In [1]:
```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns
```

In [2]:
```
# initial dataset provided by malwaredatascience.com
df = pd.read_csv('malware_data.csv')
```

Out[2]:

	positives	size	type	fs_bucket
0	45	251592	trojan	2017-01-05 00:00:00
1	32	227048	trojan	2016-06-30 00:00:00
2	53	682593	worm	2016-07-30 00:00:00
3	39	774568	trojan	2016-06-29 00:00:00
4	29	571904	trojan	2016-12-24 00:00:00
5	31	582352	trojan	2016-09-23 00:00:00
6	50	2031661	worm	2017-01-04 00:00:00
7	40	2113536	ransomware	2016-09-02 00:00:00
8	20	968216	trojan	2016-10-04 00:00:00
9	40	5260000	trojan	2016-12-29 00:00:00

In [3]:

Out[3]:

	positives	size
count	37511.000000	3.751100e+04
mean	39.446536	1.300639e+06
std	15.039759	3.006031e+06
min	3.000000	3.370000e+02
25%	32.000000	1.653960e+05
50%	45.000000	4.828160e+05
75%	51.000000	1.290056e+06
max	57.000000	1.294244e+08

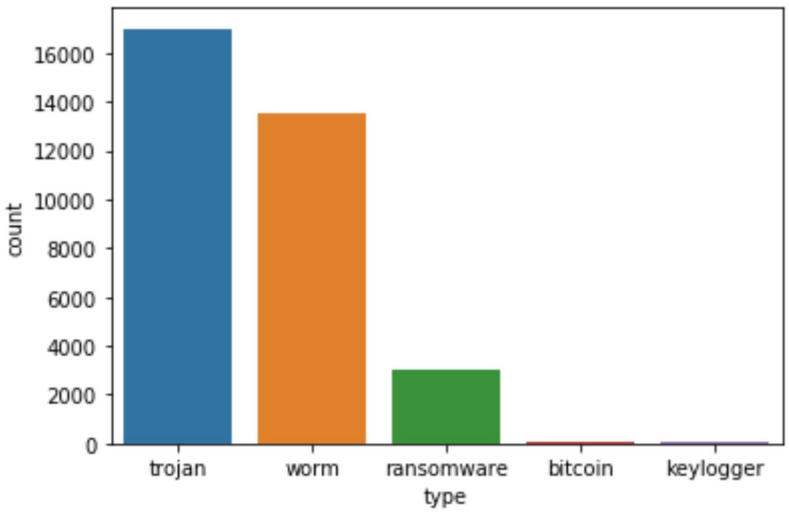
In [4]:

Out[4]:
```
Index(['positives', 'size', 'type', 'fs_bucket'], dtype='object')
```

In [5]:
```
# isolating values in type column
mal_types = []
for category in df['type'].unique():
    mal_types.append(category)
```

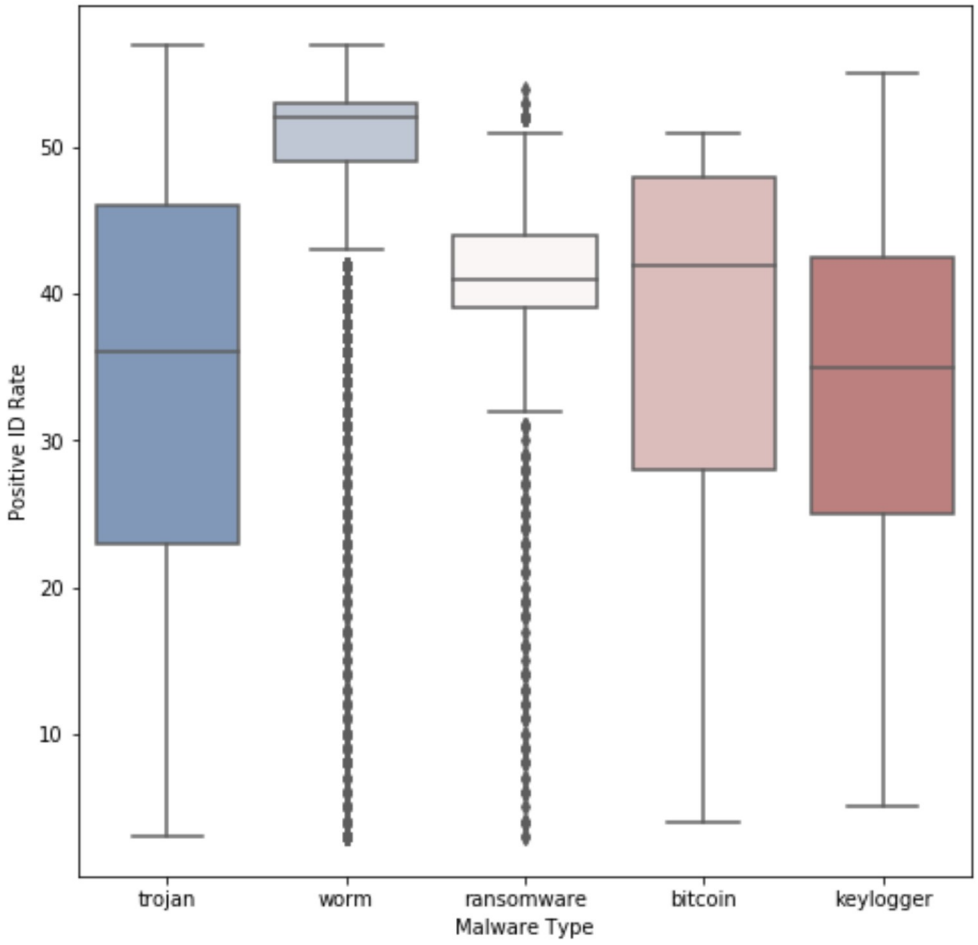
```
['trojan', 'worm', 'ransomware', nan, 'bitcoin', 'keylogger']
```

In [6]:
```
# Number identified by type
sns.countplot(x='type', data=df)
```



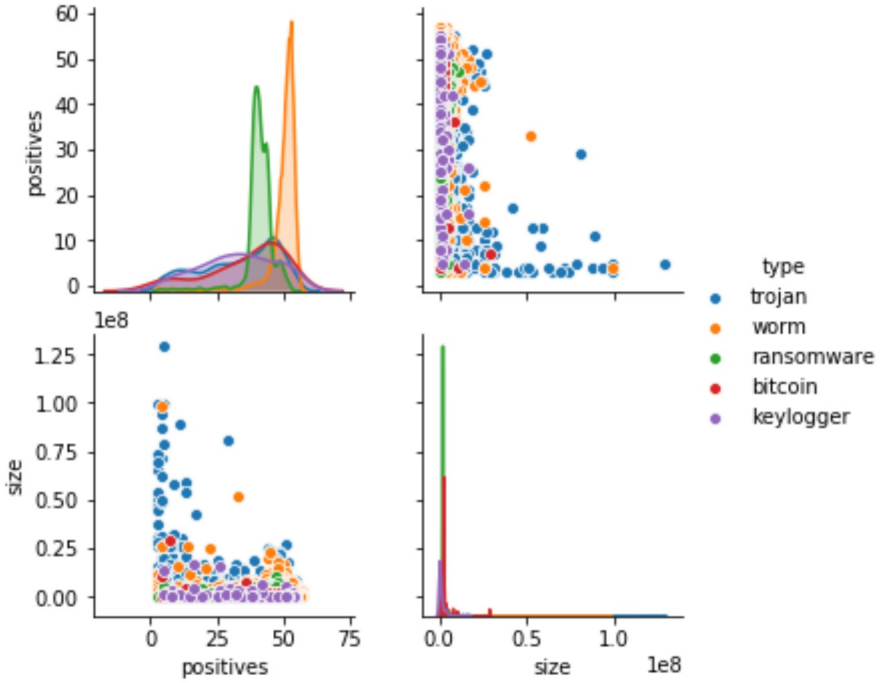
In [7]:
```
fig, ax = plt.subplots(figsize=(8,8))
sns.boxplot(x='type', y='positives', data=df, palette="vlag")
```

Out[7]:
```
[Text(0, 0.5, 'Positive ID Rate'), Text(0.5, 0, 'Malware Type')]
```



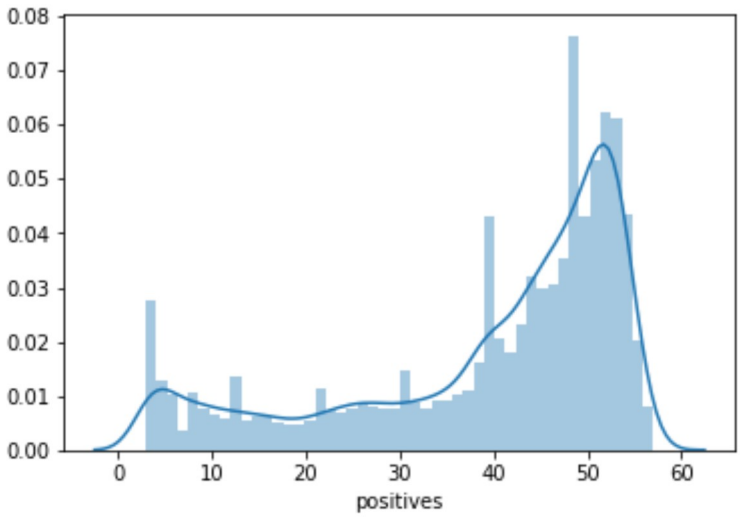
In [8]:
```
# Positive Rates by Size According to Malware Type
```

Out[8]: <seaborn.axisgrid.PairGrid at 0x27fec183c50>



In [9]:
```
# Detection Distribution each of 57 AV's tested
```

Out[9]: <matplotlib.axes._subplots.AxesSubplot at 0x27fec4c2438>



In [13]:
```
df2 = pd.pivot_table(df, values='positives', index=['fs_bucket'], colu
df2 = df2.dropna()
```

Out[13]:

type	bitcoin	keylogger	ransomware	trojan	worm	All
fs_bucket						
2016-06-13 00:00:00	37.000000	51.000000	40.361702	36.258427	50.598726	44.545455
2016-08-18 00:00:00	9.000000	36.000000	38.200000	33.640000	48.885246	40.081081
2016-08-23 00:00:00	28.000000	29.000000	39.400000	32.654545	48.055556	38.519231
2016-09-03 00:00:00	29.000000	55.000000	40.200000	32.386667	51.039604	42.854922
2016-10-04 00:00:00	45.000000	29.000000	41.875000	36.661017	49.183673	42.220339
2016-10-11 00:00:00	36.000000	25.000000	39.333333	27.523077	49.553191	36.968992
2016-10-14 00:00:00	46.000000	21.000000	39.888889	31.879518	50.784810	40.923497
2016-10-16 00:00:00	51.000000	29.000000	42.041667	37.853333	50.250000	43.768362
2016-11-16 00:00:00	47.000000	38.000000	41.600000	26.733333	51.484848	35.960000
2016-11-18 00:00:00	38.000000	38.000000	42.285714	31.900000	47.774194	37.466667
2016-11-30 00:00:00	46.000000	16.000000	43.214286	35.634921	48.939394	42.358621
2016-12-22 00:00:00	42.000000	53.000000	32.142857	27.971154	48.489362	34.425000
2017-01-06 00:00:00	49.000000	51.000000	43.684211	36.788235	51.555556	43.224852
All	35.857143	32.791209	40.708333	33.438225	49.908579	40.699866

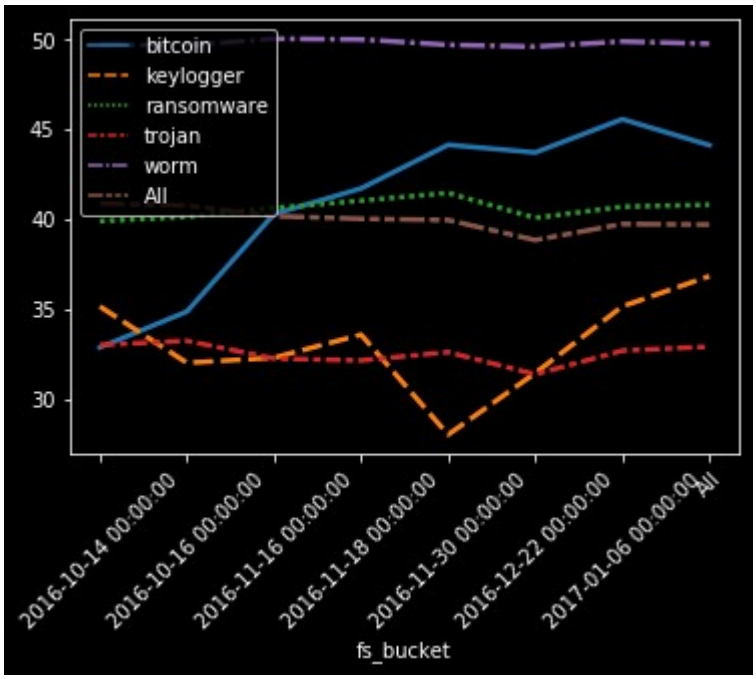
In [14]:

In [15]:
```
plt.style.use('dark_background')

sns.lineplot(data=df2, palette="tab10", linewidth=2.5)

plt.legend(loc='upper left')
```

Out[15]: ([0, 1, 2, 3, 4, 5, 6, 7], <a list of 8 Text xticklabel objects>)



In [16]:
```
# new column to parse values from range buckests
df['fs_date'] = [dateutil.parser.parse(x) for x in df['fs_bucket']]

# isolate types
trojans = df[df['type'] == 'trojan']
worms = df[df['type'] == 'worm']
```

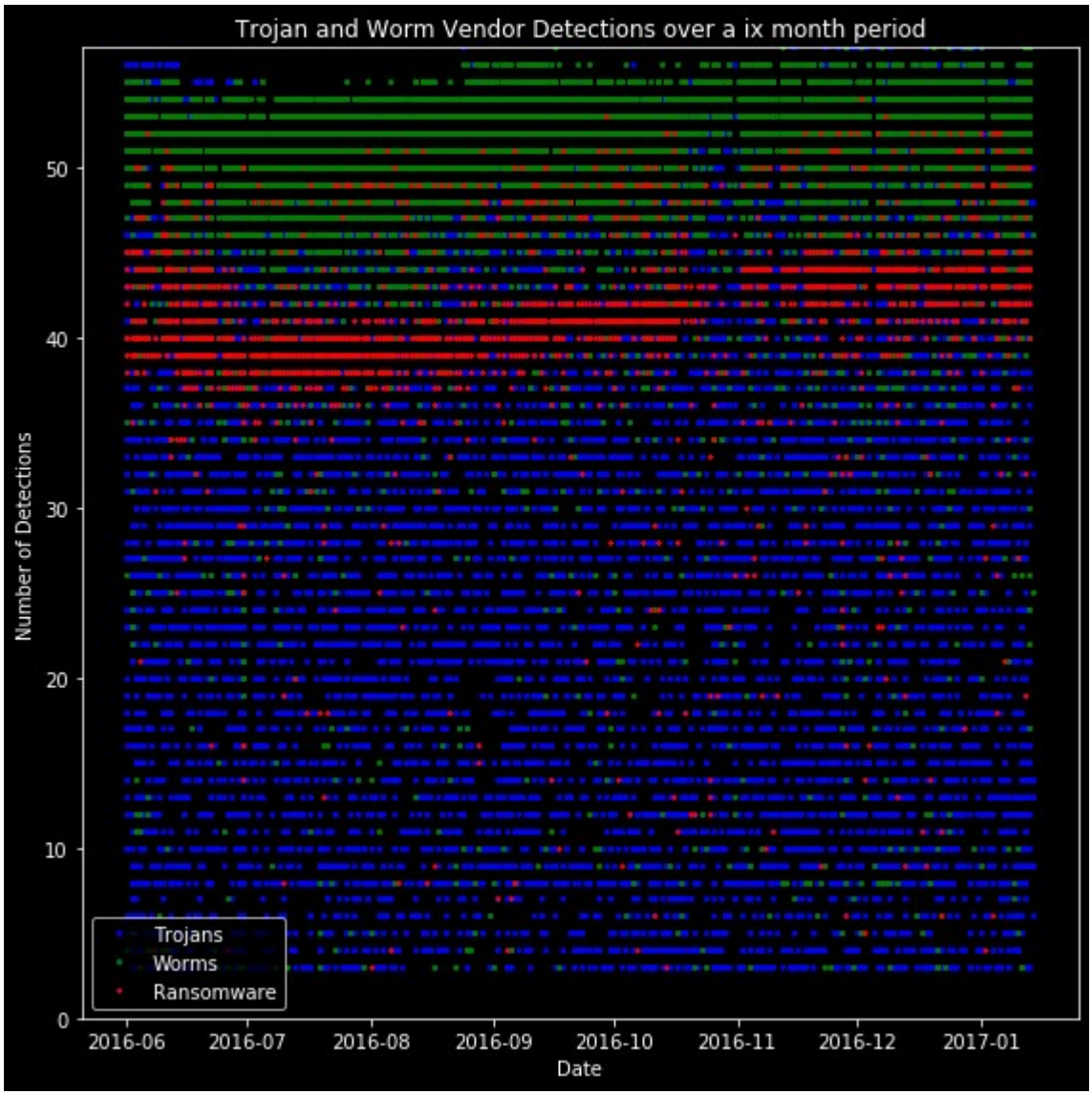
In [20]:
```
fig, ax = plt.subplots(figsize=(8,8))

plt.plot(trojans['fs_date'], trojans['positives'], 'bo', label='Trojar

plt.plot(worms['fs_date'], worms['positives'], 'go', label='Worms', ma

plt.plot(ransomware['fs_date'], ransomware['positives'], 'r*', label='

plt.legend()
plt.xlabel('Date')
plt.ylabel('Number of Detections')
plt.ylim([0,57])
plt.title('Trojan and Worm Vendor Detections over a ix month period')
plt.tight_layout()
```



In [ ]: