Now Mirai Has DGA Feature Built in

By LIU Ya

Published: 2016-12-09 · Archived: 2026-04-05 16:00:14 UTC

Update History

- 2016-12-09 first version
- 2016-12-12 fig-0 update, fix a TLD choosing error in our DGA implement

Summary

Nearly 2 weeks ago, 2 new infection vectors (aka TCP ports of 7547 and 5555) were found being used to spread MIRAI malwares

- <A Few Observations of The New Mirai Variant on Port 7547>

My colleague Genshen quickly set up some honeypots for that sort of vectors and soon had his harvests: 11 samples were captured on Nov 28th. Twill now 53 unique samples have been captured by our honeypots from 6 hosting servers.

When analyzing one of the new samples, my colleague Wenji found some DGA like code and doubted there was DGA feature there. The doubt was soon verified by evidences collected from our sandboxes. Detailed RE work shows there does exist a DGA feature in the newly distributed MIRAI samples spread through TCP ports 7547 and 5555. In this blog I would like to introduce our findings. For a quick information, the attributes of the found DGA are summarized as follow:

1. 3 TLDs are used: online/tech/support.
2. the L2 domain has a fixed length of 12-byte, with each char randomly chosen from ‘a’~’z’ ‘a’~'y'.
3. the generated domain is only determined by month, day and hardcoded seed string.
   the generated domain is determined by `year`, month, day and hardcoded seed string.
4. only one domain is generated in one single day, so the maxium DGA domain number is 365.
5. the DGA domains are only used when the hardcoded C2 domains fail to resolve.

With the learned knowledge, we re-implemented the DGA in our program, and used it to predict all 365 possible DGA domains. When looking up their registration information, we found some of them have been registered by the MIRAI author. They are:

date	L2 domain	TLD	Registrant Email
12-04	vmdefmnsndoj	tech	beaba23f49bd4c688faec8a6e5b22a23.protect@whoisguard.com
12-05	xpknpxmywqsr	online	dlinchkravitz@gmail.com
12-06	lvfjcwwobycj	tech	ac4ca107e04e4d58ad1348d5d759b3b0.protect@whoisguard.com
12-07	nympompksmfx	tech	fd444a8a2eff4676ab52907ab261fc94.protect@whoisguard.com
12-08	kedbuffigfjs	online	dlinchkravitz@gmail.com
12-14	bwhrdaumwuvn	online	dlinchkravitz@gmail.com
12-19	bpmsfckfkrpr	online	dlinchkravitz@gmail.com
12-20	oornsduuwjli	tech	dlinchkravitz@gmail.com
12-21	qjqubpciajoc	tech	dlinchkravitz@gmail.com
12-22	exvdaajegjur	online	dlinchkravitz@gmail.com
12-24	poorcetnmjfc	online	dlinchkravitz@gmail.com
12-31	vtrndmhsgada	online	vtrndmhsgada.online@domainsbyproxy.com

Fig-0, registered DGA domains

And it is worth noticing that the author dlinchkravitz@gmail.com has already registered other mirai C2 domain:

- zugzwang.me email dlinchkravitz@gmail.com

Sample and Analysis

The sample used as illustration in this blog is as follows:

- **MD5**: bf136fb3b350a96fd1003b8557bb758a
- **SHA256**: 971156ec3dca4fa5c53723863966ed165d546a184f3c8ded008b029fd59d6a5a
- **File type**: ELF 32-bit LSB executable, MIPS, MIPS-I version 1 (SYSV), statically linked, stripped

The sample is stripped but not packed. According to the experience learned from previously found samples, we soon identified its main modules. The code comparison showed that its resolv_cnc_addr function has a very

different CFG (control flow graph) from the previously found samples. The new version of CFG is shown Fig-1.

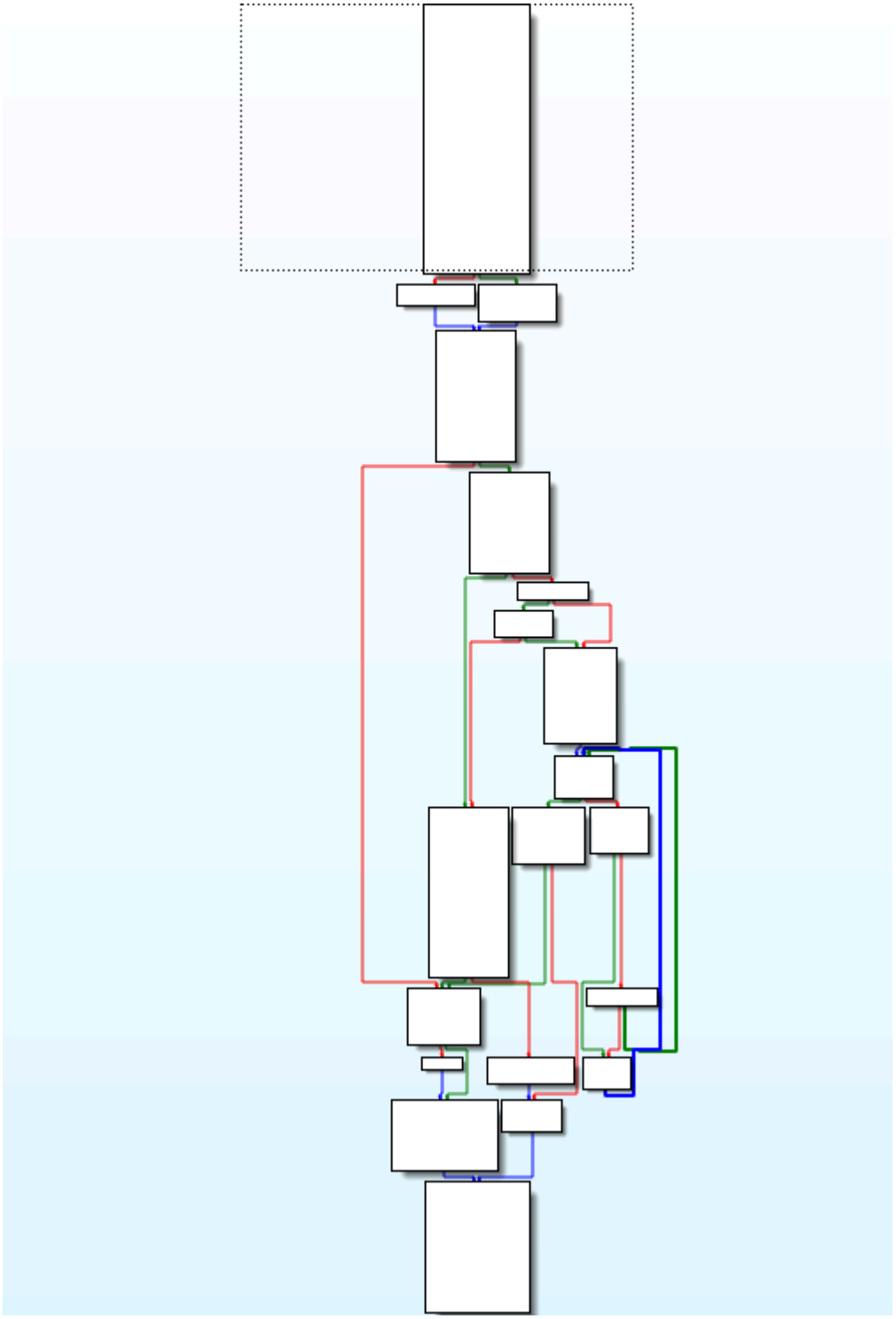


Fig-1, resolv_cnc_addr CFG

At the function beginning, since there are as much as 3 C2 controllers are hardcoded in the sample, a random number is generated to randomly select a C2 server from the first and second ones, as shown in Fig-2.

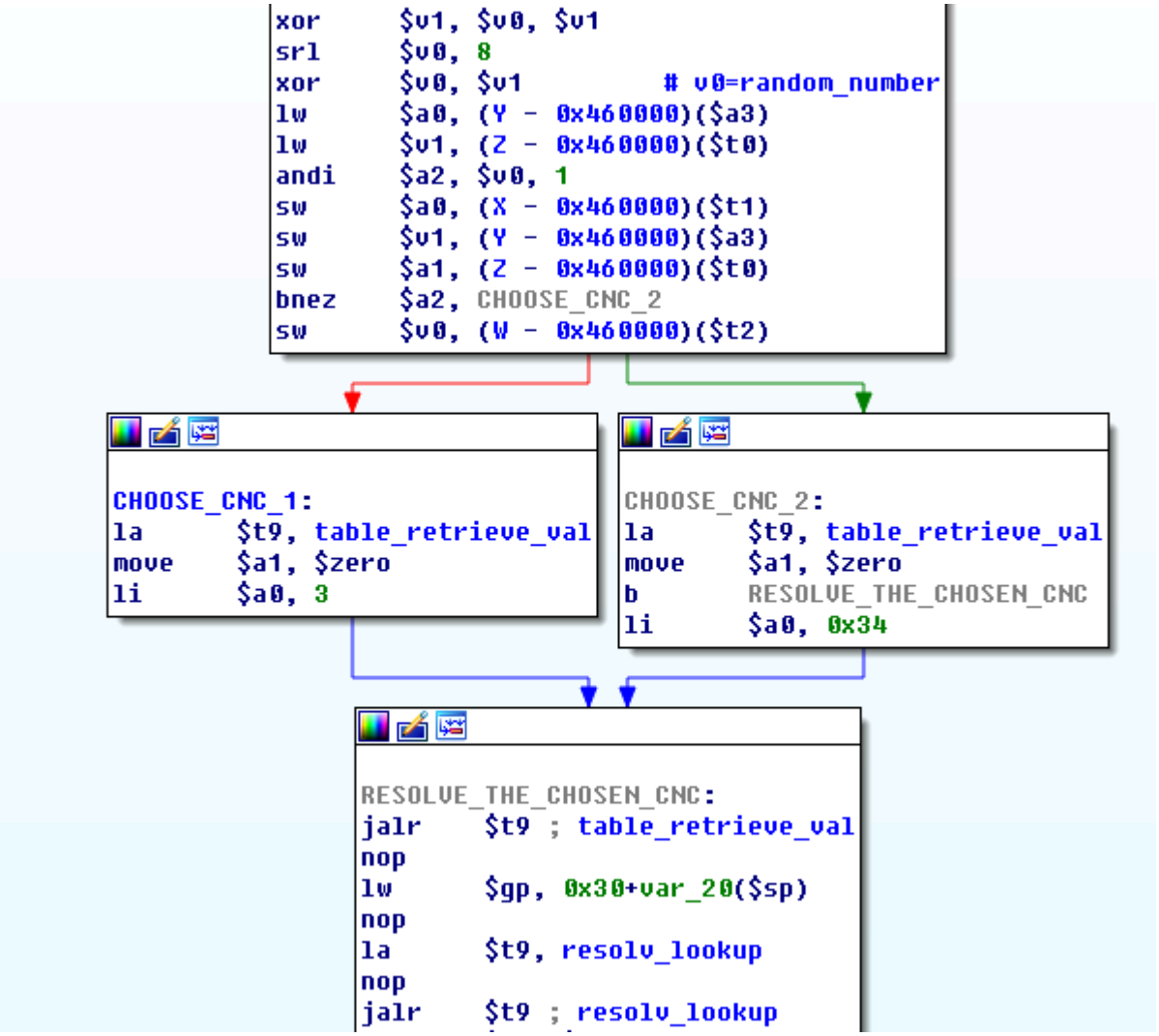


Fig-2, resolv_cnc_addr block 1

If the selected C2 domain fails to resolve, the bot will neither resolve the unselected nor the 3rd one, but will take a judge to decide whether to take the DGA branch or to resolve the 3rd C2 domain according to current date, as

shown in Fig-3.

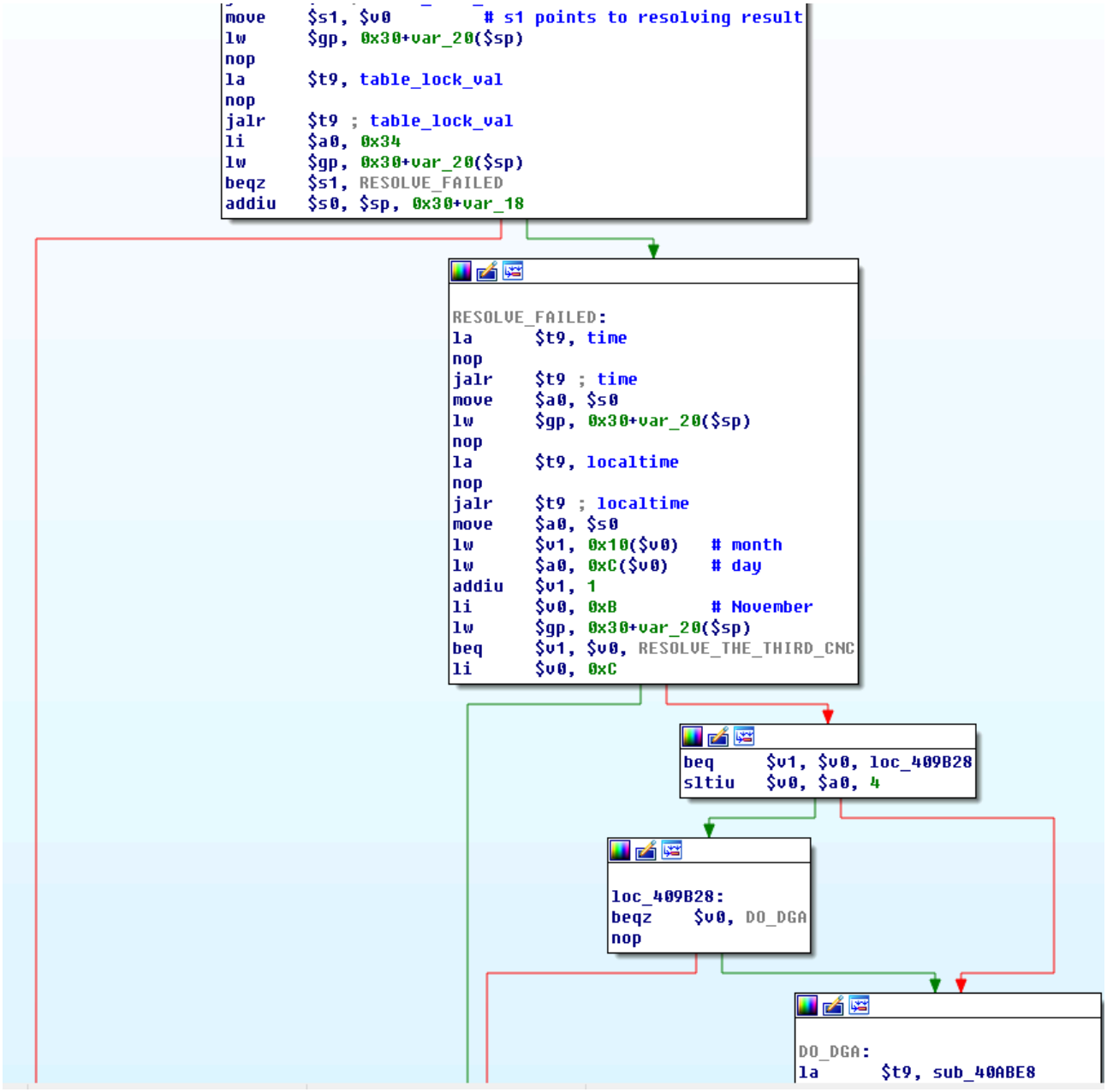


Fig-3, DGA determination

From the code snippets we can see that if current date is between Nov 1st and Dec 3rd, the 3rd CNC domain will be used. Otherwise the DGA branch will be executed. It indicates that the author doesn't want their DGA domains being used before Dec 4th, which is verified by the fact that the firstly registered MIRAI DGA domain just corresponds to Dec 4th.

The DGA main funcition is named dga_gen_domain. The domain is generated based on a seed number and current date. The seed is converted from a hardcoded hex-format string by calling strtol(). It seems a wrong string of "\x90\x91\x80\x90\x90\x91\x80\x90" was configured, which leads to the strtol() always returning 0.
The local date is got by calling C library functions of time() and localtime(). Only month and day are used here, as

shown in Fig-4.

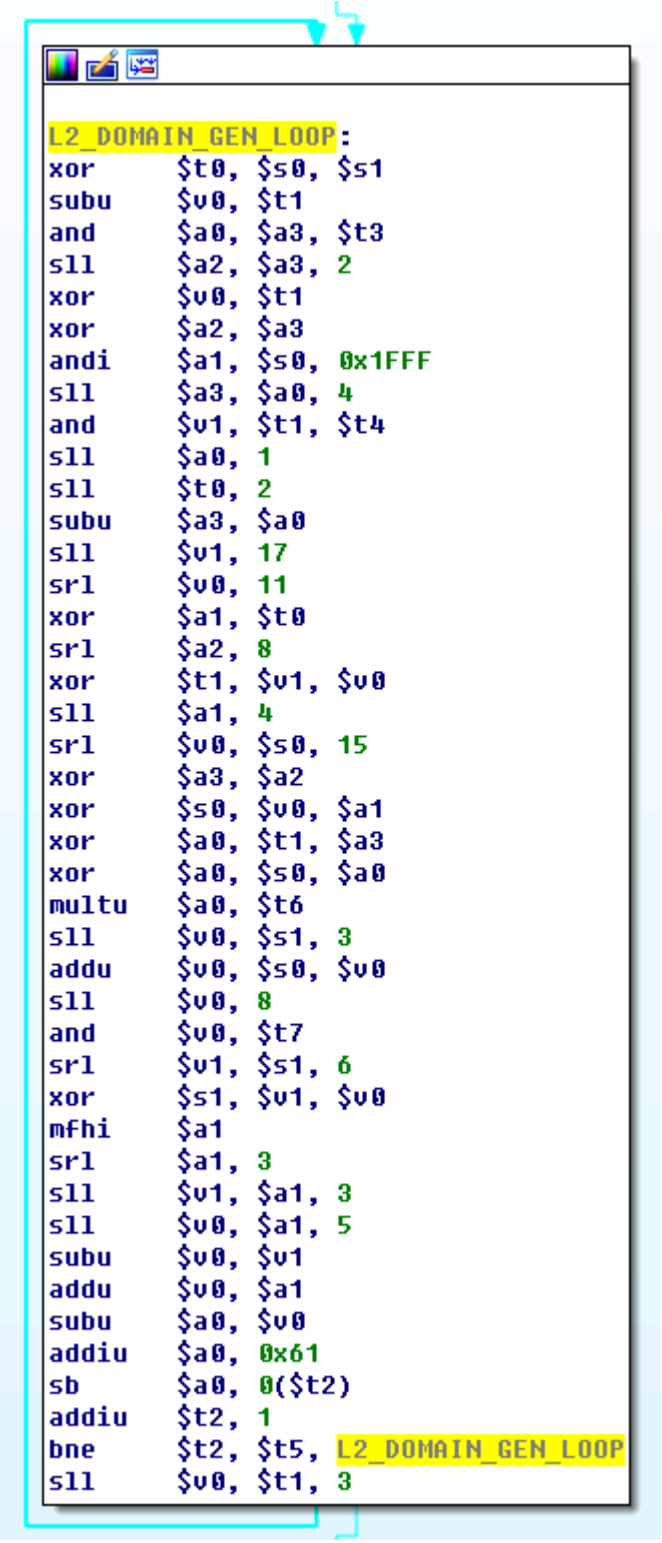
```
dga_gen_domain:

gp_= -0x30
var_time= -0x28
l2_domain= -0x24
l2_domain_end= -0x18
var_10= -0x10
var_C= -0xC
var_8= -8
var_4= -4

li      $gp, 0x57E50
addu    $gp, $t9
addiu   $sp, -0x40
sw      $ra, 0x40+var_4($sp)
sw      $s2, 0x40+var_8($sp)
sw      $s1, 0x40+var_C($sp)
sw      $s0, 0x40+var_10($sp)
sw      $gp, 0x40+gp_($sp)
la      $t9, time
addiu   $s0, $sp, 0x40+var_time
jalr    $t9 ; time
move    $a0, $s0
lw      $gp, 0x40+gp_($sp)
move    $a1, $zero
la      $a0, loc_410000
la      $t9, strtol
addiu   $a0, (dga_seed_string - 0x410000)  # "惺■惇憖
jalr    $t9 ; strtol
li      $a2, 0x10
lw      $gp, 0x40+gp_($sp)
move    $a0, $s0
la      $t9, localtime
nop
jalr    $t9 ; localtime
move    $s1, $v0          # $s1=dga_seed
lw      $v1, 0x10($v0)    # month
lw      $a0, 0x14($v0)    # year
addiu   $s2, $sp, 0x40+l2_domain
lw      $s0, 0xC($v0)     # day
addiu   $a3, $v1, 1
lui     $v0, 0x3FF
lui     $v1, 0x51EB
lw      $gp, 0x40+gp_($sp)
addiu   $t1, $a0, 0x76C
ori     $t7, $v0, 0xFF00
ori     $t6, $v1, 0x851F
move    $t2, $s2
addiu   $t5, $sp, 0x40+l2_domain_end
li      $t4, 0xFFFFFFF0
li      $t3, 0xFFFFFFFE
sll     $v0, $t1, 3
```

Fig-4, dga_gen_domain entry

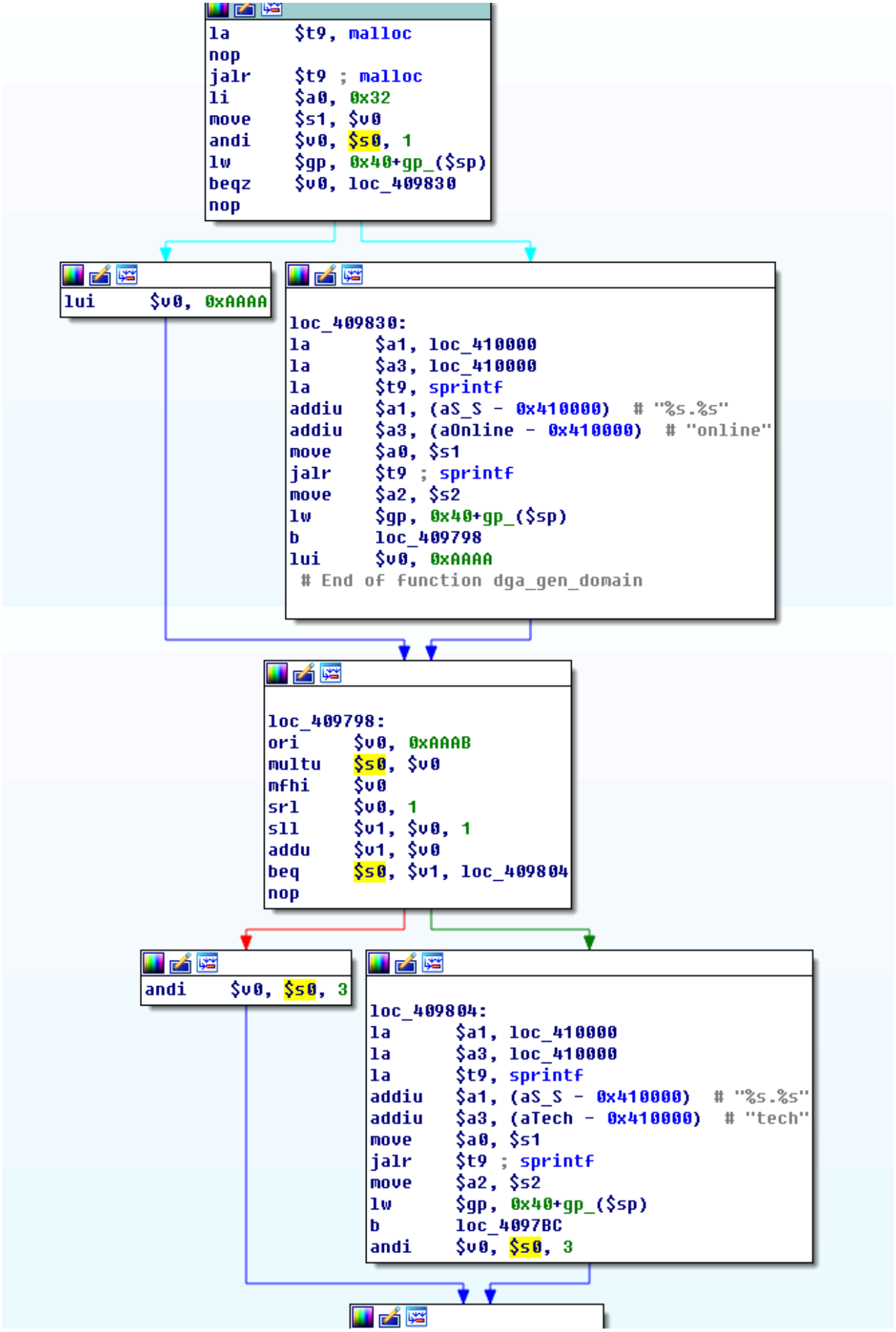
The L2 domain is generated by repeatedly executing the code block shown in Fig-5. Its length is determined by $t5 and $t2. They are set in Fig-4, from which we can tell that the L2 domain length is 12.



```
L2_DOMAIN_GEN_LOOP:
xor     $t0, $s0, $s1
subu    $v0, $t1
and     $a0, $a3, $t3
sll     $a2, $a3, 2
xor     $v0, $t1
xor     $a2, $a3
andi    $a1, $s0, 0x1FFF
sll     $a3, $a0, 4
and     $v1, $t1, $t4
sll     $a0, 1
sll     $t0, 2
subu    $a3, $a0
sll     $v1, 17
srl     $v0, 11
xor     $a1, $t0
srl     $a2, 8
xor     $t1, $v1, $v0
sll     $a1, 4
srl     $v0, $s0, 15
xor     $a3, $a2
xor     $s0, $v0, $a1
xor     $a0, $t1, $a3
xor     $a0, $s0, $a0
multu   $a0, $t6
sll     $v0, $s1, 3
addu    $v0, $s0, $v0
sll     $v0, 8
and     $v0, $t7
srl     $v1, $s1, 6
xor     $s1, $v1, $v0
mfhi    $a1
srl     $a1, 3
sll     $v1, $a1, 3
sll     $v0, $a1, 5
subu    $v0, $v1
addu    $v0, $a1
subu    $a0, $v0
addiu   $a0, 0x61
sb      $a0, 0($t2)
addiu   $t2, 1
bne     $t2, $t5, L2_DOMAIN_GEN_LOOP
sll     $v0, $t1, 3
```

Fig-5, L2 domain generation loop

The TLD is determined by the residual value in register $S0 as shown in Fig-6. We can see that 3 TLDs are used here.


la $t9, malloc
nop
jalr $t9 ; malloc
li $a0, 0x32
move $s1, $v0
andi $v0, $s0, 1
lw $gp, 0x40+gp_($sp)
beqz $v0, loc_409830
nop
lui $v0, 0xAAAA
loc_409830:
la $a1, loc_410000
la $a3, loc_410000
la $t9, sprintf
addiu $a1, (aS_S - 0x410000) # "%s.%s"
addiu $a3, (aOnline - 0x410000) # "online"
move $a0, $s1
jalr $t9 ; sprintf
move $a2, $s2
lw $gp, 0x40+gp_($sp)
b loc_409798
lui $v0, 0xAAAA
End of function dga_gen_domain
loc_409798:
ori $v0, 0xAAAB
multu $s0, $v0
mfhi $v0
srl $v0, 1
sll $v1, $v0, 1
addu $v1, $v0
beq $s0, $v1, loc_409804
nop
andi $v0, $s0, 3
loc_409804:
la $a1, loc_410000
la $a3, loc_410000
la $t9, sprintf
addiu $a1, (aS_S - 0x410000) # "%s.%s"
addiu $a3, (aTech - 0x410000) # "tech"
move $a0, $s1
jalr $t9 ; sprintf
move $a2, $s2
lw $gp, 0x40+gp_($sp)
b loc_4097BC
andi $v0, $s0, 3

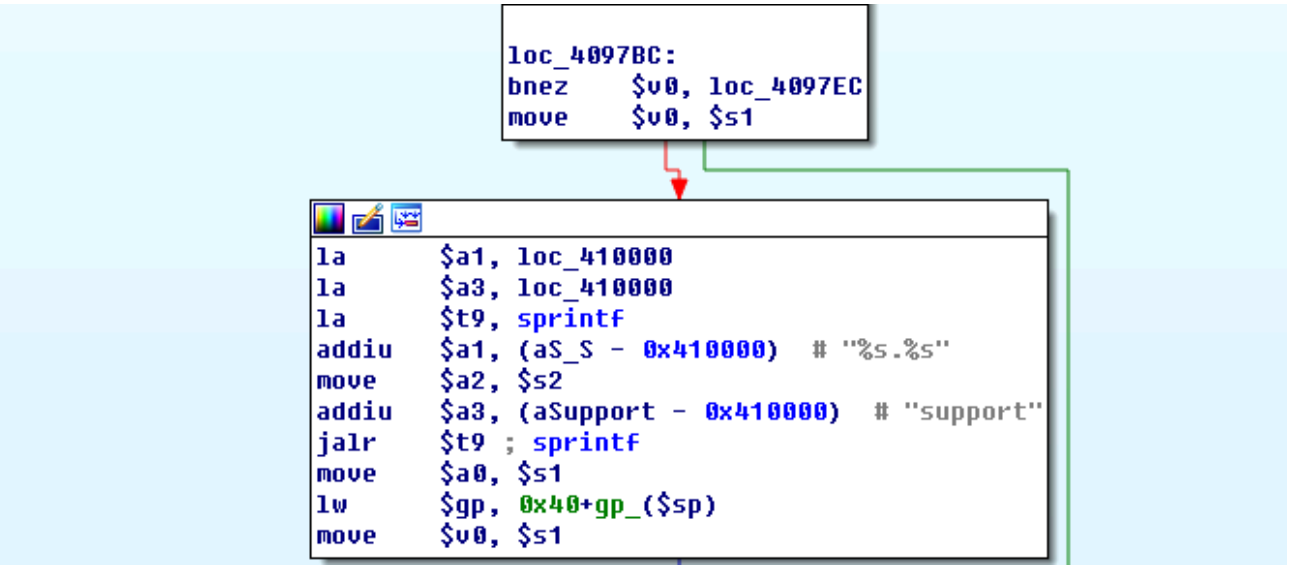


Fig-6, TLD determination

IoC

Currently the DGA feature is found in the following samples.

- 005241cf76d31673a752a76bb0ba7118
- 05891dbabc42a36f33c30535f0931555
- 0eb51d584712485300ad8e8126773941
- 15b35cfff4129b26c0f07bd4be462ba0
- 2da64ae2f8b1e8b75063760abfc94ecf
- 41ba9f3d13ce33526da52407e2f0589d
- 4a8145ae760385c1c000113a9ea00a3a
- 551380681560849cee3de36329ba4ed3
- 72bbfc1ff6621a278e16cfc91906109f
- 73f4312cc6f5067e505bc54c3b02b569
- 7d490eedc5b46aff00ffaaec7004e2a8
- 863dcf82883c885b0686dce747dcf502
- bf136fb3b350a96fd1003b8557bb758a
- bf650d39eb603d92973052ca80a4fdda
- d89b1be09de36e326611a2abbedb8751
- dbd92b08cbff8455ff76c453ff704dc6
- eba670256b816e2d11f107f629d08494

They all share the same DGA in terms of seed string and algorithm.
The hardcoded C2 domains in the samples are as follow:

- zugzwang.me
- tr069.online
- tr069.tech
- tr069.support

We will keep an eye on the progress of this DGA variant, stay tuned for future update.
