# NoName057(16) – The Pro-Russian Hacktivist Group Targeting NATO

sentinelone.com/labs/noname05716-the-pro-russian-hacktivist-group-targeting-nato/

Tom Hegel

**By Tom Hegel and Aleksandar Milenkoski**

## Executive Summary

- Pro-Russia hacktivist group NoName057(16) is conducting a campaign of DDoS attacks on Ukraine and NATO organizations that began in the early days of the war in Ukraine. Targets have included government organizations and critical infrastructure.
- NoName057(16) was responsible for disrupting services across the financial sector of Denmark this week. Other recent attacks include organizations and businesses across Poland, Lithuania and others.
- On January 11th, we observed NoName057(16) begin targeting 2023 Czech presidential election candidates' websites.
- SentinelLabs has identified how the group operates over public Telegram channels, a volunteer-fueled DDoS payment program, a multi-OS supported toolkit, and GitHub.

## What is NoName057(16)

NoName057(16), also known as NoName05716, 05716nnm or Nnm05716, is a relatively underreported hacktivist group supporting Russia since March 2022, alongside Killnet and other pro-Russian groups. In December 2022, the group was responsible for disrupting the Polish government website. As noted by the Polish government, the incident was in response to the Sejm of the Republic of Poland officially recognizing Russia as a state sponsor of terrorism in mid December 2022. More recently, the group targeted the Danish financial sector, impacting leading financial institutions as reported by Reuters.

## Motivations and Objectives

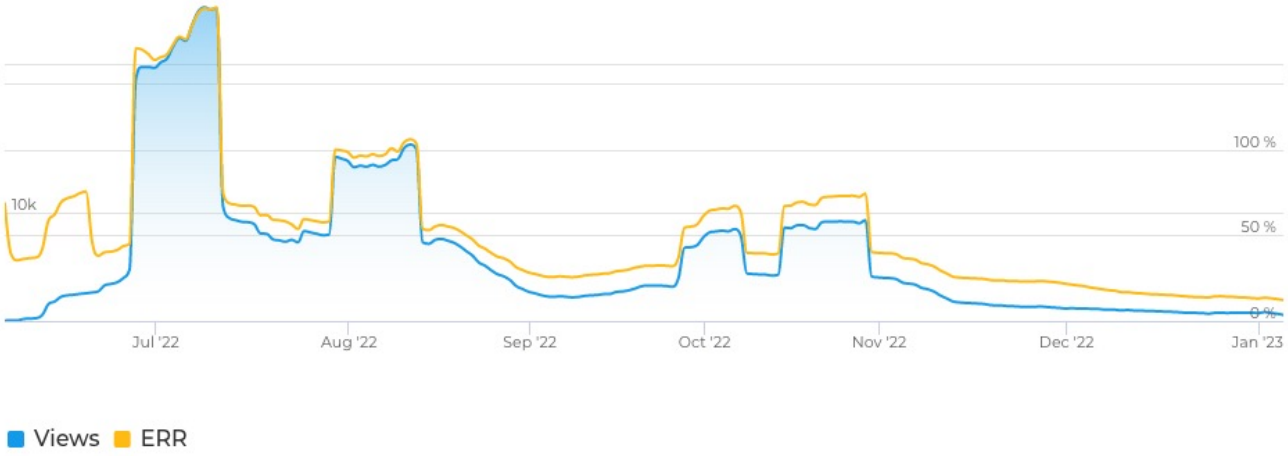
The NoName057(16) group is primarily focused on disrupting websites important to nations critical of Russia's invasion of Ukraine. Distributed Denial of Service (DDoS) attacks act as the method to conduct such disruption efforts.

Initial attacks focused on Ukrainian news websites, while later shifting to NATO associated targets. For example, the first disruption the group claimed responsibility for were the March 2022 DDoS attacks on Ukraine news and media websites Zaxid, Fakty UA, and others. Overall the motivations center around silencing what the group deems to be anti-Russian.

## Operating Methods – Telegram Channel

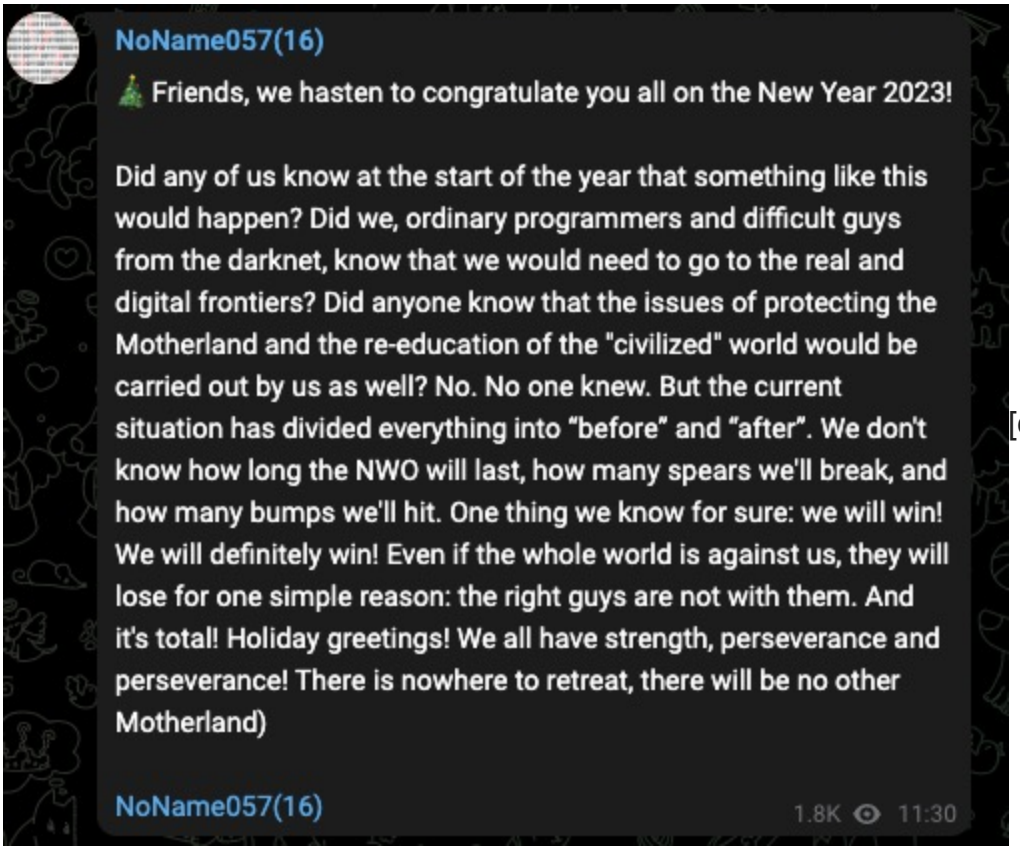
NoName057(16) operate through Telegram to claim responsibility for their attacks, mock targets, make threats, and generally justify their actions as a group. Interestingly, NoName057(16) makes attempts to teach their followers through educational content such as explaining basic industry jargon and attack concepts.

With an average of six posts per day, the overall engagement of NoName057(16)'s Telegram efforts has slowly declined over time. Peak viewership of their posts occurred in July 2022, when they reached approximately 14,000 readers with nearly 100% engagement rate. Today, daily average reach is roughly 2-3,000 and engagement in the range of 10-20%, signifying that the group is becoming less relevant to their followers and to Telegram users as a whole. This may be explained in part by the fact that many similar hacktivist groups exist, have gained more attention, and are often more impactful in their objectives.




Views and engagement rate of NoName057(16) Telegram Posts (telemetr.io)

Evidence from NoName057(16)'s Telegram channel indicates that the group values the recognition their attacks achieve through being referenced online including in Wikipedia articles. The channel also posts pro-Russian memes, motivational posts, and general status updates around the holidays. The observed Telegram activity makes it clear that the group considers itself a top tier Russian threat actor when in reality the impact of their DDoS attacks is short-lived disruption with little to no wider consequence.



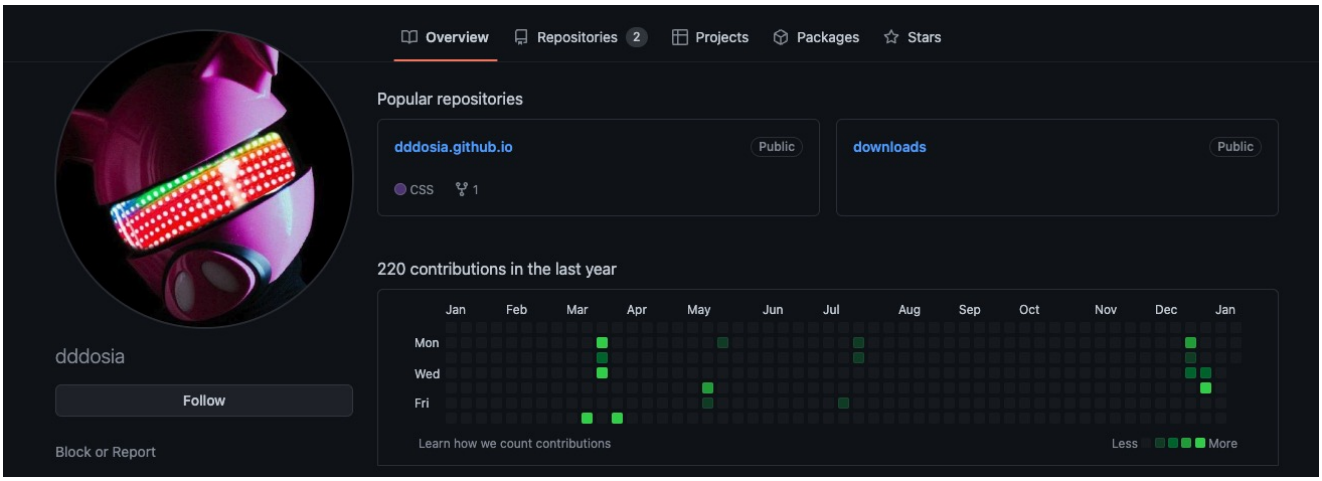

[caption]

NoName057(16) New Year Update

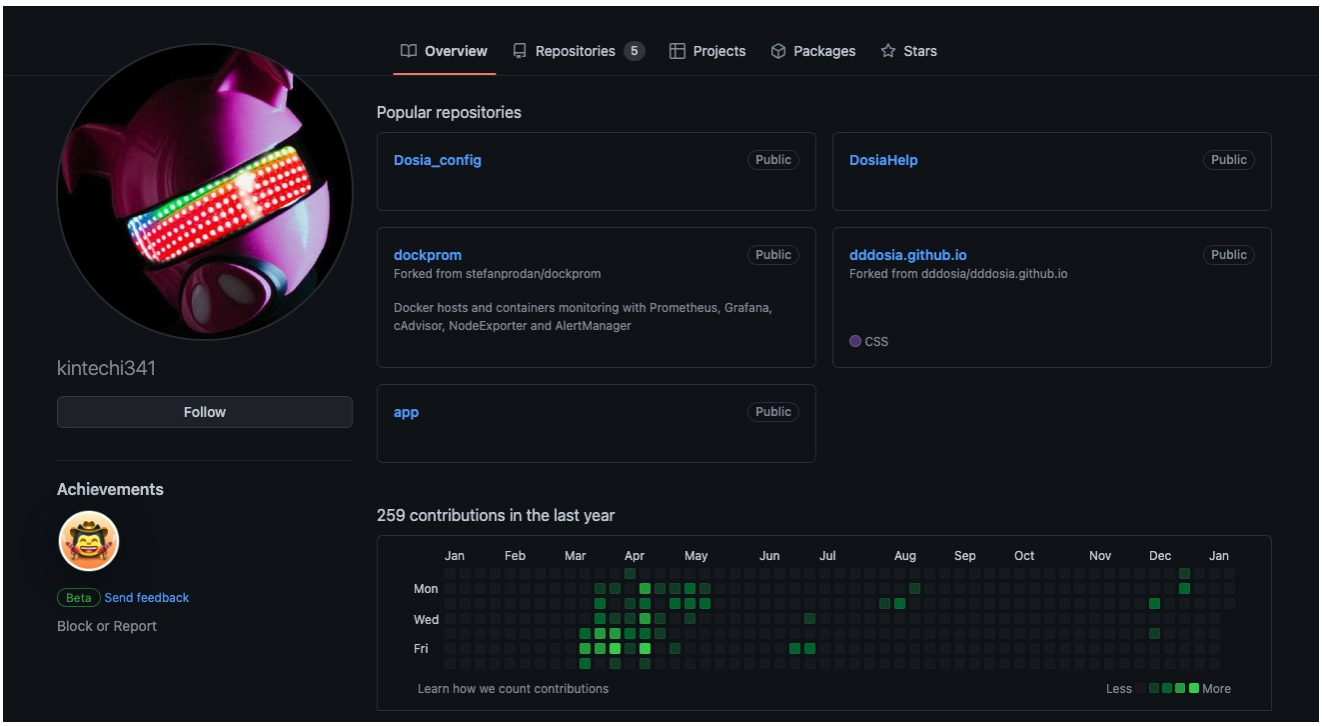
We have reported the associated accounts/channels to the Telegram Abuse team.

## Tool Hosting on GitHub

The group has also made use of GitHub to host a variety of illicit activity. This includes using GitHub Pages for freely hosting their DDoS tool website dddosia.github[.]io, and the associated GitHub repositories for hosting the latest version of their tools as advertised in the Telegram channel. Two GitHub profiles of interest are dddosia and kintechi341. Early commits to the ddos_config repo were made in the name of “Роман Омельченко”.



Associated dddosia GitHub Profile



Associated kintechi341 GitHub Profile

We reported the abuse of these services to the GitHub Trust & Safety team, who quickly took action as a violation of GitHub’s Terms of Service.

## Network

The C2 services are primarily hosted through Neterra, the Bulgarian telecommunications organization, while also making use of No-IP Dynamic DNS services. The current C2 is `zig35m48zur14nel40[.]myftp.org` at `31.13.195.87`. This server is active as of this release.

## Targets

Throughout the life of the group, NoName057(16) has focused on targeting Ukraine and NATO member countries. Organizations targeted are commonly critical infrastructure sectors whose operations are vital to the target nation.

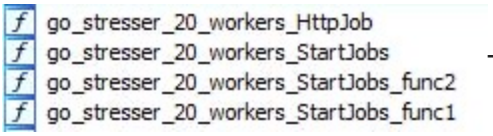
Target selection shifts according to current political events. As previously noted, the Polish government was a December target following the Sejm of the Republic of Poland officially recognizing Russia as a state sponsor of terrorism in mid December 2022. At the start of January 2023, a large focus was placed on targeting Lithuanian organizations, primarily in the cargo and shipping sectors. Most recently the actor began focusing on targeting leading Danish financial institutions including Danske Bank, Danmarks Nationalbank, and others reported in the media this week.

On January 11th 2023, we observed the actor begin targeting websites owned by multiple 2023 Czech presidential election candidates. The election is occurring on January 13th and 14th 2023, so timing of the disruption efforts can not be ignored. Specific targets include domains for candidates Pavel Fischer, Marek Hilšer, Jaroslav Bašta, General Petr Pavel, and Danuše Nerudová. Additionally, the Ministry of Foreign Affairs of the Czech Republic website was also targeted at the same time. We have notified Czech CERT upon discovery of the new target list.

## Attack Toolkit

NoName057(16) has made use of a number of different tools to conduct their attacks throughout 2022. In September, Avast reported on the threat actor using the Bobik botnet to conduct their DDoS attacks. However, the group appears to primarily seek participation voluntarily through their DDOSIA tool – also referred to by its developer as Dosia and Go Stresser, depending on versioning.

We analyzed two different implementations of DDOSIA: a Python and a Golang implementation. The Python DDOSIA implementation is delivered as a PyInstaller package. The Golang implementation refers to itself internally as Go Stresser.



The internal DDOSIA reference Go Stresser

DDOSIA is a multi-threaded application that conducts denial-of-service attacks against target sites by repeatedly issuing network requests. DDOSIA issues requests as instructed by a configuration file that the malware receives from a C2 server when started. The configuration file is in JSON format and resides at the `/client/get_targets` URL path on the C2 server. Historical configuration files can be reviewed in archived October and December 2022 server responses.

```
{
  targets: [
    {
      id: "6392ed77ac534e621b6bbc2e",
      ratio: "1",
      type: "http",
      method: "GET",
      host: "www.armee.lu",
      address: "85.93.211.246",
      port: 443,
      use_ssl: true,
      path: "/content/search?SearchButton=Recherche&SearchText=$_1",
      body: {
        type: "",
        value: ""
      },
      use_random_user_agent: true,
      timeout: 1000,
      response: true,
      headers: [],
      is_deleted: false,
      activate_by_schedule: true,
      started_at: "2022-12-09 10:00",
      finished_at: "2022-12-10 10:00"
    },
```


DDOSIA configuration file (a snippet)

For each target site, the configuration file specifies:

- A unique target identifier in the field `id`.
- Target network endpoint information in the fields `host`, `address`, and `port` – a hostname, an IP address, and a port.
- A network request type and method pairs in the fields `type` and `method`. The DDOSIA samples and configuration files we analyzed indicate that the malware supports the request types `http`, `http2`, and `tcp`, and the request methods – HTTP verbs – `GET` and `POST` (for the request types `http` or `http2`) and `syn` (for the request type `tcp`). Based on a configured type and method, DDOSIA constructs HTTP or TCP network packets (requests) for sending to a target site.
- A URL path and request body in the fields `path` and `body` for network requests of type `http` or `http2`. If the path and/or body fields have values, DDOSIA constructs and issues requests with the configured request body to the configured URL path at the target site.

```
if self._method == "syn":
    src_ip = os.urandom(4)
    src_port = random.randint(1025, 65535)
    ip_version = 4
    ip_hdr_len = 20
    ip_dsfield = 0
    ip_len = 0
    ip_id = 1
    ip_flags = 0
    ip_ttl = 64
    ip_proto = socket.IPPROTO_TCP
    ip_checksum = 0
    ip_header = struct.pack(
        '!BBHHHBBH4s4s',
        (ip_version << 4) + (ip_hdr_len // 4),
        ip_dsfield,
        ip_len,
        ip_id,
        ip_flags,
        ip_ttl,
        ip_proto,
        ip_checksum,
        src_ip,
        self._dst_ip)
    [...]
```

A Python DDOSIA implementation constructs a TCP SYN packet

```
p_http_Request = (http_Request *)runtime_newobject(&RTYPE_http_Request);
[...]
v105 = fmt_Sprintf((unsigned int)"%s%s:%v%s", 9, (unsigned int)&v112, 4, 4, v55, v56, v57, v58, v89, v94);
v106 = (url_URL *)net_url_Parse(v105, 9, v59, 4, 4, v60, v61, v62, v63, v90, v95);
if ( a15 == 4 && *(_DWORD *)target_method == 'TSOP' )
{
  if ( a18 == 6 && *(_DWORD *)a17 == 'irts' && *(_WORD *)(a17 + 4) == 'gn' )
  {
    [...]]
    v71 = (char **)net_http_NewRequestWithContext(
                     (unsigned int)go_itab__context_emptyCtx_context_Context,
                     context_background,
                     (_DWORD)target_method,
                     4,
                     v105,
                     9,
                     (unsigned int)go_itab__bytes_Reader_io_Reader,
                     (_DWORD)p_bytes_Reader,
                     v70,
                     v92,
                     v97,
                     v99);
    [...]
```

A Golang DDOSIA implementation constructs an HTTP POST request

DDOSIA replaces `$_{number}` substrings specified in the configuration file with random values that the malware generates when constructing a network request. In a DDOSIA configuration file, `$_{number}` substrings are typically placed in `path` fields. The Python implementation of DDOSIA uses templates defined in the `randoms` field in the configuration file for generating random string values.

```
host: "www.regjeringen.no",
address: "104.18.3.141",
port: 443,
use_ssl: true,
path: "/en/search/id86008/?term=$_1",
```

A *$_{number}* substring in a DDOSIA configuration file

```
randoms: [
    {
        name: "Телефон",
        id: "62d8286fddcbb37b0c77c87f",
        digit: true,
        upper: false,
        lower: false,
        min: 11,
        max: 11
    },
    {
        name: "Все символы 6-12",
        id: "62d8fccfb44b5774ee96ec0a",
        digit: true,
        upper: true,
        lower: true,
        min: 6,
        max: 12
    },
```

The *randoms* field in a DDOSIA configuration file (a snippet)

```
class Random:
    def __init__(self, digit: bool, upper: bool, lower: bool, min: int, max: int):
        temp = []
        if digit:
            temp.append(string.digits)
        if upper:
            temp.append(string.ascii_uppercase)
        [...]

    def get(self) -> str:
        temp = []
        for _ in range(random.randint(self._min, self._max)):
            temp.append(random.choice(self._chars))
        return "".join(temp)
```

A Python DDOSIA implementation generates random values

A DDOSIA configuration file specifies URL paths and request bodies that are valid at the respective target sites. This indicates that the DDOSIA operators construct configuration files by first exploring target sites. For example, the URL

`https://www.defensie[.]nl/actueel/nieuws?pagina={number}` is a valid news page iterator at the website of the Dutch Ministry of Defense.


DDOSIA configuration for targeting the Dutch Ministry of Defense

There are additional DDOSIA features to those above that a configuration file may instruct the malware to enable. For example, the `use_random_user_agent` field instructs DDOSIA to randomly select a user agent from a list of predefined user agents when constructing an HTTP request. Also, the fields `activate_by_schedule`, `started_at` and `finished_at` indicate that a DDOSIA sample can be configured to schedule the sending of network requests over specific date-time intervals. The samples we analyzed do not make use of these configuration parameters but repeatedly send network requests to each target site until terminated.

```
user_agents = [
    [...]
    "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:77.0) Gecko/20190101 Firefox/77.0",
    "Mozilla/5.0 (Windows NT 10.0; WOW64; rv:77.0) Gecko/20100101 Firefox/77.0",
    "Mozilla/5.0 (X11; Linux ppc64le; rv:75.0) Gecko/20100101 Firefox/75.0",
    "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:39.0) Gecko/20100101 Firefox/75.0",
    "Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.10; rv:75.0) Gecko/20100101 Firefox/75.0",
    "Mozilla/5.0 (X11; Linux; rv:74.0) Gecko/20100101 Firefox/74.0",
    "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.13; rv:61.0) Gecko/20100101 Firefox/73.0",
    "Mozilla/5.0 (X11; OpenBSD i386; rv:72.0) Gecko/20100101 Firefox/72.0",
    [...]
]
```

Predefined DDOSIA user agents

We note that there are differences regarding what configuration values and features are supported by different DDOSIA builds and implementations. This indicates that DDOSIA is under continuous development and is subject to frequent changes.

For example, the Golang DDOSIA implementations we analyzed support the network request type `http2`, whereas their Python counterparts do not implement this support.

```
if ( models_target_type_len == 5
    && *(_DWORD *)models_target_type == 'ptth'
    && *(_BYTE *)(models_target_type + 4) == '2' )
{
    p_http2_Transport = (http2_Transport *)runtime_newobject
    (&RTYPE_http2_Transport);
    [...]
```

An implementation of the *http2* network request type

In addition, Golang DDOSIA implementations authenticate themselves to C2 servers by issuing an HTTP POST request to the `/login_new` URL path at the servers and terminate if the authentication fails. The Python DDOSIA implementations that we analyzed do not support this feature.

```
v48 = ((__int64 (__golang *)(_DWORD,[...] __int64))go_stresser_20_models_Login)(
        (unsigned int)"/login_new",
        10,
        (_DWORD)main_BackendLink,
        [...]
if ( v48 )
{
  v109[0] = &RTYPE_string;
  v109[1] = &off_7E2E80;
  [....]
}
else
{
  [...]
  time_Sleep(0xF8475800, 1, v56, v44, (unsigned int)&off_7E2E70, v57, v58, v59, v60, v88);
  v55 = os_Exit(1, 1, v61, v44, (unsigned int)&off_7E2E70,
  v62, v63, v64, v65, v89);
}
```

```
aAvtorizaIProid db 'Авторизация пройдена успешно
```

DDOSIA authenticates itself to a C2 server (‘Авторизация пройдена успешно’ translates from Russian to ‘Authorization completed successfully’)

DDOSIA maintains statistics about its operation and success rate – the malware counts the total and the number of successful network requests sent to each target site. In the context of network requests of type `http` or `http2`, a request is considered successful if the target site returns the HTTP code `200` (`OK`).

```
v192 = net_http__ptr_Client_do(v190, v56);
if ( !v192.1.tab )
{
  v149 = v192.0;
  if ( v192.0->StatusCode == 200 )
  {
    _InterlockedExchangeAdd64(&go_stresser_20_flog_Success, 1uLL);
  [...]
```

DDOSIA counts successful HTTP network requests

DDOSIA sends the statistics to the C2 server at regular time intervals – this informs the DDOSIA operators about the overall progress and success of the denial-of-service campaign that the malware conducts. This is likely associated with how the group makes use of a volunteer profit program. They distribute cryptocurrency to the top DDoS contributors, encouraging people to contribute more technical resources for a more powerful attack.

Versions of the tool for macOS and Linux have also been developed. Android versions of the tool can also be found; however, the primary distribution of the group has not officially supported mobile.

## Conclusion

NoName057(16) is yet another hacktivist group to emerge following the war in Ukraine. While not technically sophisticated, they can have an impact on service availability– even when generally short lived. What this group represents is an increased interest in volunteer-fueled attacks, while now adding in payments to its most impactful contributors. We expect such groups to continue to thrive in today's highly contentious political climate.

We would like to thank GitHub's Trust & Safety team for a quick response following our abuse notification. The actors' accounts and pages are no longer online.

## Indicators of Compromise

| Indicator | Description |
|---|---|
| 94d7653ff2f4348ff38ff80098682242ece6c407 | DDosia.py encoded installer |
| e786c3a60e591dec8f4c15571dbb536a44f861c5 | DDosia.py encoded installer |
| c86ae9efcd838d7e0e6d5845908f7d09aa2c09f5 | December 2022 DDosia PyInstaller |
| e78ac830ddc7105290af4c1610482a41771d753f | December 2022 DDosia PyInstaller |
| 09a3b689a5077bd89331acd157ebe621c8714a89 | July 2022 DDosia PyInstaller |
| 8f0b4a8c8829a9a944b8417e1609812b2a0ebbbd | dosia_v2_macOSx64 – May 2022 |
| 717a034becc125e88dbc85de13e8d650bee907ea | dosia_v2_macOSarm64 – May 2022 |
| ef7b0c626f55e0b13fb1dcf8f6601068b75dc205 | dosia_v2_linux_x64 – May 2022 |
| b63ce73842e7662f3d48c5b6f60a47e7e2437a11 | dosia_v2.0.1.exe – May 2022 |
| 5880d25a8fbe14fe7e20d2751c2b963c85c7d8aa | dosia_v2.0.1 – May 2022 |
| 78248539792bfad732c57c4eec814531642e72a0 | dosia_v2.exe – May 2022 |

| | |
|---|---|
| 1dfc6f6c35e76239a35bfaf0b5a9ec65f8f50522 | dosia_win_x64.exe – January 2023 |
| 2.57.122.82 | C2 Server – Overlaps with Avasts Bobik findings |
| 2.57.122.243 | C2 Server – Overlaps with Avasts Bobik findings |
| 109.107.181.130 | C2 Server – October 2022 and earlier. Overlaps with Avasts Bobik findings |
| 77.91.122.69 | C2 Server – December 2022 |
| 31.13.195.87 | C2 Server – Mid December to Present Day |
| tom56gaz6poh13f28[.]myftp.org | C2 Domain |
| zig35m48zur14nel40[.]myftp.org | C2 Domain |
| [email protected][.]me | NoName057(16) Email Address |
| hxxps://t[.]me/noname05716 | NoName057(16) Primary Telegram Channel (open group) |
| hxxps://t[.]me/nn05716chat | NoName057(16) Secondary Telegram Channel (closed group) |
| hxxps://github[.]com/dddosia | Account hosting DDOSIA downloading GitHub Pages site. |
| dddosia[.]github.io | Official DDOSIA download site linked to on actors telegram page. |
| hxxps://github[.]com/kintechi341 | Contributor to the DDOSIA toolkit |