



Finding APTX

Attributing Attacks via MITRE TTPs

Lenart Bermejo, Gilbert Sison, and Buddy Tancio

TREND MICRO LEGAL DISCLAIMER

The information provided herein is for general information and educational purposes only. It is not intended and should not be construed to constitute legal advice. The information contained herein may not be applicable to all situations and may not reflect the most current situation. Nothing contained herein should be relied on or acted upon without the benefit of legal advice based on the particular facts and circumstances presented and nothing herein should be construed otherwise. Trend Micro reserves the right to modify the contents of this document at any time without prior notice.

Translations of any material into other languages are intended solely as a convenience. Translation accuracy is not guaranteed nor implied. If any questions arise related to the accuracy of a translation, please refer to the original language official version of the document. Any discrepancies or differences created in the translation are not binding and have no legal effect for compliance or enforcement purposes.

Although Trend Micro uses reasonable efforts to include accurate and up-to-date information herein, Trend Micro makes no warranties or representations of any kind as to its accuracy, currency, or completeness. You agree that access to and use of and reliance on this document and the content thereof is at your own risk. Trend Micro disclaims all warranties of any kind, express or implied. Neither Trend Micro nor any party involved in creating, producing, or delivering this document shall be liable for any consequence, loss, or damage, including direct, indirect, special, consequential, loss of business profits, or special damages, whatsoever arising out of access to, use of, or inability to use, or in connection with the use of this document, or any errors or omissions in the content thereof. Use of this information constitutes acceptance for use in an "as is" condition.

Published by

Trend Micro Research

Written by

Lenart Bermejo

Threat Researcher

Gilbert Sison

Cyber Threat Hunting Technical Lead

Buddy Tancio

Incident Response Analyst

Stock image used under license from
Shutterstock.com

For Raimund Genes (1963-2017)

Contents

4

Introduction

5

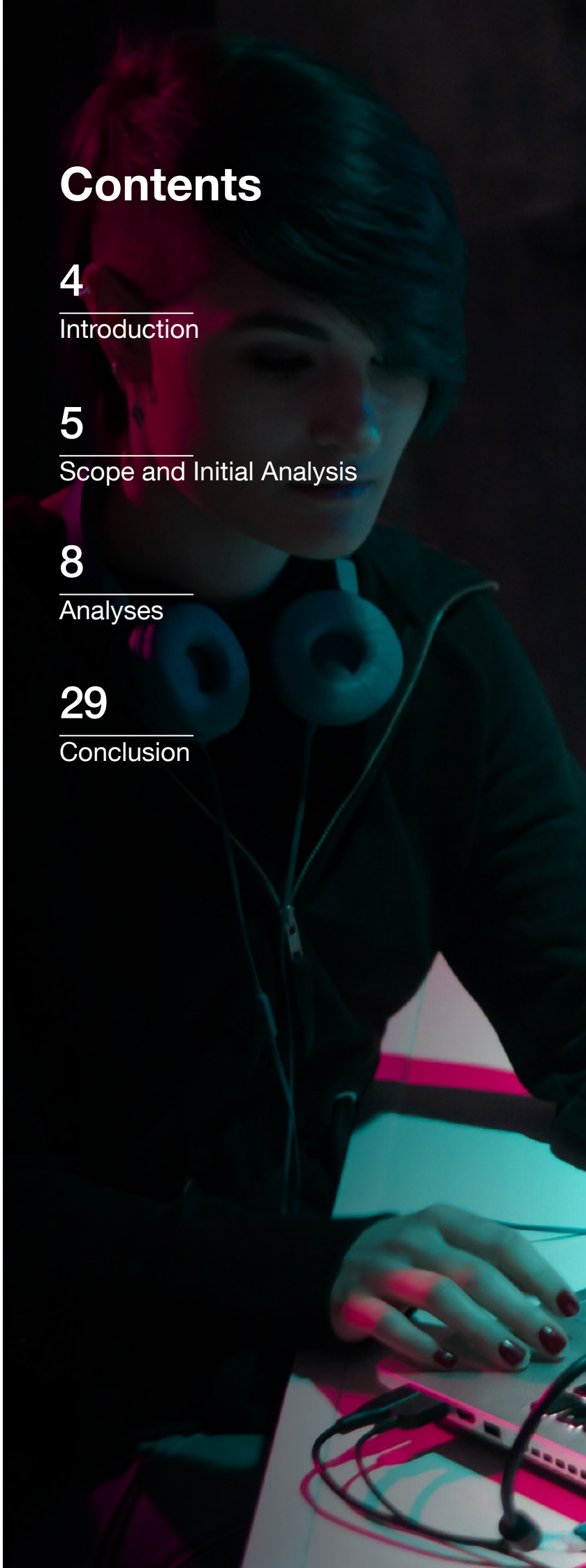
Scope and Initial Analysis

8

Analyses

29

Conclusion





Summary

We investigated an advanced persistent threat (APT) attack in an organization after the organization's security team detected an unusual series of traffic exchange with a command-and-control (C&C) server. After an analysis of the traffic and embedded codes, it appears that the intrusion has been occurring for a while and involves a number of systems within the company.

We were able to map out the tools, relationships, and behaviors of the intrusion based on memory and disk images provided by the organization. By matching the said information with publicly documented resources such as MITRE ATT&CK, we were able to narrow down the list of possible attackers despite some differences in the details. These details could have been brought about by the customized software, security posture, and hardware utilization in the organization, as specific elements unique to the company appear to have been researched by the group (or groups) responsible.

By tracking and reverse-engineering the techniques and intrusion sets, as well as comparing the published tactics, techniques, and procedures (TTPs), we tried to find the group (or groups) responsible for the persistent threat. Initial analysis using MITRE potentially showed APT32 or APT3 to be responsible for the attack. However, as we incorporated more known tools in analyzing some of the discrepancies from previously documented behaviors and tools, we came up with more detailed intrusion sets to show that there might be other potentially responsible groups. In the presence of the unidentified sets, tools, and behavior intersections, our investigation intends to add to the documented knowledge of the groups that have been studied.

While no organization hopes to be in these kinds of instances, enterprises who fall into situations like these pose a unique experience that can develop and contribute to the security landscape and defense standard of companies and countries. Technologies available in the market today can better help investigators recreate the attack scenario and improve the back-end protection that security operations centers (SOCs) depend on to prevent threats like these.

Introduction

Cybersecurity researchers and investigators use references such as MITRE ATT&CK¹ as a knowledge base for information on preventing, studying, and mitigating known attack routine techniques and tactics. Based on observed capabilities, security teams can identify the intrusion sets or groups that were used in order to gain valuable insights to defend systems against attacks. The process of identification comprises several steps, such as comparing the sets and techniques, creating detection lists, and matching the signature attacks with the specific APT groups for attribution and disclosure in the event of intrusions.

Certain pieces of information, such as the actual targets of specific groups, also help the institutions' security teams mount the necessary defenses against likely attack methods that are employed in their respective industries.

However, there is still a difference between experiencing an actual attack and reading descriptions of it. Theoretically, known attack signatures can be matched easily to identify the objective of a group, thereby immediately thwarting the said attack based on known measures. In reality, however, attacks can be modified to match the existing variables in the victim's current state, such as the existing security posture, software, or the machine environments being used. Threat actors are also known to deviate from their previously used or publicly known routines to confuse or distract security teams, for instance by planting false flags.

In the same way that an investigator plays a central role in making the connections between analyzing the details of cybersecurity incidents and identifying the significant markers for attribution, technologies that are used by the company also perform a crucial role. Often, technologies such as endpoint detection and response (EDR) are able to monitor, detect, and identify the unusual behaviors and pivot both security and investigating teams to gain a comparison and reverse-analyze the granular intersections of attack routines, tool behaviors, techniques, and relationships.

In the absence of these technologies, this paper looks into identifying and investigating both an APT attack and the actors behind the intrusion based on how MITRE grouped the techniques. Here we also compare the identified sets to the documented TTPs and the data that we were given.

Scope and Initial Analysis

After discovering a series of suspicious C&C traffic from one of their servers, we were invited by the affected company in 2019 to investigate and block the said communication and infection. Although the C&C was initially attributed to APT32 (not to mention despite detection of the malicious files), the unusual traffic warranted further investigation.

In this particular case, the investigating team encountered some challenges. The only accessible data consisted of the feedback and event logs from endpoints and agent-based servers from several identified machines to validate the compromise. Additionally, the team had access to only five machines from which disk and memory images were acquired. Lastly, there were no means to collect all of the samples and tools that were running in the systems' environment.

It is worth emphasizing here that limited data in turn limits the process of making a complete map and attribution of an attack. To improve data-collection capability, it is recommended that organizations consider EDR solutions.

Based on the detections and logs, a total of 62 machines were initially identified as infected: Among these were 10 servers, 13 machines containing binaries capable of file scraping and data exfiltration, and 22 machines that had backdoor shells. The rest hosted either other tools used in the attack or normal applications that were used to load malicious libraries.

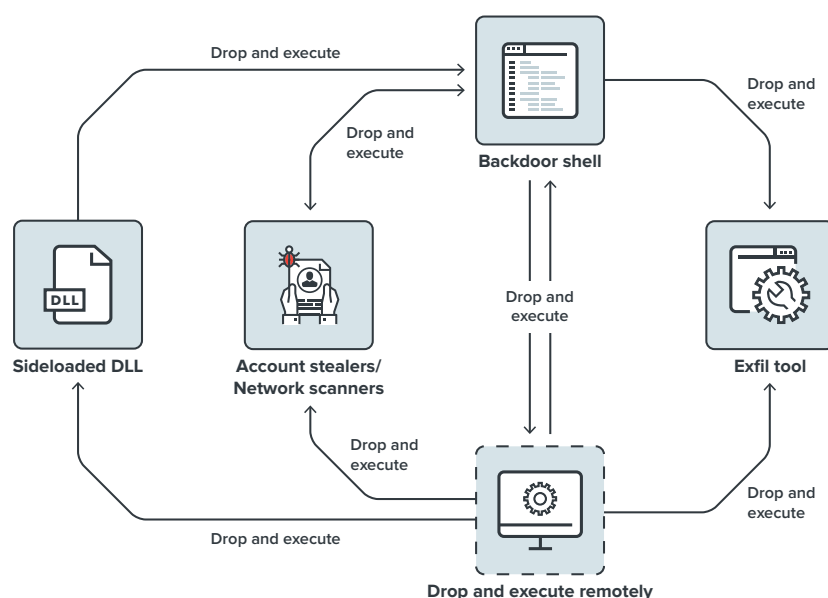


Figure 1. Initial assessment of the routine based on preliminary data collected

The backdoor shell enables the attacker to execute commands via cmd.exe, as shown when the normal process sideloaded a malicious DLL that dropped the backdoor. Tools like ProcDump and Mimikatz were

used to acquire user accounts, while network scanning tools were used to find other machines. The resulting compromised accounts were used to map the inter-process communications share (also known as IPC\$ share) of the newly discovered machines, while the mapping further allowed the backdoor to drop tools on the remote machine. To run the remotely dropped executable, two possible actions were taken: Either a scheduled task was created, or a wmic process create command was used. The tools that were dropped and executed were varied and in some cases, copies of the backdoor were dropped.

Our analysis of the exfiltration tools showed that the attackers were mostly interested in document files such as Microsoft Office documents and PDFs. It was also evident that the attack had been in progress for quite some time — likely even years — based on the timestamps of the binaries and how widespread the infection was.

Based on the mapped intrusion techniques, the group initially suspected was APT32 as referenced against MITRE ATT&CK.

MITRE ATT&CK Technique	Observed Behavior	In APT32?
T1059.003 – Windows Command Shell	Backdoor samples created cmd.exe to execute commands.	Y
T1053.005 – Scheduled Task/Job: Scheduled Task	schtask.exe created by cmd.exe to create tasks	Y
T1543.003 – Create or Modify System Process: Windows Service	Installation of a service for a tool	Y
T1047 – Windows Management Instrumentation	wmic.exe created by cmd.exe to execute scripts and executables remotely	Y
T1547.001 – Boot or Logon Autostart Execution: Registry Run Keys/Startup Folder	Creation of autostart registry keys for normal programs with sideloads and placement of samples in the startup folder	Y
T1078 – Valid Accounts	Accounts with administrator privileges were used for lateral movement.	N
T1574.002 – Hijack Execution Flow: DLL Sideload	Numerous normal binaries were used to load malicious DLLs.	Y
T1003.001 – OS Credential Dumping: LSASS Memory	procdump.exe was used to dump local security authority subsystem service (LSASS). Mimikatz was used to get the necessary information.	Y
T1069 – Permission Groups Discovery	Usage of net.exe to list groups	N
T1033 – System Owner/User Discovery	Done through T1003.001	Y
T1021.002 – Remote Services: SMB/Windows Admin Shares	IPC\$ share of remote machines were mapped and tools were dropped.	Y
T1005 – Data from Local System	Tools enumerated document/office files in the local drive.	N
T1039/T1025 – Data from Network Shared/Removable Drive	Tools enumerated documents in non-local drives.	N
T1083 – File and Directory Discovery	Tools enumerated documents.	Y
T1571 – Non-Standard Port	Detected traffic used HTTP/S.	Y

Table 1. Observed techniques associated with APT32 based on MITRE ATT&CK

Based on the available data, the techniques observed were mostly consistent with the listed intrusion set for APT32 in MITRE ATT&CK, except for four that were not associated with the group prior. However, if the set identification were to be based only on techniques, APT3 would also fit the observed map of techniques and might be a closer match than APT32, as APT3 would have only three techniques that are not associated with the group.

MITRE ATT&CK Technique	Observed Behavior	In APT32?	In APT3?
T1059.003 – Windows Command Shell	Backdoor samples created cmd.exe to execute commands.	Y	Y
T1053.005 – Scheduled Task/Job: Scheduled Task	schtask.exe created by cmd.exe to create tasks	Y	Y
T1543.003 – Create or Modify System Process: Windows Service	Installation of a service for a tool	Y	Y
T1047 – Windows Management Instrumentation	wmic.exe created by cmd.exe to execute scripts and executables remotely	Y	N
T1547.001 – Boot or Logon Autostart Execution: Registry Run Keys/Startup Folder	Creation of autostart registry keys for normal programs with sideloads and placement of samples in the startup folder	Y	Y
T1078 – Valid Accounts	Accounts with administrator privileges were used for lateral movement.	N	Y
T1574.002 – Hijack Execution Flow: DLL Sideload	Numerous normal binaries were used to load malicious DLLs.	Y	Y
T1003.001 – OS Credential Dumping: LSASS Memory	procdump.exe was used to dump local security authority subsystem service (LSASS). Mimikatz was used to get the necessary information.	Y	Y
T1069 – Permission Groups Discovery	Usage of net.exe to list groups	N	Y
T1033 – System owner/User discovery	Done through T1003.001	Y	Y
T1021.002 – Remote Services: SMB/Windows Admin Shares	IPC\$ share of remote machines were mapped and tools were dropped.	Y	Y
T1005 – Data from Local System	Tools enumerated document/office files in the local drive.	N	Y
T1039/T1025 – Data from Network Shared/Removable Drive	Tools enumerated documents in non-local drives.	N	Y
T1083 – File and Directory Discovery	Tools enumerated documents.	Y	N
T1571 – Non-Standard Port	Detected traffic used HTTP/S.	Y	Y

Table 2. Observed techniques associated with APT32 and APT3 based on MITRE ATT&CK

Analyses

Considering that both APT groups could be matched with the observed techniques, we needed to look further into the tools and the specific routines that each type underwent. Using the five endpoint targets with the most number of tools installed, we analyzed the disk images and files extracted from these machines and found a variety of tools, programs, and relationship clusters that the entire routine employed.

Tools

Exfiltration

We found six types of exfiltration tools, each with the capability to function (either with the other tools or independently), extract the files needed from a variety of locations, store or send the files, obfuscate traffic and activity, or utilize the existing programs used by the company for data exfiltration.

1. TSPY_PILFERDOC.ZBGK

This is a file scraper that harvests files, places them in a new directory, and creates a password-protected RAR archive with the use of an archiving tool. The scraper can run based on settings in a configuration file from the command-line parameters that are passed to it. In this sample, that attackers relied on the configuration file.

Command-Line Switch	Profile Key	Function
S or s	SearchPath%d	Determining the drives to be searched. This setting is a “required drives to search documents in” as the tool still enumerates logical drives that are available.
C or c	SavePath0	The path where documents will be copied to. It is set to <directory of tool>\cache. The tool appends “dir” to the value it gets here.
Y or y	Type%d	File extensions to be collected. By default this is set to “.doc”, “.pdf”, and “.txt”.
P or p	PositiveKey%d	File names to be collected. If this is not set, it will proceed to collect files that meet the other criteria.

Command-Line Switch	Profile Key	Function
N or n	NegativeKey%d	File names not to be collected. If this is not set, it will proceed to collect files that meet the other criteria.
I or i	Size0	Maximum size of file to be collected. This is required.
	Sleep0	Number of milliseconds for the tool to sleep after it finishes collecting files. The default value is 7,200,000.

Table 3. TSPY_PILFERDOC.ZBGK sample's settings

TSPY_PILFERDOC.ZBGK appears more as an archiving tool than as a piece of malware. Aside from what is given in the configuration file or the command-line switch, it enumerates the logical drives and files to be searched while simultaneously avoiding the A and B drives, as well as removable drives. The files that are discovered are checked against the criteria given by the following: the p or PositiveKey%d setting (if available), the n or NegativeKey%d setting (if available), the maximum file size restriction, and the list of file extensions to be harvested.

The files are copied to what is set as the save path sequenced with the string \dir\. After the file enumeration is done and all the target files have been copied, the tool launches rar.exe (an archiving tool) via cmd.exe. The name of the archive is a date-time string where all the “ : ” are converted to “ - ” and placed in the save path directory. It is worth noting that the save path and the name of the archive will be important in trying to correlate the tools that work together further into this analysis. A hard-coded password is also set on the archive. Afterward, all the copied files are deleted except for the RAR files.

2. TSPY_KARYAL.ZAGK

This tool complements TSPY_PILFERDOC.ZBGK as it sends files out of the network. While it can function as a standalone exfiltration tool (since it can enumerate files), it does not have the controls that TSPY_PILFERDOC.ZBGK has in sorting through the files to be collected and compressing the files before sending them out.

Switch	Description
d	Finds the IP address to be connected to and for sending the files to
p	Finds the port to be used
r	Determines the number of threads to be created
c	Clean-up flag that deletes the file after it has been sent
f	File or directory to be sent or to search in

Table 4. The command-line parameters used by TSPY_KARYAL.ZAGK

TSPY_KARYAL.ZAGK was assigned the save path directory of TSPY_PILFERDOC.ZBGK as a parameter, implying that if it works as programmed, TSPY_KARYAL.ZAGK will send the RAR archives created by the scraper. As aforementioned, this tool has the ability to enumerate files if the f switch is set to a directory. It recursively finds files in the search directory and adds each file or directory that is found in the internal structure for subsequent sending. Since TSPY_KARYAL.ZAGK sends all the files found in the set directory, TSPY_PILFERDOC.ZBGK's clean-up feature becomes important in deleting copies. If the scraper fails to delete the files archived, TSPY_KARYAL.ZAGK can end up sending all the copied files and suspiciously increasing network traffic. Evidence also shows that the tools do not have persistence mechanisms, which means that the attacker is responsible for timing the execution of each tool.

For every file it finds, a thread is created to send the file out. The tool also does both thread management and the closing of old threads.

Metadata, which includes both the computer name and the username, is sent first. The C&C response contains information on whether the contents of the file are sent or not. We also observed that there is a minimum file size requirement before a file is sent, with all packets encrypted using the exclusive operator (XOR). After the file is sent out, if the c switch is used, the tool deletes the file that it sent.

3. HKTL_FILEDOWNLD

This tool downloads a file from a site and writes that file locally on the drive. The main function of this tool initially appears to be that of a downloader of additional tools in the infected machine. However, closer inspection shows that it downloads documents — specifically PDFs — from a given site.

It also has four parameters: The first contains the complete URL of the site that the tool downloads from. The second is a date, the third is a timestamp, and the fourth is an ID. Of the four parameters, only the first is required while the rest of the parameters are optional.

This tool requires the complete URL of the target file (or files) and file names to be retrieved, both of which come from another tool. The next tool might have helped the attacker get some of the aforementioned necessary information.

4. PowerShell database tool

This is a PowerShell script that is used to query a MySQL database. The script takes in a connection string that includes the server, unique identification (UID), password, and the database. It also takes in an SQL query string that it will execute. The following is a typical example of what is passed to the tool:

```
"Run-MySQLQuery -ConnectionString 'Server=<server>;Uid=<account name>;Pwd=<password>;database=<database name>' -Query <SQL query> -Dump"
```

Figure 2. The connection string that is used to query a MySQL database

Based on an observation of the command-line parameters passed to this tool, the attacker knew the database servers, root accounts, database names, and general information that the database holds. In all observed cases, queries were used to get document records based on the date or record ID. The results of the query were written in a CSV file.

Although the tool does not get the documents, the information that it gathers can be used for another tool, such as HKTL_FILEDOWNLD. Since this tool looks into the databases, it is possible that at least one database holds the URL needed for HKTL_FILEDOWNLD. Coincidentally, we noticed a five- to seven-day gap between the execution of the tools. At this point in the investigation, however, we could not make direct connections between the two tools.

5. **HKTL_FTPEXFIL**

This tool differs slightly from TSPY_PILFERDOC with regard to how it copies files to its temporary folder. Instead of directly copying the files that it finds, it writes the file paths on a text file. The text file is read and each file path that it finds is then copied to the temporary directory of the exfiltration tool. The name of the temporary directory is based on the system date, and the resulting directory follows the format “<tool path>\yyyymmdd”. The files are archived using 7-Zip embedded in itself and encrypted by a simple XOR key. The files are sent out using FTP along with some metadata that had been written to the text file earlier. The tool takes the following parameters: the path to search files in, the IP address of the FTP server, the username and the password to access the FTP, and the hosts.

The files in the temporary directory are deleted to hide what HKTL_FTPEXFIL has exfiltrated.

6. **BKDR_NULTUS.ZCGK**

At this point in our investigation, not much can be written about this tool except that it was seen to access documents on drives in the machine. It was placed in a startup folder and proved hard to reverse-engineer. However, we will elaborate on this tool as the study unfolds.

TSPY_PILFERDOC.ZBGK works with TSPY_KARYAL.ZAGK as it is responsible for gathering the documents from the local drive with TSPY_KARYAL.ZAGK. The directory where the former creates the RAR archive is given as a parameter to the latter, meaning that the files found in that directory are sent out by TSPY_KARYAL.ZAGK. TSPY_KARYAL.ZAGK runs as a scheduled task set to run hourly. On the other hand, TSPY_PILFERDOC.ZBGK is either executed through a command via a shell backdoor or set to run once via an autostart registry entry.

The company uses a document management system where the back-end database is MySQL. As the PowerShell database tool is used to query the said type of database, it tries to get the latest document records or specific document records based on record IDs. It is possible that the information that the PowerShell script gets is used by HKTL_FILEDOWNLD.

Since we worked on the assumption that this is an attack that was perpetrated using one intrusion set, the first four tools working together therefore maximizes the attackers' search function and gathers as many documents as the tools can. The first pair of tools gathers documents on drives while the second pair gets documents saved in the document management system.

However, BKDR_NULTUS.ZCGK does not appear to fit. Considering that TSPY_PILFERDOC already has the ability to get files from drives, it seems that BKDR_NULTUS.ZCGK functions as a duplicate. We also noticed that the way BKDR_NULTUS.ZCGK is written significantly differs from the others as it employs a packing or encryption mechanism that was not observed with the others. It is also possible that TSPY_PILFERDOC is an updated tool used by the attacker.

Backdoors

Shell backdoors facilitated the routine's spread across the network, including its capabilities of moving laterally and deploying tools to gather user accounts, among others.

Here, we enumerate the backdoor tools that we found:

1. BKDR_HANNOG

This backdoor opens a cmd.exe process to execute commands received from a remote website. The commands are decrypted and passed onto cmd.exe using a pipe. The result of the execution is read from the pipe, encrypted, and sent back, with the IP address of the site given as a parameter from the two variants that we found.

These backdoor tools do not contain code to set their persistence. Instead, they receive a command to create scheduled tasks that run once at a specific time. They also have the capability to drop other tools in the system.

We discovered three distinct variants: an x86 executable, an x64 version that reflectively loads a DLL file packed in its resource section, and an x86 version with a hard-coded IP address of the C&C. We analyzed the executed commands through this backdoor:

Purpose	Command	MITRE ATT&CK Technique
Getting account information	"net user <user account> /domain"	T1087.002
Checking the currently logged user	"whoami"	T1033
Persistence mechanism for the backdoor	"schtasks /create /tn <taskname> /tr <backdoor filepath and parameters> /sc once /st <start time> /ru <user name>"	T1053.005
Persistence mechanism for a sideloaded threat	"reg.exe add hklm\software\microsoft\windows\currentversion\run /v <value> /t REG_EXPAND-ZA /d <file path of normal application>"	T1547.001

Purpose	Command	MITRE ATT&CK Technique
Executing a tool remotely via scheduled task	"schtasks /create /tn <taskname> /s <ip> /u <user account> /tr <backdoor filepath and parameters> /sc once /st <start time> /ru SYSTEM"	T1053.005
Executing an archiving/zippping tool to gather documents in the user directories	"-m5 -r -ta<yyyymmddhhmmss> -hpp<password> a <archive name> c:\users*.doc"	T1560.001
Lateral movement using the IPC\$ share with an admin account	"net use \<ip address> <password> /user:<user name>"	T1021.002

Table 5. Commands executed through BKDR_HANNOTOG

2. TROJ_SHELLLOAD

This sample reflectively loads another binary. It can be used to load other binaries or tools as long as it conforms to how the binary is unpacked and loaded in the memory. In this case, feedback data showed that this loaded a backdoor shell. In particular, x86 and x64 versions were discovered, and both versions execute commands via cmd.exe.

Purpose	Command	MITRE ATT&CK Technique
Checking the currently logged user	"/c quser", "whoami"	T1033
Getting account information	"net user <user account> /domain"	T1087.002
Getting the computer names connected to the domain	"net group Domain <Group name> /domain"	T1069.002
Displaying the running processes in a machine	"tasklist", "tasklist /svc"	T1057
Mapping the IPC\$ share using an admin account	"net use \<ip>\ipc\$ <password> /user:<user account>"	T1021.002
Executing remote scripts using wmic.exe	"wmic /node<ip> /USER:<user account> / PASSWORD:<password> process call create <file path>"	T1047

Table 6. Commands executed through TROJ_SHELLLOAD

Based on the behavior, TROJ_SHELLLOAD's final form differs from BKDR_HANNOTOG with regard to how it drops and executes tools laterally. First, it maps the IPC\$ share using an administrator account. Second, to execute a tool remotely, it uses wmic and executes a batch file while BKDR_HANNOTOG relies on the creation of the scheduled tasks remotely. Gone are the commands to manually install autostart mechanisms for tools that are dropped locally and remotely. TROJ_SHELLLOAD (or whatever it reflectively loads or drops) has its own mechanism.

3. TROJ_QUASLOAD.ZBGK

This is a .NET compiled library that loads and executes a companion binary, a compiled QUASAR remote access trojan (RAT) file. Based on feedback data, the final tool is a backdoor shell. In addition, it can also gather compiled and zipped archives. The observed commands executed by the shell are the following:

Purpose	Command	MITRE ATT&CK Technique
Getting account information	"net user <user account> /domain"	T1087.002
Mapping of remote machines using an admin account	"net use \<ip address> <password> /user:<user name>"	T1021.002

Table 7. Commands executed through TROJ_QUASLOAD.ZBGK

Though TROJ_QUASLOAD.ZBGK and TROJ_SHELLLOAD were written in different languages, the former is similar to the latter with regard to loading a binary in memory.

4. Backdoor4

Backdoor4 functions as a dropper and a shell backdoor. Based on the feedback, it can drop and execute other binaries and launch cmd.exe through the following commands:

Purpose	Command	MITRE ATT&CK Technique
Persistence mechanism for a sideloaded threat	"reg.exe add hkcu\software\microsoft\windows\currentversion\run /v <value> /t REG_EXPAND-ZA /d <file path of normal application>"	T1547.001
Adding new accounts	"net user <user name> <password> /add"	T1136.001
Modifying user accounts	"net localgroup administrators <created account> /add"	T1136.001

Table 8. Commands executed through Backdoor4

5. Web shell

Apache web servers are deployed in the environment of the organization. In this case, the attackers took advantage of these web servers to deploy web shells in some of the machines. It is possible that there is more than one type of web shell in the environment, and the following are the commands executed through these web shells:

Purpose	Command	MITRE ATT&CK Technique
Checking the currently logged user	"whoami"	T1033
Persistence mechanism for a sideloaded threat	"reg.exe add hklm\software\microsoft\windows\currentversion\run /v <value> /t REG_EXPAND-ZA /d <file path of normal application>"	T1547.001
Persistence mechanism for another tool	"schtasks /create /tn <taskname> /tr < filepath and parameters> /sc hourly /mo 2 /sd <start date> /ru system"	T1053.005
PowerShell execution of a script	-exe bypass -nop -w hidden -c Import-Module <powershell script><parameters>	T1059.001

Table 9. Commands executed through web shell

Based on the similarity of the file path parameters, T1033 and T1547.001 could have been executed through one web shell. The execution of the PowerShell stands out because the file path of the script is totally different.

6. BKDR_XRAT.ZCHB

This backdoor is a .NET-compiled shell backdoor. Based on feedback, it gets installed as a service and has exfiltration and dropping capabilities. The following are the commands that were executed:

Purpose	Command	MITRE ATT&CK Technique
Persistence mechanism for another tool	"schtasks /create /tn <taskname> /tr < filepath and parameters> /sc hourly /mo 2 /sd <start date> /ru system"	T1053.005
Persistence mechanism for a sideloaded threat	"reg.exe add hklm\software\microsoft\windows\currentversion\run /v <value> /t REG_EXPAND-ZA /d <file path of normal application>"	T1547.001
Checking for cmd.exe	"tasklist /fi "imagename eq cmd.exe""	T1057

Table 10. Commands executed through BKDR_XRAT.ZCHB

BKDR_XRAT.ZCHB closely resembles TROJ_QUASLOAD.ZBGK since both are written in .NET. However, BKDR_XRAT.ZCHB does not have to load a binary to work. Additionally, BKDR_XRAT.ZCHB is a compiled version 2.0 of XRAT, making the configuration samples different.

The variety of shell backdoors and the singularity in their purposes are noticeable. Although they were written and built differently, they both function to give the attacker the ability to run commands on the targeted machine. With the exception of BKDR_HANNOTOG and BKDR_XRAT.ZCHB, which can both drop files and send data out, the main functions of these tools are to connect outside and execute commands via cmd.exe.

Miscellaneous

Aside from the list of backdoors and exfiltration tools, we found a variety of other tools deployed in the system that support different functions for the routines. It should be noted that although this is not an exhaustive list of tools (since there could be others scattered in the network), it does contain the tools that we observed from the machines that we were given. As a result, the significance of these tools cannot be taken for granted.

1. **TROJ_CHINOXY.ZAGK**

TROJ_CHINOXY.ZAGK is a file dropper. As such, it drops a legitimate file of a phone manufacturer along with a malicious sideloaded .dll and places it in the startup folder for persistence.

2. **Malicious sideloaded DLLs**

The functions of these sideloaded DLLs are varied. We analyzed that some were used as droppers, and others were used as backdoor tools, while some were used as part of the exfiltration processes, among other functions observed.

3. **Procdump**

Procdump was used profusely during the attack to dump the memory of LSASS.

4. **Mimikatz**

Mimikatz was utilized heavily alongside Procdump.

5. **HackTool.Win32.NBTScam.A**

This was used to find name servers and open shares.

Tool Relationships

These tool relationships are based on the processes that dropped, launched, or created a persistence mechanism for another object.

1. Relationship A

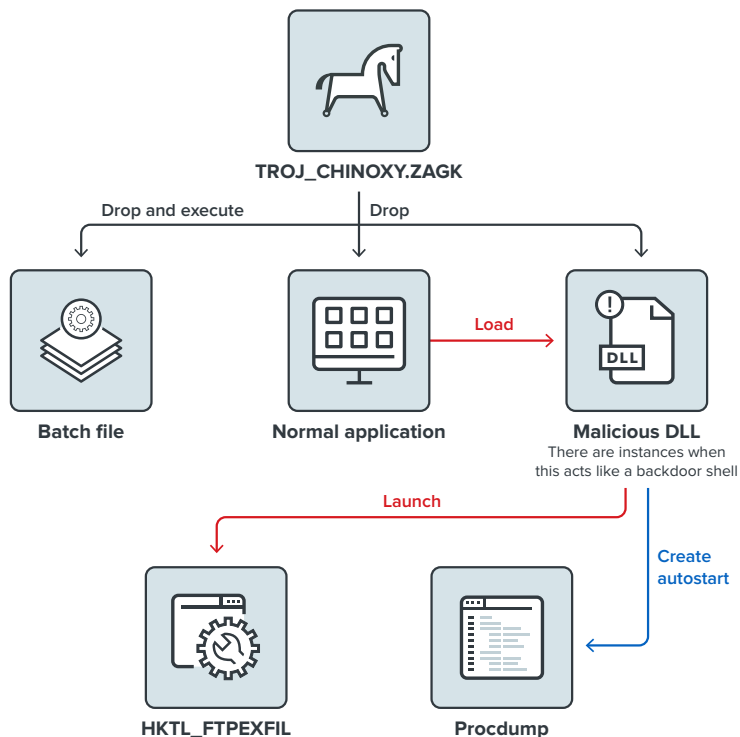


Figure 3. A relationship map of TROJ_CHINOXY to HKTL_FTPXFIL

Relationship A shows TROJ_CHINOXY.ZAGK dropping a normal application, a DLL used by the normal application, and a batch file. The normal application loads and executes the malicious DLL, while a sample shows evidence of the DLL launching HKTL_FTPXFIL.

2. Relationship B

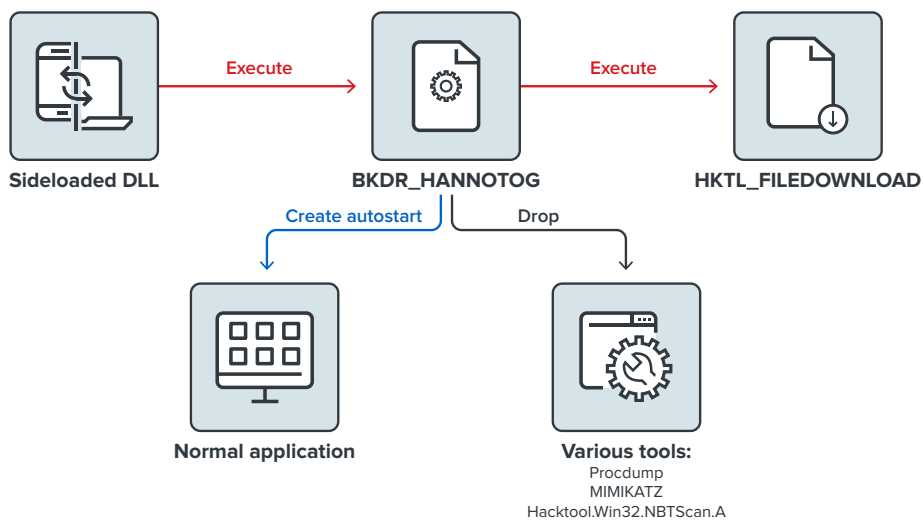


Figure 4. A relationship map of BKDR_HANNOTOG to HKTL_FILEDOWNLD

Relationship B shows BKDR_HANNOTOG, which is launched by a malicious sideloaded DLL. Considering that BKDR_HANNOTOG does not have its own persistence mechanism, this map is one piece of evidence of how it is launched.

BKDR_HANNOTOG creates autorun entries for various components, some of which are straightforward registry autorun entries for normal applications. These applications normally do not have such autostart entries, making this behavior stand out. We believe that these normal applications load malicious DLLs. There are also some instances when it created registry service entries for other malicious files.

BKDR_HANNOTOG also drops other tools such as Procdump, Mimikatz, and NBTScan to the environment. Meanwhile, the name of the file that is to be downloaded (and which will use HKTL_FILEDOWNLD), including a URL link, is missing. We consider the URL link and file names as the missing pieces of this diagram:

3. Relationship C

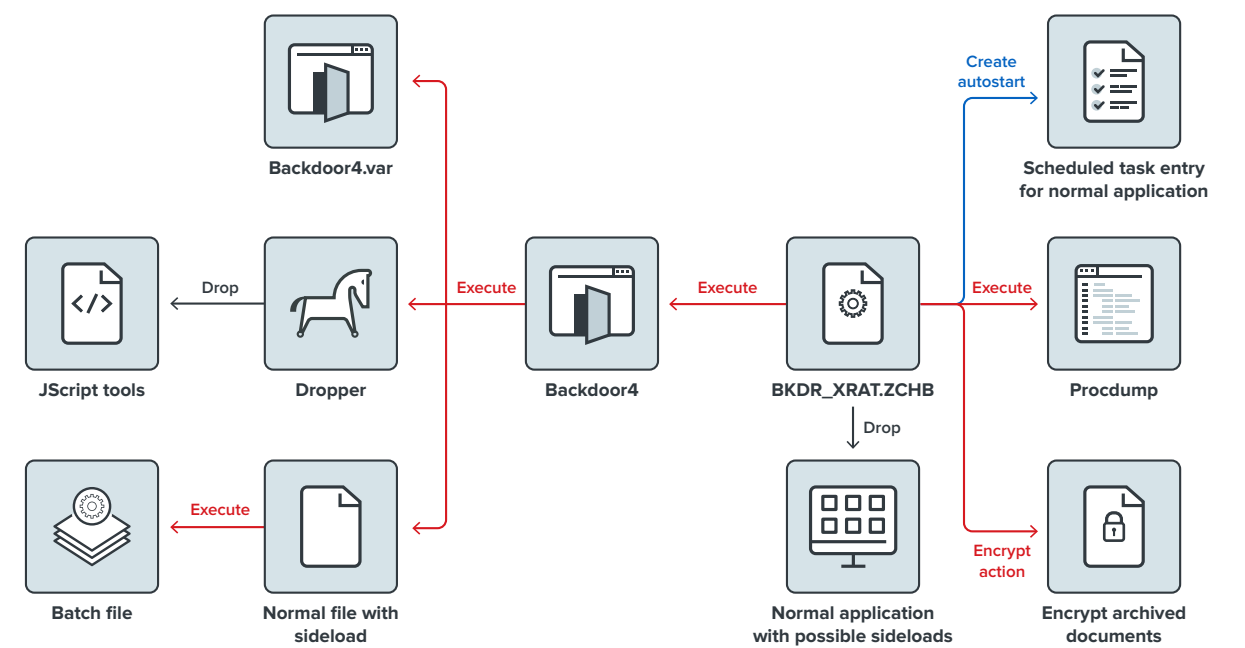


Figure 5. Relationship C shows the connections of the tools to BKDR_XRAT.ZCHB.

This diagram shows BKDR_XRAT.ZCHB as the backdoor that is responsible for remotely executing Backdoor4. It is significant to note that the behavior of BKDR_XRAT.ZCHB toward some document archives also stood out. For example, we observed it accessing and possibly encrypting .RAR files in a remote machine. Notably, the location of the said archives plays an important role later when the samples are bundled together based on locations.

4. Relationship D

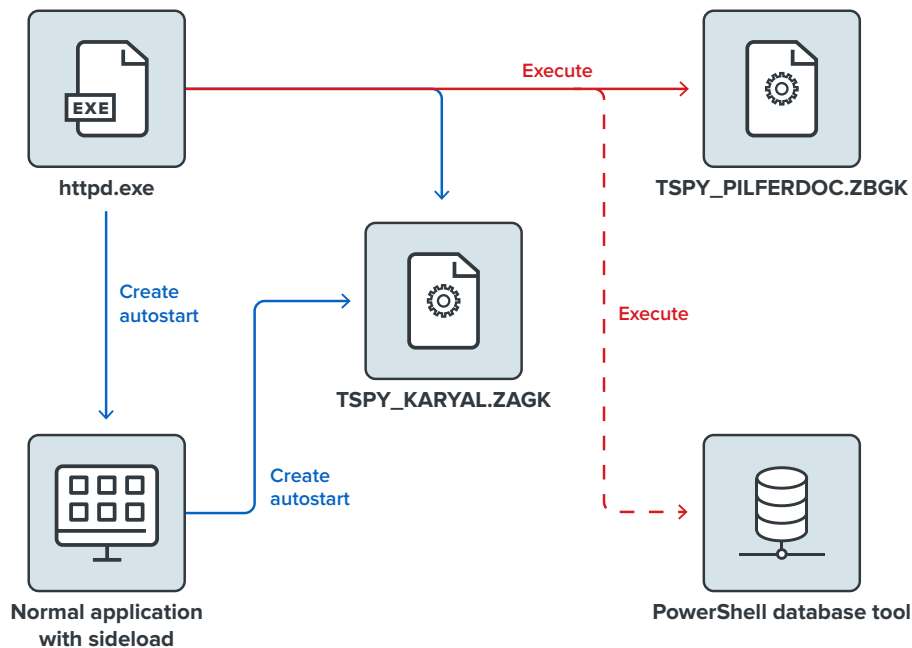


Figure 6. Modules connected to the web shells hosted by Apache

Relationship D shows the objects connected to the web shells that are hosted by Apache. There is a connection among TSPY_PILFERDOC.ZBGK, TSPY_KARYAL.ZAGK, web shells, and the sideloaded DLL. Aside from this, the PowerShell database tool appears separate from the three. TSPY_PILFERDOC.ZBGK and TSPY_KARYAL.ZAGK appear to have no use for the output of the script, which is a compilation of the records that were queried from a MySQL database in CSV format. It is possible that HKTL_FILEDOWNLD would make use of the data that is produced by the script mentioned.

Relationship A and B showed a “near complete” infection: a C&C module and a working payload, which in both cases are exfiltration tools. However, Relationship B’s missing piece is a tool for finding the documents that it is supposed to download, possibly from an internal site.

While Relationship D appears complete, there is nevertheless no proof that there is a working C&C module. Furthermore, the PowerShell script tool does not fit. On the other hand, Relationship C does not have any exfiltration tool directly linked to it.

The next set of tool relationships will show how the incomplete clusters are connected, considering the location and naming convention followed by the objects.

5. Relationship E

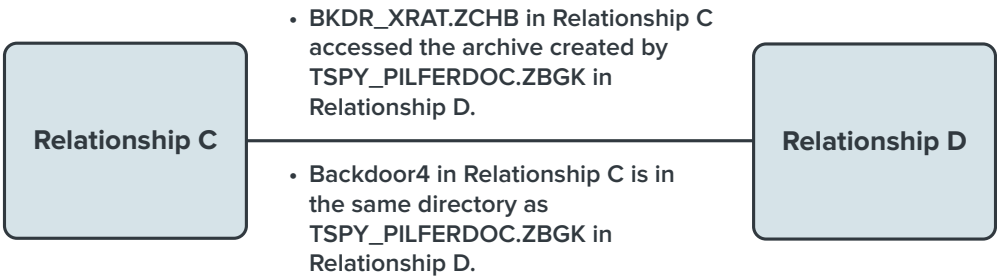


Figure 7. Relationship E shows how the objects line up in Relationships C and D.

Relationship E shows the possible connections between the components of Relationships C and D. BKDR_XRAT.ZCHB, found in Relationship C, attempts to encrypt archived files created by TSPY_PILFERDOC.ZBGK, which is in Relationship E. Backdoor4 is in a directory with the same name as TSPY_PILFERDOC. ZBGK, thereby establishing a “thin connection.” BKDR_XRAT.ZCHB, Backdoor4, and the objects in Relationship D are related.

6. Relationship F

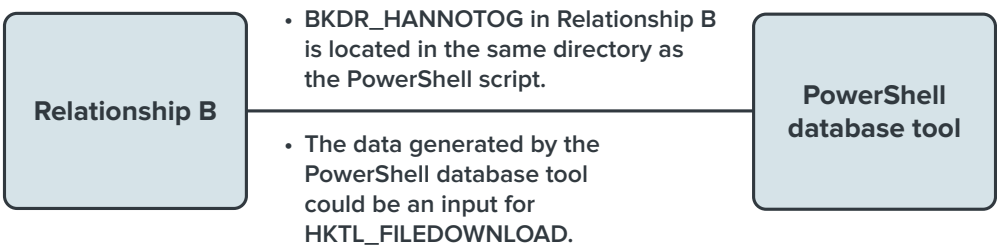


Figure 8. Relationship F shows the possible relationships between the modules in Relationship B and the PowerShell database tool.

The PowerShell database tool was originally in Relationship D with TSPY_PILFERDOC.ZBGK and TSPY_KARYAL.ZAGK. As aforementioned, neither of the tools has a need for the data generated by the PowerShell script. Additionally, neither TSPY_PILFERDOC.ZGBK nor TSPY_KARYAL.ZAGK fits with the two exfiltration tools.

However, we observed that the PowerShell script is located in the same directory as BKDR_HANNOTOG. Although the link seems superficial, considering the uniqueness of the tools’ locations increases the likelihood that the two are related. Moreover, it is possible that the required parameter of HKTL_FILEDOWNLD is generated by the PowerShell script. HKTL_FILEDOWNLD requires a complete URL and a file name to work, while the PowerShell scripts extract the file names of documents from a MySQL database that would likely be the back-end database of the organization’s document management system.

While there is no proof that whatever installed or executed BKDR_HANNOTOG or HKTL_FILEDOWNLD is the same as that which installed the PowerShell script, including the PowerShell in Relationship B completes the exfiltration capabilities of HKTL_FILEDOWNLD.

Updated MITRE ATT&CK

We looked further into the behaviors and routines to find stronger connections between these relationships, as well as a more thorough examination of both the techniques and tools that we initially found to be unconnected. As a result, we found new TTPs and listed them here:

MITRE ATT&CK Technique	Observed Behavior	In APT32?	In APT3?
T1505.003 – Server Software Component: Web Shell	Relationship E shows that either there is one web shell or two web shells installed.	Y	N
T1136.001 – Create Account: Local Account	Backdoor4 created a new account.	N	Y
T1057 – Process Discovery	BKDR_XRAT.ZCHB and TROJ_SHELLLOAD invoked tasklist via cmd.exe.	N	Y
T1560.001 – Archive Collected Data: Archive via Utility	TSPY_PILFERDOC.ZBGK and HKTL_FTPEXFIL compressed the files into archives.	N	Y
T1029 – Scheduled Transfer	TSPY_KARYAL.ZAGK was set to run every hour.	N	N
T1020 – Automated Exfiltration	TSPY_KARYAL.ZAGK was set to run every hour.	N	N

Table 11. New MITRE TTPs based on an analysis of behaviors and relationships

The additional TTPs we found, however, did not help much in identifying the intrusion set. T1136.001, T1057, and T1560.001 are the only known techniques that APT3 uses, while T1505.003 is identified with APT32. Although our initial analysis of the backdoor tool being central to the attack still stands, it appears that both the variety of the tools and the relationships that we found after investigating the infection further make the entire structure seem more complicated.

Additionally, the backdoor types looked different from each other. Although the functions that each provided as shell backdoors were the same, the way that they were written and used were different, just as the extra features that some of them had were also different.

The following are examples of these differences:

- There is a lack of a persistence mechanism in BKDR_HANNOTOG, whereas there is a presence of one in TROJ_SHELLLOAD.
- BKDR_XRAT.ZCHB and TROJ_QUASLOAD.ZBGK were written in .NET, while BKDR_HANNOTOG was written in a C-type language.

- The use of wmic by TROJ_SHELLLOAD is not found in the other backdoor types.
- The administrator shares that were used also varied.

The exfiltration tools also varied, although some of them duplicated the function of the others. For instance, HKTL_FTPXFIL has the same capability as TSPY_KARYAL and TSPY_PILFERDOC. It also appears that TSPY_KARYAL is taken over by BKDR_XRAT.ZCHB in some instances. The redundancies in the tools might be intentionally designed. In the possible instance that AV solutions detect and block at least one tool in the intrusion set, enabling a variety of functions in the tools and having more than one tool that can do the same task both ensure that the attack can still be successful.

There were also four “complete” clusters formed after grouping the tools based on the process trees and file path similarities. Based on how complete a cluster is, it could probably work independently of others. However, BKDR_NULTUS.ZCGK and TROJ_SHELLLOAD are still unaccounted for.

Overall, there seems to be a disjunction between the tools and the relationships formed. Both initial and secondary analyses give the impression that this is not a straightforward APT attack and that something else was happening in the environment at the time that the attack was occurring.

Suspected Groups and Related Campaigns

We looked into the binaries and compared them against known intrusion set tools. As a result, we learned that there were at least four distinct sets at play. Based on this, we enumerated and mapped the main sets.

Intrusion Set 1

Our analysis showed that the tools in this set can be attributed inconclusively to several groups, all of which used simplified Chinese language components. Some of the possible links that we found to be related to this particular intrusion set had already been published previously, although they can still be used by other groups.² The intrusion set is composed of a dropper, a sideloaded backdoor, and an exfiltration tool. Intrusion Set 1 seems to fit well with the behaviors that were observed in Relationship A. TROJ_CHINOXY.ZAGK, placed in the autostart folder, drops the normal application: a batch file and the malicious .dll, BKDR_CHINOXY.ZAGK. The .dll is sideloaded by the normal application. BKDR_CHINOXY.ZAGK, meanwhile, was observed as launching tools such as ProcDump and SysInfo. It also runs HKTL_FTPXFIL, which collects, archives, and sends data out.

For Intrusion Set 1, the tools identified by their detections are TROJ_CHINOXY.ZAGK, the sideloaded .dll BKDR_CHINOXY.ZAGK, and HKTL_FTPXFIL. The techniques that were mapped are:

MITRE ATT&CK Technique	Observed Behavior
T1547.001 – Boot or Logon Autostart Execution: Registry Run Keys/Startup Folder	Dropper1 was placed in a startup folder.
T1574.002 – Hijack Execution Flow: DLL Sideload	Dropper dropped the normal application in the %mytemp% folder along with the malicious .dll, BKDR_CHINOXY.ZAGK.
T1059.003 – Windows Command Shell	BKDR_CHINOXY.ZAGK launched cmd.exe, which it used to execute commands and tools.
T1053.005 – Scheduled Task/Job: Scheduled Task	Scheduled tasks were used to run tools such as ProcDump.
T1082 – System Information Discovery	SystemInfo was invoked.
T1087.002 – Account Discovery: Domain Account	Net user <user account> /dom was invoked on a couple of user accounts.
T1003.001 – OS Credential Dumping: LSASS Memory	ProcDump was used to dump the memory of LSASS.
T1560 – Archive Collected Data	HKTL_FTPEXFIL compressed the files that it collects.
T1048 – Exfiltration Over Alternative Protocol	HKTL_FTPEXFIL had the capability to send the data out by FTP.

Table 12. Intrusion Set 1 TTPs

Lotus Blossom

The tools, techniques, and motive of this intrusion set are similar to those identified with the group Thrip,³ which is known as part of the Lotus Blossom group. In our previously published works, such as our paper “Targeted Attack Trends in Asia-Pacific,” we also followed and identified this group as LStudio.⁴

Most of the tools and binaries that we found in Relationship B were clustered in this intrusion set. The binaries that belong here are either located in the system directory or in one of the subdirectories. BKDR_HANNOTOG is the shell backdoor that was widely used in this infection. Other backdoors were also used (such as BKDR64_WYZINA, TROJ64_SAGRUNEX.SMZTGD-A, and BKDR64_METREVHTTPS.ZCGK), although these were not as prominent as BKDR_HANNOTOG. Some of these tools were in fact even installed using BKDR_HANNOTOG.

For its exfiltration payload, BKDR_HANNOTOG was responsible for documents in the local file system, while HKTL_FILEDOWNLD was responsible for files in the document management system. BKDR_HANNOTOG triggered a compression tool to discover and archive files in the local machine, whereas HKTL_FILEDOWNLD downloaded document files from an internal repository. To download the documents, HKTL_FILEDOWNLD needed the names and locations of the documents. The information could have been obtained by the PowerShell script that is located in a directory where some of the BKDR_HANNOTOG samples had been placed. The parameters that were given to the PowerShell script searched for document information from the database given using either the record date (for example, the most recent documents that had been added) or the record ID. It is thus safe to assume based on this

that the attacker was looking for the latest documents that had been added to the system, or for certain documents that belong to a particular record.

If the PowerShell script is part of this intrusion set, this means that there is an unidentified web shell that should be part of this set as well. The tools that we found in this set are:

1. BKDR64_HANNOTOG.ZAGK
2. BKDR_HANNOTOG.ZBGK
3. BKDR_HANNOTOG.ZCGK
4. TROJ64_SAGRUNEX.SMZTGD-A
5. BKDR64_METREVHTTPS.ZCGK
6. HKTL_FILEDOWNLD
7. Hacktool.Win32.NBTScan.A

The next set of tools are more generic in nature, which makes it difficult to associate them with a particular set. Still, it is more likely that they are also part of the following intrusion set, since the names, as well as where they are installed, are consistent with the tools listed prior. These tools are:

8. BKDR64_WYZINA.ZBGK
9. TROJ_CMDINJECT
10. TROJ64_CMDINJECT
11. HackTool.Win64.MIHKATZ.AO
12. HKTL_MIMICLI
13. NBTScanner

The techniques used for this set are:

MITRE ATT&CK Technique	Observed Behavior
T1547.001 – Boot or Logon Autostart Execution: Registry Run Keys/Startup Folder	This was used to start a normal application with a sideload and as the autostart of CMDINJECT.
T1543.003 – Create or Modify System Process: Windows Service	Services were created for some of the tools that were deployed.
T1574.002 – Hijack Execution Flow: DLL Sideload	A normal application with a sideloaded .dll installed BKDR_HANNOTOG.
T1059.003 – Windows Command Shell	BKDR_HANNOTOG launched cmd.exe, using it to launch commands and tools.
T1053.005 – Scheduled Task/Job: Scheduled Task	Scheduled tasks were used to run tools including BKDR_HANNOTOG itself.

MITRE ATT&CK Technique	Observed Behavior
T1033 – System Owner/User Discovery	whoami was launched via BKDR_HANNOTOG.
T1087.002 – Account Discovery: Domain Account	Net user <user account> /dom was invoked on a couple of user accounts.
T1078 – Valid Accounts	Accounts with administrator privileges were used for lateral movement.
T1046 – Network Service Scanning	The intrusion set contained an NBTScanner.
T1003.001 – OS Credential Dumping: LSASS Memory	HackTool.Win64.MIMIKATZ.AO was used.
T1021.002 – Remote Services: SMB/Windows Admin Shares	Net use was abused with valid user credentials to move laterally.
T1005 – Data from Local System	BKDR_HANNOTOG, using an archiver, enumerated document files in c:\users.
T1560.001 – Archive Collected Data: Archive via Utility	Evidence suggests that BKDR_HANNOTOG used a tool to compress files from the local machine.

Table 13. Lotus Blossom TTPs

Intrusion Set 2

This is a new intrusion set that we neither traced in previous detections, nor observed with any documented APT group. It might be connected to one of the intrusion sets that we found, although we did not observe any concrete evidence that connects both Intrusion Sets 1 and 2.

Our analysis showed that the exfiltration tools that worked together belong to one set. The cluster of tools closely resembled the tools that we found in Relationship D. TSPY_PILFERDOC.ZBGK searches for documents and compresses them. The archive that is created by TSPY_PILFERDOC.ZBGK is then placed in another directory for exfiltration using TSPY_KARYAL. Both samples have .dll versions that are sideloaded by normal applications.

At this rate, it appears that the group experienced a gap between spreading and controlling the execution of the exfiltration tools. In turn, the connection of the objects in Relationships C and D responded to this gap, as also described in Relationship E.

BKDR_XRAT.ZCHB and Backdoor4 are shown in Relationship C as the C&C tools that allow the attacker to set up the exfiltration tools. Bundling these tools in this cluster is tricky, mainly because they are widely used; in fact, multiple sets use XRAT and these sets are available in online repositories.

Considering the connections made in Relationship E, the tools in this intrusion set are:

1. TSPY_PILFERDOC.ZBGK
2. TSPY_KARYAL.ZAGK
3. TSPY_PILFERUSE.ZCGK
4. BKDR_XRAT.ZCHB
5. Port scanner tool

The techniques used for this set are:

MITRE ATT&CK Technique	Observed Behavior
T1547.001 – Boot or Logon Autostart Execution: Registry Run Keys/Startup Folder	Used as a persistence mechanism for BKDR_XRAT.ZCHB. TSPY_PILFERDOC.ZBGK used a RunOnce registry value to execute pilferdoc set by another process, likely by a backdoor shell or web shell. This was not set by the exfiltration tool, however.
T1574.002 – Hijack Execution Flow: DLL Sideload	Normal application with either TSPY_PILFERDOC.ZBGK or TSPY_KARYAL.ZAGK was used.
T1059.003 – Windows Command Shell	BKDR_XRAT.ZCHB launched cmd.exe.
T1053.005 – Scheduled Task/Job: Scheduled Task	TSPY_KARYAL.ZAGK was launched using a scheduled task.
T1033 – System Owner/User Discovery	whoami was invoked by a web shell and the results were stored in the same directory where the other tools were placed.
T1087 – Account Discovery	Net user was invoked by XRAT.
T1003.001 – OS Credential Dumping: LSASS Memory	ProcDump was used at one point to dump LSASS.exe.
T1136.001 – Create Account: Local Account	An account was created by XRAT and Backdoor4 using “net user <account> /add”. This was added to the administrator group later and eventually deleted.
T1046 – Network Service Scanning	This intrusion set contained a port-scanning tool.
T1047 – Windows Management Instrumentation	XRAT launched tools remotely using wmic.
T1005 – Data from Local System	TSPY_PILFERDOC.ZBGK enumerated document/office files in the local drive.
T1083 – File and Directory Discovery	TSPY_PILFERDOC.ZBGK enumerated files using Windows APIs.
T1029 – Scheduled Transfer	TSPY_KARYAL.ZAGK was set to run every hour.
T1020 – Automated Exfiltration	TSPY_KARYAL.ZAGK was set to run every hour.
T1136.001 – Create Account: Local Account	Backdoor4 added a created account in the administrator local group.
T1057 – Process Discovery	XRAT checked for cmd.exe using tasklist.

MITRE ATT&CK Technique	Observed Behavior
T1560.001 – Archive Collected Data: Archive via Utility	TSPY_PILFERDOC compressed the files into archives.
T1505.003 – Server Software Component: Web Shell	KARYAL and PILFERDOC had some connections with a web shell.

Table 14. Intrusion Set 2 TTPs

OceanLotus

This intrusion set uses the same custom packer as KerrDown,⁵ a tool that is attributed to OceanLotus, which is also known as APT32. Moreover, the C&C used in these samples that we analyzed as OceanLotus' C&C are based on open-source intelligence (OSINT).

Based on our analysis, the attack did not require multiple tools with specific functions. Rather, it used the same set of tools with different encryptions to give the impression that it used multiple files. The backdoor tool that the attack used also doubled as the exfiltration tool, and the environment indicated the second stage payload of the attack. The tools that were found in this set are BKDR_NULTUS.ZCGK and a sideloaded .dll, BKDR_NULTUS.ZAGK.

The techniques used for this set are:

MITRE ATT&CK Techniques	Observed Behavior
T1547.001 – Boot or Logon Autostart Execution: Registry Run Keys/Startup Folder	The samples were placed in the startup folder.
T1574.002 – Hijack Execution Flow: DLL Sideload	BKDR_NULTUS.ZAGK was a sideloaded .dll.
T1005 – Data from Local System	Tools enumerated document/office files in the local drive.
T1039 – Data from Network Shared Drive T1025 – Data from Removable Media	Tools enumerated documents in non-local drives.

Table 15. OceanLotus TTPs

On the basis of OceanLotus' matched intrusion set, we suspect that APT32 could be one of the groups behind the attack on the victim organization. Notably, their toolset is vastly different from the other samples that were acquired. Aside from the feedback data being scarce, the machine name and the username were also needed to decrypt the samples.

Theoretical Attribution

There are at least four distinct intrusion sets controlled by at least one or more groups responsible for the attack on the victim organization. Among the four intrusion sets identified, two of these — Lotus Blossom and OceanLotus — have strong links to groups that other security researchers have written about. Intrusion Set 1 has been mentioned as having links to groups that use simplified Chinese language, while Intrusion Set 2 is unique in that it has not been identified or attributed to any group. It is also possible that there is a piece that we did not find, or that there is a piece that is not visible and that connects Intrusion Set 2 with the other intrusion sets.

If the main goal of the four intrusion sets was data exfiltration, then it follows that all four routines employed almost the same techniques. Additionally, almost all the techniques used in OceanLotus are present in all the other intrusion sets. The small differences would also be noticeable if the actual events were examined. For instance, the administrative share that BKDR_XRAT used was different from that which BKDR_HANNOTOG employed. No single set had exclusive rights to a technique or a set of techniques, either; instead, a tool or technique could be used at any point if it was necessary or if it made the task simpler.

The differences in the writing styles of the tools are evident. For instance, the way that the samples in OceanLotus were packed is in contrast with how “naked” the other tools were. The way that HKTL_FTPEXFIL enumerated files is also different from how TSPY_PILFERDOC proceeded with its enumeration: The former wrote the file paths in a text, while the latter kept the file paths in a structure in memory.

Were the techniques used to figure out the intrusion sets, it would be difficult to focus on one set.

Although the clustering of techniques could narrow down the suspects, it is possible that multiple sets could also be identified. In this case, the relationships and the comparisons with the existing tools provided a clearer division among the sets.

Conclusion

In studying an infection, an investigating team should be able to take a macro perspective of what is happening in the whole environment. Since one of the attack's probable main functions in order for it to work is to spread across multiple binaries and/or files (and in some instances different machines), finding the connections among the tools would give an analyst a better idea of the attack and its details.

To find these connections, the tools that provide visibility of what happens in the network, as well as any suspicious processes, would help significantly. In this particular case, for example, there was difficulty in identifying the sideloaded DLLs.

In building internal models, there is no singular procedure, model, or process for tracking and hunting intrusion sets for attribution. Security researchers, companies, and analysts could have their own models or implementations for grouping TTPs (such as the Cyber Kill Chain⁶ or the Diamond Model⁷). One suggestion would be for all these researchers, companies, and analysts to combine their models, use known tools, and share their evidence to the public for continuous investigation and intelligence gathering. Such collaborative effort would be useful, especially for large-scale enterprises or agencies where the number of components (such as memory and disk images) that analysts might have to contend with in different environments would be time-intensive, considering that the analyses have to be done on a host-to-host basis.

In bridging this gap between correlating objects and events, an EDR solution would be effective in monitoring and detecting these scenarios. EDR frameworks and services have features such as root cause analysis (RCA) that make it easier for teams to trace and rebuild tool relationships via record endpoint system-level behaviors, events (such as user, file, process, registry, memory, and network events) and command-line inputs (CLIs) used in compromised endpoints. It also has network-model capability that can show how machines interact. By deploying multiple sensors on the machines' network or endpoint activities, it can trigger an alert for any potential compromise and provide better visibility. This is significantly beneficial for analysts when correlating multiple observables and events, retracing and recreating the story of how the attack happened.

For the effective identification of intrusion sets based solely on the MITRE technique, the monitoring system installed should be able to capture and log as many events as possible. In this case, the identification

narrowed down two groups and identified these as suspects by comparing the retrieved tools against known APT tools and the clustering of tools based on their relationships with each other. These two approaches complemented each other since some of the tools are known to be used by multiple intrusion sets.

Sponsored attacks like these might have varying objectives for intrusions. The continued flow of intelligence that attackers gain from high-value targets promises a high premium for the varied potential use of the data that could be gathered in favor of the sponsoring group via espionage.

Therefore, the high regard for both the research on the tools that a targeted company uses and the personnel who might have access to critical data plays a crucial, motivational role in ensuring that the gathered intelligence is continually updated and unnoticed for as long as possible. In turn, the process of ensuring this requires the attackers to continuously adjust their tools and methods (whether these are publicly documented or not) in order to meet the sponsoring organization's needs.

TTPs, in particular, significantly change depending on a number of variables: the objective(s), the target, their existing security posture, the approaches taken for initial intrusion, the success rates of available and modifiable tools as payloads, the location of both the attacker and the target, and the formats that they would want the data to be in, among other factors.

Therefore, the victim organization is in a unique position to catalogue the indicators of compromise (IOCs), tools, techniques, and tactics that the attackers can use for future attacks. These victim organizations could gain access to a more comprehensive set compared to what has been previously reported or documented by different security vendors. Enterprises and agencies that are exposed to incidents such as these are thus privy to a huge and distinct amount of information that has to be analyzed and studied.

It is also worth taking into consideration that the process will not be straightforward.⁸ It might not follow an existing procedure that has already been used as a standard by security organizations. Such investigations require the proper tools in order to include components in the unclear area, such as undefined or unclear logs that can be used in order to prove or identify the missing links of related and unidentified IOCs and techniques that were used to facilitate the attacks. Assuming that investigating teams are knowledgeable in making sense of these logs and correlating the actions that are registered, SOCs could still learn that there are missing pieces that are necessary to uncover the entire story.

Unfortunately, in addition to possible missing technologies, not all teams or organizations are likewise equipped to do analyses such as these since correlation is not a straightforward process.

References

- 1 The MITRE Corporation. (2015). *The MITRE Corporation*. “MITRE ATT&CK.” Accessed on Sep. 11, 2020 at <https://attack.mitre.org/>.
- 2 Jon Gross and Cory Kennedy. (Aug. 19, 2020). *RiskIQ*. “RiskIQ Adventures in Cookie Land – Part 1.” Accessed on Sep. 25, 2020, at <https://community.riskiq.com/article/5fe2da7f>.
- 3 Symantec threat hunter team. (Sep. 9, 2019). *Broadcom*. “Thrip: Ambitious Attacks Against High Level Targets Continue.” Accessed on Jun. 28, 2020, at https://symantec-enterprise-blogs.security.com/blogs/threat-intelligence/thrip-apt-south-east-asia?om_ext_cid=biz_social3_AMS_NAM-US_facebook_us,blogs-social,FY20-Q1,Blog,Threat%20Intelligence,org.
- 4 TrendLabs. (2014). *Trend Micro*. “Targeted Attack Trends in Asia-Pacific.” Accessed on Sep. 2, 2020, at <https://documents.trendmicro.com/assets/threat-reports/rpt-1h-2014-targeted-attack-trends-in-asia-pacific.pdf>.
- 5 Vicky Ray and Kaoru Hayashi. (Feb. 1, 2019). *Palo Alto Networks*. “Tracking OceanLotus’ new Downloader, KerrDown.” Accessed on Jun. 15, 2020, at <https://unit42.paloaltonetworks.com/tracking-oceanlotus-new-downloader-kerrdown/>.
- 6 Instituto Nacional de Ciberseguridad de España (INCIBE-CERT). (Oct. 27, 2016). *Instituto Nacional de Ciberseguridad de España (INCIBE-CERT)*. “Cyber Kill chain applied to ICS.” Accessed on Sep. 25, 2020, at <https://www.incibe-cert.es/en/blog/cyber-kill-chain-applied-ics>.
- 7 United States Department of Defense. (Jul. 5, 2013). *United States Department of Defense*. “The Diamond Model of Intrusion Analysis.” Accessed on Nov. 9, 2020, at <https://apps.dtic.mil/dtic/tr/fulltext/u2/a586960.pdf>.
- 8 2016 Public-Private Analytic Exchange Program Team. (Sep. 9, 2016). *Office of the Director of National Intelligence*. “Cyber Attribution Using Unclassified Data.” Accessed on Sep. 25, 2020, at <https://www.dni.gov/files/PE/Documents/CyberAttribution.pdf>.



TREND MICRO™ RESEARCH

Trend Micro, a global leader in cybersecurity, helps to make the world safe for exchanging digital information.

Trend Micro Research is powered by experts who are passionate about discovering new threats, sharing key insights, and supporting efforts to stop cybercriminals. Our global team helps identify millions of threats daily, leads the industry in vulnerability disclosures, and publishes innovative research on new threat techniques. We continually work to anticipate new threats and deliver thought-provoking research.

www.trendmicro.com

