Simda Process Injection into Winlogon DGA Found

Published: 2019-08-24 · Archived: 2026-04-05 22:05:36 UTC

Overview:

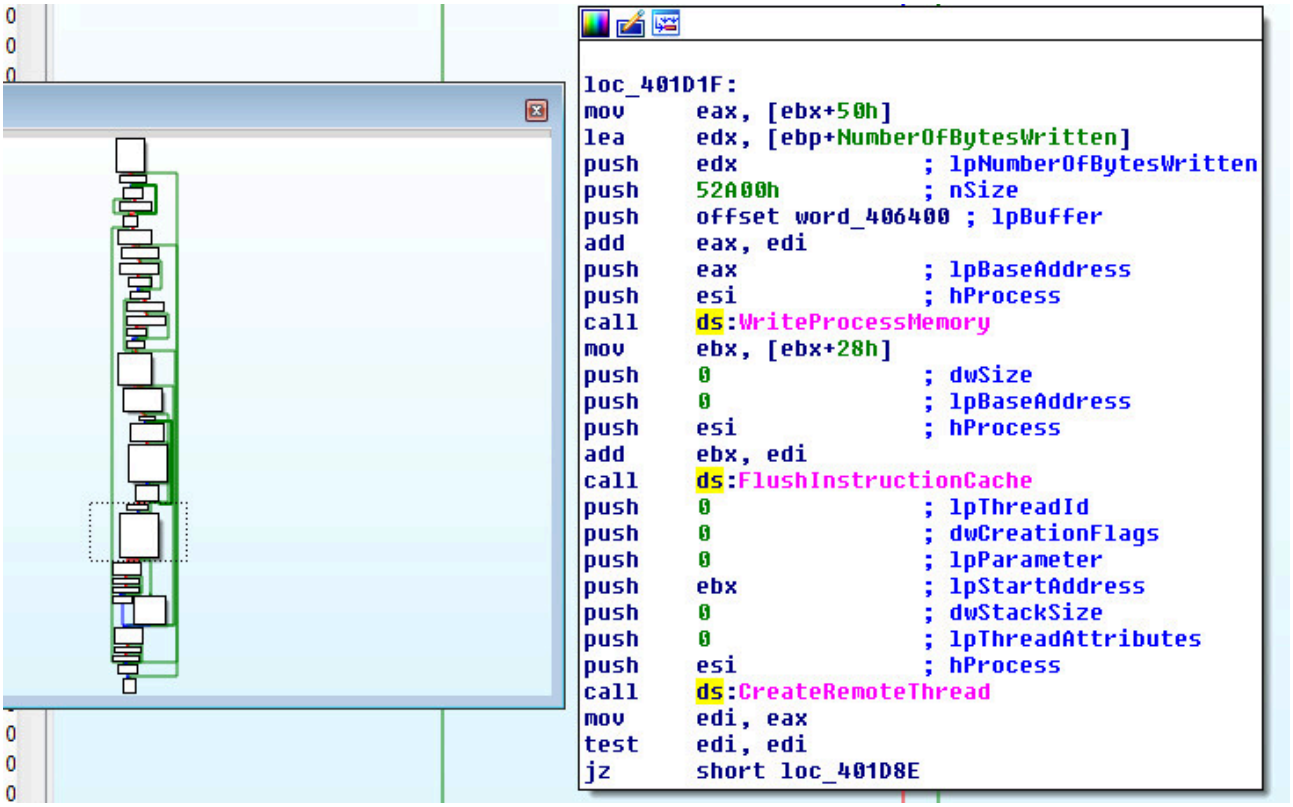
SonicWall Capture Labs Threat Research Team recently found a new sample and activity in August for Simda. Simda steals information and is capable of modifying websites through injection. Microsoft first detailed Simda long ago, the first use of the DGA was identified in 2012. However, the domains that are generated are active until the year 2106. The algorithm that generates the domain names uses an encrypted set of parameters describing how many characters the domain shall have and what TLD (Top Level Domain) to use. TLDs observed so far in this sample are ".com" only. However, other TLDs have been identified such as (.eu, .info, .com, .su, and .net).

Sample Static Information:

File Type	Portable Executable 32
File Info	UPX 2.90 [LZMA] (Delphi stub) -> Markus Oberhumer, Laszlo Molnar & John Reise
File Size	767.50 KB (785920 bytes)
PE Size	767.50 KB (785920 bytes)
Created	Friday 07 December 2018, 17.48.20
Modified	Friday 07 December 2018, 17.47.17
Accessed	Friday 07 December 2018, 17.48.20
MD5	FEE31E72C90535FC782BC8333CF4F95D
SHA-1	A3C843F323BEF582534AEEB40E27546EFB71CE46

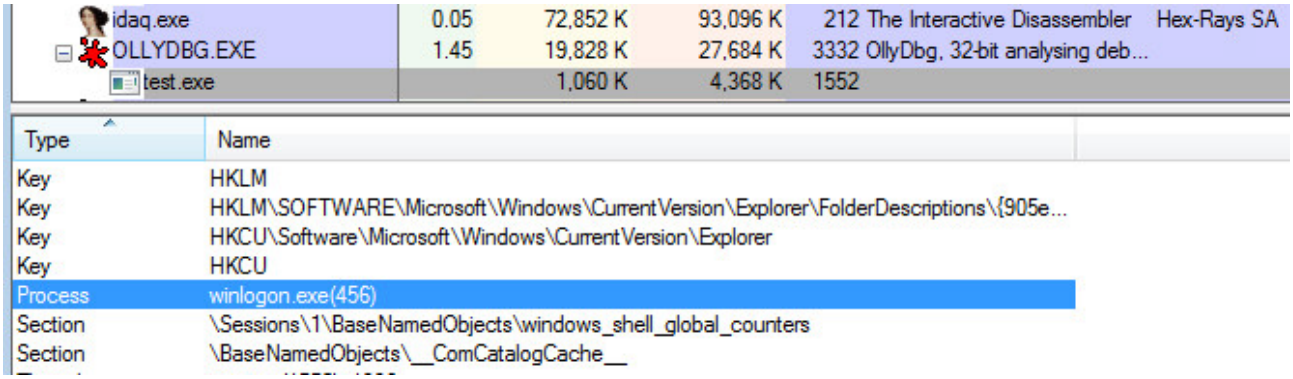
Process Injection:

Within Windows Operating Systems there are multiple approaches to injecting code into a live process. This particular sample uses Dynamic-Link Library (DLL) injection. This involves writing multiple components of the injection process into the remote process with an API named "WriteProcessMemory" and "CreateRemoteThread".



```
loc_401D1F:
mov     eax, [ebx+50h]
lea     edx, [ebp+NumberOfBytesWritten]
push    edx             ; lpNumberOfBytesWritten
push    52A00h          ; nSize
push    offset word_406400 ; lpBuffer
add     eax, edi
push    eax             ; lpBaseAddress
push    esi             ; hProcess
call    ds:WriteProcessMemory
mov     ebx, [ebx+28h]
push    0               ; dwSize
push    0               ; lpBaseAddress
push    esi             ; hProcess
add     ebx, edi
call    ds:FlushInstructionCache
push    0               ; lpThreadId
push    0               ; dwCreationFlags
push    0               ; lpParameter
push    ebx             ; lpStartAddress
push    0               ; dwStackSize
push    0               ; lpThreadAttributes
push    esi             ; hProcess
call    ds:CreateRemoteThread
mov     edi, eax
test    edi, edi
jz      short loc_401D8E
```

The remote process that will be supplying the code cave is called "Winlogon". Winlogon has multiple responsibilities: Window Station and desktop protection, Standard SAS recognition, SAS routine dispatching, User profile loading, Assignment of security to user shells, Screen Saver control, Multiple Network Provider Support. Winlogon is also responsible for loading the GINA libraries which are responsible for collecting logon credentials from the user.

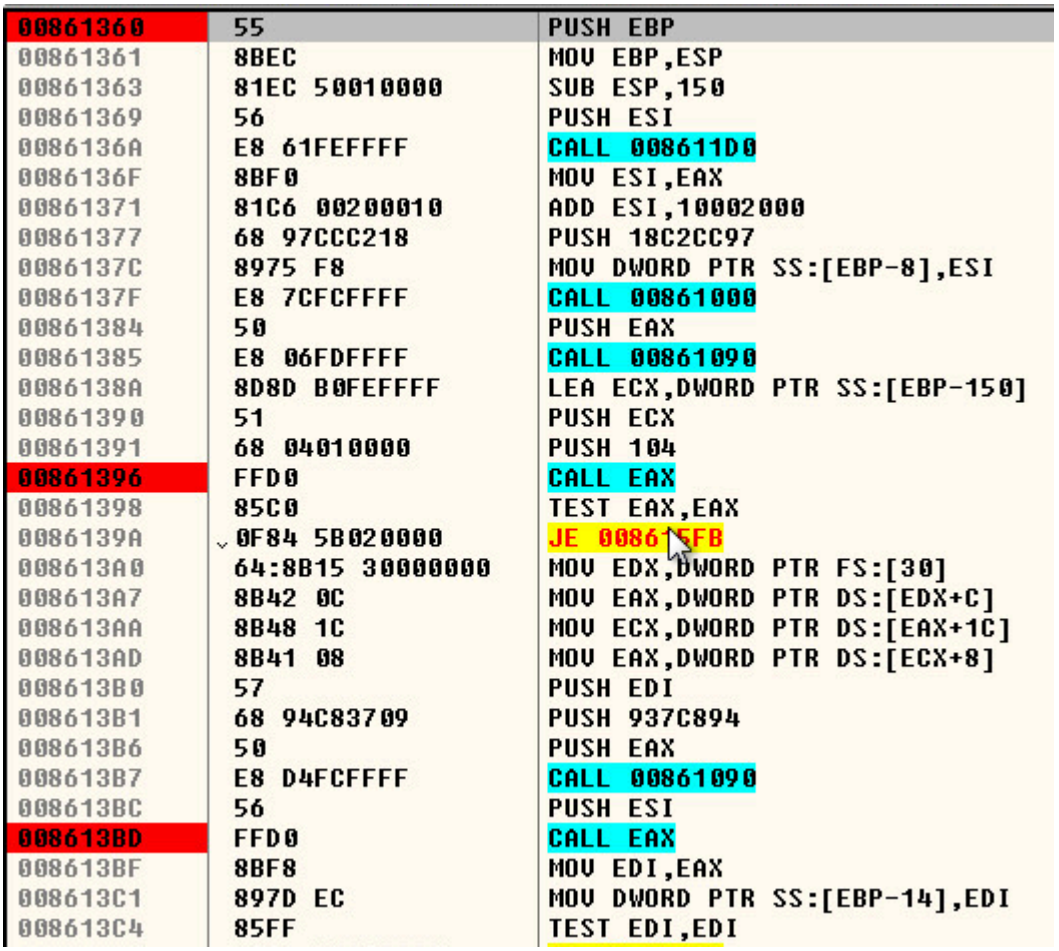


idaq.exe	0.05	72,852 K	93,096 K	212 The Interactive Disassembler	Hex-Rays SA
OLLYDBG.EXE	1.45	19,828 K	27,684 K	3332 OllyDbg, 32-bit analysing deb...	
test.exe		1,060 K	4,368 K	1552	

Type	Name
Key	HKLM
Key	HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\FolderDescriptions\{905e...
Key	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer
Key	HKCU
Process	winlogon.exe(456)
Section	\Sessions\1\BaseNamedObjects\windows_shell_global_counters
Section	\BaseNamedObjects__ComCatalogCache__

Code Cave with Stub aka ShellCode:

The code cave will call an array of Windows APIs to get the DLL loaded into the Winlogon process. Some of the APIs that are called are: RtlImageHeader, VirtualQuery, VirtualAlloc, GetModuleHandleA, LoadLibraryExA, and SetCurrentDirectoryA. The DLL that will be loaded is called "WinSCard.dll".



Here is what the code cave looks like in Ida Pro as a memory dump:

```
v0 = sub_11D0();
v1 = v0 + 0x10002000;
v60 = (char *)(v0 + 0x10002000);
v2 = sub_1000();
v3 = (int (__stdcall *)(signed int, char *))sub_1090(v2, 0x18C2CC97);
if ( v3(260, &v42) )
{
  v4 = (int (__stdcall *)(int))sub_1090(
                                 *(_DWORD *)(*(_DWORD *)(*(_DWORD *)(__readfsdword(0x30) + 12) + 28) + 8),
                                 154650772);
  v5 = v4(v1);
  v57 = v5;
  if ( v5 )
  {
    v43 = 0;
    v44 = 0;
    v45 = 0;
    v46 = 0;
    v47 = 0;
    v48 = 0;
    v49 = 0;
    v6 = sub_11D0() + 268440176;
    v7 = sub_1000();
    v8 = (void (__stdcall *)(int, int *, signed int))sub_1090(v7, 905776733);
    v8(v6, &v43, 28);
    v9 = *(_DWORD *)(v5 + 80) + *(_DWORD *)(*(_DWORD *)(v44 + 60) + v44 + 88);
    v10 = sub_1000();
    v11 = (int (__stdcall *)(_DWORD, int, signed int, signed int))sub_1090(v10, 1644136010);
    v12 = (char *)v11(0, v9, 12288, 64);
    v59 = v12;
    if ( v12 )
    {
      v58 = *(char **)(v5 + 84);
      qmemcpy(v59, v60, (unsigned int)v58);
      v13 = (_DWORD *)v57;
      v14 = *(unsigned __int16 *)(v57 + 6);
      v15 = (_DWORD *)(*(unsigned __int16 *)(v57 + 20) + v57 + 44);
      if ( v14 > 0 )
```

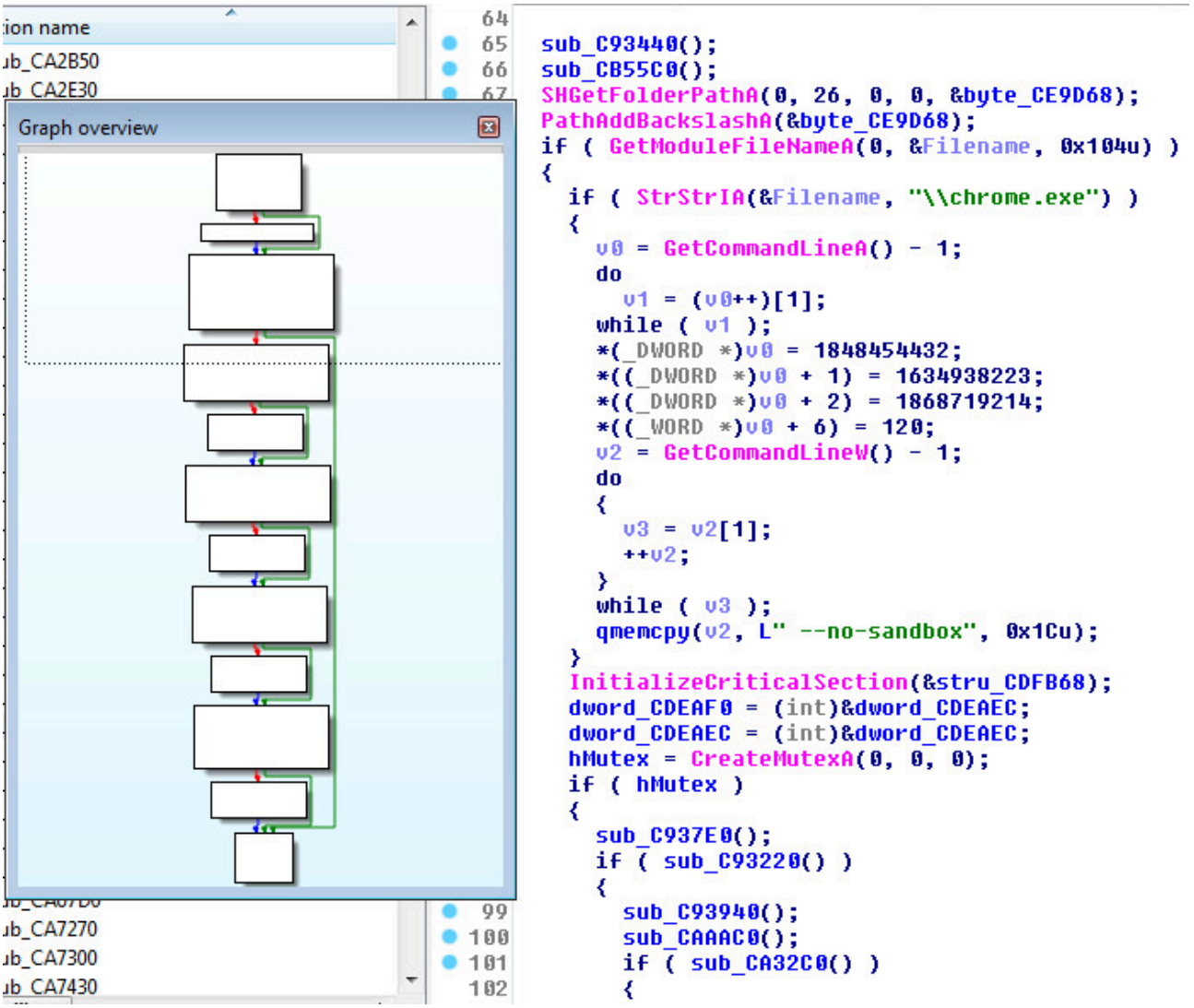
DLL Injection:

The "WinSCard.dll" dumped into Ida Pro. This shows the typical dll injection. It just calls one thread and executes everything in it.

```
BOOL __stdcall __noreturn DllEntryPoint(HINSTANCE hinstDLL, DWORD fdwReason, LPVOID lpReserved)
{
  HANDLE v3; // esi@1
  DWORD dwFlags; // [esp+4h] [ebp-4h]@2

  v3 = CreateThread(0, 0, sub_CA67D0, 0, 0, 0);
  if ( v3 )
  {
    dwFlags = 0;
    if ( GetHandleInformation(v3, &dwFlags) )
    {
      if ( !(dwFlags & 2) )
        CloseHandle(v3);
    }
  }
  ExitThread(0);
}
```

This is the top of the thread that gets called in DLL Main:



DGA (Domain Generation Algorithm) Found:

Adversaries may make use of Domain Generation Alogirthms (DGAs) to dynamically identify a destination for command and control traffic rather than relying on a list of static IP addresses or domains. This has an advantage of making it harder for defenders to block, track, or take over the command and control channel. This sample makes use of the following DGA:

```
67  i = 0x45AE94B2;
68  key = "1676d5775e05c50b46baa5579d4fc7";
69  do
70  {
71    ++key;
72    v0 += v1;
73    v1 = *key;
74  }
75  while ( *key );
76  dword_CDE610 = v0;
77  v3 = 0;
78  v4 = &v26;
79  do
80  {
81    v5 = a1676d5775e05c5[v3 + 1];
82    v42 = a1676d5775e05c5[v3];
83    v43 = v5;
84    *v4 = strtol(&v42, 0, 16);
85    v3 += 2;
86    ++v4;
87  }
88  while ( v3 < 0x1E );
89  v6 = 0;
90  v30 = 1;
91  v7 = 0;
92  do
93  {
94    *(&v28 + v7) = v7;
95    ++v7;
96  }
97  while ( v7 < 256 );
```

A little lower down in the same function:

```
if ( v13 > 0 )
{
  v16 = 1 - (_DWORD)&v24;
  for ( i = 1 - (_DWORD)&v24; ; v16 = i )
  {
    v17 = &v24 + v14;
    v19 = (unsigned int)(&v24 + v14 + v16) & 0x80000001;
    v18 = v19 == 0;
    if ( v19 < 0 )
      v18 = (((_BYTE)v19 - 1) | 0xFFFFFFFE) == -1;
    v20 = v18 ? *((_BYTE *)&v38 + dword_CDE610 / v15 % 6) : *((_BYTE *)&v31 + dword_CDE610 / v15 % 0x14);
    ++v14;
    v15 += 2;
    *v17 = v20;
    if ( v14 >= v37 )
      break;
  }
}
```

Reversing the DGA into C/C++:

```
for (int k = 0; k < 1000; k++)
{
    base += step;
    for (int j = 0; j < length; j++)
    {
        index = base / (3 + 2 * j);
        if ((j % 2) == 0)
            c = constants[index % 20];
        else
            c = vowels[index % 6];

        domain[j] = c;
    }
```

A small list from the DGA output is as follows:

A	B	C	D
gatyfus.com	qetyvep.com	lyrysor.com	puryxuq.com
lyvyxor.com	puvytuq.com	vocykem.com	gacyqob.com
vojyqem.com	gahyhob.com	qegynuv.com	lygyfex.com
qetyfuv.com	lyryvex.com	purypol.com	vowyzuk.com
puvyxil.com	vocyruk.com	gacykeh.com	qexyqog.com
gahyqah.com	qegyhig.com	lygynud.com	pufydep.com
lyryfyd.com	purycap.com	vowypit.com	gaqyzuw.com
vocyzit.com	gacyryw.com	qexykaq.com	lyxymin.com
qegyqaq.com	lygygin.com	pufybyv.com	vofydac.com
purydyv.com	vowycac.com	gaqypiz.com	qeqylyl.com
gacyzuz.com	qexyryl.com	lyxyjaj.com	puzymig.com
lygymoj.com	pufygug.com	vofybyf.com	gadydas.com
vowydef.com	gaqycos.com	qeqytup.com	lymylyr.com
qexylup.com	lyxywer.com	puzyjoq.com	volymum.com
pufymoq.com	vofygum.com	gadyveb.com	qedysov.com
gaqydeb.com	qeqyxov.com	lymytux.com	pumylel.com
lyxylux.com	puzywel.com	volyjok.com	galynuh.com
vofymik.com	gadyfuh.com	qedyveg.com	lysysod.com
qeqysag.com	lymyxid.com	pumytup.com	vonyket.com
puzylyp.com	volyqat.com	galyhiw.com	qekynuq.com
gadyniw.com	qedyfyq.com	lysyvan.com	pupypiv.com
lymysan.com	pumyxiv.com	vonyryc.com	ganykaz.com
volykyc.com	galyqaz.com	qekyhil.com	lykynyj.com
qedynul.com	lysyfyj.com	pupycag.com	vopypif.com
pumypog.com	vonyzuf.com	ganyrys.com	qebykap.com
galykes.com	qekyqop.com	lykygur.com	pujybyq.com
lysynur.com	pupydeq.com	vopycom.com	gatypub.com
vonypom.com	ganyzub.com	qebyrev.com	lyvyjox.com
qekykev.com	lykymox.com	pujygul.com	vojybek.com
pupybul.com	vopydek.com	gatycoh.com	qetytug.com
ganypih.com	qebylug.com	lyvywed.com	puvyjop.com
lykyjad.com	pujymip.com	vojygut.com	gahyvew.com
vopybyt.com	gatydaw.com	qetyxiq.com	lyrytun.com
qebytiq.com	lyvylyn.com	puvywav.com	vocyjic.com
pujyjav.com	vojymic.com	gahyfyz.com	qegyval.com
gatyvyz.com	qetysal.com	lyryxij.com	purytyg.com
lyvytuj.com	puvylyg.com	vocyqaf.com	gacyhis.com
vojyjof.com	gahynus.com	qegyfyp.com	lygyvar.com

vowyrym.com	gaqykab.com	qeqyqiv.com	lymygyx.com
qexyhuv.com	lyxynyx.com	puzydal.com	volycik.com
pufycol.com	vofypuk.com	gadyzyh.com	qedyrag.com
gaqyreh.com	qeqykog.com	lymymud.com	pumygyp.com
lyxygud.com	puzybep.com	volydot.com	galycuw.com
vofycot.com	gadypuw.com	qedyleq.com	lysywon.com
qeqyreq.com	lymyjon.com	pumymuv.com	vonygec.com
puzyguv.com	volybec.com	galydoz.com	qekyxul.com
gadyciz.com	qedytul.com	lysylej.com	pupywog.com
lymywaj.com	pumyjig.com	vonymuf.com	ganyfes.com
volygyf.com	galyvas.com	qekysip.com	lykyxur.com
qedyxip.com	lysytyr.com	pupylaq.com	vopyqim.com
pumywaq.com	vonyjim.com	ganynyb.com	qebyfav.com
galyfyb.com	qekyvav.com	lykysix.com	pujyxyl.com
lysyxux.com	pupytyl.com	vopykak.com	gatyqih.com
vonyqok.com	ganyhuh.com	qebynyg.com	lyvyfad.com
qekyfeg.com	lykyvod.com	pujypup.com	vojyzyt.com
pupyxup.com	vopyret.com	gatykow.com	qetyquq.com
ganyqow.com	qebyhuq.com	lyvynen.com	puvydov.com
lykyfen.com	pujycov.com	vojypuc.com	gahyzez.com
vopyzuc.com	gatyrez.com	qetykol.com	lyrymuj.com
qebyqil.com	lyvyguj.com	puvybeg.com	vocydof.com
pujydag.com	vojycif.com	gahypus.com	qegylep.com
gatyzys.com	qetyrap.com	lyryjir.com	purymuq.com
lyvymir.com	puvygyq.com	vocybam.com	gacydib.com
vojydam.com	gahycib.com	qegytyv.com	lygylax.com
qetylyv.com	lyrywax.com	puryjil.com	vowymyk.com
puvymul.com	vocygyk.com	gacyvah.com	qexysig.com
gahydoh.com	qegyxug.com	lygytyd.com	pufylap.com
lyryled.com	purywop.com	vowyjut.com	gaqynyw.com
vocymut.com	gacyfew.com	qexyvoq.com	lyxysun.com
qegysoq.com	lygyxun.com	pufytev.com	vofykoc.com
purylev.com	vowyqoc.com	gaqyhuz.com	qeqynel.com
gacynuz.com	qexyfel.com	lyxyvoj.com	puzypug.com
lygysij.com	pufyxug.com	vofyref.com	gadykos.com
vowykaf.com	gaqyqis.com	qeqyhup.com	lymyner.com
qexynyp.com	lyxyfar.com	puzyciq.com	volypum.com
pufypiq.com	vofyzym.com	gadyrab.com	qedykiv.com

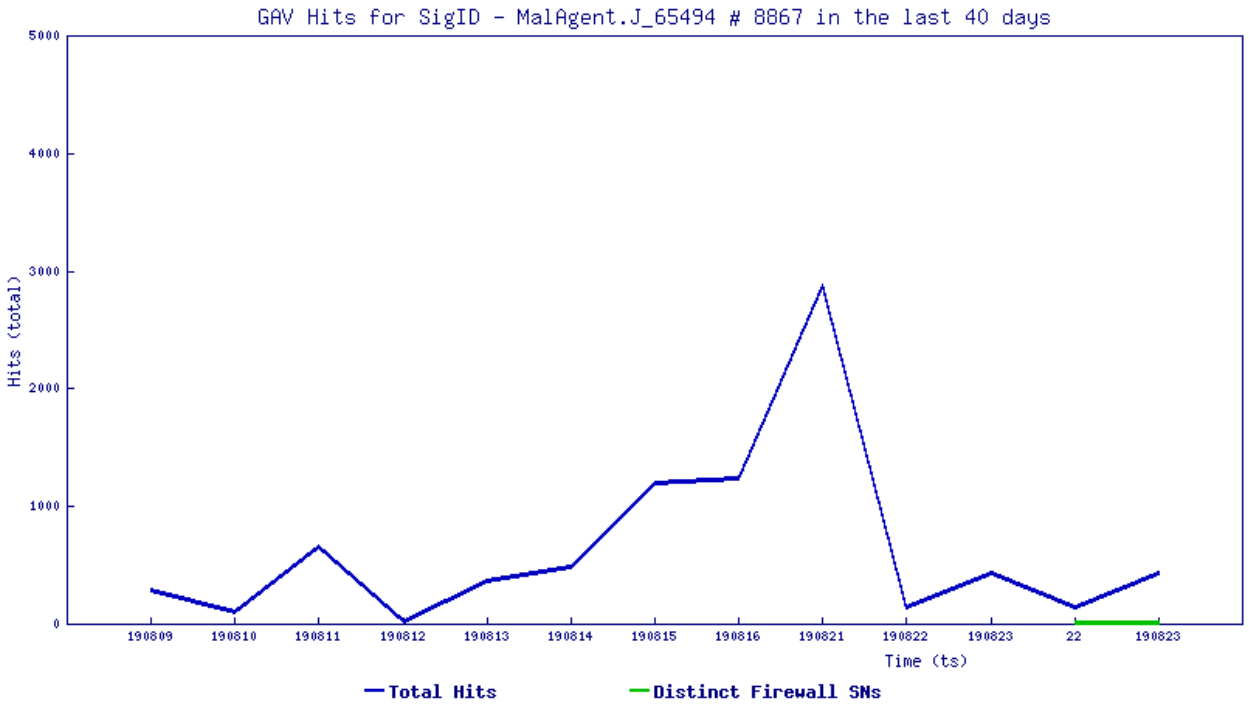
pumybal.com	vonydik.com	ganycuh.com	qebyteg.com
galypyh.com	qekylag.com	lykywid.com	pujyjup.com
lysyjid.com	pupymyp.com	vopygat.com	gatyviw.com
vonybat.com	ganydiw.com	qebyxyq.com	lyvytan.com
qekytyq.com	lykylan.com	pujywiv.com	vojyjyc.com
pupyjuv.com	vopymyc.com	gatyfaz.com	qetyvil.com
ganyvoz.com	qebysul.com	lyvyxyj.com	puvytag.com
lykytej.com	pujylog.com	vojyquf.com	gahyhys.com
vopyjuf.com	gatynes.com	qetyfop.com	lyryvur.com
qebyvop.com	lyvysur.com	puvyxeq.com	vocyrom.com
pujyteq.com	vojykom.com	gahyqub.com	qegyhev.com
gatyhub.com	qetynev.com	lyryfox.com	purycul.com
lyvyvix.com	puvypul.com	vocyzek.com	gacyroh.com
vojyrak.com	gahykih.com	qegyqug.com	lygyged.com
qetyhyg.com	yrynad.com	purydip.com	vowycut.com
puvycip.com	vocypyt.com	gacyzaw.com	qexyriq.com
gahyraw.com	qegykiq.com	lygymyn.com	pufygav.com
lyrygyn.com	purybav.com	vowydic.com	gaqycyz.com
vocycuc.com	gacypyz.com	qexylal.com	lyxywij.com
qegyrol.com	lygyjuj.com	pufymyg.com	vofygaf.com
purygeg.com	vowybof.com	gaqydus.com	qeqyxyp.com
gacycus.com	qexytep.com	lyxylor.com	puzywuq.com
lygywor.com	pufyjuq.com	vofymem.com	gadyfob.com
vowygem.com	gaqyvob.com	qeqysuv.com	lymyxex.com
qexyxuv.com	lyxytex.com	puzylol.com	volyquk.com
pufywil.com	vofyjuk.com	gadyneh.com	qedyfog.com
gaqyfah.com	qeqyvig.com	lymysud.com	pumyxep.com
lyxyxyd.com	puzytap.com	volykit.com	galyquw.com
vofyqit.com	gadyhyw.com	qedynaq.com	lysyfin.com
qeqyfaq.com	lymyvin.com	pumypyv.com	vonyzac.com
puzyxyv.com	volyrac.com	galykiz.com	qekyqyl.com
gadyquz.com	qedyhyl.com	lysynaj.com	pupydig.com
lymyfoj.com	pumycug.com	vonypyf.com	ganyzas.com
volyzef.com	galyros.com	qekykup.com	lykymyr.com
qedyqup.com	lysyger.com	pupyboq.com	vopydum.com
pumydoq.com	vonycum.com	ganypeb.com	qebylov.com
galyzeb.com	qekyrov.com	lykyjux.com	pujymel.com
lysymux.com	pupygel.com	vopybok.com	gatyduh.com

The Domain Generation Algorithm will produce 1,000 active Domains. The domains will be active until the year 2106.

SonicWall, (GAV) Gateway Anti-Virus, provides protection against this threat:

- GAV: Simda.S
- GAV: MalAgent.J_65494

Daily Stats | Hourly Stats | Slawek's Stats



Source: https://www.sonicwall.com/blog/simda-process-injection-into-winlogon-dga-found