Naikon’s Aria

securelist.com/naikons-aria/96899/

Authors

Expert GReAT

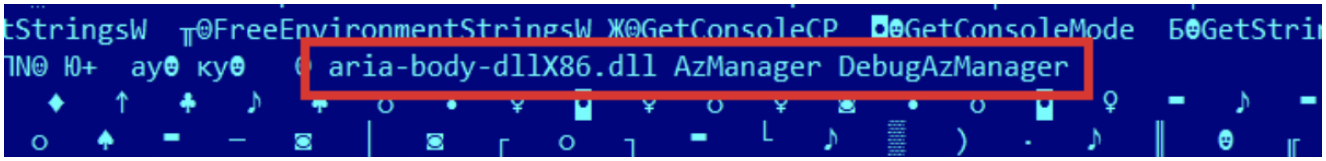
Our colleagues at Checkpoint put together a fine research writeup on some Naikon resources and activity related to “aria-body” that we detected in 2017 and similarly reported in 2018. To supplement their research findings, we are summarizing and publishing portions of the findings reported in our June 2018 “Naikon’s New AR Backdoor Deployment to Southeast Asia”. This malware and activity aligns with much of what the Checkpoint researchers brought to light today.

The Naikon APT became well-known in May 2015, when our public reporting first mentioned and then fully described the group as a long running presence in the APAC region. Even when the group shutdown much of their successful offensive activity after years of campaigns, Naikon maintained several splinter campaigns. Matching malware artifacts, functionality, and targeting demonstrates that the group continues to wage cyber-espionage campaigns in the South China Sea region during 2018.

“Aria-Body” or “AR” is a set of backdoors that maintain compilation dates between January 2017 and February 2018. It can be particularly difficult to detect, as much of this code operates in memory, injected by other loader components without touching disk. We trace portions of this codebase back to “xsFunction” exe and dll modules used in Naikon operations going back to 2012, as their compiled modules implement a subset of the xsFunction feature set. In all likelihood, this new backdoor and related activity is an extension of or merge with the group’s “Paradir Operation”. In the past, the group targeted communications and sensitive information from executive and legislative offices, law enforcement, government administrative, military and intelligence organizations within Southeast Asia. In many cases we have seen that these systems also were targeted previously with PlugX and other malware. So, the group has evolved bit since 2015, and their activity targeting these same profiles continues into 2018. We identified at least a half dozen individual variants from 2017 and 2018.

Technical Details

It seems clear that the same codebase has been reused by Naikon since at least 2012, and recent AR backdoors were built from that same code. Their use was tightly clustered in previously and heavily Naikon-targeted organizations, again lending confidence to clustering these resources and activity with previous “Naikon”.



Naikon’s new AR backdoor is a dll loaded into any one of multiple processes, providing remote access to a system. AR load attempts have been identified within processes with executable images listed here:

- c:\windows\system32\svchost.exe
- c:\windows\syswow64\svchost.exe
- c:\program files\windows nt\accessories\services.exe
- c:\users\dell\appdata\roaming\microsoft\windows\start menu\programs\startup\acrobat.exe
- c:\alphazawgyi\svchost.exe

Because this AR code is injected into processes, the yara rule provided in the Appendix is best run against memory dumps of processes maintaining a main image in the list above. The AR modules have additionally been seen in some others, including “msiexec.exe” processes.

Below are characteristics of the oldest AR and the newest known AR component in our collection.

MD5	c766e55c48a4b2e7f83bfb8b6004fc51
SHA256	357c8825b3f03414582715681e4e0316859b17e702a6d2c8ea9eb0fd467620a4
CompiledOn	Tue Jan 3 09:23:48 2017
Type	PE32 DLL
Internal name	TCPx86.dll
Size	176kb
Exports	AzManager, DebugAzManager

MD5	2ce4d68a120d76e703298f27073e1682
SHA256	4cab6bf0b63cea04c4a44af1cf25e214771c4220ed48fff5fca834efa117e5db
CompiledOn	Thu Feb 22 10:04:02 2018
Type	PE32 DLL
Internal Name	aria-body-dllX86.dll
Size	204kb
Exports	AzManager, DebugAzManager

When the dll is loaded, it registers a Windows class calling a specific Window procedure with a removable drive check, a CONNECT proxied callback to its main C2, an IP location verification against checkip.amazonaws[.]com, and further communications with a C2. Some previous modules' flow may include more or less system information collection prior to the initial callback.

The most recent version of the backdoor utilizes another Window procedure to implement a raw input device based keystroke collector. This keylogger functionality was newly introduced to the malware code in February 2018, and was not present in previous versions.

The approximately 200 – 250kb AR backdoor family provides a familiar and slightly changing functionality set per compiled module. Because Checkpoint covers the same technical points in their post, we provide this simple summary list:

- Persistence handling
- File and directory handling
- Keylogging
- Shell/Process Management

- Network activity and status listing and management
- System information collection and management
- Download management
- Windows management
- Extension management
- Location/IP verification
- Network Communications over HTTP

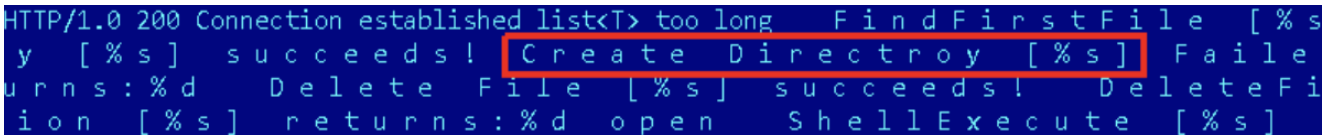
Similarities to past Naikon components

Naikon components going back to 2012 maintain heavy similarities with the current “Aria-body” modules. Not only is some of the functionality only lightly modified, but the same misspellings in error logging remains in their codebase. Let’s examine an older 2013 Naikon module and a newer 2017 Naikon AR module here.

It’s clear that the underlying codebase continues to be deployed:

e09254fa4398fccd607358b24b918b63, CompiledOn: 2013:09:10 09:00:15

c766e55c48a4b2e7f83bfb8b6004fc51, CompiledOn: 2017:01:03 09:23:48



Kudos to the Checkpoint researchers for providing new details of the Naikon story into the public discussion.

For reference, some hashes and a YARA rule are provided here. More incident, infrastructure, IOCs, and details have been and are available to our threat intel customers (please, contact intelreports@kaspersky.com).

Indicators of compromise

AR aria-body dll
c766e55c48a4b2e7f83bfb8b6004fc51
2ce4d68a120d76e703298f27073e1682

Loaders and related Naikon malware
0ed1fa2720cdab23d969e60035f05d92
3516960dd711b668783ada34286507b9

Verdicts – 2018 and Later
Trojan.Win32.Generic.gen
Trojan.Win32.SEPEH.gen
DangerousObject.Multi.Generic
Backdoor.Win64.Agent.h*
Backdoor.Win32.Agent.m*
Trojan-Downloader.Win32.Agent.x*

YARA Rules

```
1   rule apt_ZZ_Naikon_ARstrings : Naikon
2   {
3       meta:
4           copyright = "Kaspersky"
5           description = "Rule to detect Naikon aria samples"
6           hash = "2B4D3AD32C23BD492EA945EB8E59B758"
7           date = "2020-05-07"
8           version = "1.0"
9
10      strings:
11          $a1 = "Terminate Process [PID=%d] succeeds!" fullword wide
12          $a2 = "TerminateProcess [PID=%d] Failed:%d" fullword wide
13          $a3 = "Close tcp connection returns: %d!" fullword wide
14          $a4 = "Delete Directory [%s] returns:%d" fullword wide
15          $a5 = "Delete Directory [%s] succeeds!" fullword wide
16          $a6 = "Create Directory [%s] succeeds!" fullword wide
17          $a7 = "SHFileOperation [%s] returns:%d" fullword wide
18          $a8 = "SHFileOperation [%s] succeeds!" fullword wide
19          $a9 = "Close tcp connection succeeds!" fullword wide
```

```
20          $a10 = "OpenProcess [PID=%d] Failed:%d" fullword wide
21          $a11 = "ShellExecute [%s] returns:%d" fullword wide
22          $a12 = "ShellExecute [%s] succeeds!" fullword wide
23          $a13 = "FindFirstFile [%s] Error:%d" fullword wide
24          $a14 = "Delete File [%s] succeeds!" fullword wide
25          $a15 = "CreateFile [%s] Error:%d" fullword wide
26          $a16 = "DebugAzManager" fullword ascii
27          $a17 = "Create Directroy [%s] Failed:%d" fullword wide
28
29          $m1 = "TCPx86.dll" fullword wide ascii
30          $m2 = "aria-body" nocase wide ascii
31
32      condition:
33          uint16(0) == 0x5A4D and
34          filesize &lt; 450000 and
35          (2 of ($a*) and 1 of ($m*))
36  }
```

```
rule apt_ZZ_Naikon_codebase : Naikon
{
    meta:
        report = "Naikon New AR Backdoor Deployment to Southeast Asia"
        description = "Naikon typo"
        author = "Kaspersky"
        copyright = "Kaspersky"
        version = "1.0"
        date = "2018-06-28"
        last_modified = "2018-06-28"

    strings:
        $a1 = "Create Directroy [%s] Failed:%d" wide

    condition:
        uint16(0) == 0x5A4D and
        filesize &lt; 450000 and
        $a1
}
```

- APT
- Backdoor
- Cyber espionage
- Malware Technologies

Authors

GReAT

Naikon’s Aria

Your email address will not be published. Required fields are marked *