/2024
United Nations Interim Force in Lebanon	19354	11/10/2024
Alexandre de Moraes	14132	30/08/2024
Rio Grande do Sul	13852	09/05/2024

Tabela 2: Eventos com maior número de visualização em um único dia (a partir do conjunto completo).

Ao verificar os eventos aos quais estes artigos se referem e a data em que estes eventos, de fato, ocorreram, ilustrado na Tabela 3, confirma-se que os picos de interesse em um determinado evento ocorrem, em média, em até 1 dia após a data de seu acontecimento, como pode ser visto na Tabela 4.

Artigo	**Evento relacionado**	**Data do ocorrido**
CrowdStrike	Incidentes de TI relacionados a atualização falha do CrowdStrike Falcon	19/07/2024
ATR 72	Queda do Voo Voepass 2283	09/08/2024
September 2024 lunar eclipse	Eclipse lunar em Setembro de 2024	17/08/2024
Karim Ahmad Khan	Aplicação para mandato de prisão do primeiro ministro de Israel e lider do Hamas	20/05/2024
May 2024 solar storms	Tempestades solares ocorridas em Maio de 2024	10/05/2024
2024 Rio Grande do Sul floods	Enchentes no estado do RS	28/04/2024 - 07/05/2024
United Nations Interim Force in Lebanon	Ataque de tropas israelenses ao UNIFIL	10/10/2024
Alexandre de Moraes	Mandato de suspensão do Twitter/X no Brasil	30/08/2024
Rio Grande do Sul	Enchentes no estado do RS	28/04/2024 - 07/05/2024

Tabela 3: Artigos e os eventos aos quais se referem.

Para cada um destes artigos, então, foi gerada a palavra-chave mais relevante para pesquisa, como apresentado na Tabela 5.

Evento	**Data do ocorrido**	**Data com mais visualizações no artigo correspondente**
Incidentes de TI relacionados a atualização falha do CrowdStrike Falcon	19/07/2024	19/07/2024
Queda do Voo Voepass 2283	09/08/2024	09/08/2024
Eclipse lunar em Setembro de 2024	17/08/2024	18/09/2024
Aplicação para mandato de prisão do primeiro ministro de Israel e lider do Hamas	20/05/2024	20/05/2024
Tempestades solares ocorridas em Maio de 2024	10/05/2024	11/05/2024
Enchentes no estado do RS	28/04/2024 - 07/05/2024	09/05/2024
Ataque de tropas israelenses ao UNIFIL	10/10/2024	11/10/2024
Mandato de suspensão do Twitter/X no Brasil	30/08/2024	30/08/2024
Enchentes no estado do RS	28/04/2024 - 07/05/2024	09/05/2024

Tabela 4: Comparação entre datas de ocorrência de um evento e a de pico de visualizações em sua página correspondente.

Artigo	Palavra-chave
CrowdStrike	CrowdStrike
ATR 72	ATR
September 2024 lunar eclipse	lunareclipse
Karim Ahmad Khan	Khan
ATR-72	ATR
May 2024 solar storms	solar
2024 Rio Grande do Sul floods	floods
United Nations Interim Force in Lebanon	Lebanon
Alexandre de Moraes	Moraes
Rio Grande do Sul	RioGrandedoSul

Tabela 5: Artigos e suas respectivas palavras-chave.

Por fim, cada uma das palavras-chave foram submetidas ao Nevermore para verificar a presença de domínios que as contêm e, então, verificar, com auxílio da ferramenta WHOIS, se o domínio foi registrado perto da data de ocorrência do evento ao qual a palavra-chave se refere. Os resultados desta busca encontram-se na Tabela 6

Palavra-chave	Maliciosos no BD	Registrados dentro de 2 semanas	Porcentagem
CrowdStrike	157	136	86.62%
ATR	3501	54	1.54%
lunareclipse	0	0	-
Khan	231	2	0.8%
solar	461	6	1.30%
floods	0	0	-
Lebanon	19	0	0%
Moraes	5	0	0%
RioGrandedoSul	0	0	-

Tabela 6: Palavras-chave e seus respectivos resultados da pesquisa no banco de dados do Nevermore.

Discussão dos resultados

Os resultados obtidos mostram uma quantidade significativa de domínios maliciosos registrados relacionados ao incidente da CrowdStrike, evento de cunho tecnológico, mas quantidade decepcionante em relação aos outros eventos verificados. Algumas das palavras-chave não renderam um único domínio sequer, enquanto outras renderam uma quantidade ínfima, a basear-se na porcentagem que representam em relação a outros domínios em que esta palavra-chave aparece registrados em período prévio ao evento relacionado. Ao mesmo tempo, ao analisar estes termos que obtiveram baixa porcentagem, verifica-se, também, que os domínios registrados no período de 2 semanas após o acontecimento do evento relacionado não aparentam ter correlação com o evento em si.

Já nos resultados relacionados ao termo “CrowdStrike”, ao analisar, manualmente, os nomes dos domínios, é razoável concluir que, efetivamente, todos estão se aproveitando do incidente. São mostrados, na Listagem 19, alguns exemplos de domínios que foram identificados, juntamente com suas datas de registro.

```
bsodcrowdstrike.com: 2024-07-20
crowdstrikeerrorrecoveryhub.com: 2024-07-20
crowdstrikefailsafe.com: 2024-07-20
crowdstrikefix.zip: 2024-07-19
crowdstrikeglitch.com: 2024-07-19
crowdstrikerecovery.com: 2024-07-19
fix-crowdstrike.com: 2024-07-19
```

Listagem 19: Exemplos de domínios detectados se aproveitando do incidente da CrowdStrike.

Quanto à performance ínfima dos outros tópicos analisados, este fenômeno se dá, provavelmente, pelos seguintes motivos: primeiramente, como pode ser observado na Tabela 7, "CrowdStrike" era o único tópico diretamente relacionado à tecnologia. Os demais tópicos pertencem a outras categorias, como política e desastres e, em geral, não afetam diretamente a maior parte da população mundial/brasileira. Em comparação, o incidente envolvendo o CrowdStrike Falcon teve repercussões mundiais e afetou diversos aspectos da vida cotidiana do cidadão comum, como a disponibilidade de bancos e outros serviços digitais, tanto no Brasil como no mundo.

Tópico	Categoria
CrowdStrike	Tecnologia
ATR 72	Aviação/Desastre
September 2024 lunar eclipse	Evento astronômico
Karim Ahmad Khan	Política
ATR-72	Aviação/Desastre
May 2024 solar storms	Evento astronômico
2024 Rio Grande do Sul floods	Desastre
United Nations Interim Force in Lebanon	Política
Alexandre de Moraes	Política/Tecnologia
Rio Grande do Sul	Desastre

Tabela 7: Tópicos e suas respectivas categorias.

O outro possível fator que afetou os resultados é a falta de *blocklists* brasileiras. Em geral, não existem *blocklists* de domínios maliciosos com enfoque brasileiro. Desta forma, a presença de domínios referentes a fraudes aproveitando-se de tópicos relacionados a eventos nacionais em *blocklists* é extremamente escassa.

Além disso, palavras-chave curtas, como "ATR" tendem a ser genéricas demais, dado que, quanto menor o termo, maior sua chance de pertencer a grande parte do corpo léxico de uma linguagem. Desta forma, palavras-chave muito curtas mostram-se não tão eficazes para o registro de domínios fraudulentos, além de serem vagas demais para detectar as possíveis fraudes advindas delas, visto que transmitem muito pouca informação.

Por fim, é importante ressaltar que, para diversos domínios presentes nas *blocklists* do Nevermore, não foi possível recuperar informações sobre sua data de registro utilizando da ferramenta WHOIS, seja porque estas não estavam disponíveis ou pertencem a um domínio que foi removido do banco de dados, e, portanto, não foram contabilizados.

Portanto, há grande possibilidade de existirem diversos domínios que estão tomando vantagem de um determinado tópico atualmente relevante, mas não estão sendo detectados, por algum dos motivos supracitados.

5 Conclusão

Domínios utilizados para fins maliciosos existem desde os primórdios da Internet e continuam sempre evoluindo, a fim de evitar detecção e enganar mais eficientemente suas vítimas. Embora não ocorra para todo tipo de evento, a partir do exposto, pode-se conjecturar que existe, de fato, uma correlação entre a relevância de um tópico no imaginário coletivo e o registro de domínios maliciosos que se aproveitam desta popularidade momentânea, em especial para tópicos diretamente ligados à tecnologia.

A partir deste conhecimento, podem ser implementadas diversas táticas preventivas para combater atividades maliciosas na Internet. O sistema, desenvolvido neste trabalho, para identificação de tópicos e palavras-chave relevantes durante dado período de tempo pode ser utilizado de diversas maneiras, incluindo na criação momentânea de *blocklists* dinâmicas para filtrar o tráfego de DNS que circula em uma rede, a fim de evitar ataques de alto risco no momento, e na alimentação de modelos de Aprendizado de Máquina e Inteligência Artificial utilizados para classificação de domínios como maliciosos ou benignos, classificando mais rigorosamente domínios que incluem palavras-chave relacionadas a eventos relevantes recentes em seu nome.

Dificuldades encontradas

A principal dificuldade encontrada durante o desenvolvimento do trabalho foi a elaboração de uma maneira confiável de deduzir tópicos atualmente relevantes no imaginário coletivo. Inicialmente, para tal fim, estava sendo utilizado o Google Trends — serviço da Google que fornece estatísticas sobre pesquisas em seus diversos outros serviços —, por meio da API não oficial pytrends, para obter dados sobre quais pesquisas estão em alta e, a partir disso, aferir a relevância de um tópico e outros a ele relacionados. Entretanto, durante o desenvolvimento do trabalho, houveram mudanças no funcionamento do Google Trends, fazendo com que este bloqueasse qualquer tipo de *bot* e *scraper*[4], impossibilitando a extração programática de dados do serviço. Após diversas tentativas falhas de circunvir os bloqueios do Google Trends, foi necessária a elaboração de outra abordagem que não envolvesse este serviço. Eventualmente, foi elaborado o sistema utilizado neste trabalho, correlacionando diversos recursos da Wikipedia.

Além disso, como citado anteriormente, um fator que dificultou obtenção de resultados foi a escassez de *blocklists* com enfoque brasileiro. Tal falta dificulta a pesquisa e desenvolvimento na área de segurança de computadores com enfoque nacional e, consequentemente, introduz um potencial furo de segurança em nossa rede de computadores, uma vez que fraudes regionais acabam não recebendo tanta atenção quanto as internacionais.

[4]Tipo de *bot* que extrai dados de uma página *web*

Referências Bibliográficas

ALBITZ, P.; LIU, C. *DNS and BIND*. 1st. ed. USA: O'Reilly & Associates, Inc., 1993. ISBN 1565920104.

BATISTA, V. B. Framework para integração e automação de modelos de aprendizado de máquina para classificação de nomes de domínio. *Universidade Estadual Paulista (Unesp)*, 2020.

BILGE, L. et al. Exposure: A passive dns analysis service to detect and report malicious domains. *ACM Trans. Inf. Syst. Secur.*, Association for Computing Machinery, New York, NY, USA, v. 16, n. 4, apr 2014. ISSN 1094-9224. Disponível em: <https://doi.org/10.1145/2584679>.

CERT.BR. *Incidentes Notificados ao CERT.br*. 2024. Disponível em: <https://stats.cert.br/incidentes/>. Acesso em: 8 jul. 2024.

CERT.BR. *Sobre a Cartilha de Segurança para Internet*. 2024. Disponível em: <https://cartilha.cert.br/sobre/>. Acesso em: 8 jul. 2024.

CLOUDFLARE. *What is an API endpoint?* 2024. Disponível em: <https://www.cloudflare.com/learning/security/api/what-is-api-endpoint/>. Acesso em: 2 jul. 2024.

DAIGLE, L. *WHOIS Protocol Specification*. RFC Editor, 2004. RFC 3912. (Request for Comments, 3912). Disponível em: <https://www.rfc-editor.org/info/rfc3912>.

GOODWIN, M. *What is an Application Programming Interface (API)*. IBM, 2024. Disponível em: <https://www.ibm.com/topics/api>. Acesso em: 2 jul. 2024.

HOFFMAN, P. E.; SULLIVAN, A.; FUJIWARA, K. *DNS Terminology*. RFC Editor, 2019. RFC 8499. (Request for Comments, 8499). Disponível em: <https://www.rfc-editor.org/info/rfc8499>.

KEMP, S. *Digital 2024: Brazil*. DataReportal -Global Digital Insights, 2024. Disponível em: <https://datareportal.com/reports/digital-2024-brazil>.

KUROSE, J. F.; ROSS, K. W. *Computer Networking: A Top-Down Approach (6th Edition)*. 6th. ed. [S.l.]: Pearson, 2012. ISBN 0132856204.

MAJLIS martin. *Wikipedia-API: Python wrapper for Wikipedia*. GitHub, 2017. Disponível em: <https://github.com/martin-majlis/Wikipedia-API>. Acesso em: 09 out. 2024.

MOCKAPETRIS, P. *Domain names: Concepts and facilities*. RFC Editor, 1983. RFC 882. (Request for Comments, 882). Disponível em: <https://www.rfc-editor.org/info/rfc882>.

MOCKAPETRIS, P. *Domain names: Implementation specification*. RFC Editor, 1983. RFC 883. (Request for Comments, 883). Disponível em: <https://www.rfc-editor.org/info/rfc883>.

MOCKAPETRIS, P. *Domain names - concepts and facilities*. RFC Editor, 1987. RFC 1034. (Request for Comments, 1034). Disponível em: <https://www.rfc-editor.org/info/rfc1034>.

MOCKAPETRIS, P. *Domain names - implementation and specification*. RFC Editor, 1987. RFC 1035. (Request for Comments, 1035). Disponível em: <https://www.rfc-editor.org/info/rfc1035>.

REGISTRO.BR. *Estatísticas - Registro.br*. 2024. Disponível em: <https://registro.br/dominio/estatisticas>. Acesso em: 28 jun. 2024.

WIKIMEDIA. *Pageviews Analysis — Meta*. 2024. Disponível em: <https://meta.wikimedia.org/wiki/Pageviews_Analysis>. Acesso em: 09 out. 2024.

WIKIPEDIA. *Help:Categories — Wikipedia, The Free Encyclopedia*. 2024. Disponível em: <https://en.wikipedia.org/wiki/Help:Category>. Acesso em: 09 out. 2024.

WIKIPEDIA. *Wikipedia — Wikipedia, The Free Encyclopedia*. 2024. Disponível em: <https://en.wikipedia.org/wiki/Wikipedia>. Acesso em: 09 out. 2024.

WIKIPEDIA. *Wikipedia:Citing sources — Wikipedia, The Free Encyclopedia*. 2024. Disponível em: <https://en.wikipedia.org/wiki/Wikipedia:Citing_sources>. Acesso em: 09 out. 2024.

WIKIPEDIA. *Wikipedia:Five pillars — Wikipedia, The Free Encyclopedia*. 2024. Disponível em: <https://en.wikipedia.org/wiki/Wikipedia:Five_pillars>. Acesso em: 09 out. 2024.

WIKIPEDIA. *Wikipedia:Notability — Wikipedia, The Free Encyclopedia*. 2024. Disponível em: <https://en.wikipedia.org/wiki/Wikipedia:Notability>. Acesso em: 09 out. 2024.

ZHAO, H. et al. Malicious domain names detection algorithm based on lexical analysis and feature quantification. *IEEE Access*, v. 7, p. 128990–128999, 2019.