ZSCALER THREAT HUNTING ADVANCED

HuntPedia: Telegram Bot API

DOC ID:HP20240926

Title: Telegram Bot API

Document: ID: HP20240926
Created Date: 2024-09-26
Updated Date: 2024-09-26

Overview

The Telegram Bot API is an HTTP-based interface created for developers who want to build Bots for Telegram.

Telegram bots can function like apps, performing various tasks ranging from downloading YouTube videos to creating reminders. However, not all Telegram Bots are benign. Malicious Telegram bots can steal private data, propagate on-demand malware, or be used to exfiltrate data. For example, malware (such as IRRAT and TeleRat) have been known to leverage Telegram API for phishing, command and control (C2), and data exfiltrating stolen data.

Post-installation TeleRAT creates files in the app's internal directory, containing information about the device it installed on and a list of commands to execute.

The RAT announces its successful installation to the attackers by sending a message to a Telegram Bot via the Telegram Bot API with the current timestamp.The malware starts a service that listens for changes made to the Clipboard in the background. Then, the app fetches updates from the Telegram Bot API every ~4.6 seconds, listening for commands such as: `Get contacts`, `Get clipboard`, `Take photo`, etc. The malware uploads exfiltrated data using Telegram's `sendDocument` API method. This TTP eliminates the possibility of legacy network-based detection that is based on traffic to known upload servers, as all communication (including uploads) is done via the Telegram Bot API.

Each bot is given a unique authentication token when it is created. The token looks something like `123456:ABC-DEF1234ghIkl-zyx57W2v1u123ew11`. The token is required to authorize the Bot and send requests to the Bot API. Tokens are meant to be kept private as they can be used by anyone to control the Bot.

All queries to the Telegram Bot API must be served over HTTPS and need to be presented in this form: `https://api.telegram.org/bot<token>/METHOD_NAME`.

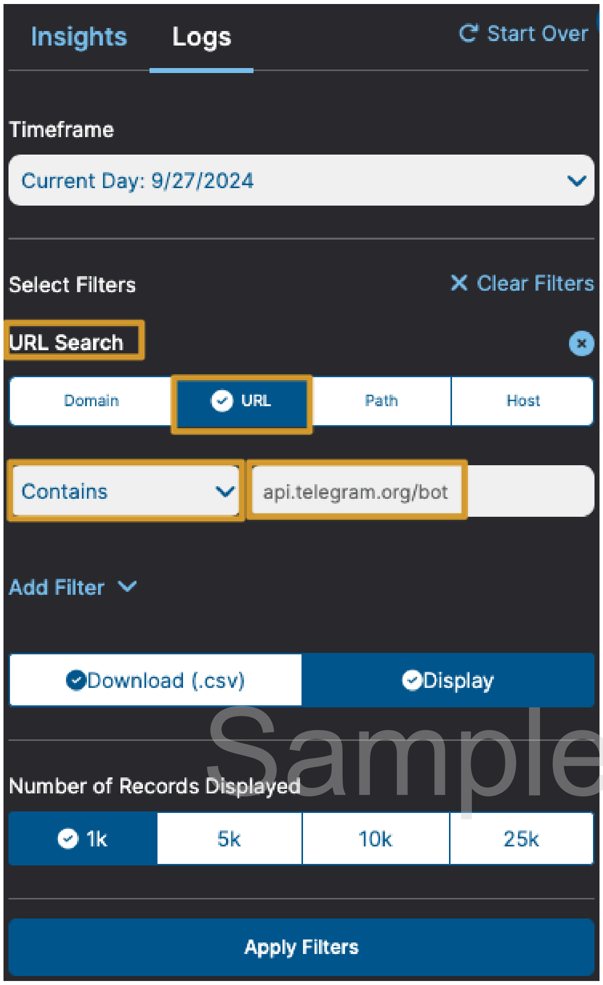
For example:

```
https://api.telegram.org/bot123456:ABC-DEF1234ghIkl-zyx57W2v1u123ew11/
getMe
```

Telegram Bot API methods can give threat hunters insight into what activity the Bot or user is performing. For example, the `sendPhoto` method is used to send images up to 10 MB in size.

Hunting Queries

The following process illustrates how to hunt for this topic in the Zscaler Admin UI:



Timeframe: Current Day: 9/27/2024

URL Search: Domain | ✓ URL | Path | Host

Contains: api.telegram.org/bot

Number of Records Displayed: ✓ 1k | 5k | 10k | 25k

Login to the Zscaler Admin UI.

Go to Analytics > Web Insights > Logs

1. Timeframe: Choose a predefined time frame or select Custom to use the calendar and time menus to define your own time frame.
2. Select Filter: Apply filters to narrow down the list or to find transactions:
   a. URL Search
   b. URL
   c. Contains: `api.telegram.org/bot`
3. Apply Filters

To learn more about Web Insights Logs, see [Web Insights Logs: Filters]()

The following regex can be used to hunt for this topic:

```
api\.telegram\.org/bot[^/\s]+
```

In this pattern:

- `api` matches the literal string "api"
- `\.` matches a literal period "."
- `telegram` matches the literal string "telegram"
- `\.` matches another literal period "."
- `org` matches the literal string "org"
- `/bot` Matches the literal string "/bot"

- `[^/\s]+` : [^/\s] is a character class that matches any character except a forward slash / or whitespace characters (spaces, tabs, etc.).
- `+` means one or more of the preceding character class, ensuring that the bot token (or any other characters following /bot) is captured.

The following Splunk Query can be used to hunt for this topic:

```
index=<your_index> sourcetype=<your_sourcetype>
(eurl="*api.telegram.org/bot*" OR ehost="*api.telegram.org")
| table time eurl cip ehost
```

Splunk Query Explained:

- `<your_index>` represents the index where your web proxy logs are stored. Replace it with the appropriate index name.
- `<your_sourcetype>` represents the sourcetype of your ZIA logs. Replace it with the appropriate sourcetype name.
- `(eurl="*api.telegram.org/bot*" OR ehost="*api.telegram.org")` : `eurl="*api.telegram.org/bot*"` : This part of the query checks if the `eurl` field contains the string `api.telegram.org/bot` anywhere within it.
- `OR ehost="*api.telegram.org"` : This part checks if the `ehost` field contains the string `api.telegram.org` anywhere within it. The `OR` operator means the query will match events if either condition is true.
- `| table time eurl cip ehost` : The table command is used to create a table of results with the specified fields.

Remember to replace `<your_index>` and `<your_sourcetype>` with the actual values specific to your Splunk deployment.

Examples of True Positives

time	eurl	cip	ehost
2024-09-24 07:00:06.000	api[.]telegram[.]org/bot5735391905:aagasectllmmlv-kaqrkh-6xmctyowulzt8/sendmessage?chat_id=1326228749&text=Dude...?	10.1.1.100	api[.]telegram[.]org
2024-09-24 08:53:49.000	api[.]telegram[.]org/bot7265790107:aae9xt3b23wybhq0fw5bww5u7wzynzt3cc/senddocument	10.1.1.105	api[.]telegram[.]org
2024-09-25 09:50:09.000	api[.]telegram[.]org/bot6374089573:aahtbtahw3nhipreweph1piowrjskkfq8fs/getme	10.1.1.110	api[.]telegram[.]org

Presence of the Telegram Bot API in your telemetry doesn’t mean malicious activity took place, however it does warrant further investigation.

To narrow down the hunt results, the Bot token can be extracted to get a distinct count of Telegram Bots in the telemetry:

```
(?<=org\/)[^\/]+(?=\/)
```

In the example true positives in the table above, we can identify three distinct Bots (`bot5735391905:aagasectllmmlv-kaqrkh-6xmctyowulzt8`, `bot7265790107:aae9xt3b23wybhq0fw5bww5u7wzynzt3cc`, and `bot6374089573:aahtbtahw3nhipreweph1piowrjskkfq8fs`).

Another method of narrowing down the hunt results is by compiling a list of potentially risky Telegram Bot API methods (found here), for example, the `sendDocument` method which is used to send files up to 50 MB in size.

Emulation & Validation

1. Create a new Telegram Bot using the `/newbot` command outline BotFather section of this knowledge base article.
    a. The token is a string, like `110201543:AAHdqTcvCH1vGWJxfSeofSAs0K5PALDsaw`, which is required to authorize the bot and send requests to the Bot API. Keep your token secure and store it safely - it can be used by anyone to control your bot.
2. In a Terminal, execute the following command to generate Bot API telemetry:
    a. `curl https://api.telegram.org/bot<TOKEN>/getUpdates | grep -o '"chat":{"id":.+?,'`
    b. Where <TOKEN> is the token provided by Telegram for your new Bot.
    c. You will get output like the one below, keep chat id:
        i. `"chat":{"id":-1001182736542,`

References

- Telegram Bot
- Telegram Bot FAQ
- Telegram Bot API

Recommendations

To mitigate risks associated with the potentially malicious Telegram Bots, consider blocking Telegram activity in your environment (CIDR range: https://core.telegram.org/resources/cidr.txt) . At a minimum, block URLs that match `api[.]telegram[.]org/bot`.

Telegram CIDR Ranges:

- 91.108.56.0/22
- 91.108.4.0/22
- 91.108.8.0/22
- 91.108.16.0/22
- 91.108.12.0/22
- 149.154.160.0/20
- 91.105.192.0/23
- 91.108.20.0/22
- 185.76.151.0/24
- 2001:b28:f23d::/48
- 2001:b28:f23f::/48
- 2001:67c:4e8::/48
- 2001:b28:f23c::/48
- 2a0a:f280::/32

Telegram Domains:

- *.t.me
- *.telegram.me
- *.telegram.org
- *.nicegram.app
- *.telesco.pe
- *.tg.dev

Please ensure to thoroughly validate this recommendation before full implementation to avoid any unintended outcomes.