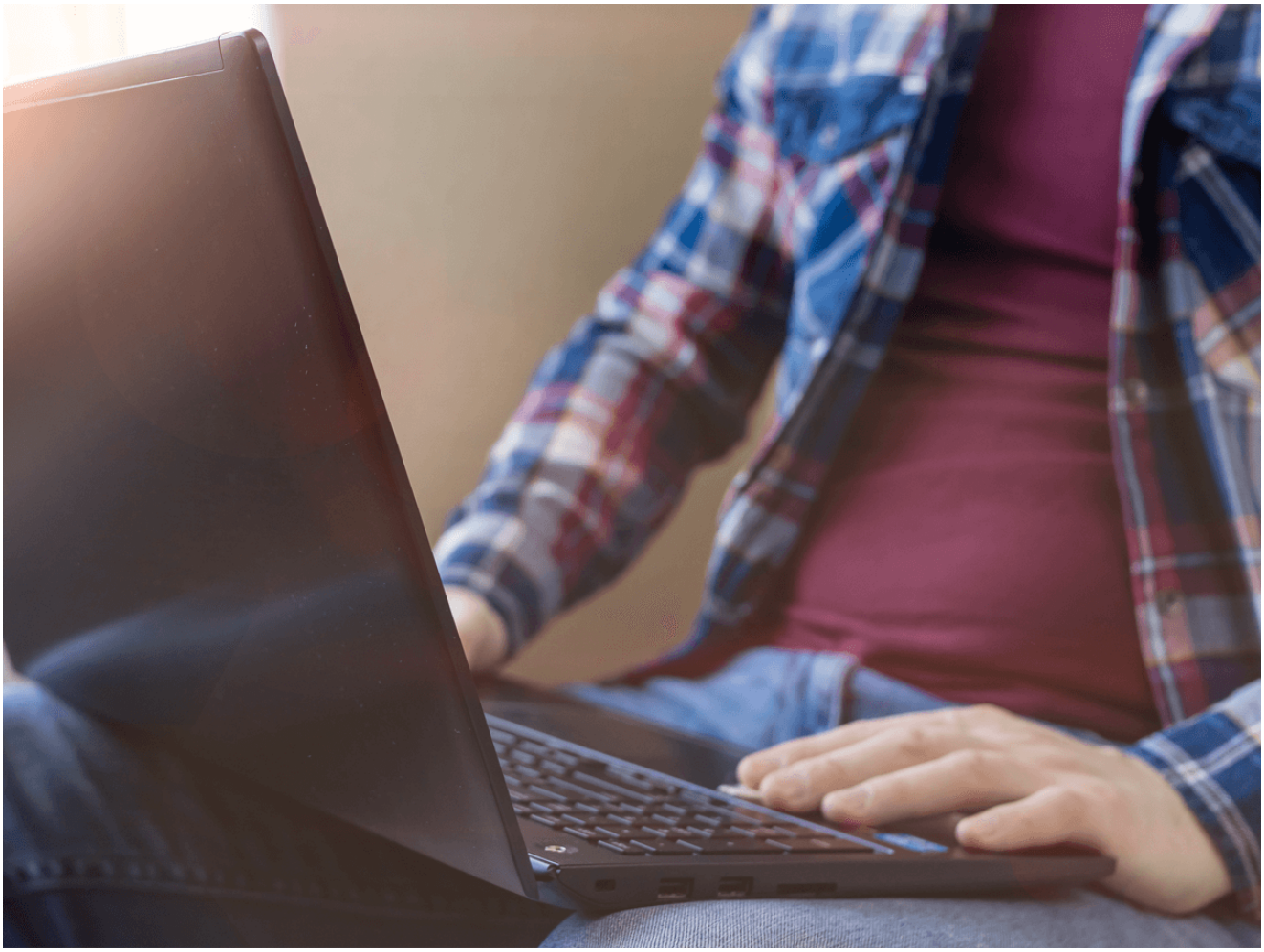
Bandook - A Persistent Threat That Keeps Evolving

fortinet.com/blog/threat-research/bandook-persistent-threat-that-keeps-evolving

December 21, 2023



Article Contents

By Pei Han Liao | December 21, 2023
Affected Platforms: Microsoft Windows
Impacted Users: Microsoft Windows
Impact: Remote attackers gain control of the infected systems
Severity Level: Critical

Bandook malware is a remote access trojan that has been continuously developed since it was first detected in 2007. It has been used in various campaigns by different threat actors over the years. FortiGuard Labs identified a new Bandook variant being distributed via a PDF file this past October. This PDF file contains a shortened URL that downloads a password-protected .7z file. After the victim extracts the malware with the password in the PDF file, the

malware injects its payload into msinfo32.exe. In this article, we will briefly introduce Bandook's behavior, provide detailed information about the modified elements of this new variant, and share some examples of the mechanism of its C2 communication.

Injector

The injector component decrypts the payload in the resource table and injects it into msinfo32.exe.

Before the injection, a registry key is created to control the behavior of the payload. The key name is the PID of msinfo32.exe, and the value contains the control code for the payload. Once executed with any argument, Bandook creates a registry key containing another control code that enables its payload to establish persistence, and it then injects the payload into a new process of msinfo32.exe. There are two registry keys, shown in Figure 1.

Figure 1: The registry keys written by Bandook.
A variant reported in 2021 required four control codes and created four processes of explorer.exe that it injected in a single execution. This new variant uses less control code and makes a more precise division of tasks.

Payload

Figure 2: The execution flow of the payload.
Figure 2 is the overview of the payload. Once injected, the payload initializes strings for the key names of registries, flags, APIs, etc. After this, it uses the PID of the injected msinfo32.exe to find the registry key and then decodes and parses the key value to perform the task specified by the control code. Figure 3 shows the relationship between the key value and the payload's behavior. The control codes play the same role as previous variants, but strings are used instead of numbers.

Figure 3: Relationship between key value, command line, and payload.
The variant we found in October 2023 has two additional control codes, but its injector doesn't create registries for them. One asks the payload to load fcd.dll, which is downloaded by another injected process and calls fcd.dll's Init function. The other mechanism establishes persistence and executes Bandook's copy.

These unused control codes have been removed from even newer variants (430b9e91a0936978757eb8c493d06cbd2869f4e332ae00be0b759f2f229ca8ce).

Of the two remaining control codes, "ACG" is the main control code for an attack, while "GUM" establishes the persistence mechanism.

GUM Control Code

When the control code is “GUM,” Bandook drops a copy to the SMC folder in the appdata folder as “SMC.exe” or “SMC.cpl” and creates a registry key to automatically execute the copy. There are three registry keys to run SMC.exe.

Software\Microsoft\Windows\CurrentVersion\Run
Key name: SMC
Value: %APPDATA%\SMC\SMC.exe

Software\Microsoft\Windows NT\CurrentVersion\Winlogon
Key name: shell
Value: explorer.exe, %APPDATA%\SMC\SMC.exe

Software\Microsoft\Windows NT\CurrentVersion\Windows\
Key name: Load
Value: short path of %APPDATA%\SMC\SMC.exe

When the copy is SMC.cpl, the registry key and value are the following:

Software\Microsoft\Windows\CurrentVersion\Run
Key name: SMC
Value: %windir%\System32\controll.exe %APPDATA%\SMC\SMC.cpl

ACG Control Code

When the control code is ACG, the payload can download files for other modules, including fcd.dll, pcd.dll, an executable file, and others. This is an optional function based on flags set when the payload initializes. The files can also be downloaded from the C2 server when necessary. If fcd.dll is downloaded, Bandook calls its functions and passes the key names of the registry key as arguments. Similarly, many registry keys store information used in other actions.

An action may separated into several parts, and it’s necessary to piece all related commands and registry keys together. For example, C2 communication may use one command to write a registry key and a separate command to read it.

C2 Communication

First, Bandook sends victim information to its C2 server:

Figure 4: Traffic capture and AES decrypted data of the victim information.
If the C2 server is available, Bandook receives commands from the server, including *DJDSR^, @0001, @0002, and so on. While the string sequence in the newest variants reaches @0155, some are only used when sending a result to the server, and others only

exist in other modules. As shown in Figure 5, the payload doesn't use the command @0133, though it can be found in fcd.dll.

Figure 5: @0133 can be found in fcd.dll.
Despite the numbering, the payload only supports 139 actions. In addition, some special commands are only sent to the server under specific conditions. Since most actions are the same as in previous variants, we will focus on communications between Bandook and the C2 server using the new commands added to the most recent variants.

These actions can be roughly categorized as file manipulation, registry manipulation, download, information stealing, file execution, invocation of functions in dlls from the C2, controlling the victim's computer, process killing, and uninstalling the malware.

The data from the C2 server has the following format:

{Command}~!{Arg2}~!{Arg3}~!{Arg4}~!{Arg5}~!{Arg6}~!

The first argument is the command, which is necessary. Arg2 to Arg6 are optional.

Below are four examples of actions that require multiple commands and actions that have complex mechanisms.

@0003, @0004

This action is about file reading. If Arg3 is R, it keeps calling the Sleep function until the C2 server sends @0004 and its related arguments to Bandook. The @0004 command gives a value to determine from where to read the file or to just do nothing.

Finally, Bandook sends the file specified by Arg2 to the C2 server.

Figure 6: Process flow when Bandook receives @0003 from the server.
@0006, @0007

This action is about file writing. Similar to @0003, @0006 waits for @0007. @0007 determines how to write data from the C2 server to a local file.

Figure 7: Process flow when Bandook receives @0007 from the server.
@0126, @0127, @0128

This action executes a Python file. The main command is @0128, which calls a ShellExecute function to run a Python file {Parent directory}\Lib\dpx.pyc with arguments Arg2~Arg6. The {Parent directory} is stored in the registry key pthma under HKCU\Software. @0126 checks pthma's value and sends the result to the server. @0127 writes its Arg2 to pthma if fcd.dll is initialized in the victim's computer.

Additionally, some commands send special data to the server:

@0124

This action monitors the victim's screen and controls the computer. When Bandook receives this command, it overwrites the config file of Firefox pref.js with code hard-coded in the payload and disables protection mechanisms in Microsoft Edge:

Registry	Key name	Value
Software\Microsoft\Internet Explorer	TabProcGrowth	0
	NoProtectedModeBanner	1
Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\3	2500	3

After this, Bandook creates a virtual desktop and assigns it to a newly created thread (Thread_Control) that establishes a new communication with the C2 server. It first sends the string AVE_MARIA, followed by another packet containing the number 1, to the server.

Figure 8: The "AVE_MARIA" and number sent by Bandook.
If the server responds, Bandook creates another thread to keep sending screenshots to the server. This thread also sends two packets: the string AVE_MARIA and the number 0. In the meantime, Thread_Control receives coordinates and control codes from the server. These tasks include:

- Open the Run dialog
- Copy user data from Chrome to another folder and open another Chrome instance using a new directory and configurations. It uses the following command to help it run faster: cmd.exe /c start chrome.exe --no-sandbox --allow-no-sandbox-job --disable-3d-apis --disable-gpu --disable-d3d11 --user-data-dir={New folder}
- Copy user data to another folder and open another Firefox instance with the copied profile
- Execute Internet Explorer
- Terminate Microsoft Edge, enable its Compatibility Mode, and open another Edge instance with a new directory and configurations. It uses the following command to help it run faster:
- C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe --no-sandbox --allow-no-sandbox-job --disable-3d-apis --disable-gpu --disable-d3d11 --user-data-dir={New folder}
- Access specified windows

In addition, there are three new commands compared to the 2021 variant:

@0138

This writes encrypted backup URLs to the registry key kPYXM under HKCU\Software\AkZhAyV0\. When the current C2 server is unavailable, Bandook will decrypt it and try to access the URLs. The format of the decrypted data will look like this:

{URL}|{URL}|{URL}|

Bandook will extract URLs and try these sequentially if the previous URL is unavailable.

@0139

This command asks Bandook to parse cookies from the browser specified by the C2, including Chrome, Edge, and Firefox, and save the result as Default.json in a .zip file.

@0140

In the previous variant, @0140 is missing. This command asks Bandook to establish a persistence mechanism with sub_13160400, also called when the control code is GUM, as shown in Figure 9.

Figure 9: The new variant uses the same function in the control code and command.

Conclusion

This article unveils new details about the C2 mechanism of this long-existing malware and the new features in its latest variant. A large number of commands for C2 communication can be found in this malware. However, the tasks performed by its payload are fewer than the number in the command. This is because multiple commands are used for a single action, some commands call functions in other modules, and some are only used to respond to the server. Though the entire system is not observed in this attack, FortiGuard will continue monitoring malware variants and provide appropriate protections.

Fortinet Protections

The malware described in this report are detected and blocked by FortiGuard Antivirus as:

PDF/Agent.1F56!tr
W32/Injector.EQDO!tr
W32/Bandok.NAT!tr

FortiGate, FortiMail, FortiClient, and FortiEDR support the FortiGuard AntiVirus service. The FortiGuard AntiVirus engine is a part of each of those solutions. As a result, customers who have these products with up-to-date protections are protected.

The FortiGuard CDR (content disarm and reconstruction) service can disarm the malicious macros in the document.

We also suggest that organizations go through Fortinet's free NSE training module: NSE 1 – Information Security Awareness. This module is designed to help end users learn how to identify and protect themselves from phishing attacks.

FortiGuard IP Reputation and Anti-Botnet Security Service proactively block these attacks by aggregating malicious source IP data from the Fortinet distributed network of threat sensors, CERTs, MITRE, cooperative competitors, and other global sources that collaborate to provide up-to-date threat intelligence about hostile sources.

If you believe this or any other cybersecurity threat has impacted your organization, please contact our Global FortiGuard Incident Response Team.

IOCs

IPs

77[.]91[.]100[.]237
45[.]67[.]34[.]219

Files

8904ce99827280e447cb19cf226f814b24b0b4eec18dd758e7fb93476b7bf8b8
d3e7b5be903eb9a596b9b2b78e5dd28390c6aadb8bdd4ea1ba3d896d99fa0057
3169171e671315e18949b2ff334db83f81a3962b8389253561c813f01974670b
e87c338d926cc32c966fce2e968cf6a20c088dc6aedf0467224725ce36c9a525
2e7998a8df9491dad978dee76c63cb1493945b9cf198d856a395ba0fae5c265a
430b9e91a0936978757eb8c493d06cbd2869f4e332ae00be0b759f2f229ca8ce
cd78f0f4869d986cf129a6c108264a3517dbcf16ecfc7c88ff3654a6c9be2bca