Yet Another Bazar Loader DGA

johannesbader.ch/blog/yet-another-bazarloader-dga/

Bazar Loader decided to change its perfectly fine domain generation algorithm (DGA) once again. The change in the algorithm is very minor, but it yields more domain names.

Sample

I looked at this sample:

MD5
9ad20d0e6da3cf135a93bf162a0a8cfb

SHA1
a97893ab95f794cabc261483423f942f552926d0

SHA256
8e244f1a5b4653d6dbb4cc3978c7dd773b227a443361fbc30265b79f102f7eed

Size
288 KB (295616 Bytes)

Compile Timestamp
2021-01-20 19:37:37 UTC

Links
MalwareBazaar, Malpedia, Dropping_sha256, Cape, VirusTotal

Filenames
Preview_report20-01.exe (VirusTotal)

Detections
MalwareBazaar: BazaLoader, **Virustotal**: 33/76 as of 2021-01-23 07:31:37 - Trojan.Win32.Zenpak.4!c (AegisLab), Backdoor:Win32/KZip.90c5e0b2 (Alibaba), BackDoor.Bazar.55 (DrWeb), Trojan.Win32.Zenpak.bfcu (Kaspersky), Trojan:Win64/Bazarldr.BMB!MSR (Microsoft), Trojan.Win32.Zenpak.bfcu (ZoneAlarm)

it unpacks to this

MD5
63784053ac2f608d94c18b17c46ab5d4

SHA1
e01c814d6a4993c74a2bfb87b1b661fe78c41291

SHA256
c0a087a520fdfb5f1e235618b3a5101969c1de85b498bc4670372c02756efd55

Size
98 KB (100864 Bytes)

Compile Timestamp
2021-01-20 19:10:11 UTC

Links
MalwareBazaar, Malpedia, Dropping_sha256, Dropping_sha256, Cape, VirusTotal

Filenames
none

Detections
MalwareBazaar: BazaLoader, **Virustotal**: 21/75 as of 2021-01-23 13:37:55 - Gen:Variant.Bulz.163525 (ALYac), Gen:Variant.Bulz.163525 (Ad-Aware), Trojan.Bulz.D27EC5 (Arcabit), Gen:Variant.Bulz.163525 (BitDefender), Gen:Variant.Bulz.163525 (B) (Emsisoft), Gen:Variant.Bulz.163525 (GData), Gen:Variant.Bulz.163525 (MicroWorld-eScan), Trojan:Win32/TrickBot.VSF!MTB (Microsoft), Trojan.TrickBot!8.E313 (TFE:5:6iToUtBEDBC) (Rising)

which finally drops

MD5
7e8eddaef14aa8de2369d1ca6347b06d

SHA1
4543e6da0515bb7d93e930c9f30e40912d495373

SHA256
f29253139dab900b763ef436931213387dc92e860b9d3abb7dcd46040ac28a0e

Size
89 KB (91136 Bytes)

Compile Timestamp
2021-01-18 14:29:29 UTC

Links
MalwareBazaar, Malpedia, Dropped_by_sha256, Cape, VirusTotal

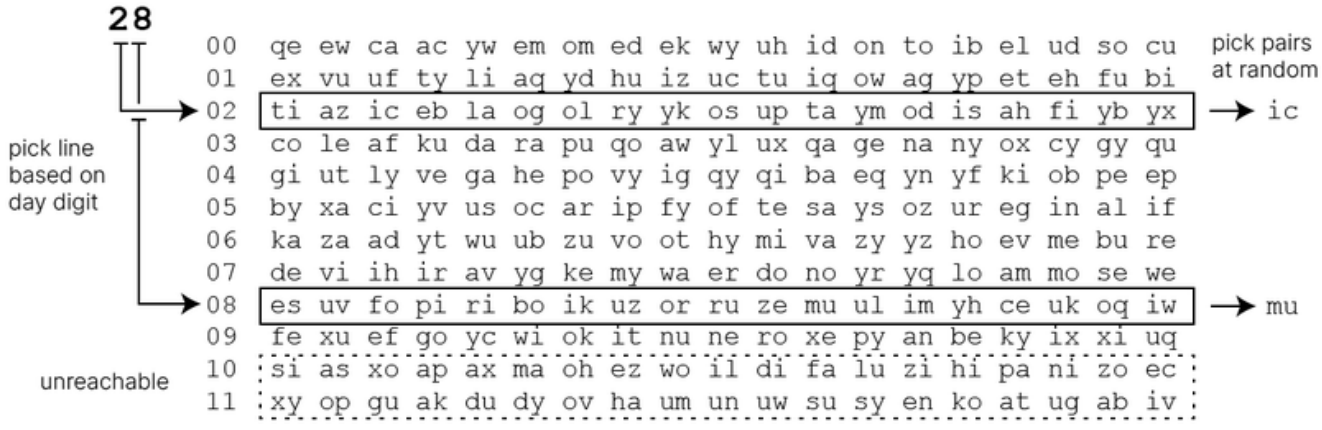
Filenames
none

Detections
MalwareBazaar: None, **Virustotal**: 19/76 as of 2021-01-23 15:04:35 - Gen:Variant.Bulz.163525 (ALYac), Gen:Variant.Bulz.163525 (Ad-Aware), Trojan.Win32.Bulz.4!c (AegisLab), Trojan.Bulz.D27EC5 (Arcabit), Gen:Variant.Bulz.163525 (BitDefender), Gen:Variant.Bulz.163525 (B) (Emsisoft), Gen:Variant.Bulz.163525 (FireEye), Gen:Variant.Bulz.163525 (GData), Gen:Variant.Bulz.163525 (MicroWorld-eScan)

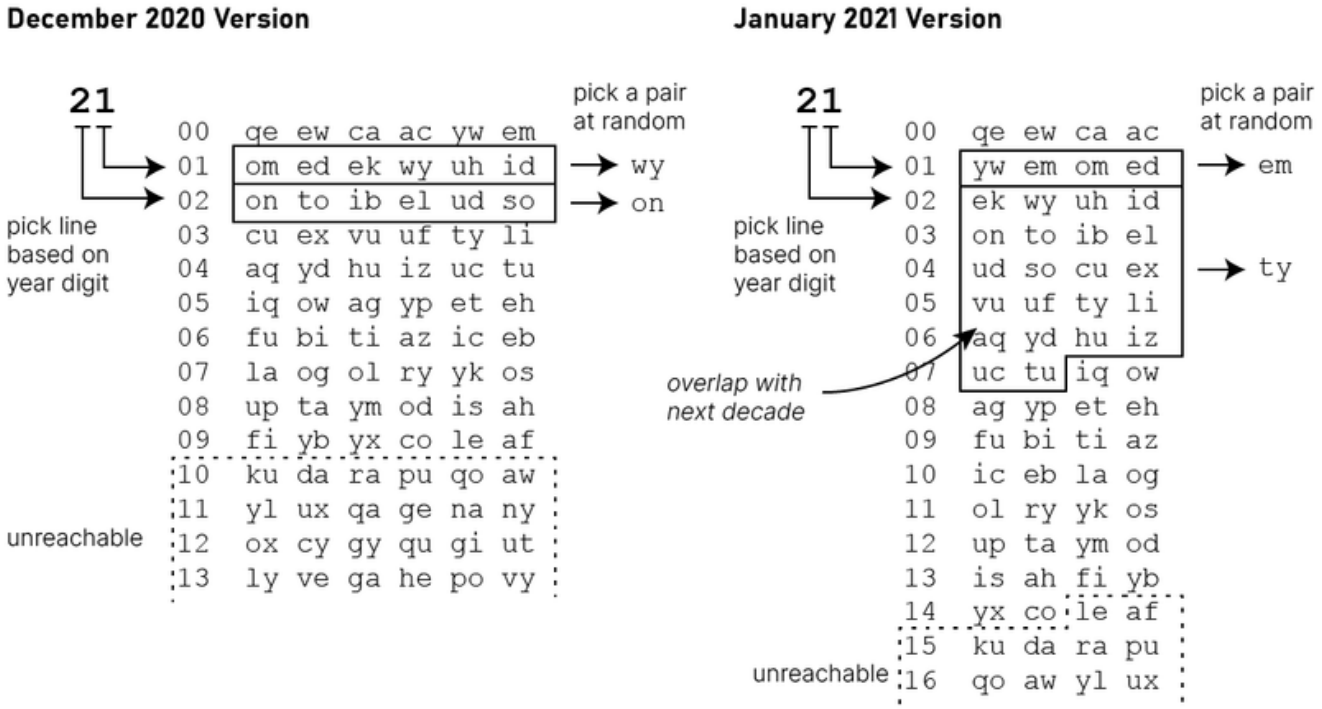
Difference from the Last Version

The current version is just a slight modification to the version from December. Like the previous version of the algorithm, this version calculates all ordered pairs of 19 consonants and 6 vowels (including *y*). These pairs are then permuted based on a fixed value. This value is the same, so the resulting list of 228 pairs is also the same.

The calculation of the first four letters is the same – that is, the selection of the first two pairs of letters: The permuted list of letters is divided into groups of 19 pairs. Then the two digits of the current month determine which group is selected. From these, one pair at a time is randomly – and unpredictably – selected.



The last four letters (two pairs) are still determined by the two year digits. However, the division of letter pairs into groups is different. Based on the current decade, two letters are chosen from a group of 22 pairs. The groups of 22 pairs partly overlap, so that theoretically after every 10 years identical domains could be generated again. This in contrast to the version from December, where the decade still determined a non-overlapping group of 6 pairs only. The last two letters are picked from groups of 4 — instead of 6 — letter pairs.



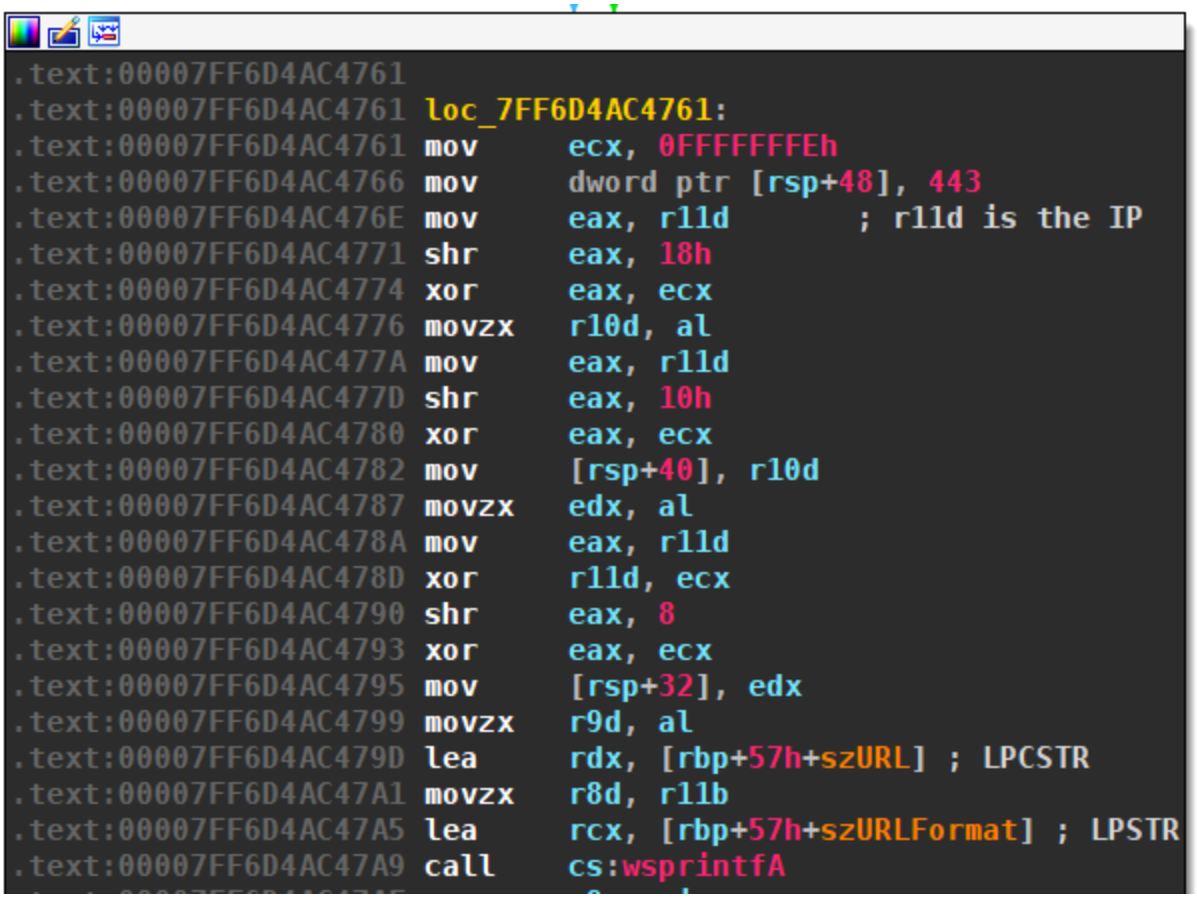
The DGA still generates 10'000 domains. But because there are 88 potential monthly combinations for the last for letters instead of just 36 previously, the excepted number of unique domains is larger:

$$ \mathbb{E} = 31768 \left(1 - \left(\frac{31768 -1}{31768 }\right)^{10000} \right) \approx 8579 $$

Since domain names partially repeat after each decade, domains can no longer be uniquely assigned to a seed. But since I strongly doubt that the domain generation algorithm will still have any relevance in a few months, let alone 10 years, the domain to seed tool assumes domains are from the 20s.

IP Transformation

The A record of the domains is encrypted, just as in previous versions. Meaning, the four bytes of the IP are XORed with `0xFE`. The IP is then used in URLs of format `https://{ip}:443`. For example, if the domain `omleekyw.bazar` has an A record of `220.39.239.27`, then the actual contacted URL is `https://34.217.17.229:443`.



```
.text:00007FF6D4AC4761
.text:00007FF6D4AC4761 loc_7FF6D4AC4761:
.text:00007FF6D4AC4761 mov     ecx, 0FFFFFFFEh
.text:00007FF6D4AC4766 mov     dword ptr [rsp+48], 443
.text:00007FF6D4AC476E mov     eax, r11d       ; r11d is the IP
.text:00007FF6D4AC4771 shr     eax, 18h
.text:00007FF6D4AC4774 xor     eax, ecx
.text:00007FF6D4AC4776 movzx   r10d, al
.text:00007FF6D4AC477A mov     eax, r11d
.text:00007FF6D4AC477D shr     eax, 10h
.text:00007FF6D4AC4780 xor     eax, ecx
.text:00007FF6D4AC4782 mov     [rsp+40], r10d
.text:00007FF6D4AC4787 movzx   edx, al
.text:00007FF6D4AC478A mov     eax, r11d
.text:00007FF6D4AC478D xor     r11d, ecx
.text:00007FF6D4AC4790 shr     eax, 8
.text:00007FF6D4AC4793 xor     eax, ecx
.text:00007FF6D4AC4795 mov     [rsp+32], edx
.text:00007FF6D4AC4799 movzx   r9d, al
.text:00007FF6D4AC479D lea     rdx, [rbp+57h+szURL] ; LPCSTR
.text:00007FF6D4AC47A1 movzx   r8d, r11b
.text:00007FF6D4AC47A5 lea     rcx, [rbp+57h+szURLFormat] ; LPSTR
.text:00007FF6D4AC47A9 call    cs:wsprintfA
```

Reimplementation in Python

This is the new version reimplemented in Python

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

Param = namedtuple('Param', 'mul mod idx')
pool = (
    "qeewcaacywemomedekwyuhidontoibeludsocuexvuuftyliaqydhuizuctuiqow"
    "agypetehfubitiaziceblaogolryykosuptaymodisahfiybyxcoleafkudarapu"
    "qoawyluxqagenanyoxcygyqugiutlyvegahepovyigqyqibaeqynyfkiobpeepby"
    "xaciyvusocaripfyoftesaysozureginalifkazaadytwuubzuvoothymivazyyz"
    "hoevmeburedeviihiravygkemywaerdonoyryqloammoseweesuvfopiriboikuz"
    "orruzemuulimyhceukoqiwfexuefgoycwiokitnuneroxepyanbekyixxiuqsias"
    "xoapaxmaohezwoildifaluzihipanizoecxyopguakdudyovhaumunuwsusyenko"
    "atugabiv"
)



def dga(date):
    seed = date.strftime("%m%Y")
    params = [
        Param(19, 19, 0),
        Param(19, 19, 1),
        Param(4, 22, 4),
        Param(4, 4, 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)
```

Edit 23 March, 2021: There is also a version with a different character pool, but otherwise same algorithm (see my GitHub repo for the full code).

```
pool = (
    "yzewevmeywreomviekwyavygontowaerudsoyr"
    "exvuamtyseweesuvizpituiqowuzoretzemuul"
    "tiazicukoqiwolxuykosupwiymitisneroxeyx"
    "anlekyixxirasiasxoapuxqaohezwooxdigyqu"
    "ziutpavezohexyvyguqyqidyovynumunuwsusy"
    "enxaatyvusivaripfyoftesaysozureginalif"
)
```

Characteristics

Except for the number of domains per month, the characteristics are the same as for the previous verion:

property	value
type	TDD (time-dependent-deterministic)
generation scheme	arithmetic
seed	current date
domain change frequency	every month
unique domains per month	19·19·22·4 = 31'768
sequence	random selection, might pick domains multiple times
wait time between domains	10 seconds
top level domain	`.bazar`
second level characters	a-z, without j
regex	`[a-ik-z]{8}\.bazar`
second level domain length	8