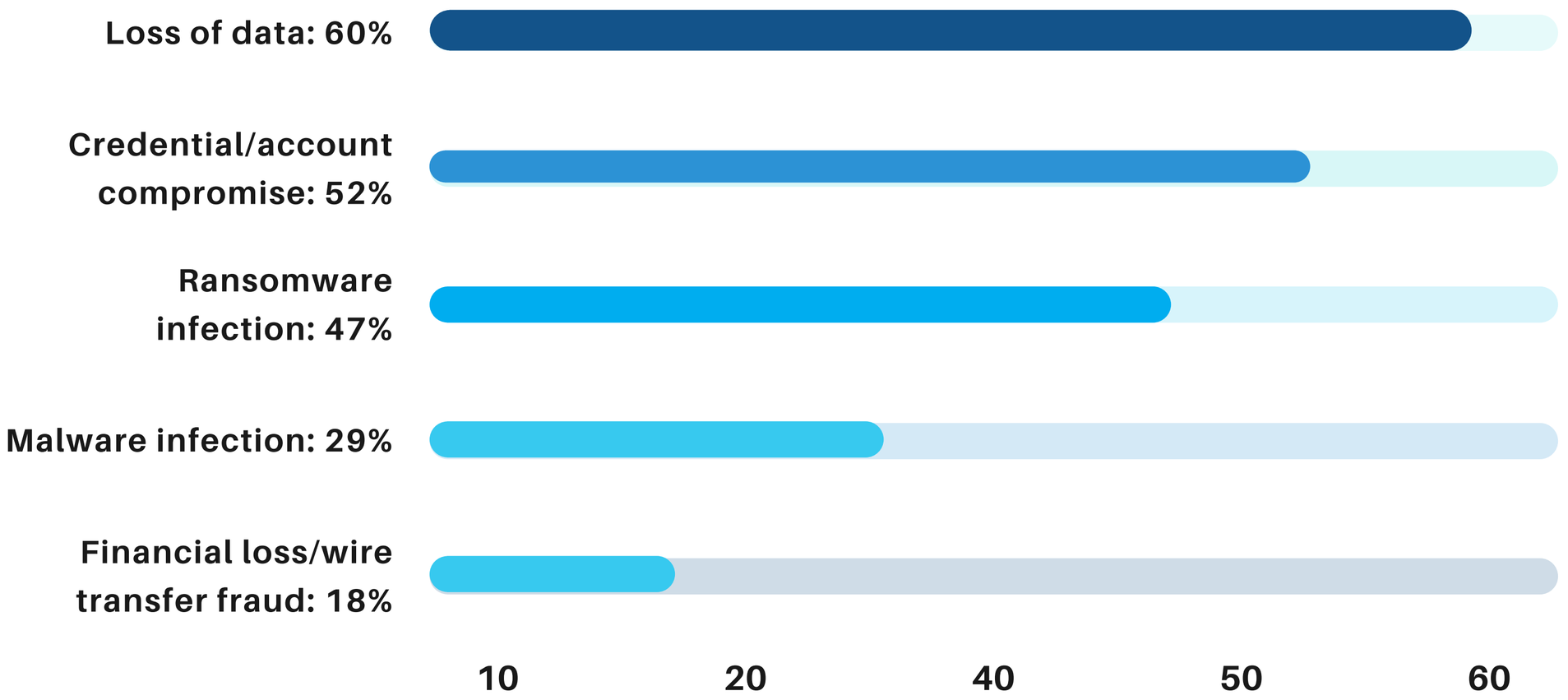
SharkDefense

Every Click Counts

What is Phishing Attack?

- Impersonation: Pretends to be trustworthy entity
- Goal: Steal sensitive information
- Common methods: Email, text messages, social media
- Indicators: Urgent messages, suspicious links, grammatical errors
- Examples: Email phishing, spear phishing, vishing
- Impact: Financial loss, identity theft, reputational damage
- Defense strategies: Security awareness training, email filtering, caution with unsolicited messages

Impacts of Successful Phishing Attacks



Defenses Against Phishing

Filtering E-mails

Filter spam and fight malicious attacks
Discard emails from untrusted sources

Blocking Malicious Attachments

.exe, .docx, .pdf.exe, .rar, .zap, .gz, etc

Challenges with Malicious URLs

It Can't do the Same With e-mails That contain Url's because It's more complex

Problems With Security Vendors

Virus Total

- Depends on IOC That comes from other security vendors
- Not trusted results every time
- Loss of Some Of Its reputation due to small incident

Urlscan.io

- It depends on the basic rules of the URL
- Threat actors could Use Any Fake IP, VPN, or proxy
- Fake TLS Certificate

What is SharkDefense?

Identifying malicious URLs to protect users from online threats, providing enhanced security for individuals and organizations based on Security Rules and Updated data set

- Website
- Mobile app
- Google extensions
- Feedback

Target Audiences

Non-Tech Users

URL from untrusted users or strange URL

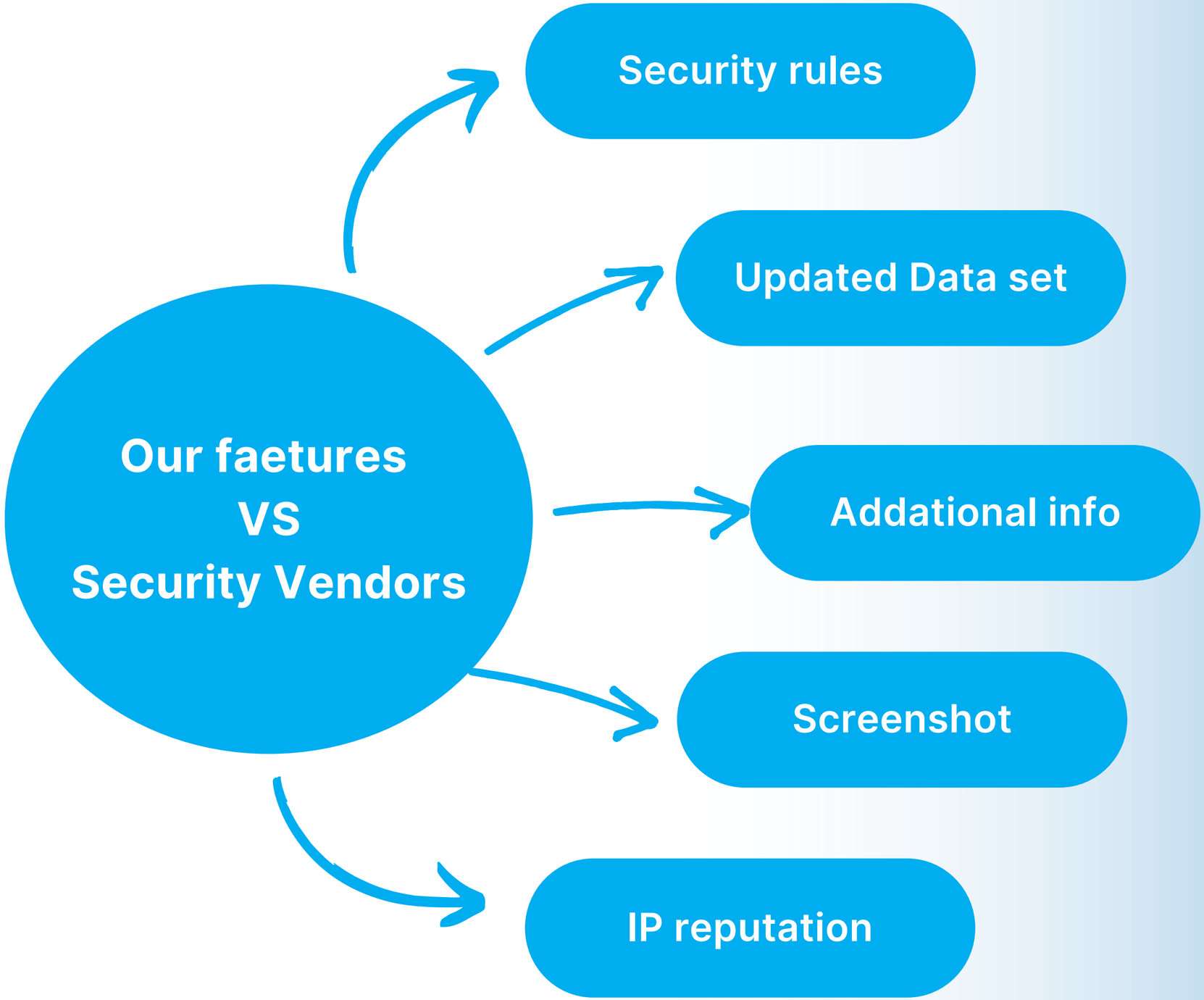
Tech Users

Developers and security engineers Info about the URL and rules

Businesses

Security Awareness Training, Phishing Simulator, and updated data set

Features


Our faetures
VS
Security Vendors
Security rules
Updated Data set
Addational info
Screenshot
IP reputation

Security Awareness Training

Protect your organization from cyber threats. Equip your employees with the skills they need through our engaging security awareness training sessions

Our Parteners

Root X

CYBER PEN

NUB (in the future)

Updated Data Sets

Our threat intelligence team updates the information every 8 hours. We work in cooperation with other companies to enhance our threat intelligence capabilities.

Threat intelligence

Our threat intelligence team updates the information every 8 hours.

Cooperation with companies

We work in cooperation with other companies to enhance our threat intelligence capabilities.

New malicious Url

When new URL detected by the security rules it's add to our Data set

Phishing simulations

Phishing simulations are controlled experiments that mimic real-life phishing attacks without malicious intent. These simulations are typically orchestrated by cybersecurity experts at Shark Eye to send emails or messages that resemble phishing attempts to the organization's staff. The goal is to see how employees react: whether they recognize the attempt as a phishing scam and report it, or if they fall for it by clicking on links, downloading attachments, or providing sensitive information.

Analysis Data

(1)

jupyter Untitled Last Checkpoint: yesterday

File Edit View Run Kernel Settings Help

Code

```
import plotly.graph_objects as go
```

[2]:
```
urls_data=pd.read_csv('malicious_phish.csv')
urls_data.head()
```

[2]:

	url	type
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

[3]:
```
urls_data.tail()
```

[3]:

	url	type
651186	xbox360.ign.com/objects/850/850402.html	phishing
651187	games.teamxbox.com/xbox-360/1860/Dead-Space/	phishing
651188	www.gamespot.com/xbox360/action/deadspace/	phishing
651189	en.wikipedia.org/wiki/Dead_Space_(video_game)	phishing
651190	www.angelfire.com/goth/devilmaycrytonite/	phishing

[4]:
```
urls_data.info()
```

(2)

jupyter Untitled Last Checkpoint: yesterday

File Edit View Run Kernel Settings Help

Code

[4]:
```
urls_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 651191 entries, 0 to 651190
Data columns (total 2 columns):
 #   Column  Non-Null Count   Dtype
---  ------  --------------   -----
 0   url     651191 non-null  object
 1   type    651191 non-null  object
dtypes: object(2)
memory usage: 9.9+ MB
```

[5]:
```
print("urls_data shape:", urls_data.shape)
```

```
urls_data shape: (651191, 2)
```

[6]:
```
urls_data.keys()
```

[6]:
```
Index(['url', 'type'], dtype='object')
```

[7]:
```
urls_data.describe()
```

[7]:

	url	type
count	651191	651191
unique	641119	4
top	http://style.org.hc360.com/css/detail/mysite/s...	benign
freq	180	428103

Analysis Data

(3)

jupyter Untitled Last Checkpoint: yesterday

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python 3 (ipykernel)

[9]:
```
urls_data["type"].value_counts()
```

[9]:
```
type
benign        428103
defacement     96457
phishing       94111
malware        32520
Name: count, dtype: int64
```

[8]:
```
urls_data.isna().sum()
```

[8]:
```
url     0
type    0
dtype: int64
```

[10]:
```
missingno.matrix(urls_data)
```

[10]: <Axes: >

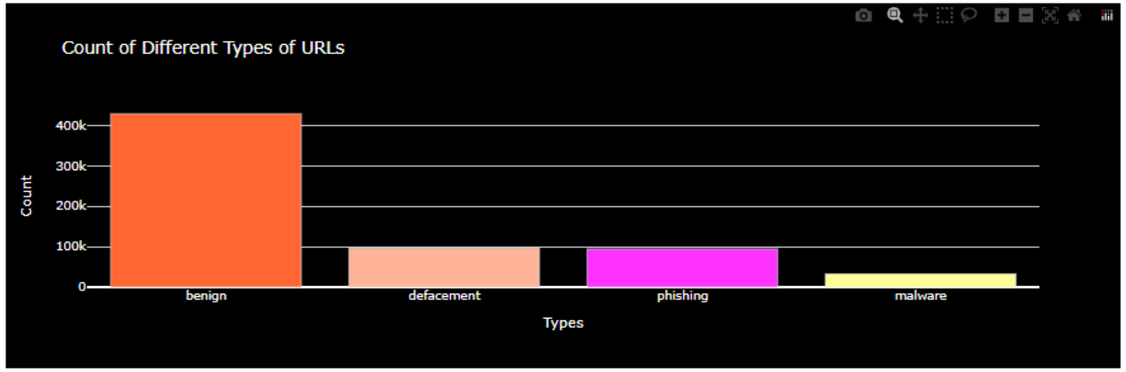
(4)

jupyter Untitled Last Checkpoint: yesterday

File Edit View Run Kernel Settings Help Trusted

Code JupyterLab Python 3 (ipykernel)

[11]:
```
count = urls_data['type'].value_counts()
colors = [
    '#FF6633', '#FFB399', '#FF33FF', '#FFFF99', '#00B3E6',
    '#E6B333', '#3366E6', '#999966', '#99FF99', '#B34D4D'
]
fig = go.Figure(data=[go.Bar(x=count.index, y=count, marker=dict(color=colors))])
fig.update_layout(
    xaxis_title='Types',
    yaxis_title='Count',
    title='Count of Different Types of URLs',
    plot_bgcolor='black',
    paper_bgcolor='black',
    font=dict(color='white')
)
fig.update_xaxes(tickfont=dict(color='white'))
fig.update_yaxes(tickfont=dict(color='white'))
fig.show()
```



Feature Engineering & Data Processing

(1)

jupyter Untitled Last Checkpoint: yesterday

File Edit View Run Kernel Settings Help

Code JupyterLab Python 3 (ipykernel)

[12]:
```
urls_data['url'] = urls_data['url'].replace('www.', '', regex=True)
urls_data.head()
```

[12]:

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

[13]:
```
urls_data["url_type"] = urls_data["type"].replace({
    'benign':0,
    'defacement':1,
    'phishing':2,
    'malware':3
});
```

```
C:\Users\H .S\AppData\Local\Temp\ipykernel_9288\3461687872.py:1: FutureWarning:

Downcasting behavior in `replace` is deprecated and will be removed in a future version. To retain the old behavior, explicitly call `result.infer_objec
ts(copy=False)`. To opt-in to the future behavior, set `pd.set_option('future.no_silent_downcasting', True)`
```

[14]:
```
urls_data.head()
```

[14]:

	url	type	url_type
0	br-icloud.com.br	phishing	2
1	mp3raid.com/music/krizz_kaliko.html	benign	0
2	bopsecrets.org/rexroth/cr/1.htm	benign	0
3	http://garage-pirenne.be/index.php?option=com_...	defacement	1

(2)

jupyter Untitled Last Checkpoint: yesterday

File Edit View Run Kernel Settings Help

Code

[15]:
```
def get_url_length(url):
    # Remove common prefixes
    prefixes = ['http://', 'https://']
    for prefix in prefixes:
        if url.startswith(prefix):
            url = url[len(prefix):]

    # Remove 'www.' if present
    url = url.replace('www.', '')

    # Return the length of the remaining URL
    return len(url)
```

[16]:
```
urls_data['url_len'] = urls_data['url'].apply(lambda x: get_url_length(str(x)))
```

[17]:
```
urls_data.head()
```

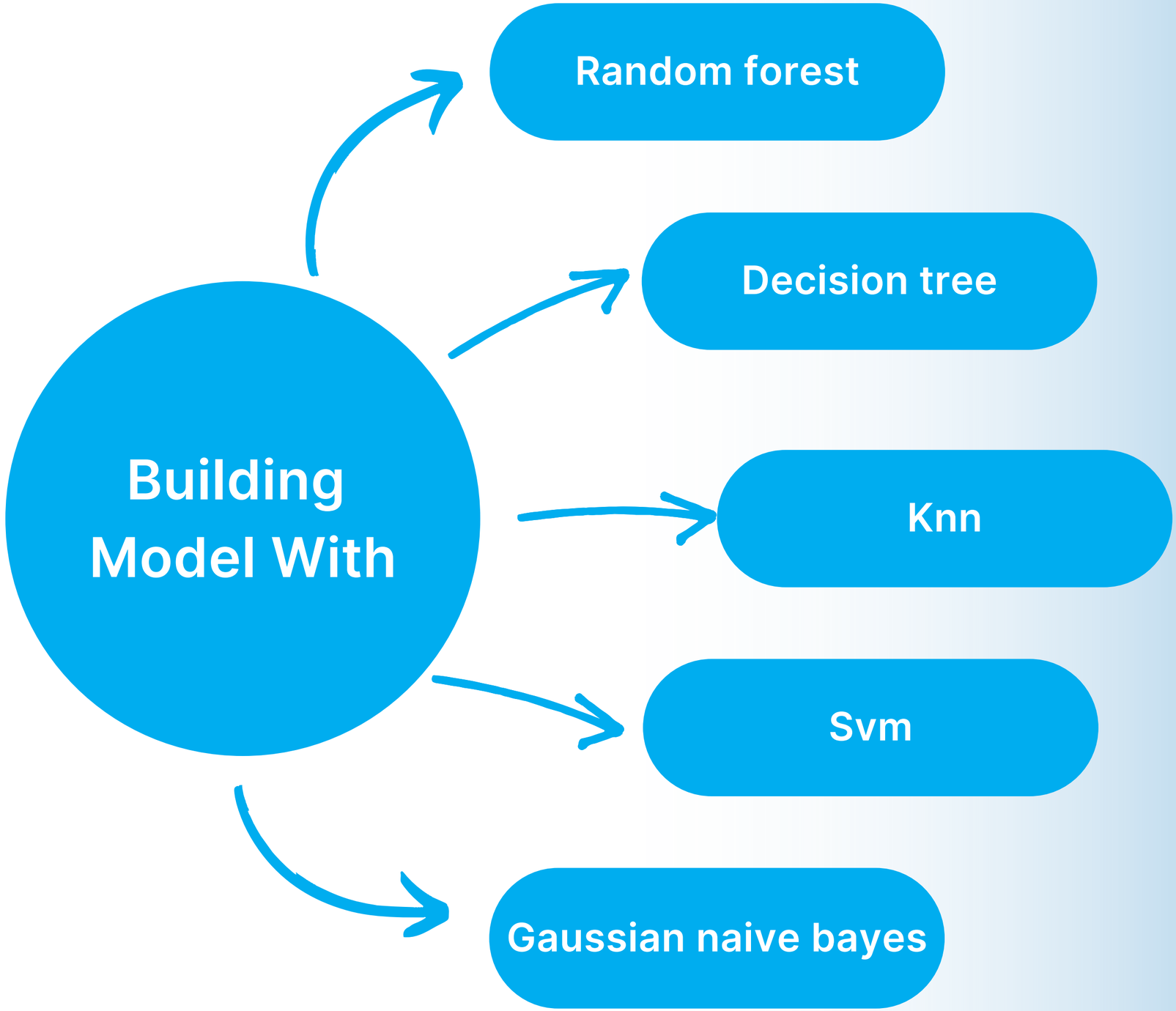
[17]:

	url	type	url_type	url_len
0	br-icloud.com.br	phishing	2	16
1	mp3raid.com/music/krizz_kaliko.html	benign	0	35
2	bopsecrets.org/rexroth/cr/1.htm	benign	0	31
3	http://garage-pirenne.be/index.php?option=com_...	defacement	1	77
4	http://adventure-nicaragua.net/index.php?optio...	defacement	1	228

[18]:
```
def extract_pri_domain(url):
    try:
        res = get_tld(url, as_object = True, fail_silently=False,fix_protocol=True)
        pri_domain= res.parsed_url.netloc
    except :
        pri_domain= None
    return pri_domain
```

[19]:
```
urls_data['pri_domain'] = urls_data['url'].apply(lambda x: extract_pri_domain(x))
```

Algorithms


Building
Model With
Random forest
Decision tree
Knn
Svm
Gaussian naive bayes

Backend Technologies

RESTful API

- Receive a URL as a post request
- input validation and error handling
- invoke a machine learning model to get the classification result

Flask

Flask is lightweight and requires minimal setup, making it a great choice for building small to medium-sized APIs. This makes Flask an ideal choice for our project to build robust and scalable APIs

Mongodb

MongoDB is a popular, open-source NoSQL database management system that stores data in a flexible, JSON format

- Performance.
- High availability
- Horizontal Scaling.

Code snippets

Graph visualization

```python
import requests
import networkx as nx
import pyvis.network as net

def get_subdomains(api_key, domain):
    api_url = f'https://api.securitytrails.com/v1/domain/{domain}/subdomains'
    headers = {'APIKEY': api_key}

    response = requests.get(api_url, headers=headers)

    if response.status_code == 200:
        subdomains = response.json().get('subdomains', [])
        return subdomains
    else:
        print(f"Failed to fetch subdomains. Status code: {response.status_code}")
        return []

def generate_graph(main_domain,subdomains):
    # Initialize a directed graph
    graph = nx.DiGraph()

    # Add URL as the central node
    graph.add_node(main_domain)

    # Add subdomains as nodes and edges
    for subdomain in subdomains[:40]:
        graph.add_node(subdomain)
        graph.add_edge(main_domain, subdomain)

    # Create the graph visualization
    pyvis_graph = net.Network(height="500px", width="100%", directed=True, notebook=False)
    pyvis_graph.from_nx(graph)
    pyvis_graph.show_buttons(filter_=['nodes'])
    html = pyvis_graph.generate_html()

    return html

#usage example
api_key = 'API_KEY'
main_domain = 'google.com'
subdomains = get_subdomains(api_key, main_domain)
graph_html=generate_graph(main_domain,subdomains)

# Save the HTML to a file
with open('subdomain_graph.html', 'w') as file:
    file.write(graph_html)

```

Screenshot

```python
from selenium import webdriver
from selenium.webdriver.chrome.options import Options

def capture_screenshot_as_base64(url):
    chrome_options = Options()
    chrome_options.add_argument('--headless')  # Run Chrome in headless mode (without UI)

    driver = webdriver.Chrome(options=chrome_options)

    try:
        driver.get(url)

    finally:
        # Capture a screenshot and convert it to base64
        screenshot_base64 = driver.get_screenshot_as_base64()
        driver.quit()

    return screenshot_base64


if __name__ == "__main__":
    url_to_capture = "https://example.com"
    screenshot_data = capture_screenshot_as_base64(url_to_capture)

    print(screenshot_data)

```

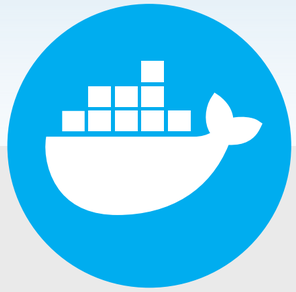
IP reputation

```python
import requests
import json
import socket
from urllib.parse import urlparse

def extract_domain_from_url(url):
    domain = None
    try:
        parsed_url = urlparse(url)
        domain = parsed_url.netloc
    except Exception as e:
        print(f"Error occurred while parsing the URL: {e}")
    return domain

def get_ip_address(domain):
    try:
        ip_address = socket.gethostbyname(domain)
        return ip_address
    except socket.gaierror as e:
        print(f"Error occurred while resolving IP address: {e}")
        return None

def check_ip_reputation(ip_address):

            try:
                url = 'https://api.abuseipdb.com/api/v2/check'

                querystring = {
                    'ipAddress': f'{ip_address}',
                    'maxAgeInDays': '90'
                }

                headers = {
                    'Accept': 'application/json',
                    'Key': 'api key'
                }

                response = requests.request(method='GET', url=url, headers=headers, params=querystring)

                if response.status_code == 200:
                        result = response.json()
                        return result

                else:
                    print(f"Error occurred while retrieving reputation information. Status code: {response.status_code}")

            except requests.exceptions.RequestException as e:
                print(f"Error occurred while retrieving reputation information: {e}")

url='https://google.com'
domain=extract_domain_from_url(url)
ip=get_ip_address(domain)
reputation=check_ip_reputation(ip)
formatted_reputation = json.dumps(reputation, indent=4)
print(formatted_reputation)

```

DEVOPS AND CLOUD TECHNOLOGIES

Git is a distributed version control system for tracking changes in source code during software development

GitHub is a web-based hosting service for version control using Git.



Docker is an open-source platform for developing, shipping, and running applications in containers

DEVOPS AND CLOUD TECHNOLOGIES

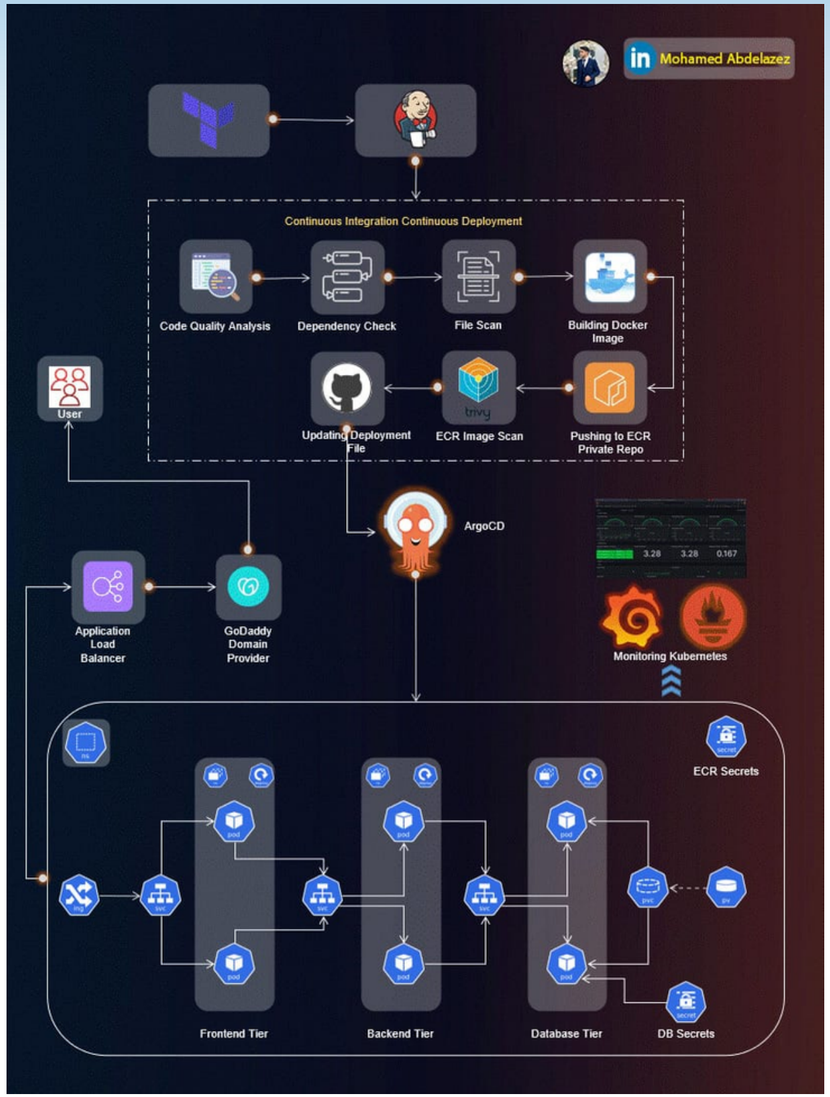
Amazon Elastic Kubernetes Service (EKS) is a managed Kubernetes service provided by AWS.

Prometheus is an open-source monitoring and alerting toolkit. Grafana is an open-source platform for monitoring and observability

ELK Stack is a combination of three open-source projects: Elasticsearch, Logstash, and Kibana.

DEVOPS AND CLOUD TECHNOLOGIES

CI/CD Pipeline Flow



(UI) / (UX) Design

1

visually appealing and user-friendly interfaces visually appealing and user-friendly interfaces

2

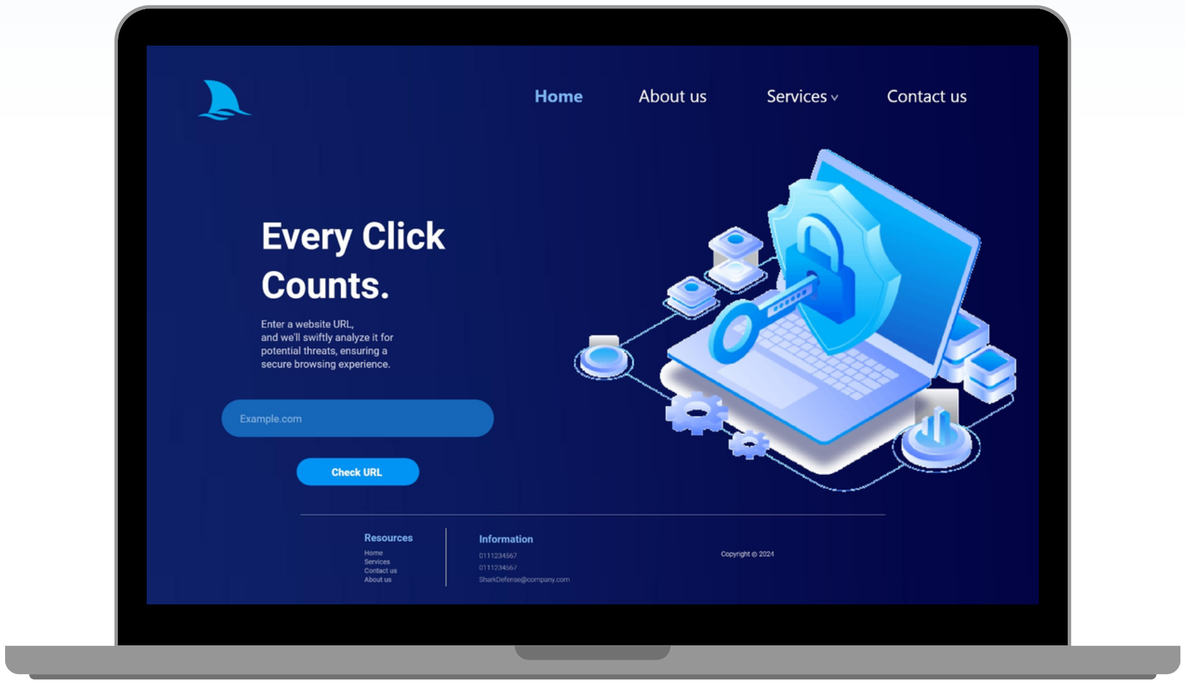
Utilizing clear and concise messages to educate users about potential phishing threats.

3

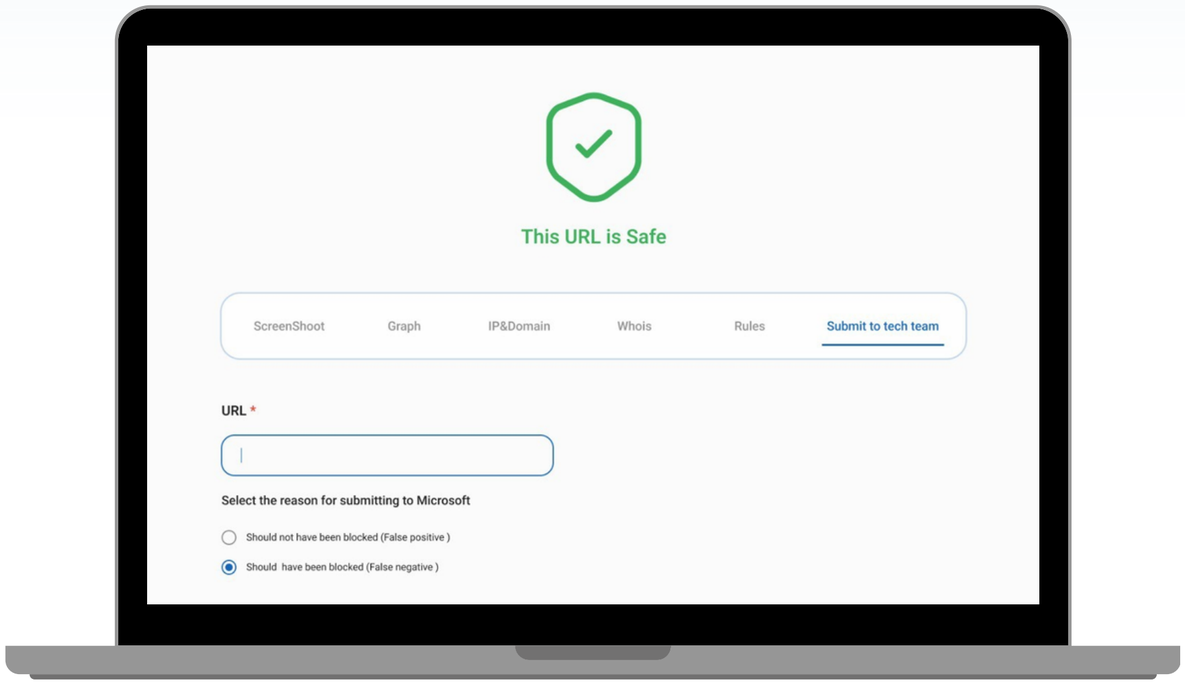
Utilizing clear icons to warn users about potential phishing threats

Website

Home Page



Testing Page



Mobile App (Flutter)

1

- Develop applications on various platforms
- App performance is close to the original
- Developer-friendly tools

2

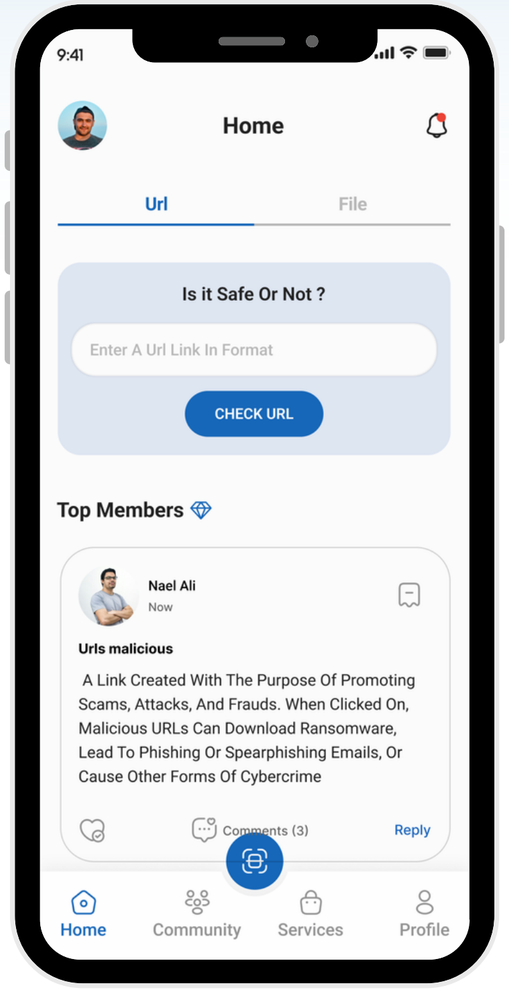
Flutter simplifies the process of creating consistent, attractive user interfaces for an app across the six platforms it supports.

3

The shark eye mobile app allows users to access our services anytime, anywhere, to test malicious URL

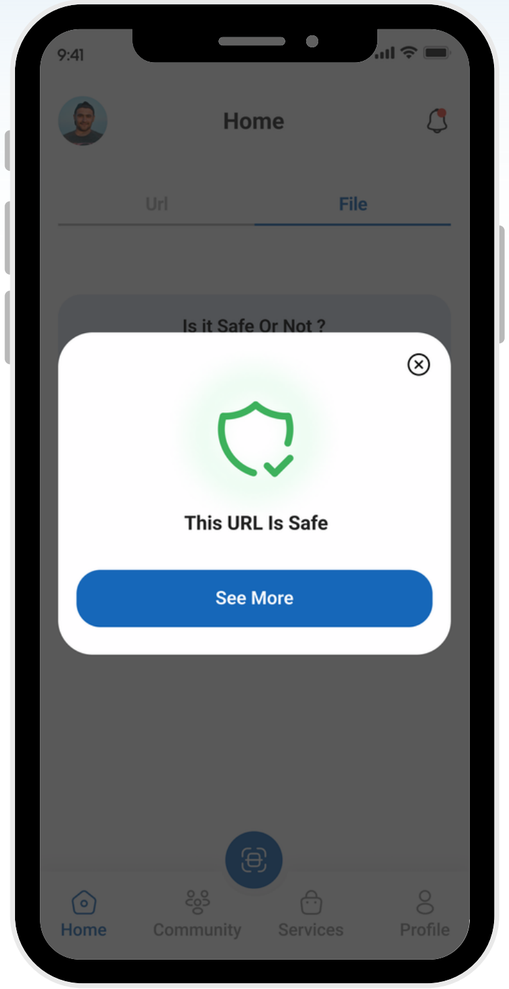
Mobile App

Home Page



Scanning Page

Result Page



Stay safe with SharkDefense

Thank you