```
In [14]: import pandas as pd

         df = pd.read_csv('dataset_link_phishing.csv', usecols=['url', 'status'])
         print(df.head())

         print(df.info())
         print(df['status'].value_counts())
```

```
                                                 url      status
0     http://www.progarchives.com/album.asp?id=61737    phishing
1  http://signin.eday.co.uk.ws.edayisapi.dllsign....    phishing
2  http://www.avevaconstruction.com/blesstool/ima...    phishing
3                               http://www.jp519.com/  legitimate
4                      https://www.velocidrone.com/  legitimate
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 19431 entries, 0 to 19430
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   url     19431 non-null  object
 1   status  19431 non-null  object
dtypes: object(2)
memory usage: 303.7+ KB
None
status
legitimate    9716
phishing      9715
Name: count, dtype: int64
```

```
In [15]: import re
         import numpy as np

         # url len
         df['url_length'] = df['url'].apply(len)

         # Домен (com, ru, org, etc)
         df['tld'] = df['url'].str.extract(
             r'(?:https?://)?(?:www\.)?[^/\.]+\.([^/\.]+)(?=/|$|\?)'
         )
         df['tld'].fillna('undefined', inplace=True)

         df['extension'] = df['url'].str.extract(r'(\.[a-zA-Z]{2,4})(?:\?|$)')  # рас
         df['extension'].fillna('none', inplace=True)


         df['tls'] = np.where(
             df['url'].str.startswith('https://'), 1,  # Если HTTPS
             np.where(
                 df['url'].str.startswith('http://'), 0,  # Если HTTP
                 'undefined'
             )
         )

         spec_chars = r'[?/.&=%-_+@]'  # символы
         df['special_chars_count'] = df['url'].apply(lambda x: len(re.findall(spec_ch
```

```python
print(df.head())

print(df.info())
```

```
                                                 url      status  url_length  \
0     http://www.progarchives.com/album.asp?id=61737    phishing          46   
1  http://signin.eday.co.uk.ws.edayisapi.dllsign....    phishing         128   
2  http://www.avevaconstruction.com/blesstool/ima...    phishing          52   
3                              http://www.jp519.com/  legitimate          21   
4                         https://www.velocidrone.com/  legitimate          28   

   tld extension tls  special_chars_count  
0  com      .asp   0                   14  
1  com      none   0                   21  
2  com      .htm   0                    8  
3  com      none   0                    9  
4  com      none   1                    6  
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 19431 entries, 0 to 19430
Data columns (total 7 columns):
 #   Column               Non-Null Count  Dtype 
---  ------               --------------  ----- 
 0   url                  19431 non-null  object
 1   status               19431 non-null  object
 2   url_length           19431 non-null  int64 
 3   tld                  19431 non-null  object
 4   extension            19431 non-null  object
 5   tls                  19431 non-null  object
 6   special_chars_count  19431 non-null  int64 
dtypes: int64(2), object(5)
memory usage: 1.0+ MB
None
```

```
/tmp/ipykernel_7961/2095398962.py:11: FutureWarning: A value is trying to be
set on a copy of a DataFrame or Series through chained assignment using an i
nplace method.
The behavior will change in pandas 3.0. This inplace method will never work
because the intermediate object on which we are setting values always behave
s as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'd
f.method({col: value}, inplace=True)' or df[col] = df[col].method(value) ins
tead, to perform the operation inplace on the original object.


  df['tld'].fillna('undefined', inplace=True)
/tmp/ipykernel_7961/2095398962.py:14: FutureWarning: A value is trying to be
set on a copy of a DataFrame or Series through chained assignment using an i
nplace method.
The behavior will change in pandas 3.0. This inplace method will never work
because the intermediate object on which we are setting values always behave
s as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'd
f.method({col: value}, inplace=True)' or df[col] = df[col].method(value) ins
tead, to perform the operation inplace on the original object.


  df['extension'].fillna('none', inplace=True)
```

In [16]:

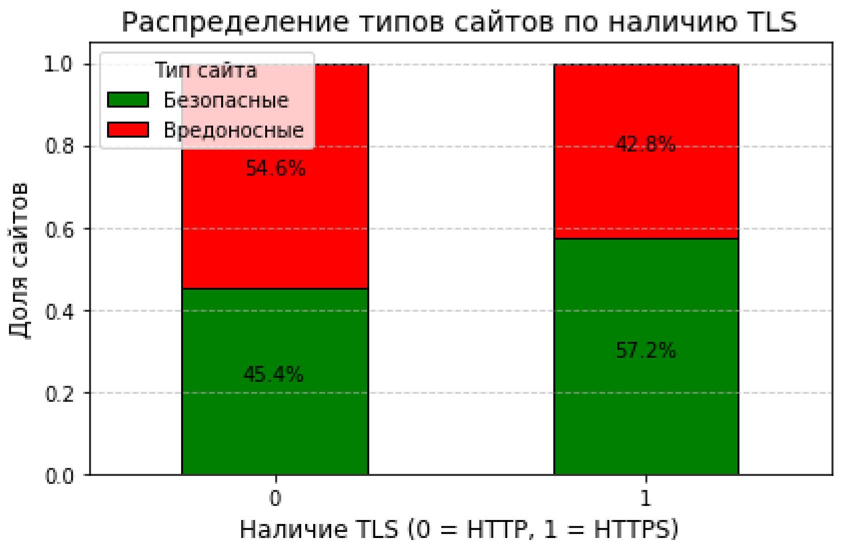
```python
from sklearn.preprocessing import LabelEncoder, StandardScaler, OneHotEncode
from sklearn.compose import ColumnTransformer

le = LabelEncoder()
df['status_encoded'] = le.fit_transform(df['status'])

# One-Hot Encoding для категориальных признаков
#df = pd.get_dummies(df, columns=['extension'], prefix=['ext'])

# масштабирование
numeric_features = ['special_chars_count', 'url_length']
scaler = StandardScaler()
df[numeric_features] = scaler.fit_transform(df[numeric_features])

# 6. Итоговый датасет
print("\nОбработанный датасет:")
print(df.head())
print("\nКолонки после обработки:")
print(df.columns.tolist())
```

```
Обработанный датасет:
                                                   url      status  url_length  \
0     http://www.progarchives.com/album.asp?id=61737    phishing   -0.269167
1  http://signin.eday.co.uk.ws.edayisapi.dllsign....    phishing    1.189954
2  http://www.avevaconstruction.com/blesstool/ima...    phishing   -0.162402
3                              http://www.jp519.com/  legitimate   -0.714021
4                       https://www.velocidrone.com/  legitimate   -0.589462

   tld extension tls  special_chars_count  status_encoded
0  com      .asp   0            -0.137333               1
1  com      none   0             0.101217               1
2  com      .htm   0            -0.341804               1
3  com      none   0            -0.307726               0
4  com      none   1            -0.409961               0

Колонки после обработки:
['url', 'status', 'url_length', 'tld', 'extension', 'tls', 'special_chars_co
unt', 'status_encoded']
```

In [17]:

```python
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd

df['is_malicious'] = df['status'].isin(['phishing', 'defacement', 'malware']
cross_tab = pd.crosstab(df['tls'], df['is_malicious'], normalize='index')

plt.figure(figsize=(10, 6))
ax = cross_tab.plot(kind='bar', stacked=True, color=['green', 'red'], edgecc

plt.title('Распределение типов сайтов по наличию TLS', fontsize=14)
plt.xlabel('Наличие TLS (0 = HTTP, 1 = HTTPS)', fontsize=12)
plt.ylabel('Доля сайтов', fontsize=12)
plt.xticks(rotation=0)
plt.legend(['Безопасные', 'Вредоносные'], title='Тип сайта')

for p in ax.patches:
    width, height = p.get_width(), p.get_height()
    x, y = p.get_xy()
    ax.annotate(f'{height:.1%}', (x + width/2, y + height/2), ha='center', f

plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```

```
<Figure size 720x432 with 0 Axes>
```



```
In [18]: df.to_csv('filtered_dataset.csv', index=False)
```

```
In [ ]:
```

This notebook was converted with convert.ploomber.io