

Analysis of Andariel's New Attack Activities

ASEC asec.ahnlab.com/en/56405/

By Sanseo

August 31, 2023



Contents

1. Past attack cases
 - 1.1. Cases of Innorix Agent abuse
 - 1.1.1. NukeSped variant – Volgmer
 - 1.1.2. Andardoor
 - 1.1.3. 1th Troy Reverse Shell
 - 1.2. Cases of attacks against Korean corporations
 - 1.2.1. TigerRat
 - 1.2.2. Black RAT
 - 1.2.3. NukeSped variants
2. Cases of recent attacks
 - 2.1. Cases of Innorix Agent abuse
 - 2.1.1. Goat RAT
 - 2.2. Cases of attacks against Korean corporations
 - 2.2.1. AndarLoader
 - 2.2.2. DurianBeacon
3. Connections to recent attack cases
4. Connections to past attack cases of the Andariel group
5. Conclusion

The Andariel threat group which usually targets Korean corporations and organizations is known to be affiliated with the Lazarus threat group or one of its subsidiaries. Attacks against Korean targets have been identified since 2008. Major target industries are those related to

national security such as national defense, political organizations, shipbuilding, energy, and communications. Various other companies and institutes in Korea including universities, logistics, and ICT companies are also becoming attack targets. [1] (this report only supports the Korean version)

During the initial compromise stage, the Andariel threat group usually employs spear phishing, watering hole, and supply chain attacks. Additionally, there are cases where the group abuses central management solutions during the malware installation process. [2] A notable fact about the group is its creation and use of various malware types in its attacks. There are many backdoor types, including Andarat, Andaratm, Phandoor, and Rifdoor used in the past attacks, as well as TigerRAT [3] and MagicRAT [4] which have been detected for the past few years.

AhnLab Security Emergency response Center (ASEC) is continuously monitoring the attacks of the Andariel threat group. This blog post will cover details surrounding the recently identified attacks deemed to be perpetrated by the Andariel group. Note that because the malware strains and C&C servers identified in past attack cases were not used in the aforementioned attacks, there is no direct connection. Thus, in order to identify the connection between the recent attacks and the Andariel threat group, this post will first analyze the cases of attacks by the Andariel group in the first half of 2023. Then the analysis will be used to identify the possible link between the attacks and the threat group. Details confirmed in the past attack cases will be included if necessary.

One characteristic of the attacks identified in 2023 is that there are numerous malware strains developed in the Go language. In an attack case where Innorix Agent was used, a Reverse Shell developed in Go was used. Black RAT was used in attacks targeting Korean companies afterward. Such trends continued into the recent cases, where other malware strains developed in Go such as Goat RAT and DurianBeacon are being used in attacks. Besides the Go version, DurianBeacon has a version developed in the Rust language as well.

```
.rdata:00000000006CFED3 aGDevGoDurianbe db 'G:/Dev/Go/DurianBeacon/Command.go',0
.rdata:00000000006CFED3 ; DATA XREF: .rdata:00000000006CAA64↑o
.rdata:00000000006CFEF5 aGDevGoDurianbe_0 db 'G:/Dev/Go/DurianBeacon/SSL.go',0
.rdata:00000000006CFEF5 ; DATA XREF: .rdata:00000000006CAAD8↑o
.rdata:00000000006CFF13 aGDevGoDurianbe_1 db 'G:/Dev/Go/DurianBeacon/Utils.go',0
.rdata:00000000006CFF13 ; DATA XREF: .rdata:00000000006CAB94↑o
.rdata:00000000006CFF33 aGDevGoDurianbe_2 db 'G:/Dev/Go/DurianBeacon/main.go',0
```



Figure 1. Source code information of DurianBeacon developed in Go

Because the initial distribution case could not be identified directly, this post will conduct an analysis based on the malware strains used in the attacks. Note that various malware types are being used in the attacks. When a name given by the malware creator can be confirmed, the said name will be used. If not, the names of similar malware types or AhnLab's detection name will be used.

1. Past attack cases

1.1. Cases of Innorix Agent abuse

In February 2023, ASEC shared the case where the Andariel threat group distributed malware to users with a vulnerable version of Innorix Agent in the blog post “Distribution of Malware Exploiting Vulnerable Innorix: Andariel.” [5] The Innorix Agent program abused in distribution is a file transfer solution client program. According to the post regarding the vulnerability by the Korea Internet & Security Agency (KISA), the affected versions were found to be INNORIX Agent 9.2.18.450 or earlier, which were advised to be applied with the security update. [6] (this content only supports the Korean version)

Target Type	File Name	File Size	File Path ⓘ
Current	 innorixas.exe	8.17 MB	%SystemDrive%\innorix_agent\innorixas.exe
Target	 msdes.exe.irx	40.5 KB	%SystemDrive%\users\%ASD%\msdes.exe.irx




Process	Module	Target	Data
 innorixas.exe	N/A	N/A	 msdes.exe.irx
 innorixas.exe	N/A	N/A	http://4.246.144.112/update.exe

Figure 2. Malware being distributed using Innorix Agent which had been vulnerable in the past

An investigation of the malware strains used in the attacks based on past attack cases revealed that multiple Korean universities were infected with malware strains. Most malware types used in the attacks were backdoors, and no previously identified type was present. However, because there is a connection with other malware strains used in the past or those used in subsequent attacks, a brief summary of their characteristics will be given.

1.1.1. NukeSped variant – Volgmer

As covered in the ASEC Blog before, this malware strain uses the following 0x10 byte key in the process of communicating with the C&C server to encrypt packets. The key value in question is the same as the one employed in Volgmer used by the Hidden Cobra (Lazarus) threat group, as stated in a report by the United States Cybersecurity & Infrastructure Security Agency (CISA). [7] (page currently unavailable)

Key: 74 61 51 04 77 32 54 45 89 95 12 52 12 02 32 73

Volgmer was also used in comparatively recent attacks. It runs by reading the configuration data saved in the registry key “HKLM\SYSTEM\CurrentControlSet\Control\WMI\Security” and uses the HTTP protocol to communicate with the C&C server. Such characteristics are highly similar to the type mentioned in the CISA report in the past, which means that the malware

continues to be used in attacks with no significant variants being released. While the same key value was used in both the malware mentioned in this post and Volgmer, there is a difference: the malware used in the current attack cases uses the key value to encrypt the packets used to communicate with the C&C server. Meanwhile, Volgmer uses the value to decrypt the encrypted configuration data saved in the registry.

Accordingly, it is not entirely accurate to categorize the above malware strain as a type of Volgmer, so it was categorized as a variant of NukeSped instead. The malware is a comparatively simple backdoor that only provides basic features. Notably, the Batch script used in the self-deletion process is similar to the one used in NukeSped types in the past.

```
.rdata:000000014001E248 aHelloServer db 'Hello Server' ; DATA XREF: fn_initComm+29↑r
.rdata:000000014001E248 ; fn_initComm+3A↑r
.rdata:000000014001E254 byte_14001E254 db 0 ; DATA XREF: fn_initComm+43↑r
.rdata:000000014001E255 align 8
.rdata:000000014001E258 ; const char Str2[]
.rdata:000000014001E258 Str2 db 'Hello Client',0 ; DATA XREF: fn_initComm:loc_14000143A↑o
.rdata:000000014001E265 align 8
.rdata:000000014001E268 ; const char Source[]
.rdata:000000014001E268 Source db 'uninstall.bat',0 ; DATA XREF: sub_1400017A0+41↑o
.rdata:000000014001E276 align 8
.rdata:000000014001E278 ; const char data_SelfDelBat[]
.rdata:000000014001E278 data_SelfDelBat db ':L1',0Dh,0Ah ; DATA XREF: sub_1400017A0+B1↑o
.rdata:000000014001E27D db 'del /F "%s"',0Dh,0Ah
.rdata:000000014001E28A db 'if exist "%s" goto L1',0Dh,0Ah
.rdata:000000014001E2A1 db 'del /F "%s"',0Dh,0Ah,0
.rdata:000000014001E2AF align 10h
```

Figure 3. Batch script used in the self-deletion process

1.1.2. Andardoor

Developed in .NET, this malware is a backdoor that uses the name TestProgram. Based on AhnLab’s detection name, it is classified as Andardoor. It is notable for being obfuscated using the Dotfuscator tool. It offers various features for controlling the infected system, such as file and process tasks, executing commands, and capturing screenshots. SSL encryption is used for communication with the C&C server. For the server name, it designated the “clientName” string.

```
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
}
tcpClient = new TcpClient();
tcpClient.Connect(new IPEndPoint(address, (int)this.eval_...));
num = (short)1056307212;
num2 = (int)((IntPtr)num);
continue;
IL_153:
int a_ = this.eval_....ReadInt32();
int a_2 = this.eval_....ReadInt32();
num3 = this.eval_....ReadInt32();
num = (short)1911422983;
num2 = (int)((IntPtr)num);
continue;
IL_340:
goto IL_153;
IL_226:
networkStream = tcpClient.GetStream();
this.eval_... = new SslStream(networkStream, false, new RemoteCertificateValidationCallback(this.eval_...));
this.eval_... AuthenticateAsClient("clientName");
this.eval_... = new byte[this.eval_...];
this.eval_... = new BinaryReader(this.eval_...);
this.eval_... = new BinaryWriter(this.eval_...);
this.eval_... Write(this.eval_...);
```

Figure 4. SSL communications routine with the C&C server

1.1.3. 1th Troy Reverse Shell

1th Troy is a Reverse Shell malware developed in Go. The following string included in the binary shows that the malware has the simple name of “Reverse_Base64_Pipe” and the malware’s creator classified the malware as “1th Troy”.

G:/Code/01__1th Troy/Go/Reverse_Base64_Pipe/Client/client.go

Being a Reverse Shell that only provides basic commands, the commands supported include “cmd”, “exit”, and “self delete”. They support the command execution, process termination, and self-deletion features respectively.

<pre>f os_exec_ptr_Cmd_writerDescriptor_func1 .text f os_exec_ptr_Cmd_Start_func1 .text f os_exec_ptr_Cmd_Start_func2 .text f os_exec_init_0_func1 .text f os_exec_init .text f type_eq_os_exec_Error .text f main_RandomString .text f main_SelfDelete .text f main_backdoor .text f main_main .text f main_backdoor_func1 .text f type_eq_6_interface_ .text</pre>	<pre>mov [rsp+248h+var_108], rax lea rcx, off_553020 ; "c" mov [rsp+248h+var_108+8], rcx mov qword ptr [rsp+248h+var_F8], rax lea rcx, off_553040 ; "m" mov qword ptr [rsp+248h+var_F8+8], rcx mov qword ptr [rsp+248h+var_E8], rax lea rcx, off_553050 ; "d" mov qword ptr [rsp+248h+var_E8+8], rcx lea rcx, a555 ; "%s%s%s" mov [rsp+248h+var_248], rcx ; __int64</pre>
--	---

Figure 5. Reverse Shell with a simple structure

1.2. Cases of attacks against Korean corporations

The Andariel group also distributed malware in March 2023 in its attacks against the Korean defense industry and an electronics device manufacturer. The method of initial compromise has not yet been identified, but logs of the mshta.exe process installing TigerRat and the mshta.exe process being terminated were confirmed through the AhnLab Smart Defense (ASD) infrastructure. This means that the malware strains were installed through a script-type malware with the spear phishing attack method.

Process	Module	Behavior	Data
mshta.exe	N/A	Creates process	Target Process certsvc.exe

Figure 6. Mshta process

installing TigerRat

Malware strains used in attacks were generally backdoor types. TigerRat, which has been used by the Andariel group since the past, was also included.

1.2.1. TigerRat

TigerRat is a RAT-type malware with its name given by KISA [8] and has been consistently employed by the Andariel threat group since 2020. It is known to be generally distributed through malicious document files containing macros that are attached to spear phishing

emails, or through watering hole attacks. [9] There are also cases where the Andariel group targeted Korean corporations that use vulnerable versions of VMware Horizon and launched Log4Shell vulnerability attacks to install TigerRat. [10]

Besides offering basic features such as file tasks and executing commands, TigerRat is a backdoor that supports other various features such as collecting information, keylogging, capturing screenshots, and port forwarding. One of its characteristics is that there is an authentication process upon the first communication session with the C&C server. In past versions, the string shown below disguised as SSL communications was used in the authentication process. Depending on the malware version, either the string “HTTP 1.1 /member.php SSL3.4” or “HTTP 1.1 /index.php?member=sbi2009 SSL3.3.7” must be sent to the C&C server, and the string “HTTP 1.1 200 OK SSL2.1” should be sent in return for authentication to be successful.

```
Stream Content
00000000 48 54 54 50 20 31 2e 31 20 2f 6d 65 6d 62 65 72 HTTP 1.1 /member
00000010 2e 70 68 70 20 53 53 4c 33 2e 34 00 .php SSL 3.4.
00000000 48 54 54 50 20 31 2e 31 20 32 30 30 20 4f 4b 20 HTTP 1.1 200 OK
00000010 53 53 4c 32 2e 31 00 SSL2.1.
0000001c 18 00 00 00 fc 7c c4 38 3a 32 37 7f fd 34 80 40 .....|.8 :27..4.@
0000002c ee 11 4a 1d a1 8e 48 6f a7 de 99 14 ..J...Ho ....
```

Figure 7. String used in the authentication process for the C&C server – past version

However, in the recently identified NukeSped type, the following random strings 0x20 in size are used. These strings are thought to be the MD5 hash for “fool” (dd7b696b96434d2bf07b34f9c125d51d) and “iwan” (01ccce480c60fcdb67b54f4509ffdb56). It seems that the threat actor used random strings in the authentication process to evade network detection.

```
Stream Content
00000000 64 64 37 62 36 39 36 62 39 36 34 33 34 64 32 62 dd7b696b 96434d2b
00000010 66 30 37 62 33 34 66 39 63 31 32 35 64 35 31 64 f07b34f9 c125d51d
00000020 00
00000000 30 31 63 63 63 65 34 38 30 63 36 30 66 63 64 62 01ccce48 0c60fcdb
00000010 36 37 62 35 34 66 34 35 30 39 66 66 64 62 35 36 67b54f45 09ffdb56
00000020 00
00000021 18 00 00 00
00000031
```

Figure 8. String used in authentication to the C&C server – latest version

- **C&C request string: dd7b696b96434d2bf07b34f9c125d51d**
- **C&C response string: 01ccce480c60fcdb67b54f4509ffdb56**

1.2.2. Black RAT

Black Rat is a backdoor-type malware that is likely created by the threat actor. Like other malware strains, it was developed in Go. While the 1th Troy Reverse Shell identified in the previous case only supports a basic command execution feature, Black Rat provides many additional features such as downloading files and capturing screenshots.

f	main_Send	.text
f	main_SendPacket	.text
f	main_Recv	.text
f	main_RecvPacket	.text
f	main_SelfDelete	.text
f	main_CmdShell	.text
f	main_CmdShell_func1	.text
f	main_RunTask	.text
f	main_getDriveType	.text
f	main_GetLogicalDrives	.text
f	main_GetAllFoldersAndFiles	.text
f	main_ScreenMonitThread	.text
f	main_FileDownload	.text
f	main_Handshake	.text
f	main_main	.text
f	main_MultiByteToWideChar	.text
f	main_MultiByteToWideChar_func1	.text
f	main_WideCharToMultiByte	.text
f	main_WideCharToMultiByte_func1	.text
f	main_NewMultiByteToWideChar	.text
f	main_NewWideCharToMultiByte	.text
f	main_ScreenRect	.text
f	main_ScreenRect_func1	.text
f	main_CaptureScreen	.text
f	main_CaptureRect	.text
f	main_CaptureRect_func4	.text
f	main_CaptureRect_func3	.text
f	main_CaptureRect_func2	.text
f	main_CaptureRect_func1	.text
f	main_ReleaseDC	.text
f	main_DeleteDC	.text
f	main_BitBlt	.text
f	main_DeleteObject	.text
f	main_init	.text

Figure 9. Features supported by Black RAT

Examining the following string included in the binary shows that the malware creator classified the malware as a RAT type and named it Black.

```
I:/01___Tools/02__RAT/Black/Client_Go/Client.go
```

1.2.3. NukeSped variants

A typical NukeSped-type backdoor was also used in this attack. Supported features include network scanning, process and file lookup, file upload/download, and command execution. The names of the APIs to be used are encrypted as shown below. These are decrypted and the API names are taken from somewhere else. A key with a size of 0x26 is used for decryption.

```

str_kernel32_dll = fn_decStr("wWN7BLxxfV1HBby");
kernel32_dll = LoadLibraryA(str_kernel32_dll);
free(str_kernel32_dll);
if ( kernel32_dll )
{
  GetProcAddress = fn_decStr("zVWL0gXy1YY/HAj70lk=");
  GetProcAddress_0 = ::GetProcAddress(kernel32_dll, GetProcAddress);
  free(GetProcAddress);
  LoadLibrary = fn_decStr("x1+e5jv01LU6CgPf");
  ::LoadLibrary = GetProcAddress_0(kernel32_dll, LoadLibrary);
  free(LoadLibrary);
  GetModuleFileNameA = fn_decStr("zVWLzXj5g6s+PhPyLGTnHtvG");
  ::GetModuleFileNameA = GetProcAddress_0(kernel32_dll, GetModuleFileNameA);
  free(GetModuleFileNameA);
  DeleteFileW = fn_decStr("z1WT5wP4sK43HS0=");
  ::DeleteFileW = GetProcAddress_0(kernel32_dll, DeleteFileW);
  free(DeleteFileW);
  CreateThread = fn_decStr("yUKa4wP4oq8pHRv6");
  ::CreateThread = GetProcAddress_0(kernel32_dll, CreateThread);
  free(CreateThread);
  CreateFileA = fn_decStr("yUKa4wP4sK43HTs=");
  ::CreateFileA = GetProcAddress_0(kernel32_dll, CreateFileA);
  free(CreateFileA);
  v11 = fn_decStr("yUKa4wP4sK43HS0=");
}

```

Figure 10.

Obfuscated API string

Key value used for decryption: i<6fu>-0|HSLRCqd.xHqMB]4H#axZ%5!5!?!SQ&

This NukeSped variant also uses a Batch script for self-deletion, but it is slightly different from the one used in the previous attacks.

```

.rdata:0000000140020590 aSSCS: ; DATA XREF: comm_execCmd+196fo
.rdata:0000000140020590 text "UTF-16LE", '%s\%S /c "%s"',0
.rdata:00000001400205AC align 10h
.rdata:00000001400205B0 ; const size_t BufferCount
.rdata:00000001400205B0 BufferCount db '@echo off',0Dh,0Ah ; DATA XREF: fn_selfDel+1E8to
.rdata:00000001400205B8 db ':L1',0Dh,0Ah
.rdata:00000001400205C0 db 'del "%s"%s "%s" goto L1',0Dh,0Ah
.rdata:00000001400205D9 db 'del "%s"',0Dh,0Ah,0
.rdata:00000001400205E4 align 8

```

Figure 11. Batch script used in the self-deletion process

There are two types of identified NukeSped variants: Reverse Shell and Bind Shell types. Both listen to port number 10443. This NukeSped variant has an authentication process before communicating with the C&C server like TigerRat. Yet whereas TigerRat disguised the process as SSL communications, NukeSped disguised it as HTTP communications. Thus, after sending the following POST request, an accurately matching HTTP response must be received for the malware to commence communications with the C&C server.


```

00000000 50 4f 53 54 20 2f 69 6e 64 65 78 2e 70 68 70 20 POST /in dex.php
00000010 48 54 54 50 2f 31 2e 31 5c 72 5c 6e 41 63 63 65 HTTP/1.1 \r\nAcce
00000020 70 74 3a 20 61 70 70 6c 69 63 61 74 69 6f 6e 2f pt: appl ication/
00000030 78 2d 6d 73 2d 61 70 70 6c 69 63 61 74 69 6f 6e x-ms-app lication
00000040 2c 20 69 6d 61 67 65 2f 6a 70 65 67 2c 20 2a 2f , image/ jpeg, */
00000050 2a 5c 72 5c 6e 41 63 63 65 70 74 2d 4c 61 6e 67 *\r\nAcc ept-Lang
00000060 75 61 67 65 3a 20 6b 6f 2d 4b 52 5c 72 5c 6e 55 uage: ko -KR\r\nU
00000070 73 65 72 2d 41 67 65 6e 74 3a 20 4d 6f 7a 69 6c ser-Agen t: Mozil
00000080 6c 61 2f 33 2e 36 32 20 28 63 6f 6d 70 61 74 69 la/3.62 (compati
00000090 62 6c 65 3b 20 4d 53 49 45 20 38 2e 33 32 3b 20 ble; MSI E 8.32;
000000A0 57 69 6e 64 6f 77 73 20 4e 54 20 36 2e 31 3b 20 Windows NT 6.1;
000000B0 57 4f 57 36 34 3b 20 54 72 69 64 65 6e 74 2f 34 WOW64; T rident/4
000000C0 2e 30 29 43 6f 6e 74 65 6e 74 2d 54 79 70 65 3a .0)Conte nt-Type:
000000D0 20 61 70 70 6c 69 63 61 74 69 6f 6e 2f 78 2d 77 applica tion/x-w
000000E0 77 77 2d 66 6f 72 6d 2d 75 72 6c 65 6e 63 6f 64 ww-form- urlencod
000000F0 65 64 5c 72 5c 6e 41 63 63 65 70 74 2d 45 6e 63 ed\r\nAc cept-Enc
00000100 6f 64 69 6e 67 3a 20 67 7a 69 70 2c 20 64 65 66 oding: g zip, def
00000110 6c 61 74 65 5c 72 5c 6e 48 6f 73 74 3a 20 77 77 late\r\n Host: ww
00000120 77 2e 62 69 6e 67 2e 63 6f 6d 5c 72 5c 6e 43 6f w.bing.c om\r\nCo
00000130 6e 74 65 6e 74 2d 4c 65 6e 67 74 68 3a 20 39 39 ntent-Le ngth: 99
00000140 39 39 39 39 39 39 39 5c 72 5c 6e 43 6f 6e 6e 65 9999999\ r\nConne
00000150 63 74 69 6f 6e 3a 20 4b 65 65 70 2d 41 6c 69 76 ction: K eep-Aliv
00000160 65 5c 72 5c 6e 43 61 63 68 65 2d 43 6f 6e 74 72 e\r\nCac he-Contr
00000170 6f 6c 3a 20 6e 6f 2d 63 61 63 68 65 5c 72 5c 6e ol: no-c ache\r\n
00000180 5c 72 5c 6e 00 \r\n.
00000000 48 54 54 50 2f 31 2e 31 20 32 30 30 20 4f 4b 5c HTTP/1.1 200 OK\
00000010 72 5c 6e 53 65 72 76 65 72 3a 20 41 70 61 63 68 r\nServe r: Apach
00000020 65 5c 72 5c 6e 4b 65 65 70 2d 41 6c 69 76 65 3a e\r\nKee p-Alive:
00000030 20 74 69 6d 65 6f 75 74 3d 35 2c 20 6d 61 78 3d timeout =5, max=
00000040 31 30 30 5c 72 5c 6e 43 6f 6e 6e 65 63 74 69 6f 100\r\nC onnectio
00000050 6e 3a 20 4b 65 65 70 2d 41 6c 69 76 65 5c 72 5c n: Keep- Alive\r\
00000060 6e 43 61 63 68 65 2d 43 6f 6e 74 72 6f 6c 43 6f nCache-C ontrolCo
00000070 6e 74 65 6e 74 2d 54 79 70 65 3a 20 6f 63 74 65 ntent-Ty pe: octe
00000080 74 2d 73 74 72 65 61 6d 5c 72 5c 6e 5c 72 5c 6e t-stream \r\n\r\n
00000090 00 .

```

Figure 12. HTTP packet used in authentication

2. Cases of Recent Attacks

ASEC is monitoring attacks of the Andariel group and has recently identified cases of Innorix Agent being abused to install malware. Unlike past cases where Innorix Agent was downloading malware strains, the recent case directly creates the malware file, so it is not certain whether the attacks are vulnerability attacks or if Innorix Agent was simply abused.

Malware strains identified in these attacks are not those that had been used by the Andariel group in the past, but aside from the fact that Innorix was used in the attacks, the current attack is similar to past attack cases in that the attack targets are Korean universities. While the attack was being perpetrated, attack cases against Korean ICT companies, electronic

device manufacturers, the shipbuilding industry, and the manufacturing industry were identified as well. Analysis showed that there was a connection with the malware strains used in attack cases where Innorix was abused.

This part will analyze each attack case and malware strains used in the attacks. Afterward, a summary will be given of the conclusion that the same threat actor is behind these attacks and the basis behind the claim, as well as the relationship between the current attacks and past attack cases of the Andariel threat group.

2.1. Cases of Innorix Agent abuse

2.1.1. Goat RAT

In recent attacks against Korean universities, there were cases where Innorix Agent installed malware strains. Innorix Agent installed the malware strains under the name “iexplorer.exe”. This is one of the names that has been often used by the Andariel group.

Target Type	File Name	File Size	File Path
Current	■ innorixas.exe	7.93 MB	%SystemDrive%\innorix_agent\innorixas.exe
Target	■ iexplore.exe.irx	2.07 MB	%SystemDrive%\users\%ASD%\downloads\iexplore.exe.irx

Process	Module	Target	Behavior	Data
■ innorixas.exe	N/A	N/A	Creates executable file	■ iexplore.exe.irx
■ innorixas.exe	N/A	N/A	Changes executable file name	■ iexplore.exe.irx

Figure 13. Using Innorix Agent to install Goat RAT

```
E:/Projects/Malware/6_Goat_23/Goat/Goat.go
E:/Projects/Malware/6_Goat_23/Goat/define.go
E:/Projects/Malware/6_Goat_23/Goat/anti-vaccine.go
E:/Projects/Malware/6_Goat_23/Goat/command.go
```

Although the recent version is obfuscated unlike the Go-based backdoor-type malware used in past attacks, basic file tasks, self-deletion features, etc. can be identified. There are also logs where the following commands were executed.

```
> cmd /c tasklist
> cmd /c ipconfig /all
```

f	._00000000000000000000	.text	21	if (v14 <= 1)
f	0000000000000000	.text	22	000000000000000000();
f	00000000000000	.text	23	*(_BYTE*)(v12 + 1) = a1;
f	._000000000000000000000000	.text	24	v6 = 000000000000(34LL, 128LL);
f	00000000000000	.text	25	if (v14 <= 2)
f	0000000000	.text	26	000000000000000000();
f	0000000000	.text	27	*(_BYTE*)(v12 + 2) = v6;
f	._00000000000000000000	.text	28	if (v14 < 7)
f	0000000000000000000000	.text	29	0000000000000000000000();
f	0000000000000000000000	.text	30	*(_DWORD*)(v12 + 3) = a5;
f	0000000000000000	.text	31	v7 = 000000000000(34LL, 128LL);
f	0000000000000000000000	.text	32	v8 = v14;
f	0000000000000000000000	.text	33	if (v14 <= 7)
f	._000000000000000000000000	.text	34	000000000000000000();
f	0000000000000000000000	.text	35	v9 = v12;
f	0000000000000000000000	.text	36	*(_BYTE*)(v12 + 7) = v7;

Figure 14. Obfuscated function name

2.2. Cases of attacks against Korean corporations

2.2.1. AndarLoader

Aside from the attack cases where Innorix Agent was abused, ASEC identified another type of attack in a similar period of time. While the initial distribution route has not yet been ascertained, the malware strains used in the attacks were obfuscated with a tool called Dotfuscator like the .NET malware strains classified as Andardoor. Another common trait is that both types use SSL communications with the C&C server. Unlike Andardoor which used “clientName” when connecting to the C&C server, this attack case used the string “sslClient”.

```

485      if (!IPAddress.TryParse(array[num5], out address))
486      {
487          num = (short)1567817729;
488          num2 = (int)(IntPtr)num;
489          continue;
490      }
491      tcpClient = new TcpClient();
492      tcpClient.Connect(new IPEndPoint(address, array2[num5]));
493      num = (short)507379715;
494      num2 = (int)(IntPtr)num;
495      continue;
496      IL_A25:
497      networkStream = tcpClient.GetStream();
498      sslStream = new SslStream(networkStream, false, new RemoteCertificateValidationCallback(RuntimeBrokerEval_0.eval_0));
499      sslStream.AuthenticateAsClient("<Module>.b("01000000", e_));
500      binaryReader = new BinaryReader(sslStream, Encoding.UTF8);
501      binaryWriter = new BinaryWriter(sslStream, Encoding.UTF8);
502      binaryWriter.Write(text3);

```

Name	Value	Type
<Module>.b returned	"sslClient"	string
A_0	{string[0x00000000]}	string[]

Figure 15. SSL connection process with the C&C server

Whereas Andardoor had most of its features already implemented, this malware strain only has a downloader feature to download and execute executable data such as .NET assemblies from external sources. Out of the commands sent from the C&C server, the commands shown below can be used to execute or terminate the received code. Behaviors performed by the threat actor using AndarLoader include installing Mimikatz in the infected system, which has been confirmed through a recorded log.

At the time of analysis, the C&C server was unavailable for access and the part in charge of the actual features could not be investigated, so no direct similarity with Andardoor could be confirmed. However, the use of the same obfuscation tool or the similarities in the communication process with the C&C server led AhnLab to categorize this malware as the AndarLoader type.

Command	Feature
alibaba	Execute the downloaded .NET assemblies
facebook	Execute the downloaded .NET method
exit	Terminate
vanish	Self-delete and terminate

Table 1. List of commands that can be executed

Among the commands given by the treat actor that AndarLoader executes, there is a command to terminate the mshta.exe process. The fact that AndarLoader was installed via PowerShell and the mshta.exe process was involved leaves the possibility that this is a spear phishing attack like the cases of attacks covered above.

```
Execute the Process

CMDLine
schtasks /delete /tn "\microsoft\windows\authservice" /f
schtasks /create /tn "\microsoft\windows\creditservice" /tr "c:\windows\system32\creditsvc.exe" /sc daily /st 10:35:20 /ru
taskkill /f /im mshta.exe
"cmd.exe"
schtasks /delete /tn "\microsoft\windows\creditservice" /f
schtasks /query
```

Figure 16. Commands executed by AndarLoader

Additionally, logs of the mshta.exe process connecting to the C&C server can be found in systems infected with AndarLoader.

Process	Module	Target	Behavior	Data
mshta.exe	N/A	N/A	Connects to network	http://www.ipservice.kro.kr/index.php
mshta.exe	N/A	N/A	Connects to network	http://www.ipservice.kro.kr/view.php
mshta.exe	N/A	N/A	Connects to network	http://www.ipservice.kro.kr/modeRead.php
powershell.exe	N/A	N/A	Downloads executable file	http://www.ipservice.kro.kr/dataSeq.exe Target creditsvc.exe
powershell.exe	N/A	N/A	Downloads executable file	http://www.ipservice.kro.kr/creditsvc.exe Target creditsvc.exe

Figure 17. Network communications log

The domain kro.kr was used as the C&C and download URLs. This is a domain generally used by the Kimsuky threat group. Also, the fact that Ngrok was installed for RDP connection during the attack process shows how the case is similar to the attack pattern of the Kimsuky group.

```

"targetProcess": {
  "imageInfo": {
    "fileObj": {
      "fileName": "service.exe",
      "fileSize": 24948456,
      "filePath": "%SystemDrive%\\users\\%ASD%\\service.exe",
      "commandLine": "%SystemDrive%\\users\\%ASD%\\service.exe tcp 3389"
    }
  },
  "currentProcess": {
    "imageInfo": {
      "fileObj": {
        "fileName": "cmd.exe",
        "fileSize": 289792,
        "filePath": "%SystemRoot%\\system32\\cmd.exe",
      }
    }
  },
  "parentProcess": {
    "imageInfo": {
      "fileObj": {
        "fileName": "mshta.exe",
        "fileSize": 43520,
        "filePath": "%SystemRoot%\\system32\\mshta.exe",
      }
    }
  }
}

```

Figure 18. Log

showing the installed Ngrok being executed

2.2.2. DurianBeacon

While investigating the AndarLoader malware, AhnLab identified that a malware strain named DurianBeacon was also used in the attack process. There are two versions of DurianBeacon, one developed in Go and the other developed in Rust. Both are backdoors that can perform malicious behaviors by receiving the threat actor's commands from the C&C server.

A. Go Version

Examining the following strings included in the binary indicates that the malware creator named this malware strain DurianBeacon.

```
G:/Dev/Go/DurianBeacon/Command.go
G:/Dev/Go/DurianBeacon/SSL.go
G:/Dev/Go/DurianBeacon/Utils.go
G:/Dev/Go/DurianBeacon/main.go
```

The Go version of DurianBeacon uses the SSL protocol to communicate with the C&C server. After initial access, it sends the infected system's IP information, user name, desktop name, architecture, and file names before awaiting commands. When a command is sent, it returns a result. Supported features besides collecting basic information about the infected system include file download/upload, lookup, and command execution features.

```
f main_ProcessCommand .text
f main_ProcessCommand_func3 .text
f main_ProcessCommand_func2 .text
f main_ProcessCommand_func1 .text
f main_ProcessCommand_MakeDir .text
f main_ProcessCommand_Remove .text
f main_ProcessCommand_Execute .text
f main_ProcessCommand_DownloadStart .text
f main_ProcossCommand_UploadStart .text
f main_ProcessCommand_Ls .text
f main_ProcessCommand_Drives .text
f main_ProcessCommand_ExecuteJob .text
f main_ProcessCommand_ExecuteJob_func1 .text
f main_ProcessCommand_Hibernate .text
f main_ptr_SSLclient_Handshake .text
f main_ptr_SSLclient_Close .text
f main_ptr_SSLclient_SendResult .text
f main_ptr_SSLclient_ReceiveCommand .text
f main_GenerateSessionMetaData .text
f main_GetImageName .text
f main_GetInternallP .text
f main_GetUsername .text
f main_GetComputerName .text
f main_GetArchitecture .text
f main_getVolumeInfoJson .text
f main_GetVolumeInfo .text
f main_GetVolumeInfo_func1 .text
f main_GetVolumeInformation .text
f main_GetVolumeInformation_func1 .text
f main_main .text
```

Figure 19. Features supported by

DurianBeacon

Because the SSL protocol is used, communications packets are encrypted. The following packet structure is used internally.

Offset	Size	Description
0x00	0x04	Command number
0x04	0x04	Size of the command argument
0x08	Variable	Command argument

Table 2. Command packet structure of DurianBeacon
The features corresponding to each command code are as follows.

Command	Feature
0x00	Hibernate
0x01	Interval
0x02	Execute commands (return results)
0x03	Look up directories
0x04	Drive information
0x05, 0x06, 0x07, 0x08	Upload files
0x09, 0x0A, 0x0B	Download files
0x0C	Create directories
0x0D	Delete files
0x0E	Run commands
0x0F	Terminate

Table 3. List of DurianBeacon commands
After executing commands, the malware sends the success status or the command execution results to the threat actor's C&C server. The response is also similar to the command packet.

Offset	Size	Description
0x00	0x04	Response number
0x04	0x04	Size of the command execution results

Offset	Size	Description
0x08	Variable	Command execution results

Table 4. Structure of the DurianBeacon response packet

Response	Description
0x00	Return command results
0x01, 0x02, 0x03	Look up directories (start, terminate, etc.)
0x04	Drive information
0x05, 0x06, 0x07	Upload files (error, success, etc.)
0x08, 0x09, 0x0A	Download files (error, success, etc.)
0x0B, 0x0C	Create directories (failure or success)
0x0D, 0x0E	Delete files (failure or success)
0x0F, 0x10	Run commands (failure or success)

Table 5. DurianBeacon's response list

B. Rust Version

Investigation of related files revealed that the Rust version of DurianBeacon was also used in attacks.

PDB information: C:\Users\Anna\Documents\DurianBeacon\target\x86_64-pc-windows-msvc\release\deps\DurianBeacon.pdb

DurianBeacon supports packet encryption using XOR aside from SSL to communicate with the C&C server, using the key 0x57.

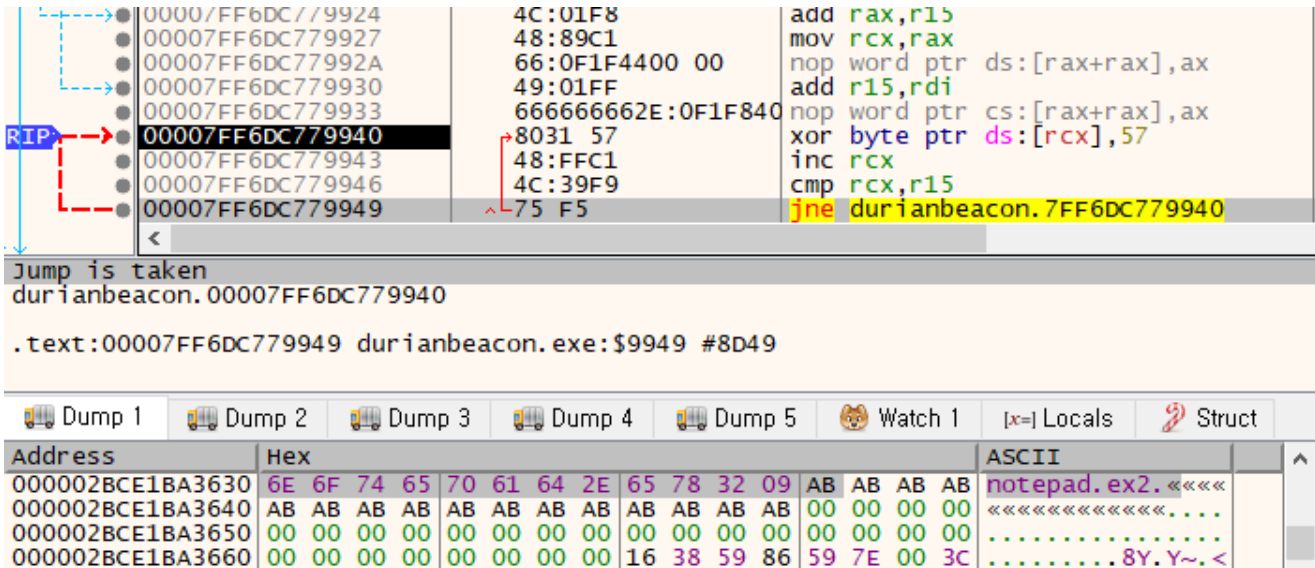


Figure 20. Rust version of DurianBeacon supporting XOR encryption
The packet structure and commands are also the same as the Go version. The Rust version of DurianBeacon sends the keyword “durian2023” alongside the infected system’s IP information, user name, desktop name, architecture, and file names before awaiting command. When a command is sent, it returns the results.

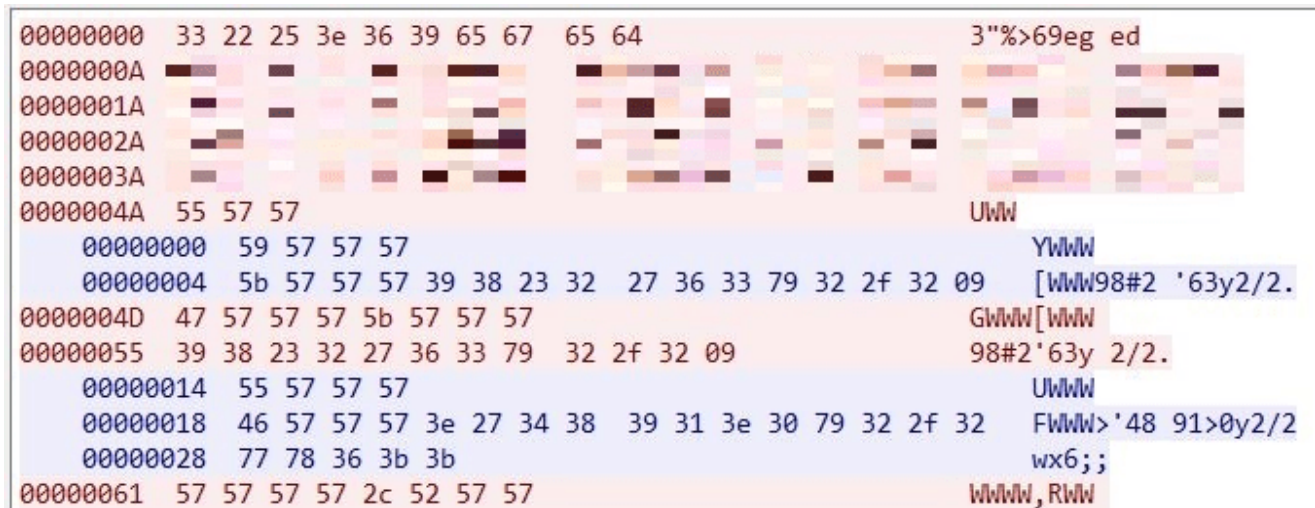


Figure 21. Communications packet of the Rust version – test



3. Connections to recent attack cases

The above section covered the recently identified two cases where universities in Korea were attacked through abusing Innorix Agent and where malware strains were installed in Korean corporations through presumably spear phishing attacks. This part will explain why AhnLab considers the same threat actor to be behind both types of attacks.

First, there are cases in AhnLab’s ASD logs where Durian, Goat RAT, and AndarLoader were collected together in a similar period. The system in question is thought to be the threat actor’s test PC because the path name of AndarLoader was as follows.

AndarLoader collection path: d:\01__developing\99__c#_obfuscated\runtime broker.exe

There are also cases where the C&C servers of backdoor-type malware strains were the same. When the threat actor used Innorix Agent to install malware, Goat RAT was mainly employed, but there is a significant portion where other malware strains were installed. While such malware samples could not be collected, there are recorded communications logs with the C&C server. Also, the URL in question was the same as the DurianBeacon C&C server URL used in other attacks.

Target Type	File Name	File Size	File Path ⓘ
Current	 innorixas.exe	7.94 MB	%SystemDrive%\innorix_agent\innorixas.exe
Target	 iexpe.exe.irx	3.84 MB	%SystemDrive%\users\%ASD%\downloads\iexpe.exe.irx







Process	Module	Target	Behavior	Data
 innorixas.exe	N/A	N/A	Creates executable file	N/A
 File Less Submit	iexp.exe	N/A	Connects to network	8.213.128.76:53
 innorixas.exe	N/A	N/A	Changes executable file name	N/A

Figure 22. C&C communications log of the malware installed through Innorix Agent
 Finally, there was a log where DurianBeacon installed AndarLoader. This means that these attacks happened around a similar time period, and the attacks might be related to each other as the installation processes and the C&C server URLs used tend to be similar.

Target Type	File Name	File Size	File Path ⓘ
Current	 creditsvc.exe	11 KB	%SystemRoot%\system32\creditsvc.exe
Parent	 svchost.exe	54.02 KB	%SystemRoot%\system32\svchost.exe
DropperOfCurrent	 agent.exe	427.5 KB	%SystemRoot%\agent.exe




Process	Module	Target	Behavior	Data
 creditsvc.exe	N/A	N/A	Connects to network	10.10.10.10:3389
 creditsvc.exe	N/A	N/A	Creates executable file	 mimi.exe

Figure 23. Log showing DurianBeacon creating AndarLoader

4. Connections to past attack cases of the Andariel group

The recently identified two attack cases are likely done by the same threat actor. This section will cover the relationship between these attacks and the Andariel threat group.

A. Attack Targets

Attacked universities, the national defense industry, electronic device manufacturers, ICT companies, etc. in Korea.

B. Attack Methods

- Abused Innorix Agent like in past cases
- Probably employed spear phishing method like in past cases
- Similarities between the path and file names used when installing the malware strains

C. Malware Types Used

- Malware strains developed in Go were used
- Similarities between Andardoor and AndarLoader
- Malware types similar to the Infostealer used in previous attacks were identified

First, there are the facts that the industries and sectors that became attack targets were the same as the targets identified in past attack cases and the same attack methods used in previous attacks were employed in recent cases. AhnLab identified cases where Innorix Agent was used. While not confirmed, many logs showed circumstances of spear phishing attacks.

The file name “iexplorer.exe” used to install malware has been identified from Andariel’s past attack cases to the present. Besides “iexplorer.exe”, names including the “svc” keyword such as “authsvc.exe” and “creditsvc.exe” has been continuously used. Also, aside from “mainsvc.exe” and “certsvc.exe”, there are cases where similar names such as “netsvc.exe” and “srvcredit.exe” were used.

As covered in the corresponding section, AndarLoader was obfuscated with the trial version of Dotfuscator, the tool used in Andardoor in previous attacks. It also uses SSL encryption to communicate with the C&C server, again showing similarities with past attack cases. Two other malware strains developed in Go were used as well. These align with the trend of malware strains developed in Go such as 1th Troy Reverse Shell and Black RAT continuously being used since the early part of this year.

Finally, there is also the system thought to be the threat actor’s test PC and Infostealer strains presumably created by the threat actor during the attack process. In fact, the Andariel group in the past installed malware strains responsible for stealing account credentials during the attack process, exfiltrating account credentials saved in Internet Explorer, Chrome, and Firefox web browsers. Such malware strains are command line tools that output the extracted account credentials via command lines. It seems that the threat actor used a backdoor to send the results to the C&C server.

```
-----Google Chrome Password-----
-----Mozilla Firefox Password-----
Mozilla Firefox isn't install..
-----Internet Explorer Password-----
Internet Explorer => uname: justtest   pwd: testpass   site: https://www.ahnlab.com/
-----Opera < v60-----
-----Opera < v80-----
opera isn't install..
-----Naver Whale-----
whale browser isn't install..
-----Outlook-----
```

Figure 24. Infostealer identified in past attack cases

The Infostealer used in the recent attacks has a similar format. The only difference is that it only targets web browsers and steals account credentials and histories. Additionally, unlike the past cases where results were outputted by command lines, the recent version saves the stolen information in the same path under the file name “error.log”.

```

2  ▾ *****Chromium*****
3  |
4  |
5  |
6  |     =====Credentials=====
7  |
8  | Url: https://www.kim.com
9  | Username: [redacted]
10 | Password:
11 | ▾ -----
12 |     ===== History =====
13 |
14 | Url: h1
15 | Title: [redacted]
16 | LastVisitedTime: [redacted] -----
17 |
18 | ▾ *****Firefox*****
19 | |
20 | |
21 | |
22 | |     =====Credentials=====
23 | |
24 | ▾ 64bit Firefox is only available!
25 | |
26 | |     ===== History =====
27 | |
28 | |
29 | ▾ *****Internet explorer*****
30 | |
31 | |
32 | |
33 | |     =====Credentials=====
34 | |
35 | |     ===== History =====

```

Figure 25. Infostealer identified in recent attack cases

5. Conclusion

The Andariel group is one of the highly active threat groups targeting Korea along with Kimsuky and Lazarus. The group launched attacks to gain information related to national security in the early days but now carries out attacks for financial gains. [11] The group is known to employ spear phishing attacks, watering hole attacks, and vulnerability exploits for their initial infiltration process. There have also been cases where it used other vulnerabilities in the attack process to distribute malware strains.

Users must be particularly cautious about attachments to emails with unknown sources or executable files downloaded from websites. Users should also apply the latest patch for OS and programs such as internet browsers and update V3 to the latest version to prevent malware infection in advance.

File Detection

- Backdoor/Win.Agent.R562183 (2023.03.14.00)
- Backdoor/Win.Andardoor.C5381120 (2023.02.16.01)
- Backdoor/Win.Andardoor.R558252 (2023.02.16.01)
- Backdoor/Win.AndarGodoor.C5405584 (2023.04.05.03)
- Backdoor/Win.DurianBeacon.C5472659 (2023.08.18.02)
- Backdoor/Win.DurianBeacon.C5472662 (2023.08.18.02)
- Backdoor/Win.DurianBeacon.C5472665 (2023.08.18.03)
- Backdoor/Win.Goat.C5472627 (2023.08.18.02)
- Backdoor/Win.Goat.C5472628 (2023.08.18.02)
- Backdoor/Win.Goat.C5472629 (2023.08.18.02)
- Backdoor/Win.NukeSped.C5404471 (2023.04.03.02)
- Backdoor/Win.NukeSped.C5409470 (2023.04.12.00)
- Backdoor/Win.NukeSped.C5409543 (2023.04.12.00)
- Infostealer/Win.Agent.C5472631 (2023.08.18.02)
- Trojan/Win.Agent.C5393280 (2023.03.11.00)
- Trojan/Win.Agent.C5451550 (2023.07.11.00)
- Trojan/Win.Andarinodoor.C5382101 (2023.02.16.01)
- Trojan/Win.Andarinodoor.C5382103 (2023.02.16.01)
- Trojan/Win32.RL_Mimikatz.R366782 (2021.02.18.01)

Behavior Detection

- Suspicious/MDP.Download.M1004
- Infostealer/MDP.Behavior.M1965

IOC

MD5

- 0a09b7f2317b3d5f057180be6b6d0755: NukeSped variant – Volgmer (%SystemRoot%\mstc.exe.irx)
- 1ffccc23fef2964e9b1747098c19d956: NukeSped Variant – Volgmer (%SystemRoot%\msnox.exe.irx)
- 9112efb49cae021abebd3e9a564e6ca4: NukeSped variant – Volgmer (%SystemRoot%\system32\mscert.exe)
- bcac28919fa33704a01d7a9e5e3ddf3f: NukeSped variant – Volgmer (%SystemRoot%\msnoxe.exe.irx)
- ac0ada011f1544aa3a1cf27a26f2e288: Andardoor (%SystemDrive%\users\%ASD%\msdes.exe.irx)
- c892c60817e6399f939987bd2bf5dee0: Andardoor (%SystemDrive%\users\%ASD%\msdes.exe.irx)
- 0211a3160cc5871cbcd4e5514449162b: Andardoor (%SystemDrive%\users\%ASD%\msdes.exe.irx)
- e5410abaaac69c88db84ab3d0e9485ac: 1st Troy Reverse Shell

(%SystemRoot%\msnox.exe.irx)
 – 88a7c84ac7f7ed310b5ee791ec8bd6c5: 1st Troy Reverse Shell
 (%SystemRoot%\msnox.exe.irx)
 – eb35b75369805e7a6371577b1d2c4531: TigerRat (%SystemRoot%\system32\hl_cl.exe)
 – 5a3f3f75048b9cec177838fb8b40b945: TigerRat
 (%SystemDrive%\users\%ASD%\larabar.exe, %SystemDrive%\users\%ASD%\mainsvc.exe,
 %SystemDrive%\users\%ASD%\certsvc.exe)
 – 9d7bd0caed10cc002670faff7ca130f5: Black RAT
 (%SystemRoot%\syswow64\mbcbuilder.exe, %SystemRoot%\syswow64\msinfo.exe)
 – 8434cdd34425916be234b19f933ad7ea: Black RAT
 (%SystemRoot%\system32\shamon.exe)
 – bbaee4fe73ccff1097d635422fdc0483: NukeSped Variant
 (%SystemDrive%\users\%ASD%\update.exe)
 – 79e474e056b4798e0a3e7c60dd67fd28: NukeSped variant (%SystemRoot%\hl_cl.exe)
 – 3ec3c9e9a1ad0e6a6bd75d00d616936bc: Goat RAT
 (%SystemDrive%\users\%ASD%\downloads\iexplore.exe)
 – 95c276215dcc1bd7606c0cb2be06bf70: Goat RAT
 (%SystemDrive%\users\%ASD%\downloads\iexplore.exe)
 – 426bb55531e8e3055c942a1a035e46b9: Goat RAT
 (%SystemDrive%\users\%ASD%\downloads\iexplore.exe)
 – cfae52529468034dbbb40c9a985fa504: Goat RAT
 (%SystemDrive%\users\%ASD%\downloads\iexplore.exe)
 – deae4be61c90ad6d499f5bdac5dad242: Goat RAT
 (%SystemDrive%\users\%ASD%\downloads\iexplore.exe)
 – 6ab4eb4c23c9e419fbba85884ea141f4: AndarLoader
 ((SystemDrive%\users\%ASD%\pictures\runtime broker.exe,
 %SystemRoot%\system32\authsvc.exe, %SystemRoot%\system32\creditsvc.exe,
 %ProgramFiles%\smartplant\svchost.exe)
 – bda0686d02a8b7685adf937cbcd35f46: DurianBeacon Go (a.exe)
 – 6de6c27ca8f4e00f0b3e8ff5185a59d1: DurianBeacon Go
 (%SystemDrive%\users\%ASD%\pictures\xxx.exe)
 – c61a8c4f6f6870c7ca0013e084b893d2: DurianBeacon Rust
 (%SystemDrive%\users\%ASD%\documents\d.exe)
 – 5291aed100cc48415636c4875592f70c: Mimikatz
 (%SystemDrive%\users\%ASD%\mimi.exe)
 – f4795f7aec4389c8323f7f40b50ae46f: malware collecting account credentials
 (%SystemDrive%\users\%ASD%\documents\mshelp.exe)

Download URLs

– hxxp://27.102.113[.]88/client.exe: NukeSped variant – Volgmer
 – hxxp://27.102.107[.]230/mstcs.exe: NukeSped variant – Volgmer
 – hxxp://27.102.107[.]233/update.exe: NukeSped variant – Volgmer

- hxxp://27.102.107[.]233/client.exe: NukeSped variant – Volgmer
- hxxp://27.102.107[.]234/update.exe: NukeSped variant – Volgmer
- hxxp://27.102.107[.]235/mstcs.exe: NukeSped variant – Volgmer
- hxxp://139.177.190[.]243/update.exe: Andardoor
- hxxp://4.246.144[.]112/update.exe: Andardoor
- hxxp://27.102.113[.]88/update.exe: 1st Troy Reverse Shell
- hxxp://27.102.107[.]224/update.exe: 1st Troy Reverse Shell
- hxxp://27.102.107[.]230/update.exe: 1st Troy Reverse Shell
- hxxp://www.ipservice.kro[.]kr/dataSeq.exe: AndarLoader
- hxxp://www.ipservice.kro[.]kr/creditsvc.exe: AndarLoader

C&C URLs

- 27.102.113[.]88:5443: NukeSped variant – Volgmer
- 27.102.107[.]234:8443: NukeSped variant – Volgmer
- 27.102.107[.]224:5443: NukeSped variant – Volgmer
- 109.248.150[.]179:443: NukeSped variant – Volgmer
- 139.177.190[.]243:443: Andardoor
- 4.246.144[.]112:443: Andardoor
- 27.102.113[.]88:21: 1st Troy Reverse Shell
- 27.102.107[.]224:8443: 1st Troy Reverse Shell
- 4.246.149[.]227:8080: TigerRat
- 13.76.133[.]68:8080: TigerRat
- 217.195.153[.]233:443: TigerRat
- bbs.topignorsvin.com[.]ec:8080: Black RAT
- 27.102.129[.]196:8088: Black RAT
- 13.76.133[.]68:10443: NukeSped variant
- 46.183.223[.]21:8080: Goat RAT
- chinesekungfu[.]org:443: AndarLoader
- privatemake.bounceme[.]net:443: AndarLoader
- 8.213.128[.]76:1012: DurianBeacon Go

Subscribe to AhnLab's next-generation threat intelligence platform 'AhnLab TIP' to check related IOC and detailed analysis information.

Categories:[Malware Information](#)

Tagged as:[Andardoor](#),[Andariel](#),[BlackRAT](#),[DurianBeacon](#),[GoatRAT](#),[Lazarus](#),[TigerRat](#),[Volgmer](#)