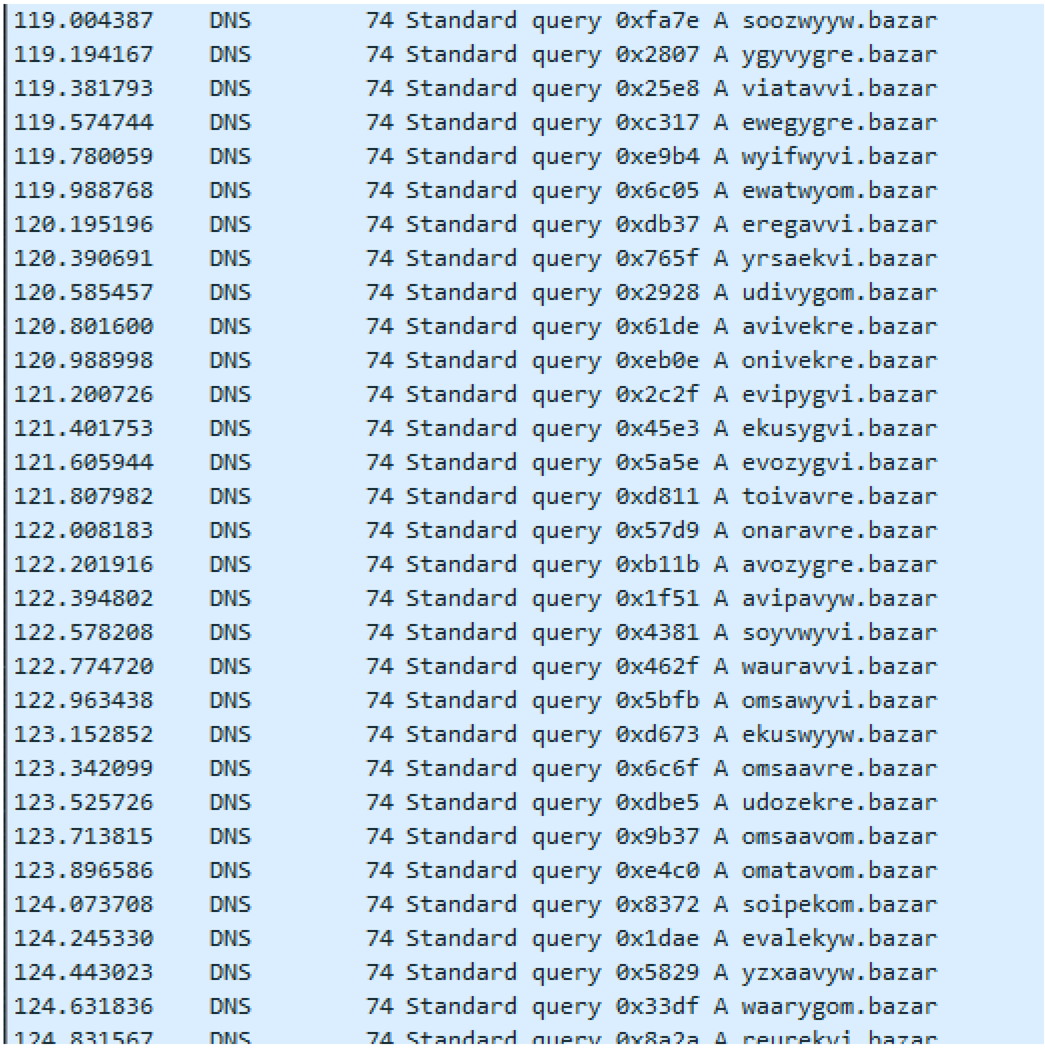
A BazarLoader DGA that Breaks Down in the Summer

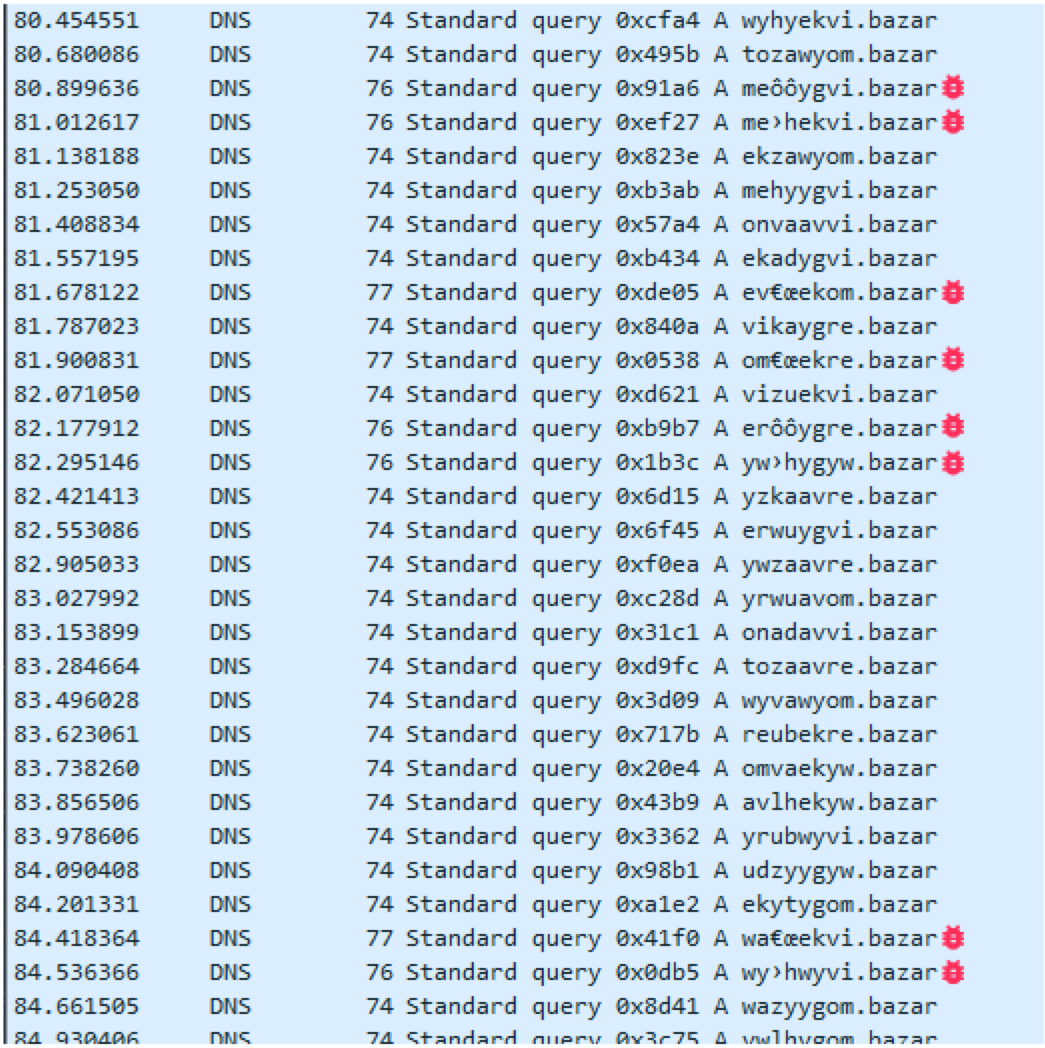
johannesbader.ch/blog/a-bazarloader-dga-that-breaks-during-summer-months/

André Tavares sent me a Bazar Loader sample whose Domain Generation Algorithm (DGA) shows some interesting behavior. In May, it generates valid domain names with the eponymous top level domain *.bazar*:



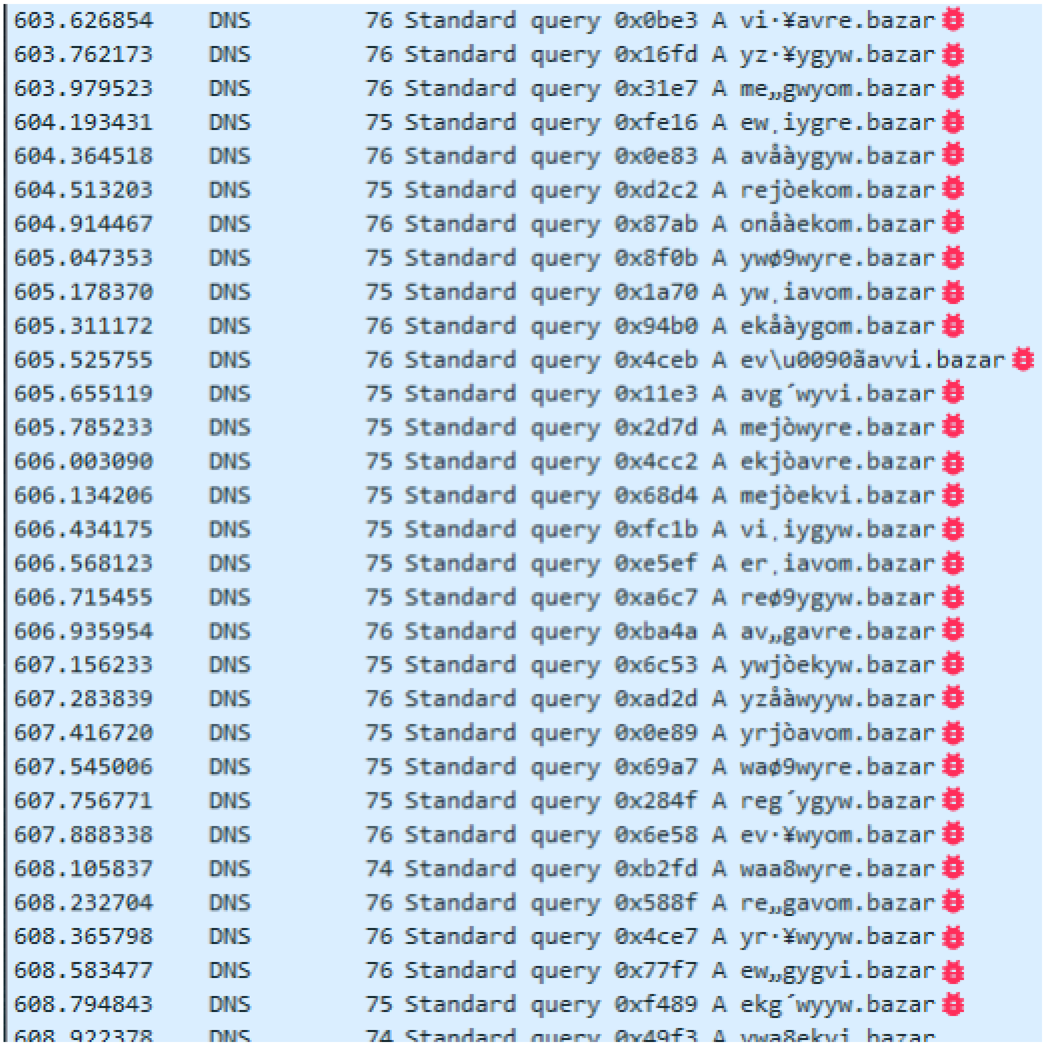
```
119.004387    DNS         74 Standard query 0xfa7e A soozwyyw.bazar
119.194167    DNS         74 Standard query 0x2807 A ygyvygre.bazar
119.381793    DNS         74 Standard query 0x25e8 A viatavvi.bazar
119.574744    DNS         74 Standard query 0xc317 A ewegygre.bazar
119.780059    DNS         74 Standard query 0xe9b4 A wyifwyvi.bazar
119.988768    DNS         74 Standard query 0x6c05 A ewatwyom.bazar
120.195196    DNS         74 Standard query 0xdb37 A eregavvi.bazar
120.390691    DNS         74 Standard query 0x765f A yrsaekvi.bazar
120.585457    DNS         74 Standard query 0x2928 A udivygom.bazar
120.801600    DNS         74 Standard query 0x61de A avivekre.bazar
120.988998    DNS         74 Standard query 0xeb0e A onivekre.bazar
121.200726    DNS         74 Standard query 0x2c2f A evipygvi.bazar
121.401753    DNS         74 Standard query 0x45e3 A ekusygvi.bazar
121.605944    DNS         74 Standard query 0x5a5e A evozygvi.bazar
121.807982    DNS         74 Standard query 0xd811 A toivavre.bazar
122.008183    DNS         74 Standard query 0x57d9 A onaravre.bazar
122.201916    DNS         74 Standard query 0xb11b A avozygre.bazar
122.394802    DNS         74 Standard query 0x1f51 A avipavyw.bazar
122.578208    DNS         74 Standard query 0x4381 A soyvwyvi.bazar
122.774720    DNS         74 Standard query 0x462f A wauravvi.bazar
122.963438    DNS         74 Standard query 0x5bfb A omsawyvi.bazar
123.152852    DNS         74 Standard query 0xd673 A ekuswyyw.bazar
123.342099    DNS         74 Standard query 0x6c6f A omsaavre.bazar
123.525726    DNS         74 Standard query 0xdbe5 A udozekre.bazar
123.713815    DNS         74 Standard query 0x9b37 A omsaavom.bazar
123.896586    DNS         74 Standard query 0xe4c0 A omatavom.bazar
124.073708    DNS         74 Standard query 0x8372 A soipekom.bazar
124.245330    DNS         74 Standard query 0x1dae A evalekyw.bazar
124.443023    DNS         74 Standard query 0x5829 A yzxaavyw.bazar
124.631836    DNS         74 Standard query 0x33df A waarygom.bazar
124.831567    DNS         74 Standard query 0x8a2a A reurekvi.bazar
```

But as soon as June comes around, some generated domains contain invalid characters:



```
80.454551     DNS        74 Standard query 0xcfa4 A wyhyekvi.bazar
80.680086     DNS        74 Standard query 0x495b A tozawyom.bazar
80.899636     DNS        76 Standard query 0x91a6 A meôôygvi.bazar
81.012617     DNS        76 Standard query 0xef27 A me›hekvi.bazar
81.138188     DNS        74 Standard query 0x823e A ekzawyom.bazar
81.253050     DNS        74 Standard query 0xb3ab A mehyygvi.bazar
81.408834     DNS        74 Standard query 0x57a4 A onvaavvi.bazar
81.557195     DNS        74 Standard query 0xb434 A ekadygvi.bazar
81.678122     DNS        77 Standard query 0xde05 A ev€œekom.bazar
81.787023     DNS        74 Standard query 0x840a A vikaygre.bazar
81.900831     DNS        77 Standard query 0x0538 A om€œekre.bazar
82.071050     DNS        74 Standard query 0xd621 A vizuekvi.bazar
82.177912     DNS        76 Standard query 0xb9b7 A erôôygre.bazar
82.295146     DNS        76 Standard query 0x1b3c A yw›hygyw.bazar
82.421413     DNS        74 Standard query 0x6d15 A yzkaavre.bazar
82.553086     DNS        74 Standard query 0x6f45 A erwuygvi.bazar
82.905033     DNS        74 Standard query 0xf0ea A ywzaavre.bazar
83.027992     DNS        74 Standard query 0xc28d A yrwuavom.bazar
83.153899     DNS        74 Standard query 0x31c1 A onadavvi.bazar
83.284664     DNS        74 Standard query 0xd9fc A tozaavre.bazar
83.496028     DNS        74 Standard query 0x3d09 A wyvawyom.bazar
83.623061     DNS        74 Standard query 0x717b A reubekre.bazar
83.738260     DNS        74 Standard query 0x20e4 A omvaekyw.bazar
83.856506     DNS        74 Standard query 0x43b9 A avlhekyw.bazar
83.978606     DNS        74 Standard query 0x3362 A yrubwyvi.bazar
84.090408     DNS        74 Standard query 0x98b1 A udzyygyw.bazar
84.201331     DNS        74 Standard query 0xa1e2 A ekytygom.bazar
84.418364     DNS        77 Standard query 0x41f0 A wa€œekvi.bazar
84.536366     DNS        76 Standard query 0x0db5 A wy›hwyvi.bazar
84.661505     DNS        74 Standard query 0x8d41 A wazyygom.bazar
84.930406     DNS        74 Standard query 0x3c75 A ywlhygom.bazar
```

And as it gets to July, all domain names are invalid (with very few exceptions):



```
603.626854    DNS         76 Standard query 0x0be3 A vi·¥avre.bazar
603.762173    DNS         76 Standard query 0x16fd A yz·¥ygyw.bazar
603.979523    DNS         76 Standard query 0x31e7 A me„gwyom.bazar
604.193431    DNS         75 Standard query 0xfe16 A ew¸iygre.bazar
604.364518    DNS         76 Standard query 0x0e83 A avååygyw.bazar
604.513203    DNS         75 Standard query 0xd2c2 A rejòekom.bazar
604.914467    DNS         76 Standard query 0x87ab A onååekom.bazar
605.047353    DNS         75 Standard query 0x8f0b A ywø9wyre.bazar
605.178370    DNS         75 Standard query 0x1a70 A yw¸iavom.bazar
605.311172    DNS         76 Standard query 0x94b0 A ekååygom.bazar
605.525755    DNS         76 Standard query 0x4ceb A ev\u0090ãavvi.bazar
605.655119    DNS         75 Standard query 0x11e3 A avg´wyvi.bazar
605.785233    DNS         75 Standard query 0x2d7d A mejòwyre.bazar
606.003090    DNS         75 Standard query 0x4cc2 A ekjòavre.bazar
606.134206    DNS         75 Standard query 0x68d4 A mejòekvi.bazar
606.434175    DNS         75 Standard query 0xfc1b A vi¸iygyw.bazar
606.568123    DNS         75 Standard query 0xe5ef A er¸iavom.bazar
606.715455    DNS         75 Standard query 0xa6c7 A reø9ygyw.bazar
606.935954    DNS         76 Standard query 0xba4a A av„gavre.bazar
607.156233    DNS         75 Standard query 0x6c53 A ywjòekyw.bazar
607.283839    DNS         76 Standard query 0xad2d A yzååwyyw.bazar
607.416720    DNS         75 Standard query 0x0e89 A yrjòavom.bazar
607.545006    DNS         75 Standard query 0x69a7 A waø9wyre.bazar
607.756771    DNS         75 Standard query 0x284f A reg´ygyw.bazar
607.888338    DNS         76 Standard query 0x6e58 A ev·¥wyom.bazar
608.105837    DNS         74 Standard query 0xb2fd A waa8wyre.bazar
608.232704    DNS         76 Standard query 0x588f A re„gavom.bazar
608.365798    DNS         76 Standard query 0x4ce7 A yr·¥wyyw.bazar
608.583477    DNS         76 Standard query 0x77f7 A ew„gygvi.bazar
608.794843    DNS         75 Standard query 0xf489 A ekg´wyyw.bazar
608.922378    DNS         74 Standard query 0x49f3 A ywa8ekvi.bazar
```

The DGA also fails during August and September. But when October rolls around, all domains are valid again. This continues until next June, when the DGA has problems all over again.

This short blog post explores what causes the DGA to stop working properly in the summer, of all times.

The Sample Examined

I reverse engineered the DGA of the following sample:

MD5
5f11f2db1295fa419b190bd7478d9b23

SHA1
96d6c37fa0046a8dc1c520249dc94122e0fb3f52

SHA256
86d2aa04988befc74eccca5d99550f67093969b31aafa11cdce3476a4c59ba74

Size

248 KB (254474 Bytes)

Compile Timestamp
2021-07-13 08:22:30 UTC

Links
MalwareBazaar, Cape, VirusTotal

Filename
5f11f2db1295fa419b190bd7478d9b23.dll (MalwareBazaar), (VirusTotal)

Detections
MalwareBazaar: BazaLoader, **Virustotal**: 47/75 as of 2021-08-05 11:35:35 - Gen:Variant.Razy.892983 (MicroWorld-eScan), Trojan.Agent (CAT-QuickHeal), Backdoor.Win64.Bazdor.ah (Sangfor), Backdoor:Win64/Bazdor.ae3c68af (Alibaba), Trojan (0057f6941) (K7GW), Trojan (0057f6941) (K7AntiVirus), W64/Trojan.FRTN-3244 (Cyren), Win64/BazarLoader.AP (ESET-NOD32), generic.ml (Paloalto), Backdoor.Win64.Bazdor.ah (Kaspersky), Gen:Variant.Razy.892983 (BitDefender), Win64:DropperX-gen [Drp] (Avast), Gen:Variant.Razy.892983 (Ad-Aware), Gen:Variant.Razy.892983 (B) (Emsisoft), Trojan.Agent.Win64.8672 (Zillya), Artemis!Trojan (McAfee-GW-Edition), Trojan.Agent.dkxh (Jiangmin), TR/Redcap.ntozn (Avira), malware (ai score=88) (MAX), Win32.Troj.Undef.(kcloud) (Kingsoft), Trojan.Win64.Agent.oa (Gridinsoft), Trojan:Win64/Cobaltstrike.A!MSR (Microsoft), Backdoor.Win64.Bazdor.ah (ZoneAlarm), Gen:Variant.Razy.892983 (GData), Trojan.Win64.Convagent (VBA32), Gen:Variant.Razy.892983 (ALYac), Trojan.Bazar (Malwarebytes), Trojan.Agent!v7VRXZm6ckQ (Yandex), Trojan.Win64.Bazarloader (Ikarus), Win64:DropperX-gen [Drp] (AVG), Trj/CI.A (Panda)

I have unpacked it to the following state:

MD5
7c64ea7c4a229414b6048d18ab0836fd

SHA1
f10621be9bfee0152931f7790c2cbff022611f62

SHA256
d15dbfb7ef0511556a3527cc98d09145a56302bdd19a6083ee6d007af3352434

Size
113 KB (116224 Bytes)

Compile Timestamp
2021-07-12 13:27:57 UTC

Links
MalwareBazaar, Cape, VirusTotal

Detections

MalwareBazaar: BazaLoader, **Virustotal**: 40/75 as of 2021-08-05 19:07:37 - Trojan.Win32.Razy.4!c (Lionic), Gen:Variant.Razy.891147 (MicroWorld-eScan), Gen:Variant.Razy.891147 (FireEye), Backdoor.Bazdor.Win64.3 (Zillya), Backdoor:Win64/Bazdor.9312a6ac (Alibaba), Trojan (0057f6941) (K7GW), Trojan (0057f6941) (K7AntiVirus), W64/Trojan.QFLC-7900 (Cyren), Win64/BazarLoader.AP (ESET-NOD32), Backdoor.Win64.Bazdor.ax (Kaspersky), Gen:Variant.Razy.891147 (BitDefender), Gen:Variant.Razy.891147 (Ad-Aware), BehavesLike.Win64.Trojan.ch (McAfee-GW-Edition), Gen:Variant.Razy.891147 (B) (Emsisoft), Trojan.Win64.Bazarloader (Ikarus), TR/Redcap.rlvgc (Avira), malware (ai score=81) (MAX), Win32.Hack.Undef.(kcloud) (Kingsoft), Trojan.Win64.Agent.oa (Gridinsoft), Trojan:Win32/Tiggre!rfn (Microsoft), Gen:Variant.Razy.891147 (GData), Backdoor.Win64.Bazdor (VBA32), Gen:Variant.Razy.891147 (ALYac), Trojan.Bazar (Malwarebytes), Win64.Backdoor.Bazdor.Ajls (Tencent), W64/BazarLoader.AP!tr (Fortinet), Trj/CI.A (Panda)

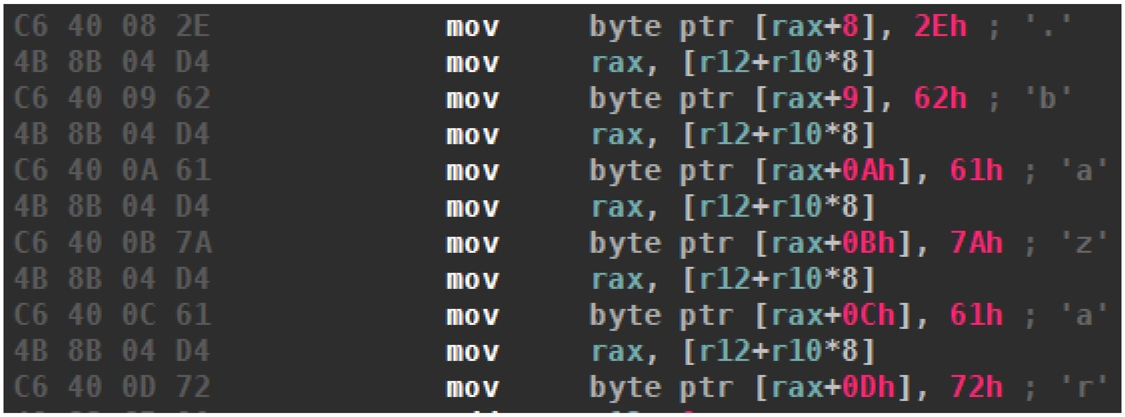
The Domain Generation Algorithm

The DGA can be easily be located in the unpacked sample based on the *.bazar* TLD, for example with this Yara rule:

```
rule BazarDGA
{
    strings:
        $bazar_tld= { 2E [4-12] 62 [4-12] 61 [4-12] 7A [4-12] 61 [4-12] 72 }

    condition:
        $bazar_tld
}
```

The rule triggers at the following location, which adds the top level domain to the generated domain (pointed to by `rax`) at the end of the DGA function:



```
C6 40 08 2E          mov     byte ptr [rax+8], 2Eh ; '.'
4B 8B 04 D4          mov     rax, [r12+r10*8]
C6 40 09 62          mov     byte ptr [rax+9], 62h ; 'b'
4B 8B 04 D4          mov     rax, [r12+r10*8]
C6 40 0A 61          mov     byte ptr [rax+0Ah], 61h ; 'a'
4B 8B 04 D4          mov     rax, [r12+r10*8]
C6 40 0B 7A          mov     byte ptr [rax+0Bh], 7Ah ; 'z'
4B 8B 04 D4          mov     rax, [r12+r10*8]
C6 40 0C 61          mov     byte ptr [rax+0Ch], 61h ; 'a'
4B 8B 04 D4          mov     rax, [r12+r10*8]
C6 40 0D 72          mov     byte ptr [rax+0Dh], 72h ; 'r'
```

Here is how the DGA works:

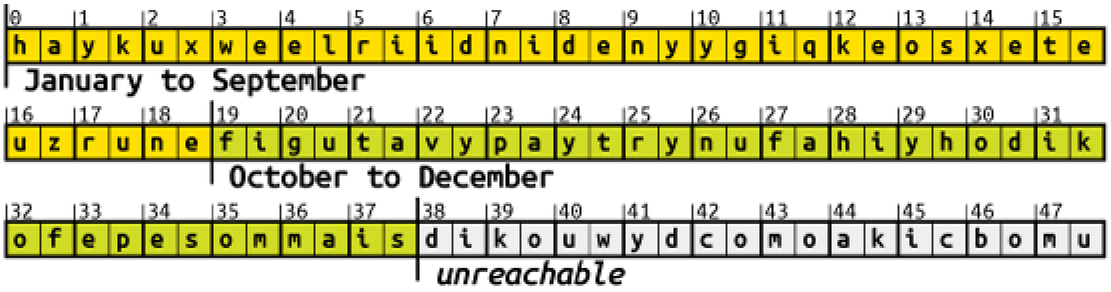
1. BazarLoader divides the letters – except J, which was omitted for unknown reasons – into two character classes:

   - the 6 vowels *aeiouy*
   - the 19 consonants *bcdfghklmnpqrstvwxz*
2. The two sets are then combined into all 2·6·19 ordered pairs that contain one vowel and one consonant: `ab` , `ba` , `eb` , `be` , `ib` , `bi` , `ob` , `bo` , …, `oz` , `zo` , `uz` , `zu` , `yz` , `zy` .

3. These 228 pairs are then rearranged with a permutation that is hard-coded into the malware. The permutation is the seed of the BazarLoader DGA and offers the possibility to generate a different set of domains with the same algorithm. The permutation is stored as an array of 228 bytes that represent the one-line notation of the permutation. So for example, a permutation of 27, 119, 38, … would place the first pair `ab` at position 27, the second pair `ba` at 119, and so on (0 being the first position).

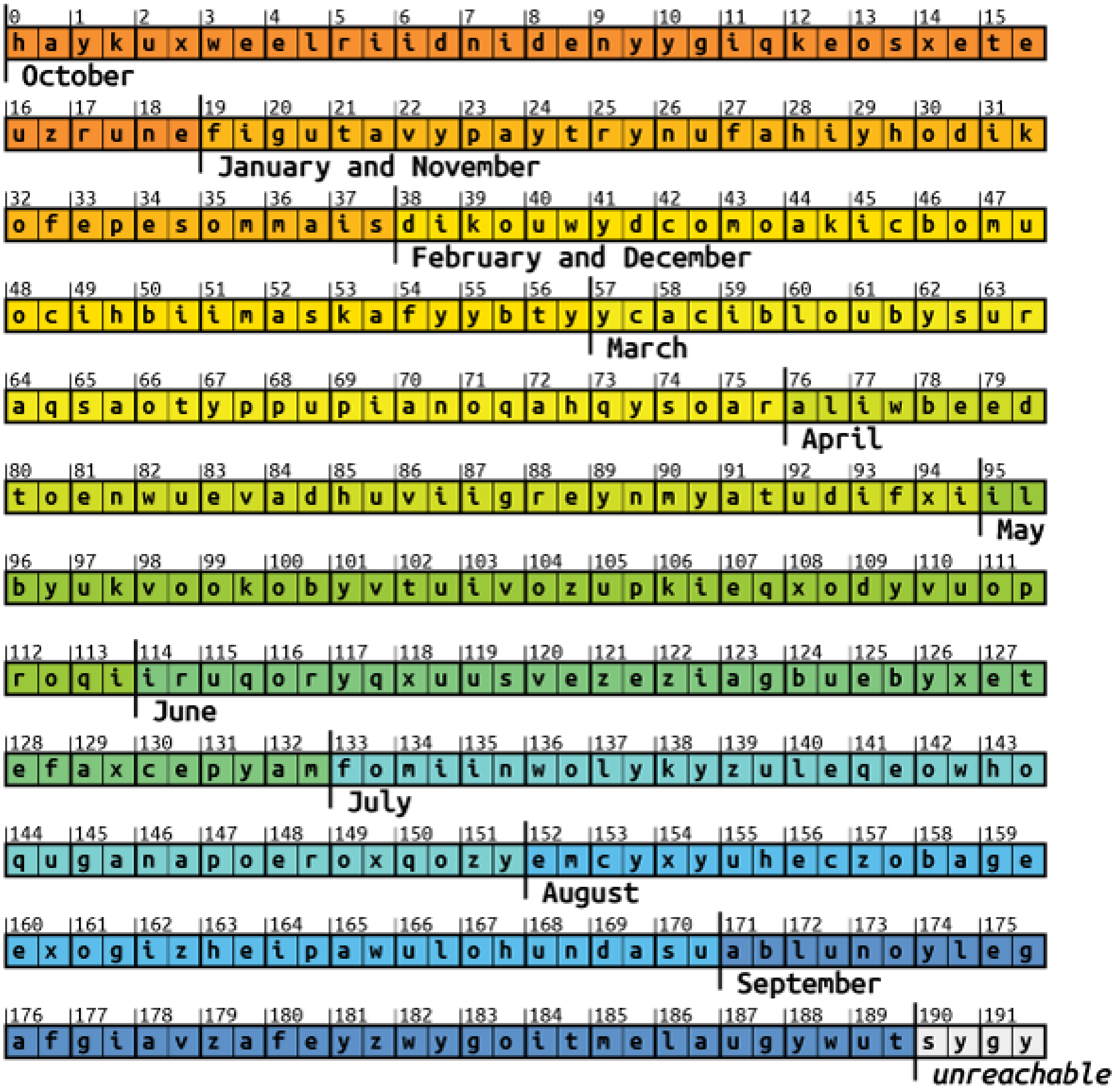
4. Four pairs are then picked from the 228 permutated pairs, and strung together to form the 8 letter long second level domain. Which pairs are selected depends on the current date. The date is formatted as `%m%y` , where `%m` is the zero-padded month and `%y` is the two digit year. For example, December 5, 2035 would be `1235` . The four digits, e.g., 1, 2, 3 and 5, then define which pairs will be selected for the first, second, third and fourth pair respectively.

5. The **first pair** is selected by first splitting the pairs into groups of 19 pairs. The first digit derived from the current date then serves as the index of the groups to select. Since the first digit can only be 0 or 1, only two groups are possible [1]

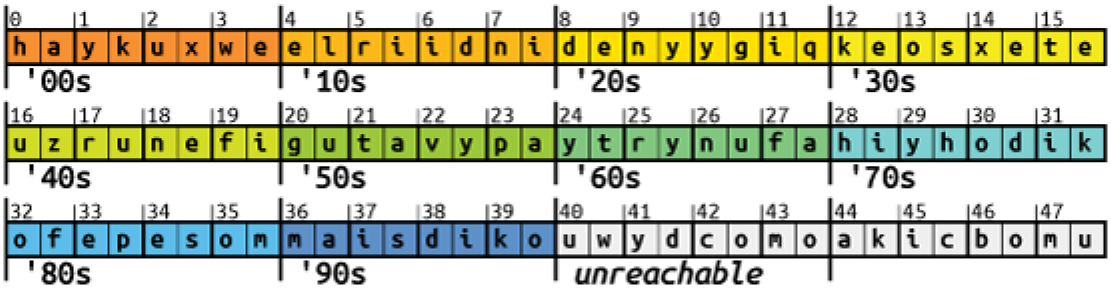
   BazarLoader then picks a pair at random from the 19 pairs of the given group.

6. The **second pair** is selected like the first pair, except the groups are picked based on the second date digit. This digit can be any value from 0 to 9, so ten different groups are possible:



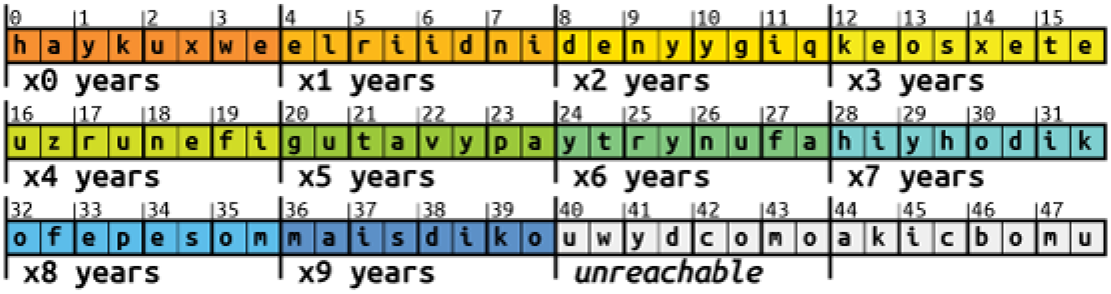
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
h a y k u x w e e l r i i d n i d e n y y g i q k e o s x e t e
October
16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
u z r u n e f i g u t a v y p a y t r y n u f a h i y h o d i k
January and November
32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47
o f e p e s o m m a i s d i k o u w y d c o m o a k i c b o m u
February and December
48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63
o c i h b i i m a s k a f y y b t y y c a c i b l o u b y s u r
March
64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79
a q s a o t y p p u p i a n o q a h q y s o a r a l i w b e e d
April
80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95
t o e n w u e v a d h u v i i g r e y n m y a t u d i f x i i l
May
96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111
b y u k v o o k o b y v t u i v o z u p k i e q x o d y v u o p
112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127
r o q i i r u q o r y q x u u s v e z e z i a g b u e b y x e t
June
128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143
e f a x c e p y a m f o m i i n w o l y k y z u l e q e o w h o
July
144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159
q u g a n a p o e r o x q o z y e m c y x y u h e c z o b a g e
August
160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175
e x o g i z h e i p a w u l o h u n d a s u a b l u n o y l e g
September
176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191
a f g i a v z a f e y z w y g o i t m e l a u g y w u t s y g y
unreachable

7. For the **third pair**, the groups only have a size of 4 pairs. Since the third date digit represents the decade, the same group will be selected for years to come.



0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
h a y k u x w e e l r i i d n i d e n y y g i q k e o s x e t e
'00s
'10s
'20s
'30s
16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
u z r u n e f i g u t a v y p a y t r y n u f a h i y h o d i k
'40s
'50s
'60s
'70s
32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47
o f e p e s o m m a i s d i k o u w y d c o m o a k i c b o m u
'80s
'90s
unreachable

8. The **fourth pair** is also picked from groups of 4 pairs, based on the least significant digit of the year.



9. The four picked pairs are concatenated into an 8-letter second level domain, and the top level domain `.bazar` is appended.

As can be seen from the illustrations above, pairs at higher positions are selected only as a second pair and only during the summer months. And that is exactly what causes the bug.

The Bug - A Faulty Permutation

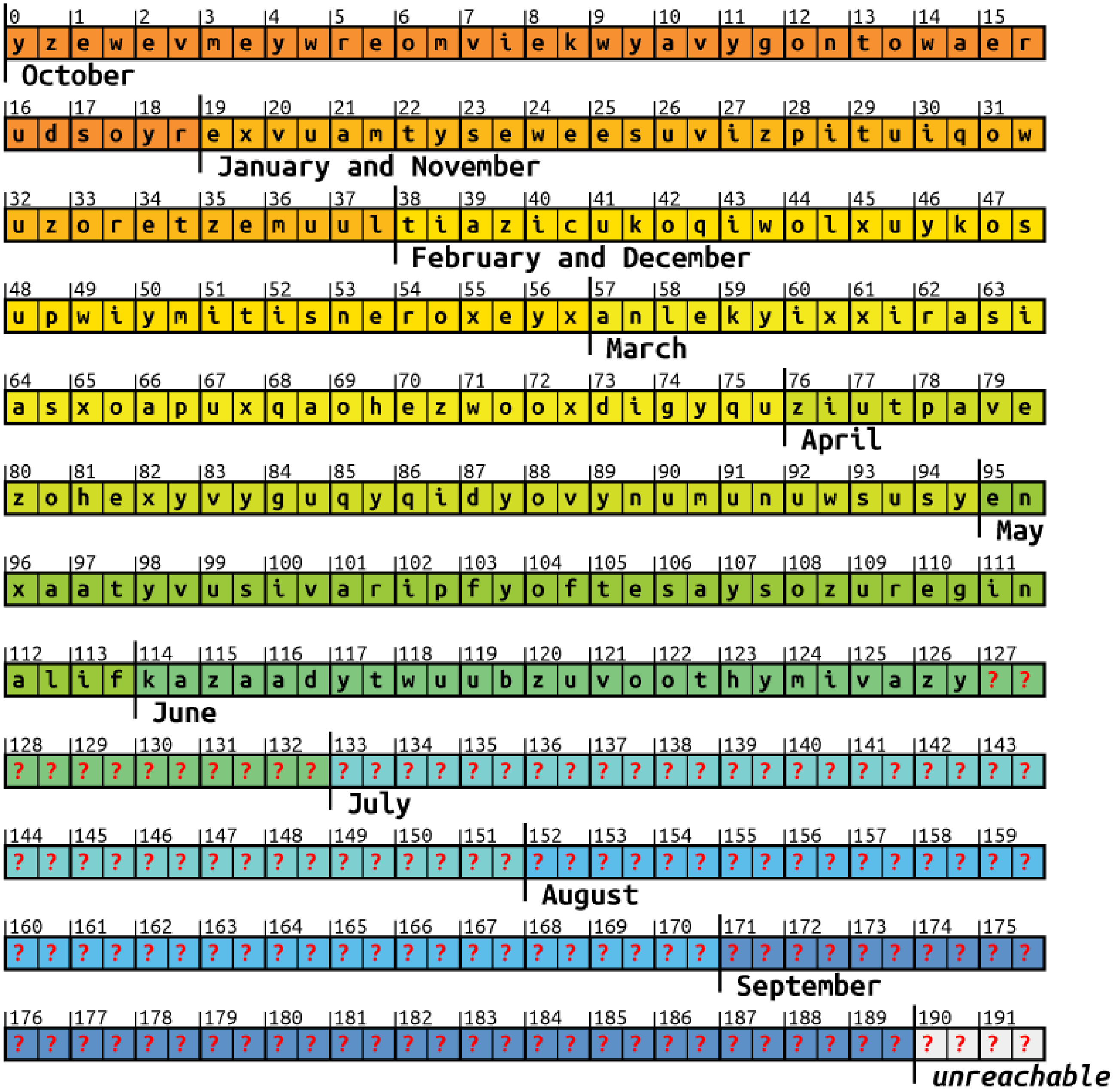
The DGA is implemented exactly as described above. The hard-coded permutation, however, is incorrect:

```
57 63 3A 29 25 0E 1E 5C  04 77 5F 37 02 03 28 51
61 28 39 64 12 1C 49 30  3D 74 06 07 49 0B 10 33
56 10 57 19 4A 3B 2C 2E  36 71 1B 68 24 15 67 5A
50 20 45 6E 4C 54 2F 2B  54 62 4A 0B 59 35 51 23
4D 08 01 45 1A 0A 7B 27  72 55 0C 08 5B 1F 60 32
3C 29 3B 2E 2A 70 3A 0F  17 48 14 2C 4B 25 4E 42
44 15 03 05 7C 26 16 06  24 5A 0D 32 46 39 35 5F
4F 6F 11 0C 34 5B 47 59  4E 42 5D 5E 1C 66 52 53
3F 30 38 21 44 18 00 58  56 1E 40 2A 4B 3E 55 13
3E 65 05 0F 1D 09 36 21  22 6D 2D 12 6A 40 17 19
3F 34 11 2F 5D 63 5E 6B  31 61 69 22 26 33 0D 7A
1D 4D 16 75 7D 0A 4F 02  07 64 79 58 14 1A 53 62
0E 41 18 01 31 2B 47 1F  76 5C 09 04 60 43 37 13
3D 3C 41 48 2D 43 52 38  73 27 23 46 4C 1B 50 6C
78 20 7E 00
```

For the permutation to be valid, i.e., bijective, it would need to contain all numbers from `0x00` to `0xe3` (227). But the largest number in the above list of numbers is only `0x7E` (126). Possibly the wrong data type was chosen when generating the permutation. For example, a signed char to store the numbers 1-228.

Instead of permuting the pairs, the DGA places them all in the first 127 places. Some pairs will therefore be overwritten by another pair placed in the same spot. For instance the first pair `ab` is placed at position `0x57` (first number of the "permutation") . But since `0x57` appears a second time (35th number of the "permutation"), the pair `ab` will be overwritten.

Similarly, all spots above 127 are never filled. So with the actual “permutation” applied, the illustration for picking the second pair looks as follows, where `?` denotes undefined memory:



All pairs in July, August and September are undefined and will likely result in invalid domains. In June, only 13 out of 19 pairs are undefined, hence some domains come out correct. All other months are not affected by the bug.

Reimplementation in Python

The following Python script will generate all possible domains for a given date. When it is run for months affected by the bug, the resulting domains will contain two `??` that represent characters from undefined memory.

```
from datetime import datetime
import argparse
from collections import namedtuple

Param = namedtuple('Param', 'block idx')
pool = (
    "yzewevmeywreomvi"
    "ekwyavygontowaer"
    "udsoyrexvuamtyse"
    "weesuvizpituiqow"
    "uzoretzemuultiaz"
    "icukoqiwolxuykos"
    "upwiymitisneroxe"
    "yxanlekyixxirasi"
    "asxoapuxqaohezwo"
    "oxdigyquziutpave"
    "zohexyvyguqyqidy"
    "ovynumunuwsusyen"
    "xaatyvusivaripfy"
    "oftesaysozuregin"
    "alifkazaadytwuub"
    "zuvoothymivazy"
)

pool +=(10*19*2 - len(pool))*"?"

def dga(date):
    seed = date.strftime("%m%Y")
    params = [
        Param(19, 0),
        Param(19, 1),
        Param(4, 4),
        Param(4, 5)
    ]

    ranges = []
    for p in params:
        s = int(seed[p.idx])
        lower = p.block*s
        upper = lower + p.block
        ranges.append(list(range(lower, upper)))

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

    return domains


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)
```

Characteristics of the DGA

The following table summarizes the properties of the BazarLoader DGA when it is working as intended, i.e., October through May.

property	**value**
type	TDD (time-dependent-deterministic)
generation scheme	arithmetic
seed	current date
domain change frequency	every month
unique domains per month	5776
sequence	random selection, might pick domains multiple times
wait time between domains	none
top level domain	.bazar
second level characters	a-z, without j
regex	[a-ik-z]{8}.bazar
second level domain length	8

1. note that the letters used in the illustrations are randomly placed and not the actual letter pairs that BazarLoader uses. ↩