GROUP-IB

← Blog

Rustam Mirkasymov
Senior Targeted Threat Researcher

Martijn van den Berk
Cyber Threat Intelligence Analyst

Boolka Unveiled: From web attacks to modular malware

Uncovering the operations of threat actor Boolka, driven by the creation of malicious scripts, malware trojans, sophisticated malware delivery platforms, and more.

June 21, 2024 · min to read · Malware Analysis

Boolka Malware Threat Intelligence

Introduction

In January 2024, during the analysis of the infrastructure used by ShadowSyndicate Group-IB Threat Intelligence analysts detected a landing page designed to distribute the BMANAGER modular trojan, created by **threat actor dubbed Boolka**. Further analysis revealed that this landing page served as a test run for a malware delivery platform based on BeEF framework. The threat actor behind this campaign has been carrying out opportunistic SQL injection attacks against websites in various countries since at least 2022. Over the last three years, the threat actor have been infecting vulnerable websites with malicious JavaScript scripts capable of intercepting any data entered on an infected website.

This blogpost contains a description of:

- injected JS snippets used by the attacker we named Boolka
- a newly discovered trojan we dubbed BMANAGER

YARA rules are available for Group-IB Threat Intelligence customers.

If you have any information which can help to shed more light on this threat and enrich current research, please join our Cybercrime Fighters Club. We would appreciate any useful information to update the current blog post.

Description

Discovery via InfraStorm connection

In January 2024 Group-IB detected a new ShadowSyndicate server with IP address 45.182.189[.]109 by SSH fingerprint 1ca4cbac895fc3bd12417b77fc6ed31d. This server was used to host a website with domain name updatebrower[.]com. Further analysis showed that this website serves a modified version of Django admin page with injected script loaded from hXXps://beef[.]beonlineboo[.]com/hook.js.

The SSH key was mentioned in Group-IB blogpost. Based on that, an assumption was made that **ShadowSyndicate is a RaaS affiliate that uses various types of ransomware**, which is the most plausible case.

However, the information obtained during this research decreased the chance of this assumption being correct. We will continue to monitor InfraStorm assets to clarify the attribution. At the moment it looks like the aforementioned SSH belongs to some bulletproof hosting provider or VPN.

Web attacks

Threat actor Boolka started his activities in 2022 by infecting websites with malicious form stealing JavaScript script. The threat actor injected the following script tag into HTML code of websites (Picture 1).

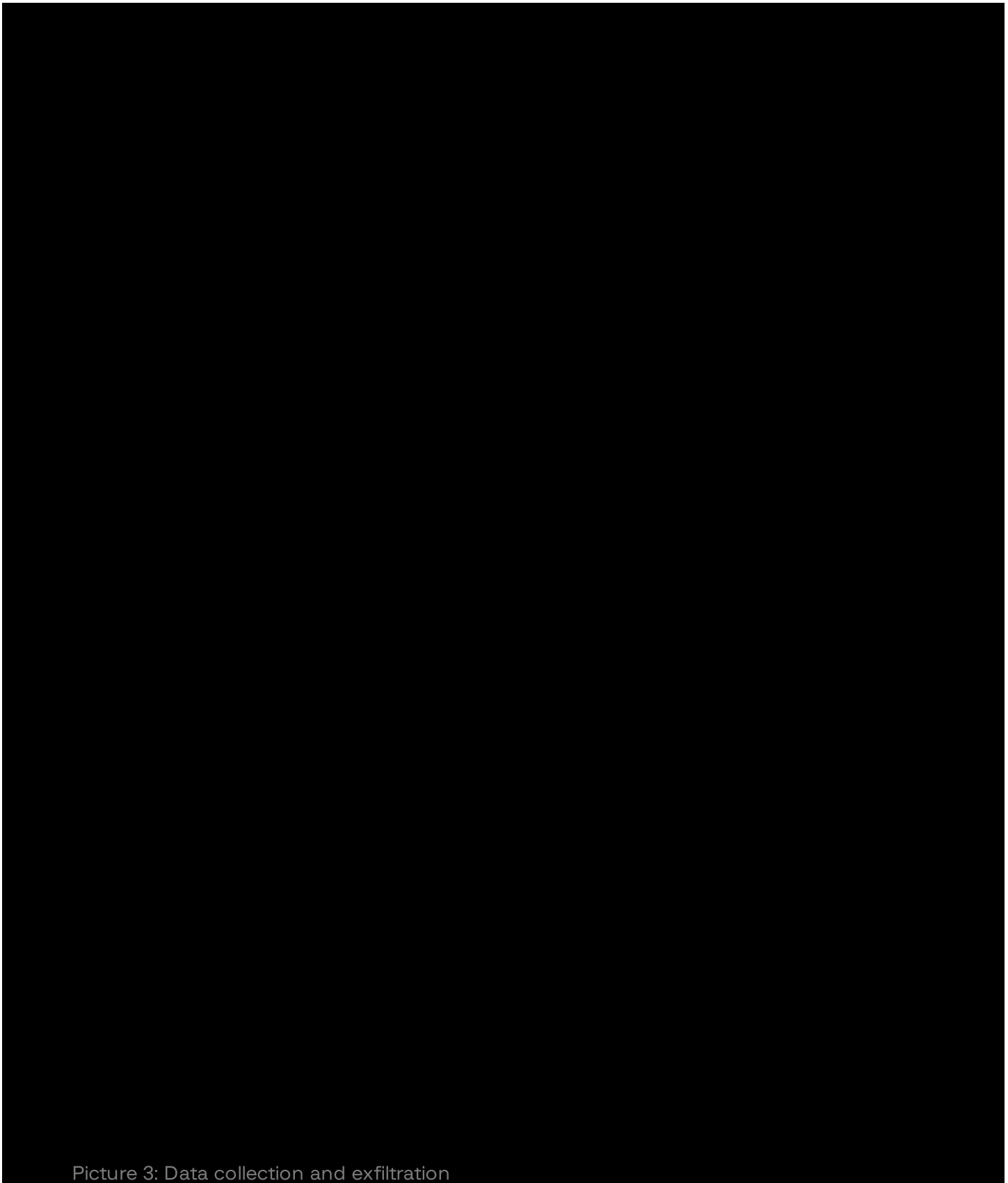
Picture 1: Injected script tag

When a user visits the infected website, the script will be downloaded and executed. During execution it performs two main actions.

First, it sends a request to the threat actor's server to notify it that the script was executed. It utilizes HTTP GET parameters with "document.location.hostname" returning the hostname of the infected website; and the current URL being Base64-encoded (Picture 2).

Picture 2: Sending a beacon to C2

Second, it collects and exfiltrates user input from infected website (Picture 3)



Picture 3: Data collection and exfiltration

The Boolka formstealing JavaScript script actively monitors user interactions, capturing and encoding input data from forms into session storage when form elements like inputs, selects, and buttons are changed or clicked. It sends all stored session data (collected form values) encoded in

Base64 format back to the threat actor's server. This behavior suggests that the script is designed for data exfiltration, potentially capturing sensitive user inputs such as passwords and usernames.

Since at least November 24th 2023, the payload loaded by the script tag was updated. Let's compare two snippets used by Boolka before and after this update:
https://urlscan.io/responses/420d8d83d5b98d959f7c62c2043b0cc2511385d4cab722b23ef4b39da5
https://urlscan.io/responses/e6bc4f2ca5bf36fae278cbbc12bbacc12f475cd92f194a79c24afe384af3e
The updated version of this malicious script includes several modifications. Notably, it now checks for the presence of a specific div element with the ID "hookwork" on the page (Picture 4). If this div is not found, it creates one and sets it to be hidden.

Picture 4: Snippet for creating div element

The code now includes additional checks within the cbClickButton function to exclude certain sessionStorage properties (key, getItem, setItem, removeItem, clear) from being sent to the server (Picture 5).

Picture 5: Updated collection and exfiltration code

The event listeners for user interactions with input fields, buttons, and select elements remain, capturing user input and sending it to the remote server.

The IP addresses of servers hosting the Boolka infrastructure were reported for multiple SQL injection attempts. The number and locations of reporters allow us to speculate that these attacks were opportunistic since there was no particular pattern in regions attacked by threat actor. Based on this information we can infer that the infection of compromised websites was the result of exploitation of vulnerabilities detected during this opportunistic vulnerability scanning.

Example SQL Injection payload used by attacker:

```
AND 1=1 UNION ALL SELECT
1,NULL,'<script>alert("XSS")</script>',table_name FROM
```

```
information_schema.tables WHERE 2>1--/**/; EXEC
xp_cmdshell('cat ../../../etc/passwd'
```

Malware delivery

The landing page updatebrower[.]com (Picture 6) detected in **January 2024** was a test run of a malware delivery platform created by Boolka. This platform was based on open source tool BeEF (The Browser Exploitation Framework). In addition to the use of the obvious subdomain “beef” and default BeEF filename “hook.js” VirusTotal also detected and saved default hook.js version.

Picture 6: Screenshot of first detected test landing page created by Boolka

In total threat actor created 3 domain names for landing pages but used only one of them:

- updatebrower.com
- 1-update-soft.com
- update-brower.com

In March 2024, **Group-IB Threat Intelligence analysts detected the first use of Boolka’s malware delivery platform in the wild**. While there are multiple overlaps between the list of

websites infected with Boolka's formstealing JS and Boolka's BeEF payload, we can assume that during this campaign the threat actor used the same approach for website infection that he tested during early stages of his activities.

In analyzed cases BeEF-based malware delivery platform created by Boolka was used to distribute a downloader for the BMANAGER trojan.

Malware

Different malware samples were discovered during analysis. Infection starts with the BMANAGER dropper which will attempt to download the BMANAGER malware from a hard-coded URL.

The following malware samples have been discovered as being used by Boolka.

BMANAGER downloader

Downloader

BMANAGER

Downloader

BMREADER

Data exfiltration

BMLOG

Keylogger

BMHOOK

Records which applications are running and have keyboard focus

BMBACKUP

File stealer

All samples found thus far have been created with PyInstaller. The Python scripts used rely on Python 3.11.

BMANAGER downloader

The BMANAGER downloader attempts to download, configure persistence for, and execute the BMANAGER malware.

It downloads the BMANAGER from a URL hard-coded into the dropper using a HTTP(S) GET request.

The response to this request is a list of Base64 encoded strings. These strings are decoded, ZLIB decompressed, and appended to the BMANAGER executable file.

By default it drops the BMANAGER malware at: *C:|Program Files|Full Browser Manager|1.0.0|bmanager.exe*

BMANAGER persistence & execution

Persistence is achieved via Windows tasks. This starts the BMANAGER malware when the user logs into Windows.

```
schtasks /create /sc onlogon /tn bmanager  /tr "'C:\\Program Files\\Full
Browser Manager\\1.0.0\\bmanager.exe'" /it /rl HIGHEST
```

The task is executed immediately after creation.

```
schtasks /run /tn bmanager
```

These values are hard-coded into the downloader.

BMANAGER

BMANAGER is capable of downloading files from a hard-coded C2, creating startup tasks, deleting startup tasks, and running executables.

Features

- Download executables from a hard-coded C2 address
- Create Windows tasks to allow executables to run on login
- Create Windows tasks to run executables
- Delete Windows tasks

Windows tasks & persistence

Persistence is achieved by creating Windows tasks. Individual malware samples do not have the capability to achieve persistence. This is done for them by the BMANAGER malware. The BMANAGER malware will execute the following command to achieve persistence:

```
schtasks /create /sc onlogon /tn {task_name}  /tr
"'{path_to_executable}.exe'" /it /rl HIGHEST
```

With task_name being replaced by a name for the task as defined by the C2. And path_to_executable being replaced with the path to and name of the executable to configure the persistence for.

C2 communication

The malware communicates with the C2 via HTTP(S) GET requests.

Register client

On startup the malware will send messages to the C2 to register it using a GUID randomly generated by the malware. This GUID is stored in a local SQL database.

The initial C2 this request is sent to is hard-coded into the sample.

1. /client?guid={guid}
    1. Expects a string “success” to be returned.
2. /getmainnodes?guid={guid}
    1. Expects a list of potential C2s to be returned.
3. /
    1. This request is sent to each C2 in the received list to determine response time.
    2. List of C2s is sorted based on response time from low to high.
4. /client?guid={guid}
    1. Request is executed for each C2 in the returned list.
    2. Expects a string “success” to be returned.
    3. If “success” is returned the C2 is selected as the active C2 and it stops going through the list of C2s.

The list of C2s is stored in a locally kept SQL database. The active C2 is marked as such in this SQL database.

Get target applications

Next the malware will attempt to retrieve a list of applications which are targets. This request is made to the active C2.

```
/getprogramms?guid={guid}
```

The response is a single string containing comma separated executable names.

```
opera.exe,msedge.exe,chrome.exe,firefox.exe,HxOutlook.exe,HxAccounts.exe,EXCEL.EXE,Search
```

Response of C2 during time of analysis (29/02/2024)

This list of applications is stored in the local SQL database. The information can then be used by other modules to determine what applications to target.

Get additional malware

Last but not least the malware will attempt to retrieve additional executables from the active C2. These executables have thus far always been other malware samples. These samples are:

BMREADER

Data exfiltration module

BMLOG

Keylogger module

BMHOOK

Windows hooking module

BMBACKUP

File stealer module

It will send a GET request to the C2 to retrieve the applications to download and install.

/getinstall?guid={guid}

```
hxxps://updatebrower[.]com/download/bmbackup.txt;bmbackup;C:\Program Files\Full Browser Ma
hxxps://updatebrower[.]com/download/bmhook.txt;bmhook;C:\Program Files\Full Browser Manage
hxxps://updatebrower[.]com/download/bmlog.txt;bmlog;C:\Program Files\Full Browser Manager\
hxxps://updatebrower[.]com/download/bmreader.txt;bmreader;C:\Program Files\Full Browser Ma
```

Response of C2 during the time of analysis (29/02/2024).

These strings consist of parameters used by the BMANAGER malware. These parameters are separated using the semicolon (;) character. The parameters are as follows:

Download URL

The URL from where to download the executable.

Windows task name

The name of the Windows task to create/run/delete.

Executable dump path

Where the downloaded executable is dumped on the victim device.

Function

Whether to create a new Windows task for the executable, to run an existing Windows task, to create and run a Windows task, or to delete an existing Windows task.

Possible values:

1

Create new Windows task (which is set to start on login)

This will download the executable.

2

Delete an existing Windows task

3

Create a new Windows task (which is set to start on login) and run it immediately

This will download the executable.

4

Run an existing Windows task

5

Stop a currently running Windows task

This will also delete the executable.

Version

A string value. This value is used to distinguish between versions of the malware.

To download an executable the malware sends a GET request to the given URL. The response is a list of Base64 encoded strings. These strings are decoded, ZLIB decompressed, and appended to the final executable file.

A new Windows task is created for this executable to start on login, and optionally the executable is started immediately.

After all applications have been downloaded, and all tasks have been performed, a message is sent back to the C2.

/install?guid={guid}&name={version}

The version being the version string found in the C2 response.

BMREADER

The BMREADER malware sends stolen data stored in the local SQL database to the active C2.

Features

Exfiltrates data stored in the local SQL database

C2 communication

Communication with the C2 is done via HTTP(S) GET requests.

Register with C2

On start-up the malware will retrieve a C2 to use for further communication. To make the first request the initial C2 that is used is set to the active C2 in the local SQL database.

1. /getnodes?guid={guid}&type=2
    1. Expects a list of C2s as response.
2. /usednodes?guid={guid}&t=0&node={resultnode}
    1. resultnode is set to the initial C2 address.
    2. Only called if 1 did not return a list of C2s.
    3. Expects a list of C2s as response.
3. /
    1. Called for every C2 in the list.
    2. Measures response time of C2s.
    3. List of C2s is sorted based on response time from low to high.
4. /client?guid={guid}
    1. Called for every C2 in the list.
    2. Expects string “success”.
    3. If “success” is returned it will stop going through the list of C2s.
5. /usednodes?guid={guid}&t=0&node={resultnode}
    1. resultnode is set to the C2 the malware has chosen to connect to.
    2. Sent to the initial C2.
    3. If no C2 returns “success”, the initial C2 is used.

Sending stolen inputs

One of the values stored in the local SQL database that is exfiltrated by the BMREADER is a list of keyboard inputs. These keyboard inputs have been obtained by the BMLOG (keylogger) malware.

The following GET request is made to the connected C2.

/clientdata?guid={guid}&programm={programm}&title={titleencode}&vars={resultencode}

guid being the GUID retrieved from the local SQL database

programm being the path of the application from which the keys were logged

titleencode being a ZLIB compressed and Base64 encoded string that is the window title from which the keys were logged

resultencode being a ZLIB compressed and Base64 encoded string that is a combination of a number of values.

The “resultencode” string is created as follows:

“eventid={eventid}|||recid={recid}|||data={data}|||”

eventid being the ID of the event that triggered the keylogging

recid being the ID of the keylogging.

data being the actual string of inputs stolen from the victim.

The logged keys sent are then removed from the local SQL database.

Sending known applications

Another value stored in the local SQL database, and sent to the C2 by the malware, are applications found to be running on the victim device. These applications are collected by the BMHOOK malware.

A GET request is made to the C2:

/clientprogramm?guid={guid}&vars={resultencode}

guid being the random GUID obtained from the local SQL database.

resultencode being a ZLIB compressed and Base64 encoded string consisting of all programs stored in the local SQL database

When the response to this request is a string value of “success” the SQL database is updated. This update sets all applications as having been sent. This prevents entries from being sent twice.

BMLOG

The BMLOG malware is a keylogger. It stores logged keys in a local SQL database.

It performs the keylogging using the Python keyboard module.

Due to the keyboard module logging keys globally, not per window, it uses the BMHOOK malware to record which window currently has keyboard focus.

It will only log keys for applications that have been set as targets. These targets are received by the BMANAGER malware from the C2 and stored in the local SQL database. The BMLOG malware reads these targets from that same database.

Features

- Record keyboard inputs

Storing logged keys

Instead of sending logged keys to a C2 it stores them in a local SQL database.

The keylogger will continually log keys until either:

1. 60 seconds of logging have passed
2. A different window gains keyboard focus

If either of these events occurs all inputs are stored as a single string in the local SQL database. After storage the keylogger will begin logging again.

The inputs are translated as follows:

- For inputs a single character long (a, b, 1, 2, etc.) they are put in the string as is.
- For space inputs a whitespace is appended to the string.
- For tab inputs a “\t” character is appended to the string.
- For other inputs the input is capitalized and placed between square brackets before being appended to the string.

Additional values stored alongside the input string are:

- The event ID
- The amount of recordings made for the logged application
- The path to the logged application

- The title of the window being keylogged
- 0 value to indicate the information has not yet been sent to the C2

The BMREADER application sends the logged keys to the C2.

BMHOOK

The BMHOOK malware uses Windows hooks to discover which applications are running on a victim device and which window/application has keyboard focus.

This sample stands out in its implementation in that it uses CPython and Windows APIs to install Windows hooks. This makes the sample function only on Windows.

Features

- Install a Windows hook to trigger on a window receiving keyboard focus

Windows hooks

The BMHOOK malware uses the SetWinEventHook function to install a Windows hook. This hook is configured to trigger on win32con.EVENT_OBJECT_FOCUS events. This type of event occurs when a window receives keyboard focus.

The following actions occur when this event is triggered:

- Use GetWindowTextW to retrieve the title of the hooked window.
- Obtain the full path of the executable the window belongs to.
- Insert these two values, and a unique ID value, into the local SQL database.
- Insert the path to the application into the local SQL database, if it does not exist there already.

The BMREADER malware uses the information stored in the local SQL database to send to the C2. The BMLOG malware uses the information to determine which window/application is being keylogged.

BMBACKUP

The BMBACKUP malware is a file stealer. It checks for specific files retrieved from a C2. If it finds the files it will read them and send them to the C2.

Features

- Retrieve paths of files to steal from C2
- Exfiltrate stolen files to C2

C2 communication

Communication with the C2 occurs via HTTP(S) GET requests.

Register with C2

On start-up the malware will retrieve a C2 to use for further communication. To make the first request the initial C2 that is used is set to the active C2 in the local SQL database.

1. /getnodes?guid={guid}&type=2
    1. Expects a list of C2s as response.
2. /usednodes?guid={guid}&t=0&node={resultnode}
    1. Only called if 1 did not return a list of C2s.
    2. Expects a list of C2s as response.
3. /
    1. Called for every C2 in the list.
    2. Measures response time of C2s.
    3. List of C2s is sorted based on response time from low to high.
4. /client?guid={guid}
    1. Called for every C2 in the list.
    2. Expects string “success”.
    3. If “success” is returned it will stop going through the list of C2s.
5. /usednodes?guid={guid}&t=0&node={resultnode}
    1. Sent to the initial used for the first request.
    2. resultnode is set to the C2 the malware has chosen to connect to.
    3. If no C2 returns “success”, the initial C2 is used.

Get target files

The malware sends a request to the C2 every 60 seconds to retrieve a list of files to exfiltrate.

/getpaths?guid={guid}

The response consists of a list of strings. Each being an absolute path to a file to exfiltrate.

```
C:\*\*\AppData\Roaming\Bitcoin\wallets\*\wallet.dat
C:\*\*\AppData\Roaming\Bitcoin\wallets\wallet.dat
```

Response from C2 during the time of analysis (29/02/2024).

After making this request it will check each of these files whether they exist or not. If a file is found to exist the exfiltration process is initiated.

Exfiltrating files

The malware will go through the list of files to exfiltrate and check if they exist. When a file exists it will begin the exfiltration process.

1. A copy of the target file is made with a randomized name. This randomized name is a random UUID value ending with ".tmp". This copy is placed in the users temporary directory (*C:|Users|*|AppData|Local|Temp*).
2. The copy file is read in 16384 byte chunks. Each of these chunks is sent to the C2 via a GET request.
    1. /clientfiledata?guid={guid}&vars={resultencode}
    2. resultencode being a Base64 encoded string containing the byte data.

resultencode is created in the following manner:

Up to 16384 bytes are read from the target backup file and converted to a hexadecimal string

The info string is created

"partid={partid}|||partcount={partcount}|||hex={hex}|||fn={file}|||

partid is which chunk of the file this object is

partcount are the total amount of chunks the file consists of

hex are the bytes read from the file

file is the path and name of the original file (not the path and name of the backup file)

This info string is ZLIB compressed, Base64 encoded, and then made URL safe

This is the final resultencode object that is sent as a URL parameter

SQL database

Most samples make use of a local SQL database. The path and name of this database is hard-coded in the samples to be located at: *C:|Users|{user}|AppData|Local|Temp|coollog.db,* with user being the username of the logged in user.

The following is a map of the SQL database. This map contains all tables and fields used by the different malware samples. Do note that the tables are created by each sample as they use them. Thus if certain samples are not present on a device, these tables may not be present.

Tables

clientguid

Contains the randomly generated GUID used to identify the sample to the C2.

Created by BMANAGER

mainnodes

Contains a list of C2s, in particular the currently active C2.

Created by BMANAGER

log

Contains the keylogger data.

Created by BMLOG

event

Contains which applications/windows have/had keyboard focus.

Created by BMHOOK

allprogramm

Contains a list of applications whose window has received keyboard focus at one point.

Created by BMHOOK

programms

Contains a list of all applications that are to be targeted by other modules.

Created by BMANAGER

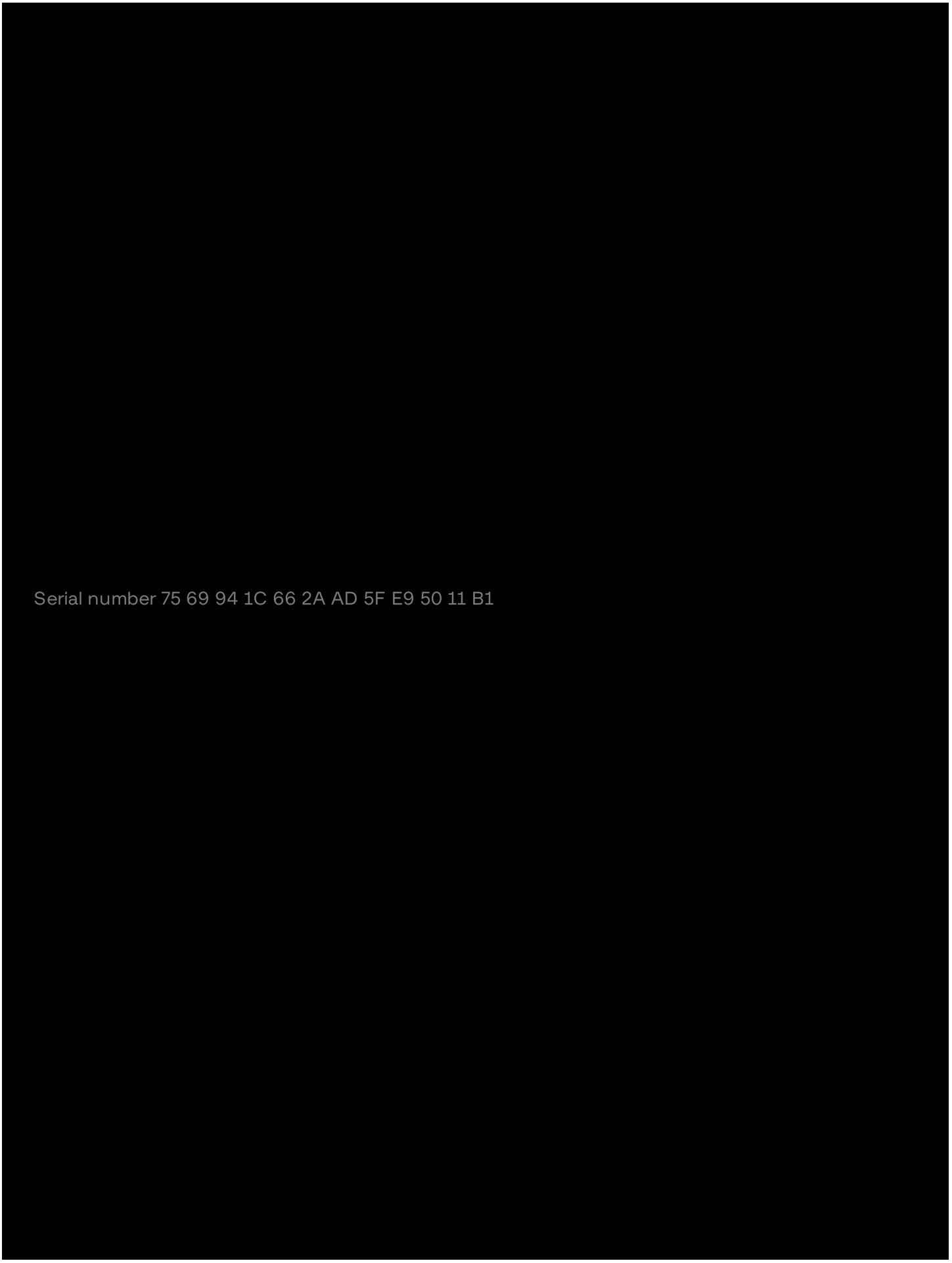
files

Contains a list of files that need to be exfiltrated to the C2.

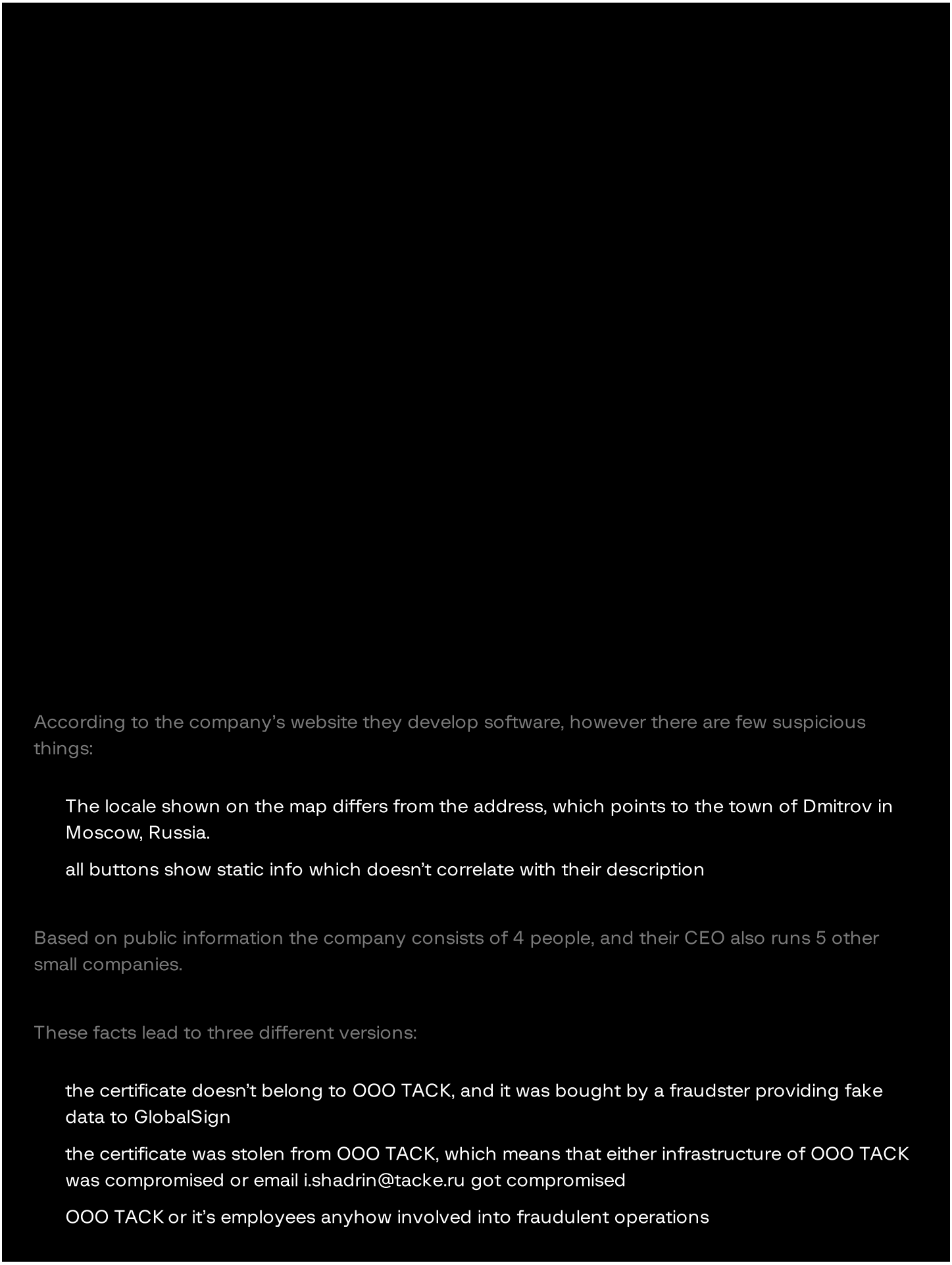
Created by BMBACKUP

Signing certificate

BMANAGER 2f10a81bc5a1aad7230cec197af987d00e5008edca205141ac74bc6219ea1802 is signed with a valid certificate by OOO TACK:


Serial number 75 69 94 1C 66 2A AD 5F E9 50 11 B1


According to its metadata the signer is i.shadrin@tacke.ru.



According to the company's website they develop software, however there are few suspicious things:

- The locale shown on the map differs from the address, which points to the town of Dmitrov in Moscow, Russia.
- all buttons show static info which doesn't correlate with their description

Based on public information the company consists of 4 people, and their CEO also runs 5 other small companies.

These facts lead to three different versions:

- the certificate doesn't belong to OOO TACK, and it was bought by a fraudster providing fake data to GlobalSign
- the certificate was stolen from OOO TACK, which means that either infrastructure of OOO TACK was compromised or email i.shadrin@tacke.ru got compromised
- OOO TACK or it's employees anyhow involved into fraudulent operations

We can not confirm any of these versions. However we checked domain tacke.ru in the stealer logs cloud and didn't find any occurrence.

Conclusion

The discovery of the Boolka's activities sheds light on the evolving landscape of cyber threats. Starting from opportunistic SQL injection attacks in 2022 to the development of his own malware delivery platform and trojans like BMANAGER, Boolka's operations demonstrate the group's tactics have grown more sophisticated over time. The injection of malicious JavaScript snippets into vulnerable websites for data exfiltration, and then the use of the BeEF framework for malware delivery, reflects the step-by-step development of the attacker's competencies.

The analysis reveals **the complexity of the malware ecosystem employed by Boolka**, with various components such as formstealing scripts, keyloggers, and file stealers orchestrated to achieve malicious objectives. Additionally, the investigation into the signing certificate used by the BMANAGER malware underscores the challenges in attribution and the potential involvement of legitimate entities in illicit activities.

Recommendations

Recommendations for end users:

- Avoid clicking on suspicious links or downloading files from unknown sources.
- Download apps and updates only from official sources.
- Ensure that your operating systems, browsers, and all software are regularly updated.
- Employ strong, unique passwords for different accounts and use a reputable password manager to keep track of them.
- Enhance security by enabling multi-factor authentication (MFA) on your accounts wherever possible.
- Ensure you have reliable and up-to-date security measures like anti-virus software in place to detect and remove threats.

Recommendations for website owners:

Conduct frequent security audits and vulnerability assessments to identify and fix potential weaknesses. **Group-IB's Penetration Testing services** can help you minimize your susceptibility to web attacks. Our experts work with the latest methods and techniques curated by **Group-IB Threat Intelligence** to pinpoint assets vulnerable to web injection attacks, and more.

Use robust authentication protocols and require strong passwords for all users, along with multi-factor authentication.

Ensure all software, including plugins and content management systems, are updated with the latest security patches.

Deploy a WAF to monitor and filter malicious traffic targeting your web applications.

For advanced cybersecurity teams, we recommend using **Group-IB's Threat Intelligence system**, which can be used to detect relevant threats as early as during their preparation stage. The built-in graph analysis tool enriched by data from the largest threat-actor database reveals links between attackers, their infrastructures, and their tools. Enriching cybersecurity with threat intelligence helps significantly strengthen an organization's ability to counter attacks, including ones carried out by state-sponsored groups.

Provide regular training for your staff on the latest security practices and threat awareness.

Set up continuous website monitoring for suspicious activities and have an incident response plan ready in case of a breach.

MITRE ATT&CK

T1583.001 – Acquire Infrastructure: Domains

T1583.004 – Acquire Infrastructure: Virtual Private Server

T1584.003 – Compromise Infrastructure: Botnet

T1587.001 – Develop Capabilities: Malware

T1588.002 – Obtain Capabilities: Tool

T1189 – Drive-by Compromise

T1190 – Exploit Public-Facing Application

T1059.007 – Command and Scripting Interpreter: JavaScript

T1203 – Exploitation for Client Execution

T1204.002 – User Execution: Malicious File

T1569 – System Services

T1569.002 – System Services: Service Execution

T1543 – Create or Modify System Process

T1543.003 – Create or Modify System Process: Windows Service

T1001 – Data Obfuscation

T1657 – Static Analysis Evasion

T1056 – Input Capture

T1056.001 – Input Capture: Keylogging

T1082 – System Information Discovery

T1083 – File and Directory Discovery

T1210 – Exploitation of Remote Services

T1005 – Data from Local System

T1213 – Data from Information Repositories

T1071.001 – Application Layer Protocol: Web Protocols

T1041 – Exfiltration Over C2 Channel

T1565 – Data Manipulation

T1565.002 – Data Manipulation: Transmitted Data Manipulation

T1608 – Stage Capabilities

T1608.004 – Stage Capabilities: Upload Malware

IoCs

File hashes

```
2f10a81bc5a1aad7230cec197af987d00e5008edca205141ac74bc6219ea1802 - Dropper
7266f20123edcb2e0b92ac0b63225b8db2c5ff349818b339ef1553bff06719e4 - BMANAGER
9434e2f277f764bb75302cd5355ed45f7624f1d993a454a7dbaf68b7e9b4b3a2 - BMBACKUP
b2dbd3187c67883c0f77c17530f41e05950e9e38b2798773770fe37f5985e367 - BMHOOK
94430690ac9516a25ca764bae8c4b5a88d6f0308f558aea43ca50b5f750685ee - BMLOG
227b8233071da4d3015cb04b69285885100c9f2e5d98b803b37d23afb798375a - BMREADER
```

Domains & IPs

Domain	Registration date	Registrar	Description
boolka.tk	–	Freenom	Form stealing JS C2
boolka24.tk	–	Freenom	Form stealing JS C2
beonlineboo.com	30.06.2022	CNOBIN INFORMATION TECHNOLOGY LIMITED	Form stealing JS C2
beef.beonlineboo.com	30.06.2022	CNOBIN INFORMATION TECHNOLOGY LIMITED	BeEF-based malware delivery platform C2
mainnode.beonlineboo.com	30.06.2022	CNOBIN INFORMATION TECHNOLOGY LIMITED	Boolka malware C2 (BManager, BMReader, BMBackup)
		CNOBIN	

IP	Domain Name

194.165.16.68	boolka.tk
141.98.81.23	boolka24.tk
179.60.150.123	beonlineboo.com
141.98.9.152	mainnode.beonlineboo.com
92.51.2.78	beef.beonlineboo.com
179.60.147.74	node.beonlineboo.com
45.182.189.109	updatebrower.com

URLs

```
https://mainnode.beonlineboo.com
https://mainnode.beonlineboo.com/client?guid={guid}
https://mainnode.beonlineboo.com/getmainnodes?guid={guid}
https://mainnode.beonlineboo.com/getprogramms?guid={guid}
https://mainnode.beonlineboo.com/getinstall?guid={guid}
https://mainnode.beonlineboo.com/install?guid={guid}&name={version}
https://mainnode.beonlineboo.com/usednodes?guid={guid}&t={nodeping}&node=https://node.beo
https://node.beonlineboo.com
https://node.beonlineboo.com/client?guid={guid}
https://node.beonlineboo.com/clientdata?guid={guid}&programm={programm}&title={titleencod
https://node.beonlineboo.com/clientprogramm?guid={guid}&vars={resultencode}
https://node.beonlineboo.com/clientfiledata?guid={guid}&vars={resultencode}
https://updatebrower.com/download/bmanager.txt
https://updatebrower.com/download/bmbackup.txt
https://updatebrower.com/download/bmhook.txt
https://updatebrower.com/download/bmlog.txt
https://updatebrower.com/download/bmreader.txt
http://boolka.tk/js/support.js?host=
https://beef.beonlineboo.com/check?url=
https://beef.beonlineboo.com/hook.js
https://beonlineboo.com/js/support.js?host=
https://boolka24.tk/js/support.js?host=
```

File artifacts

```
C:\Users\*\AppData\Local\Temp\coollog.db
C:\Users\*\AppData\Local\Temp\coollog.db-journal
C:\Program Files\Full Browser Manager\1.0.0\bmanager.exe
C:\Program Files\Full Browser Manager\1.0.0\bmbackup.exe
C:\Program Files\Full Browser Manager\1.0.0\bmhook.exe
C:\Program Files\Full Browser Manager\1.0.0\bmlog.exe
C:\Program Files\Full Browser Manager\1.0.0\bmreader.exe
C:\Users\*\AppData\Local\Temp\{UUID}.tmp
        {UUID} being a randomly generated UUID value.
```

Persistence

To achieve persistence, a Windows task is created by the BMANAGER malware.

```
schtasks /create /sc onlogon /tn bmanager  /tr "'C:\\Program Files\\Full Browser Manager\'
schtasks /create /sc onlogon /tn bmreader  /tr "'C:\\Program Files\\Full Browser Manager\'
schtasks /create /sc onlogon /tn bmlog  /tr "'C:\\Program Files\\Full Browser Manager\\1.(
schtasks /create /sc onlogon /tn bmhook  /tr "'C:\\Program Files\\Full Browser Manager\\1
schtasks /create /sc onlogon /tn bmbackup  /tr "'C:\\Program Files\\Full Browser Manager\'
```

GROUP-IB

Products

- Threat Intelligence
- Fraud Protection
- Managed XDR
- Attack Surface Management
- Digital Risk Protection
- Business Email Protection
- Cyber Fraud Intelligence Platform
- Unified Risk Platform
- Integrations

Resources

- Research Hub
- Success Stories
- Knowledge Hub
- Certificates
- Webinars
- Podcasts
- TOP Investigations
- Ransomware Notes
- AI Cybersecurity Hub

Partners

- Partner Program
- MSSP and MDR Partner Program
- Technology Partners
- Partner Locator

Company

- About Group-IB
- Team
- CERT-GIB
- Careers
- Internship
- Academic Aliance
- Sustainability
- Media Center
- Contact

Subscription plans

Services

Resource Center

Contact

APAC: +65 3159 3798

EU & NA: +31 20 226 90 90

MEA: +971 4 568 1785

info@group-ib.com

Subscribe to stay up to date with the latest cyber threat trends