David Pekerow
Web Page Phishing Project

Description: Analyze web page URL phishing data to determine which fields in a URL suggest that the URL may be phishing.

Data download:
https://www.kaggle.com/datasets/danielfernandon/web-page-phishing-dataset?resource=download

Task:
Take the above CSV file and convert it into a relational database. The first table should be named *web_page_fishing* and contain the following fields:

1. unique_id
2. url_length
3. n_redirection
4. phishing

The second table should be named *phishing_dataset* and contain the following fields:

1. unique_id
2. n_dots
3. n_hyphens
4. n_underline
5. n_slash
6. n_questionmark
7. n_equal
8. n_at
9. n_and
10. n_exclamation
11. n_space
12. n_tilde
13. n_comma
14. n_plus
15. n_asterisk
16. n_hashtag
17. n_dollar
18. n_percent

What SQL code would you write to query all of the data contained in the provided dataset?

How I Constructed the Database

This process was relatively straightforward, involving just a few blocks of SQL code. The initial data was given in a CSV file containing 100,077 rows of data, which will comprise the database records. The CSV presumably represents a hypothetical IT department's analysis of its website traffic, with each row containing data about a single URL. Except for the column listing the URL's length, each column lists possible characters (e.g., '#', '$', '%') that may be present in a URL. A value of '1' denotes the presence of that character and '0' denotes its absence.

After downloading the file to my local machine, I divided the data up into 2 CSV files, each forming the basis for a separate database table. Then I imported the records into a postgresql database using pgAdmin and the following SQL commands:

```
CREATE DATABASE web_page_phishing;

CREATE TABLE web_page_fishing (
    id SERIAL PRIMARY KEY,
    url_length INTEGER,
    n_redirection INTEGER,
    phishing INTEGER
);

COPY web_page_fishing (url_length, n_redirection, phishing)
FROM 'C:\\temp\\web-page-phishing-table1.csv'
DELIMITER ','
CSV HEADER;

CREATE TABLE phishing_dataset (
      phishing_id SERIAL PRIMARY KEY,
      n_dots INTEGER,
      n_hyphens INTEGER,
      n_underline INTEGER,
      n_slash INTEGER,
      n_questionmark INTEGER,
      n_equal INTEGER,
      n_at INTEGER,
      n_and INTEGER,
      n_exclamation INTEGER,
      n_space INTEGER,
      n_tilde INTEGER,
      n_comma INTEGER,
      n_plus INTEGER,
      n_asterisk INTEGER,
      n_hashtag INTEGER,
      n_dollar INTEGER,
      n_percent INTEGER
);

COPY phishing_dataset (n_dots, n_hyphens, n_underline, n_slash,
n_questionmark, n_equal, n_at, n_and, n_exclamation, n_space, n_tilde,
n_comma, n_plus, n_asterisk, n_hashtag, n_dollar, n_percent)
FROM 'C:\\temp\\web-page-phishing-table2.csv'
DELIMITER ','
CSV HEADER;
```

Now we have an exact replica of the CSV data in two tables, **`web_page_fishing`** and **`phishing_dataset,`** each having a primary key **`id`** and **`phishing_id`**, respectively.

(We need to use differently-named primary keys in order to perform joins in SQL without issue.) We can now use SQL to query the data and analyze it.

Business Questions:

1. Is URL length a strong indicator of whether or not the URL is phishing? Why or why not? What metrics do you have to support your answer?

- The average URL length for all sites recorded is 39 characters, the max being over 4,000 characters long. (Nothing suspicious there.) Intuitively, 100 characters seems like a good benchmark for a long URL length.
- There are 36,362 URLs identified as phishing, or 36.3% of the total.
- Of these, 5,555 have a URL length >= 100, or 15.3% of all phishing sites. Let's call these "very long URLs."
- Of the remaining 63,715 non-phishing sites, just 472 have very long URLs, or less than 1%.
- This gives us 6,027 very long URLs in total, of which 92.1% are phishing.

Conclusion: Yes, a 92.1% positive rate shows that most very long URLs *are* indicative of phishing. However, very long URLs only identify 15.3% of all phishing sites. While a clear red flag, we need to do more work beyond URL length-counting to find the other 84.7% of phishing sites.

SQL queries used:

```
SELECT MIN(url_length), MAX(url_length), AVG(url_length) FROM
web_page_fishing;
4, 4165, 39.1
```

```
SELECT COUNT(*) FROM web_page_fishing AS COUNT
WHERE phishing = 1
36362
```

```
SELECT COUNT(*) FROM web_page_fishing AS COUNT
WHERE phishing = 1 AND url_length >= 100;
5555
```

```
SELECT COUNT(*) FROM web_page_fishing AS COUNT
WHERE phishing = 0 AND url_length >= 100;
472
```

2. Is the number of redirections a strong indicator of whether or not the URL is phishing? Why or why not? What metrics do you have to support your answer?

- For phishing URLs, the number of redirects is 7,858.
- For non-phishing URLs, the number of redirects is 19,798.

Conclusion: There is no correlation. With non-phishing URLs redirecting to other URLs more frequently than phishing URLs, there is no basis for using the number of redirects as a criterion for flagging URLs for phishing.

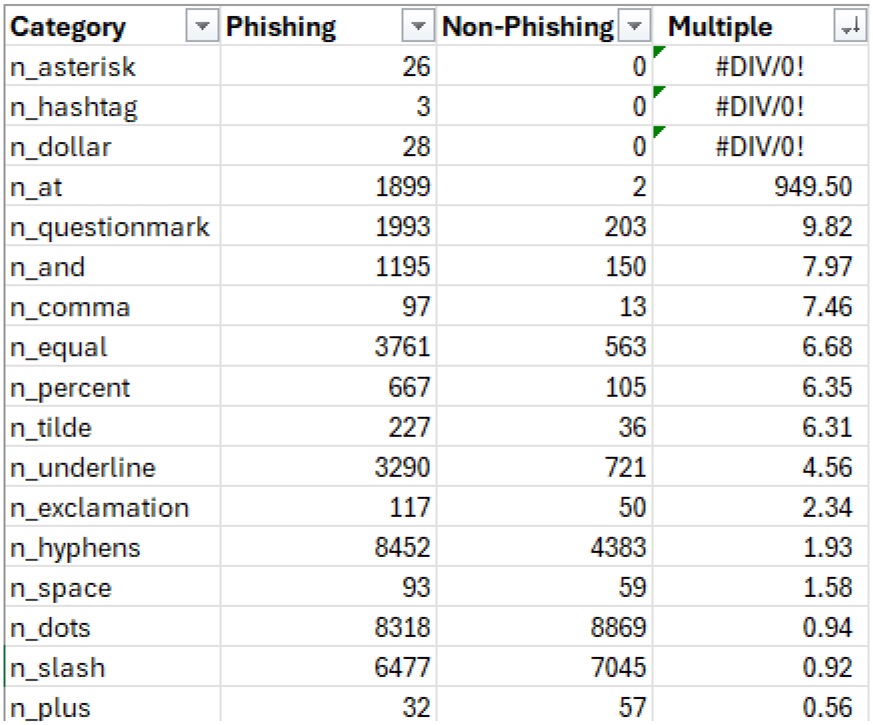
SQL queries used:

```
SELECT COUNT(*) FROM web_page_fishing AS count
WHERE phishing = 1 AND n_redirection = 1;
7858
```

```
SELECT COUNT(*) FROM web_page_fishing AS count
WHERE phishing = 0 AND n_redirection = 1;
19798
```

3. Which field(s) has/have the strongest correlation with the "phishing" field? Which field(s) has/have the weakest correlation with the "phishing" field?

To answer this question, I first counted the number of instances when the various URL attributes occur in both phishing and non-phishing sites:



Category	Phishing	Non-Phishing	Multiple
n_asterisk	26	0	#DIV/0!
n_hashtag	3	0	#DIV/0!
n_dollar	28	0	#DIV/0!
n_at	1899	2	949.50
n_questionmark	1993	203	9.82
n_and	1195	150	7.97
n_comma	97	13	7.46
n_equal	3761	563	6.68
n_percent	667	105	6.35
n_tilde	227	36	6.31
n_underline	3290	721	4.56
n_exclamation	117	50	2.34
n_hyphens	8452	4383	1.93
n_space	93	59	1.58
n_dots	8318	8869	0.94
n_slash	6477	7045	0.92
n_plus	32	57	0.56

Figure 1: What is this $#%? Special characters in phishing and non-phishing URLs.*

Having pulled the above figures from the database using SQL, I then exported them into a CSV file, sorting on the ratio of their appearance in phishing vs. non-phishing sites. This lets us see clearly how certain kinds of characters could be telltale for phishing URLs. The rightmost column indicates the ratio of phishing to non-phishing URLs where special characters appear. The findings are as follows:

- Asterisks, hashtags, and dollar signs *never* appear in any legitimate URL.
- Don't '@' me. The '@' character only appears just twice in legitimate URLs, but 1,899 times in phishing URLs.
- The humble hyphen appears nearly twice as frequently in phishing sites than non-phishing sites, and is also the character that appears *most* frequently in phishing URLs, with a 23.2% occurrence.
- Except for dots, slashes, and plus signs, all of the other special characters above have a significantly higher correlation with phishing. On the high end, question marks occur 9.82 times more frequently in phishing URLs than in non-phishing URLs, while on the low end empty spaces occur 1.58 times more frequently.
- Dots, slashes, and plus signs are characters that do not correlate with phishing, appearing slightly more frequently in non-phishing URLs than phishing URLs.

SQL queries used:

I used an INNER JOIN on each table's primary key to view all of the data across both tables. Also, using "CREATE VIEW" allows repeated operations on the joined tables so long as the query tool in pgadmin remains active; we'll call this view *combined_phishing*:

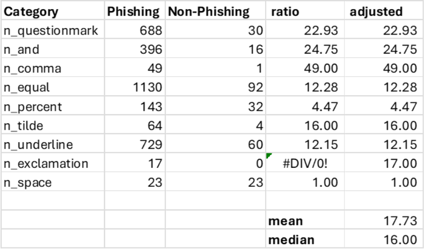
```
CREATE VIEW combined_phishing AS
SELECT *
FROM web_page_fishing
INNER JOIN phishing_dataset
ON web_page_fishing.id = phishing_dataset.phishing_id;
```

To count the number of occurrences of special characters in both the phishing and non-phishing records, I constructed the following complex SQL statement:

```
SELECT
    SUM(CASE WHEN n_dots = 1 THEN 1 ELSE 0 END) AS count_n_dots,
    SUM(CASE WHEN n_hyphens = 1 THEN 1 ELSE 0 END) AS count_n_hyphens,
    SUM(CASE WHEN n_underline = 1 THEN 1 ELSE 0 END) AS
count_n_underline,
    SUM(CASE WHEN n_slash = 1 THEN 1 ELSE 0 END) AS count_n_slash,
    SUM(CASE WHEN n_questionmark = 1 THEN 1 ELSE 0 END) AS
count_n_questionmark,
    SUM(CASE WHEN n_equal = 1 THEN 1 ELSE 0 END) AS count_n_equal,
    SUM(CASE WHEN n_at = 1 THEN 1 ELSE 0 END) AS count_n_at,
    SUM(CASE WHEN n_and = 1 THEN 1 ELSE 0 END) AS count_n_and,
    SUM(CASE WHEN n_exclamation = 1 THEN 1 ELSE 0 END) AS
count_n_exclamation,
    SUM(CASE WHEN n_space = 1 THEN 1 ELSE 0 END) AS count_n_space,
    SUM(CASE WHEN n_tilde = 1 THEN 1 ELSE 0 END) AS count_n_tilde,
    SUM(CASE WHEN n_comma = 1 THEN 1 ELSE 0 END) AS count_n_comma,
    SUM(CASE WHEN n_plus = 1 THEN 1 ELSE 0 END) AS count_n_plus,
    SUM(CASE WHEN n_asterisk = 1 THEN 1 ELSE 0 END) AS
count_n_asterisk,
    SUM(CASE WHEN n_hashtag = 1 THEN 1 ELSE 0 END) AS count_n_hashtag,
    SUM(CASE WHEN n_dollar = 1 THEN 1 ELSE 0 END) AS count_n_dollar,
    SUM(CASE WHEN n_percent = 1 THEN 1 ELSE 0 END) AS count_n_percent
FROM combined_phishing
WHERE phishing = 1;
```

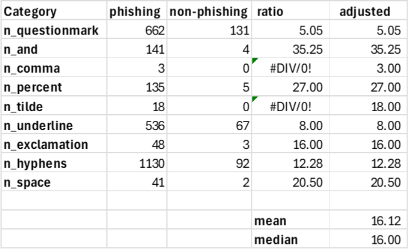
We are looking for counts of records according to the delimited criteria. For each character count, we sum only if there is a ‘1’ in that row, further culling only those records that have “1” in the phishing column. Once we obtain the counts for phishing sites and export them to CSV, we then change the final line to `‘WHERE phishing = 0’` and run the code a second time to get the counts for non-phishing sites, which we export to a second CSV file. The above screenshot reflects a manual paste of the data from this second CSV file to column C in the first file.

After completing this analysis, I looked for combinations of characters, focusing on the 3 that occur most frequently: hyphens, equal signs, and underlines.



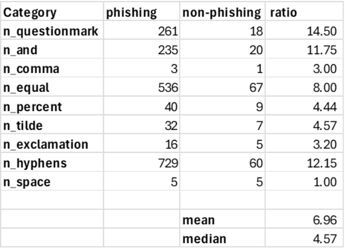
Category	Phishing	Non-Phishing	ratio	adjusted
n_questionmark	688	30	22.93	22.93
n_and	396	16	24.75	24.75
n_comma	49	1	49.00	49.00
n_equal	1130	92	12.28	12.28
n_percent	143	32	4.47	4.47
n_tilde	64	4	16.00	16.00
n_underline	729	60	12.15	12.15
n_exclamation	17	0	#DIV/0!	17.00
n_space	23	23	1.00	1.00
			mean	17.73
			median	16.00

Figure 2: Hyphens plus other chars



Category	phishing	non-phishing	ratio	adjusted
n_questionmark	662	131	5.05	5.05
n_and	141	4	35.25	35.25
n_comma	3	0	#DIV/0!	3.00
n_percent	135	5	27.00	27.00
n_tilde	18	0	#DIV/0!	18.00
n_underline	536	67	8.00	8.00
n_exclamation	48	3	16.00	16.00
n_hyphens	1130	92	12.28	12.28
n_space	41	2	20.50	20.50
			mean	16.12
			median	16.00

Figure 3: Equals plus other chars



Category	phishing	non-phishing	ratio
n_questionmark	261	18	14.50
n_and	235	20	11.75
n_comma	3	1	3.00
n_equal	536	67	8.00
n_percent	40	9	4.44
n_tilde	32	7	4.57
n_exclamation	16	5	3.20
n_hyphens	729	60	12.15
n_space	5	5	1.00
		mean	6.96
		median	4.57

Figure 4: Underline plus other chars

When these characters appeared alongside another special character, the likelihood of that URL being associated with a phishing site increased by several orders of magnitude in nearly all cases.

SQL queries used

This is a sample for checking to see how frequently hyphens appeared together with question marks in both phishing and non-phishing URLs:

```
SELECT
    (SELECT COUNT(*)
     FROM combined_phishing
      WHERE phishing = 1 AND n_questionmark = 1 AND n_hyphens = 1) AS
num_phishing,
      (SELECT COUNT(*)
     FROM combined_phishing
      WHERE phishing = 0 AND n_questionmark = 1 AND n_hyphens = 1) AS
num_non_phishing;
```

4. Based on your analysis, what advice would you give to others for deciphering whether or not a URL is phishing?

- A very long URL is a good first red flag, but its presence doesn't automatically rule a site in or out as engaging in phishing activity. It will also catch only a fraction of phishing sites. These sites should be flagged for review but also should be examined to rule them *out* as phishing sites, as close to 10% in this sample proved to be non-phishing.
- Any site that has a URL containing an asterisk, dollar sign, hashtag, or '@' should be flagged and immediately blocked. Since the likelihood of such sites being legitimate is near-zero, these should be the lowest priority for IT review, only to be done if there's nothing else to do.
- Any site that has *one* special character other than a slash, dot, or plus sign should be immediately flagged and blocked until further review, allowable only after phishing has been ruled out.
- Websites with *more than one* special character in their URL should be the last to be reviewed and unblocked as they are much more likely to be phishing than sites with just one special character. The legitimate sites in this category will be blocked for longer, with IT catching up as time permits. This is the price to be paid by these sites for not following the conventions of legitimate URLs.
- Important sites known to be non-phishing should be added to the company firewall "allow" list to prevent preemptive blocking, and this list should be reviewed frequently to ensure that the company's operations aren't negatively impacted.