

Next Version of the Bazar Loader DGA

Archived: 2026-04-05 18:50:50 UTC

Last week, a new version of the Bazar Loader Domain Generation Algorithm (DGA) appeared. I already analyzed two [previous versions](#), so I'm keeping this post short.

The DGA still uses the eponymous *.bazar* top level domain, but the second level domains are shorter with 8 characters instead of 12 for the previous versions:

```
liybelac.bazar  
izryudew.bazar  
biymudqe.bazar  
fuicibem.bazar  
biykonem.bazar  
aqtielew.bazar  
yptaonem.bazar  
exyxtoca.bazar  
iqfisoew.bazar  
aguponew.bazar  
exogelqe.bazar  
exybonyw.bazar  
etymonac.bazar
```

I analysed the following sample without much obfuscation. There are many other samples that have additional reverse engineering counter measures such as junk code, but a quick comparison revealed no functional differences.

MD5

c6502d4dd27a434167686bfa4d183e89

SHA1

bddbceefe4185693ef9015d0a535eb7e034b9ec3

SHA256

35683ac5bbcc63eb33d552878d02ff44582161d1ea1ff969b14ea326083ea780

Size

336 KB (344576 Bytes)

Compile Timestamp

2020-12-10 13:05:18 UTC

Links

[MalwareBazaar](#), [Malpedia](#), [Cape](#), [VirusTotal](#)

Filenames

1ld.3.v1.exe, 35683ac5bbcc63eb33d552878d02ff44582161d1ea1ff969b14ea326083ea780 (VirusTotal)

Detections

Virustotal: 8/72 as of 2020-12-11 02:58:32 - Win64/Bazar.Y (ESET-NOD32), Backdoor.Win32.Bazdor.co (Kaspersky), Trojan.Win64.BAZALoader.SMYAAJ-A (TrendMicro), Trojan.Win64.BAZALoader.SMYAAJ-A (TrendMicro-HouseCall)

Unpacking the sample leads to this:

MD5

e44cfd6ecc1ea0015c28a75964d19799

SHA1

cb294c79b5d48840382a06c4021bc2772fdbcf63

SHA256

52e72513fe2a38707aa63fbc52dabd7c7d2c5809ed7e27f384315375426f57bf

Size

96 KB (98816 Bytes)

Compile Timestamp

2020-12-09 10:16:56 UTC

Links

[MalwareBazaar](#), [Malpedia](#), [Cape](#), [VirusTotal](#)

Filenames

content.28641.20903.13470.9122.7127 (VirusTotal)

Detections

Virustotal: 4/75 as of 2020-12-15 21:30:37

Reverse Engineering

Apart from the common dynamic loading of Windows API functions and encrypted strings, Bazar Loader relies on *arithmetic substitution via identities* to obfuscate the code. The following relationship is particularly often used:

$$a \oplus b = (\sim a \cdot b) + (a \cdot \sim b)$$

The same obfuscation is also used by [Zloader](#). It makes the code very hard to read. Here is a small snippet from the DGA:

Hex Ray's decompiler also produces really messy code because the arithmetic identities are not simplified:

The DGA uses the current month and year as the seed. The seed is stored as a string, and its four ASCII characters are the basis for picking four character pairs. These four pairs are joined to form the 8 second level characters.

The list of character pairs is generated by calculating the cartesian product of the consonants "bcdfghklmnpqrstvwxyz" and vowels (with y) "aeiouy". The product is calculated both ways, leading to $19 \cdot 6 \cdot 2$ character pairs. These pairs are then concatenated into a large string of 456 characters by using a hardcoded sequence of random numbers:

```
qeewcaacywemomedekwyuhidontoibeludsocuexvuuffyliagydhuiuctuiqow  
agypetehfubitiaziceblaogolrykosuptaymodisahfiybyxcoleafkudarapu
```

```
qoawyluxqagenanyoxcygyqugiutlyvegahepovyigqyqibaeqynyfkiobpeepby  
xacyivusocaripfyoftesaysozureginalifkazaadytwubzuvoothymivazyyz  
hoevmeburedeviiravygkemywaerdonoyryqloamosewesuvfopiriboikuz  
orruzemuulimyhceukoqiwfexuefgoycwiokitnuneroxepyanbekyixxiuqsias
```

The string is then encrypted using a random xor key of the same length.

Apart from the date-based seed, the DGA also uses a standard linear congruential generator (LCG) to pick the four character pairs. The LCG is seeded with the current processor tick count and thus unpredictable. For the first two character pairs, the random number is taken mod 19, and for the remaining two pairs mod 6. These numbers correspond to the length of the consonants and vowels array, but make no sense in this context. Because the random numbers are unpredictable, any combination of the $19 \cdot 19 \cdot 6 \cdot 6 = 12996$ character pairs could be picked. Bazar Loader generates 10'000 domains per run, but does not guarantee they are unique. On average, 6975 unique domains are expected:

$$E = 12996(1 - (\frac{12996 - 1}{12966})^{10000}) = 6975$$

Even with the short waiting time between resolving domains, the malware will need to run a long time to get through the list of domains.

Reimplementation in Python

The following Python code shows how the domains are generated:

```
from itertools import product
from datetime import datetime
import argparse
from collections import namedtuple

Param = namedtuple('Param', 'mul mod idx')
pool = (
    "qewcaacywemomedekwyuhidontoibeludsocuexvuuflyliaqydhuzuctuiqow"
    "agypetehfubitiaziceblaogolrykosuptaymodisahfiybyxcoleafkudarapu"
    "qoawyluxqagenanyoxcygyqugiutlyvegahepovyigqyqibaeqynyfkiobpeepby"
    "xacyivusocaripfyoftesaysozureginalifkazaadytwubzuvoothymivazyyz"
    "hoevmeburedeviiravygkemywaerdonoyryqloamosewesuvfopiriboikuz"
    "orruzemuulimyhceukoqiwfexuefgoycwiokitnuneroxepyanbekyixxiuqsias"
    "xoapaxmaohezwoildifaluzihpanizoecxyopguakdudyovhaumunuwsusyenko"
    "atugabiv"
)

def dga(date):
    seed = date.strftime("%m%Y")
    params = [
        Param(19, 19, 0),
```

```
    Param(19, 19, 1),
    Param(6, 6, 4),
    Param(6, 6, 5)
]
ranges = []
for p in params:
    s = int(seed[p.idx])
    lower = p.mul*s
    upper = lower + p.mod
    ranges.append(list(range(lower, upper)))

for indices in product(*ranges):
    domain = ""
    for index in indices:
        domain += pool[index*2:index*2 + 2]
    domain += ".bazar"
    yield domain

if __name__ == "__main__":
    parser = argparse.ArgumentParser()
    parser.add_argument(
        "-d", "--date", help="date used for seeding, e.g., 2020-06-28",
        default=datetime.now().strftime('%Y-%m-%d'))
    args = parser.parse_args()
    d = datetime.strptime(args.date, "%Y-%m-%d")
    for domain in dga(d):
        print(domain)
```

Here are all the domains for [December 2020](#), [January 2021](#), [February 2021](#), and [March 2021](#).

Characteristics

The following table summarizes the properties of the new Bazar Loader DGA.

property	value
type	TDD (time-dependent-deterministic)
generation scheme	arithmetic
seed	current date
domain change frequency	every month
unique domains per month	12996
sequence	random selection, might pick domains multiple times

property	value
wait time between domains	10 seconds
top level domain	<code>.bazar</code>
second level characters	a-z, without j
regex	<code>[a-ik-z]{8}\.bazar</code>
second level domain length	8

Source: <https://johannesbader.ch/blog/next-version-of-the-bazarloader-dga/>