UNIVERSIDAD PANAMERICANA

UNIVERSIDAD PANAMERICANA
Campus México

Class: Gestión de Análisis y Diseño de Comercialización(COM145)

Professor: Sarahí Aguilar González

Delivery date: 24 mayo, 2022

Cycle: 1222

Project Name: Identification of Malicious URL Patterns

Miembros del Equipo		
ID	**Nombre**	**Carrera**
0213358	Flores Peregrina, Ricardo Ariel	LTISC
0212614	Mayen Soto, Esteban	LTISC

Rúbricas				
ID	**2-social**		**7-knowledge**	
	D	**C**	**A**	**JI**

Abstract — The timely detection of malicious URLs would be of great value to society since many Internet users surf the Internet unconsciously and later find themselves paying for it. Therefore, through our Data Science model and tools, we seek to categorize what really distinguishes a harmless URL from one that could have a malicious purpose..

Keywords— link, URL, link, malware

Introduction

Research question

"Can the links we receive on a day-to-day basis with abnormal patterns be trusted?"

Every day, each individual person has thousands of interactions with the Internet in different ways, but the most common of them are through links that they can receive or simply access because they appear somewhere. This is where the problem that we want to solve with our project comes in, "how do we know what we can trust and what we can't?" For this reason we have decided, through the use of Data Science tools, to identify patterns in access links to Internet addresses (URLs) in order to identify particular characteristics in links that could lead to malicious sites or directly to the download of malware to the personal devices of Internet surfers.

Development

Using Data Science tools and applications, we plan to come up with a pattern classification model to get a closer look at what distinguishes a possibly malicious link from a benign or harmless link that fulfills its purpose of directing users to a page or content they are looking for. Within our research we came across the different types of malware that can be hosted within a URL. The number of users who are aware of the dangers that can exist on the Internet is really small, compared to the number of people who surf without the slightest care, who access any type of link and later find themselves in the situation that they have fallen into a type of malware that seeks to misuse their information or simply seeks to damage their personal computers.

By means of a database of URLs classified with their malware categories, we will make use of Data Science techniques and tools to perform analysis and be able to reach a conclusion where we can identify particular patterns that could be found in a malicious URL. As we well know, not everything on the Internet is always good, and that is why one of the easiest methods for spreading malware is through links, since it is enough for the user to click on the link he receives and it is more than enough for a malware to take possession of his device and with it, the personal and sensitive information it may contain.

```
!pip install -U -q kaggle
!mkdir -p ~/.kaggle

from google.colab import files
files.upload()

Choose Files No file chosen
Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

Saving kaggle.json to kaggle.json

{'kaggle.json': b'{"username":"estebanmayen","key":"77a752ce280a3a9ff128ef1fd8a84cfa"}'}

!cp kaggle.json ~/.kaggle/

!kaggle datasets download -d estebanmayen/bm-data

Downloading bm-data.zip to /content
 73% 16.0M/22.0M [00:00<00:00, 20.8MB/s]
100% 22.0M/22.0M [00:00<00:00, 30.8MB/s]

!chmod 600 /root/.kaggle/kaggle.json

!unzip bm-data.zip

Archive:  bm-data.zip
  inflating: Dataset_BM_1.csv
```

We started our analysis by taking as a basis a dataset containing approximately one million malicious links with their malware categories, as well as benign links to help us have a clear comparison of how to distinguish by means of patterns whether a link contains malware or is a harmless site. In order to have a much more efficient management and not having to rely on having the dataset locally, we chose to mount the dataset in a Kaggle, in order to simply call it through where it is hosted. Once we selected our dataset to work with, we previously had to clean it of null values and some extra values that we decided to exclude from the analysis.

```
data = pd.read_csv('/content/Dataset_BM_1.csv')
data.head()
```

	URL	TIPO
0	http://66.208.203.190:36841/malware.a	malware
1	http://58.255.129.35:53862/malware.a	malware
2	http://60.25.156.155:47183/malware.m	malware
3	http://192.72.17.236:35284/malware.a	malware
4	http://27.41.38.130:50541/malware.m	malware

```
data.info()
```

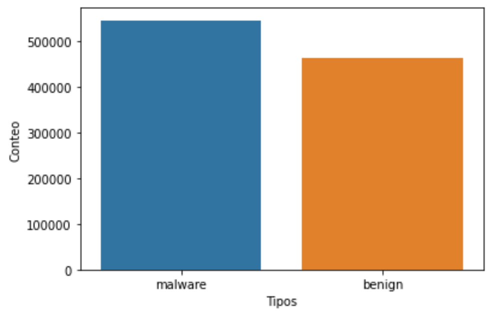
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1008870 entries, 0 to 1008869
Data columns (total 2 columns):
 #   Column  Non-Null Count    Dtype
---  ------  --------------    -----
 0   URL     1008870 non-null  object
 1   TIPO    1008870 non-null  object
dtypes: object(2)
memory usage: 15.4+ MB
```

Once we had our dataset ready to work with, we processed it in a Jupyter notebook and by means of data management libraries we were able to have clearer visualizations of the data we were going to work with. Through the visualizations we had a much clearer notion of the data that is contained in the dataset and with which we work to train our model. Our dataset is quite simple, because although it has a lot of data, it only has 2 columns per record, one being the URL and the other TYPE, which is the type of URL that is that record.

```
count = data.TIPO.value_counts()
count
```

```
malware    545389
benign     463481
Name: TIPO, dtype: int64
```

```
sns.barplot(x=count.index, y=count)
plt.xlabel('Tipos')
plt.ylabel('Conteo');
```



As mentioned above, the dataset contains both malware URLs as well as benign URLs. Having these two types of URLs in our project gives us a clearer view that our model will be able to identify patterns of a malicious link from a harmless one. For the data processing we did some research and there already existed certain libraries which help us to perform a better analysis of the data, some of them were urlparse, get_tld, is_tld.

```
rem = {"Categoria": {"benign": 0, "malware": 1}}
data['Categoria'] = data['TIPO']
data = data.replace(rem)
```

```
data['URL_LEN'] = data['URL'].apply(lambda x: len(str(x)))
```

```
def process_tld(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
```

```
data['DOMAIN'] = data['URL'].apply(lambda i: process_tld(i))
```

```
data.head(200)
```

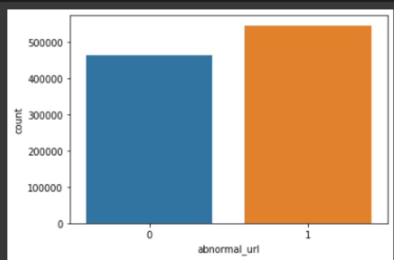
	URL	TIPO	Categoria	URL_LEN	DOMAIN
0	http://66.208.203.190:36841/malware.a	malware	1	37	None
1	http://58.255.129.35:53862/malware.a	malware	1	36	None
2	http://60.25.156.155:47183/malware.m	malware	1	36	None
3	http://192.72.17.236:35284/malware.a	malware	1	36	None
4	http://27.41.38.130:50541/malware.m	malware	1	35	None
...	...	...	...	...	...
195	http://164.163.25.165:41491/bin.sh	malware	1	34	None
196	http://37.120.222.60/mysite/catimages/243.malware	malware	1	49	None
197	http://37.120.222.60/mysite/catimages/244.malware	malware	1	49	None
198	http://37.120.222.60/mysite/catimages/242.malware	malware	1	49	None
199	http://37.120.222.60/mysite/catimages/246.malware	malware	1	49	None

200 rows × 5 columns

```
def abnormal_url(URL):
    hostname = urlparse(URL).hostname
    hostname = str(hostname)
    match = re.search(hostname, URL)
    if match:
        #hay una relacion
        return 1
    else:
        #no se encontre relacion
        return 0
```

```
[24] data['abnormal_url'] = data['URL'].apply(lambda i: abnormal_url(i))
```

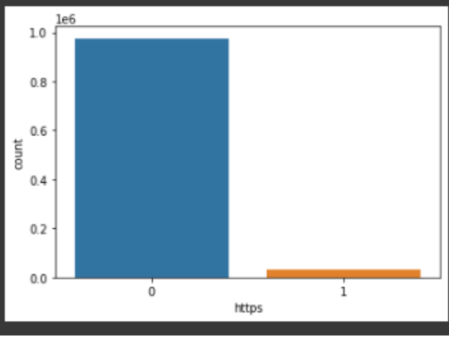
```
[25] sns.countplot(x='abnormal_url', data=data);
```



```
[26] def httpSecure(URL):
         htp = urlparse(URL).scheme
         match = str(htp)
         if match=='https':
             #hay una relacion
             return 1
         else:
             # no hay ninguna relacion
             return 0
```

```
[27] data['https'] = data['URL'].apply(lambda i: httpSecure(i))
```

```
[28] sns.countplot(x='https', data=data);
```




```
def digit_count(URL):
    digits = 0
    for i in URL:
        if i.isnumeric():
            digits = digits + 1
    return digits
```

```
[30] data['digits'] = data['URL'].apply(lambda i: digit_count(i))
```

```
[31] def letter_count(URL):
         letters = 0
         for i in URL:
             if i.isalpha():
                 letters = letters + 1
         return letters
```

```
[32] data['letters']= data['URL'].apply(lambda i: letter_count(i))
```

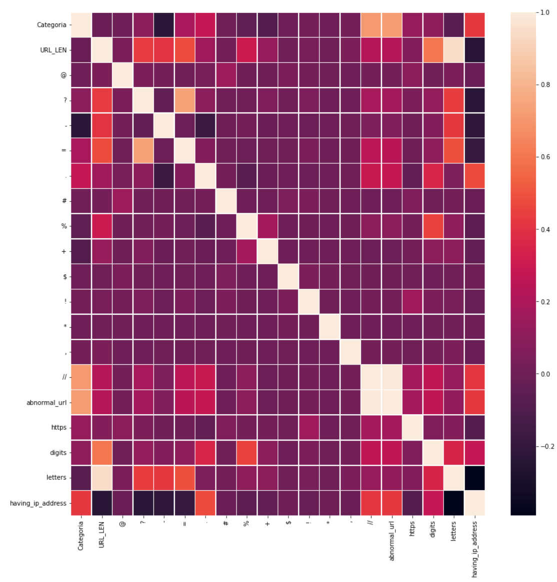
```
[33] def having_ip_address(URL):
         match = re.search(
             '(([01]?\\d\\d?|2[0-4]\\d|25[0-5])\\.([01]?\\d\\d?|2[0-4]\\d|25[0-5])\\.(
             '([01]?\\d\\d?|2[0-4]\\d|25[0-5])\\/)|'  #identificador de IPV4
             '(([01]?\\d\\d?|2[0-4]\\d|25[0-5])\\.([01]?\\d\\d?|2[0-4]\\d|25[0-5])\\.(
             '([01]?\\d\\d?|2[0-4]\\d|25[0-5])\\/)|'  # identificador de IPv4 con puer
             '((0x[0-9a-fA-F]{1,2})\\.(0x[0-9a-fA-F]{1,2})\\.(0x[0-9a-fA-F]{1,2})\\.(0
             '(?:[a-fA-F0-9]{1,4}:){7}[a-fA-F0-9]{1,4}|'
             '([0-9]+(?:\.[0-9]+){3}:[0-9]+)|'
             '((?:(?:\d|[01]?\d\d|2[0-4]\d|25[0-5])\.){3}(?:25[0-5]|2[0-4]\d|[01]?\d\d
         if match:
             return 1
         else:
             return 0
```

```
[34] data['having_ip_address'] = data['URL'].apply(lambda i: having_ip_address(i))
```

```
[35] data['having_ip_address'].value_counts()

     0    832348
     1    176522
     Name: having_ip_address, dtype: int64
```

Once the type of data contained in the dataset is known, we can begin to identify patterns that could make up a malicious URL. To identify components of a malicious URL, we use binary identification in order to gradually highlight patterns characteristic of these malicious links within the dataset. Also with the processing of the data, columns were added that could give us more details about each record within the dataset, such as length, if they have a domain, the distinctive number to categorize binary way if the link is benign or malicious, etc.. As well as a filter to detect that a link had an abnormal structure and that could be evident for our analysis. Also another factor that we wanted to highlight within the URLs was the fact if it came from a domain with the https tag, which is a good factor to highlight since usually this tag is only given to sites that have security certificates and likewise this field was added to our dataset to use it in the training of our model. And finally one of the last fields we added to the dataset was the factor of how many digits, letters and if they contained directly in the URL any IP address.



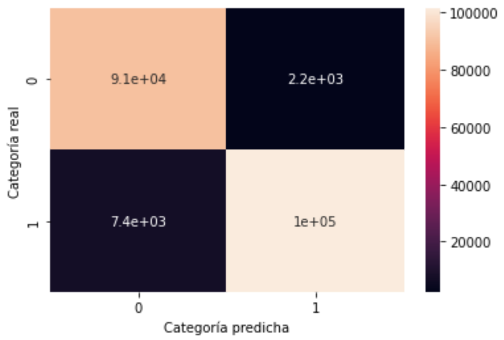
Once we have the identifiers in our dataset, we can put together our correlation matrix to see in more detail which values will help us in our classification.

For the implementation of the model we decided to choose "Decision Trees" since we are managing the project as a supervised model in which we tell it which URLs are benign or malignant.

In our case before working with the model we must define the dependent variable (category) and the independent variables

(URL, type, domain) after this we will divide the data sets for the training then we must create the SVM (Support Vector Machine) object that will be the Decision Tree Classifier, once with this we will adjust the model with the X and Y training, once with this we will make the prediction of the values of the independent variable, once with this we will make a confusion matrix with y_test and y_predict, once with this we will make the graph of the confusion matrix where we will be able to see the classification in 0 or 1, if the url that we are working is benign or malignant, with this we obtain the following:

```
array([[ 90554,   2230],
       [  7416, 101574]])
```



Here we can see the four options that arise:

1. The URL is benign and the model classified it as benign (+) . This would be a true positive or VP .
2. The URL is malignant and the model classified it as malignant (-) . This would be a true negative or VN.
3. The URL is malignant and the model classified it as benign (-) . This would be a type II error or a false negative or FN.
4. The URL is benign and the model classified it as malignant (+) . This is a type I error, or a false positive or FP.

To finalize the classification we must calculate the quality metrics which are Accuracy, Precision, Recall and F1-Score and we obtain the following:

```
Accuracy: 0.9521940388751772
Precisión: 0.9785172055026782
Recall: 0.9319570602807598
F1-score: 0.9546697745237178
```

- Accuracy is the ratio of correctly classified observations to all classified observations.
- Precision is the ratio of correctly classified positive observations to all positive classified observations.
- Recall is the ratio of correctly classified positive observations to all true positive observations.
- F1-score is the harmonic mean of precision and recall.

Conclusions

With the deep analysis of our project, we can conclude that our model worked successfully to determine and differentiate malicious URLs from harmless ones by means of the characteristics that can compose a malware URL. The tools and techniques used for Data Science are really useful but require some knowledge about them to be applied in their own way and with the purpose we want to give them. Later and by means of the defined characteristics, we could seek to implement our model in general systems such as those of messaging (email or communication platforms) so that any URL that is received in the person's device, is briefly compared and in case of being a malware suspicion, warn the user and that at his own risk access the link. Another factor that we do not rule out for a future improvement would be to compare our model against any other that may exist on this issue.

Bibliography

- **IBM. (2021, 12 marzo).** ***Supervised vs. Unsupervised Learning: What's the Difference?*** https://www.ibm.com/cloud/blog/supervised-vs-unsupervised-learning

- ***Malware & ransomware*. (2018, 5 enero). Australian Competition and Consumer Commission. https://www.scamwatch.gov.au/types-of-scams/threats-extortion/malware-ransomware#:%7E:text=Malware%20scams%20work%20by%20installing,details%20and%20commit%20fraudulent%20activities**

- **Security, S. (2021, 22 abril). *What Is a Malicious URL? (And How You Can Avoid Them)*. Savvy Security. https://cheapsslsecurity.com/blog/what-is-a-malicious-url/**