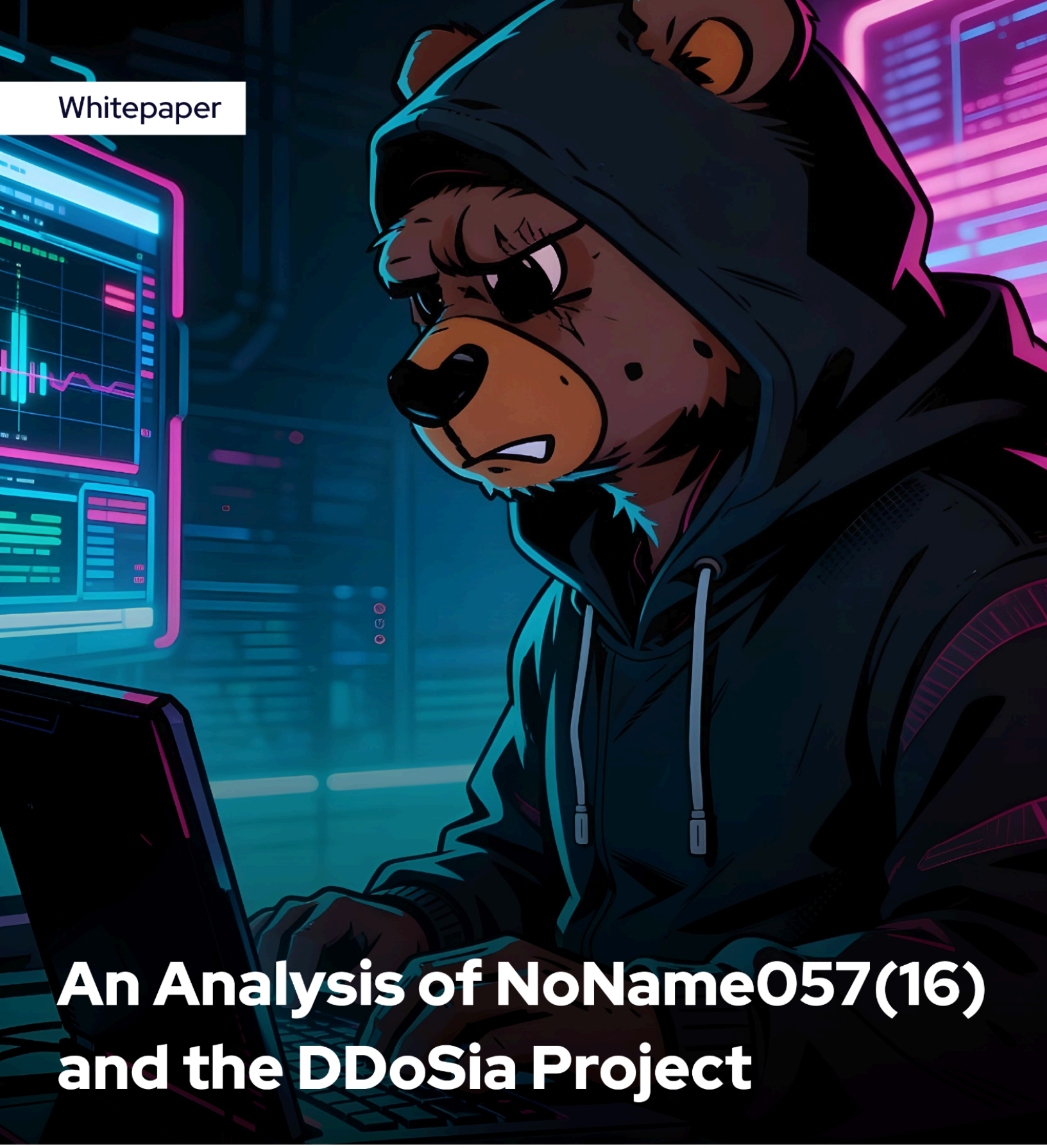
Whitepaper



An Analysis of NoName057(16) and the DDoSia Project

SOCRadar®
Your Eyes Beyond

socradar.io

Executive Summary

NoName057(16) is a pro-Russian hacktivist group active since 2022. It uses a custom denial-of-service tool called **DDoSia** to disrupt online services. The group mainly targets governments, public institutions, media outlets, and organizations that support Ukraine or oppose Russian political interests.

Threat actor card of NoName057(16)

DDoSia operates as a **voluntary botnet**. Participants knowingly install the tool and take part in attacks. The group encourages participation through propaganda, political messaging, and a reward system based on points, rankings, and occasional payments. This approach allows people with limited technical skills to contribute.

The tool supports Windows, Linux, and Android systems. It connects to command-and-control servers to receive targets, attack settings, and updates. Over time, DDoSia has evolved to include stronger evasion techniques, encrypted communication, user-agent rotation, proxy use, and anti-analysis features, making detection and mitigation more difficult.

Attacks are often linked to geopolitical events such as NATO actions, sanctions, elections, or military aid to Ukraine. Campaigns usually begin shortly after these events to increase visibility and public impact.

Overall, NoName057(16) demonstrates a high level of organization and adaptability. As long as the Russia–Ukraine conflict and related geopolitical tensions continue, the group is likely to remain active and further enhance its operational capabilities.

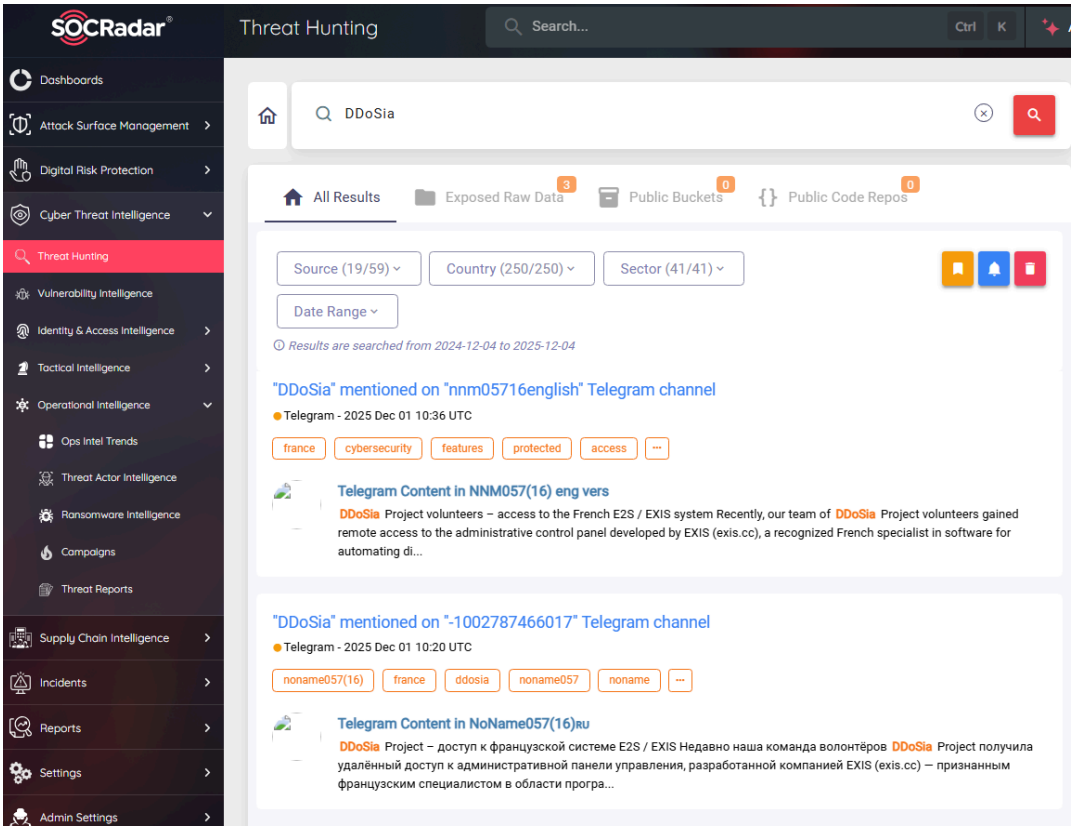
Key Points

- NoName057(16) is an ideologically driven hacktivist group aligned with pro-Russian interests.
- DDoSia is a custom-built DDoS tool used in coordinated campaigns.
- The botnet is voluntary, with users knowingly participating in attacks.
- Recruitment relies on propaganda, gamification, and rewards.
- The tool supports Windows, Linux, and Android devices.
- Communication with servers is encrypted and dynamically updated.
- Attacks often follow geopolitical events related to Ukraine, NATO, or sanctions.
- Targets mainly include government bodies, public services, media, and financial institutions.
- The threat is ongoing and likely to persist while current geopolitical tensions remain.

What is DDoSia?

DDoSia is a project implemented as a tool by the pro-Russian hacktivist group NoName057(16), which has been active since mid-2022. Its primary characteristic is denial-of-service (DoS) or **distributed denial-of-service (DDoS)** attacks, which concurrently target preselected targets by the threat actor. This project operates as a **botnet** with a notable difference from others: the nodes that launch the attacks are aware they are part of the operation, because they belong to a "voluntary botnet" model in which the people who control the nodes knowingly download and use the tool to contribute to the group's designated targets, participating in the operations and aligning with the group's goals.

To recruit and align volunteers, NoName057(16) heavily promotes **ideological propaganda** about alleged adversaries of Russia, remaining very active across its networks and channels and justifying each attack. The group grows this network by incentivizing affiliates with various rewards, which has increased its popularity and the use of its modus operandi over the months.



SOCRadar Threat Hunting

What are its components?

DDoSia is possible thanks to several pieces that fit together to make the project feasible; the sum of these parts produces a functional tool that is constantly evolving.

- **Architecture of voluntary participation**: As noted above, the group maintains a network of affiliates who sympathize with its cause and install the software on various devices (including Windows, Linux, and Android) to create nodes in the network and establish a large distribution of attackers. This effectively transforms any user device in the world into an active node serving an ideologically motivated botnet run by third parties aligned with the cause.

- **Rewards**: Another feature that explains its success and is presented as an innovation is the implementation of mechanics similar to a video game, where affiliates accumulate points based on their contributions to the generated traffic, creating ranking systems and competition among them. Beyond competitiveness, elements such as cryptocurrency-based rewards and visualization of the real impact created generate a pull effect that attracts more supporters to the tool.

- **Centralization:** Tools installed on the various nodes connect to C2 servers managed by the adversary, which provide updated target lists, attack parameters, and tool updates.

- **Propaganda:** A factor as important as the technical or innovative components is the reach of the pro-Russian narrative the actor pushes, focusing heavily on the war against Ukraine and on opposing allies, justifying attacks and discrediting geopolitical adversaries wherever they intend to strike, while producing a recruitment effect among potential affiliates.

- **Evolution:** Another key element is the actor's ability to rapidly evolve DDoSia, adding evasion capabilities and improving technical aspects that enhance attacks in each new version.

In addition to the technical and conceptual components that enable DDoSia, it is necessary to understand the actor's strong emphasis on social networks and channels, where, at an organizational level, it uses multiple sources and strategies to support its operations.

- **Telegram:** The actor runs multiple highly active Telegram groups with a sizable fandom, sometimes exceeding 20,000 followers, maintaining constant creator-to-affiliate communication

- **X (Formerly Twitter):** The actor also maintains social accounts, such as X, where they announce and partially showcase operations, keeping a large follower base informed when an attack is launched or when accounts are changed.

- **Repositories:** The actor keeps repositories where files or information are uploaded and periodically rotated.

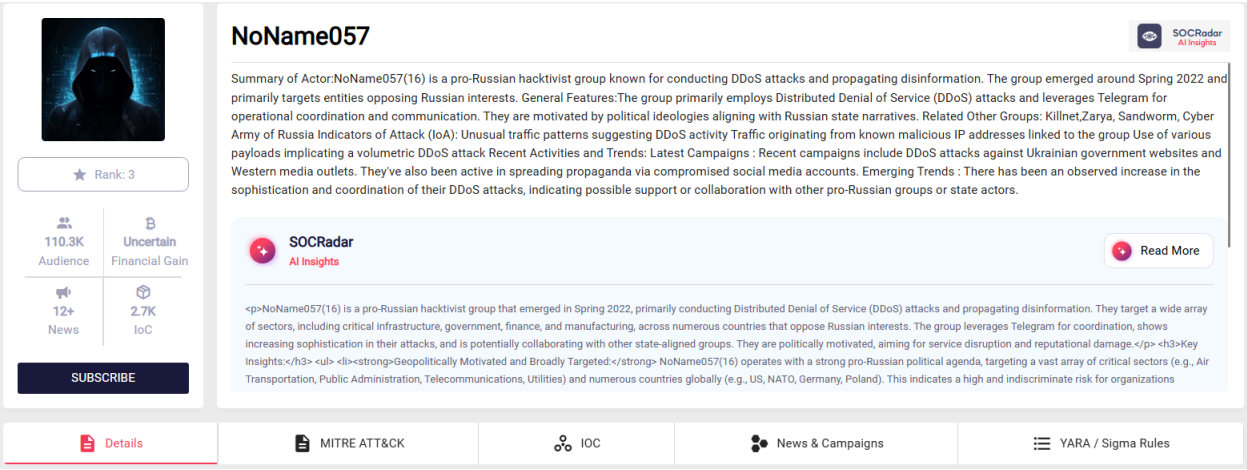
- **Distribution:** The adversary has sometimes provided affiliates with Dockerized versions to simplify installation and regularly publishes instructions and information about targets, as well as new versions of the tools, as described above.

The model supporting NoName057(16) is **atypical** compared to other groups of the same kind because it treats **affiliates like clients,** providing technical support, updates, and openly publishing information while boasting about operations, results, and attacks carried out by the Adversary-Client collective. This generates controversy because these botnets are not executed unconsciously; everyone who contributes becomes complicit in the operation.

Moreover, this modus operandi has inspired other lines of activity, and several hacktivist groups are attempting to adopt similar models with comparable tools, potentially increasing the overall threat surface.

NoName057(16) & Hacktivism

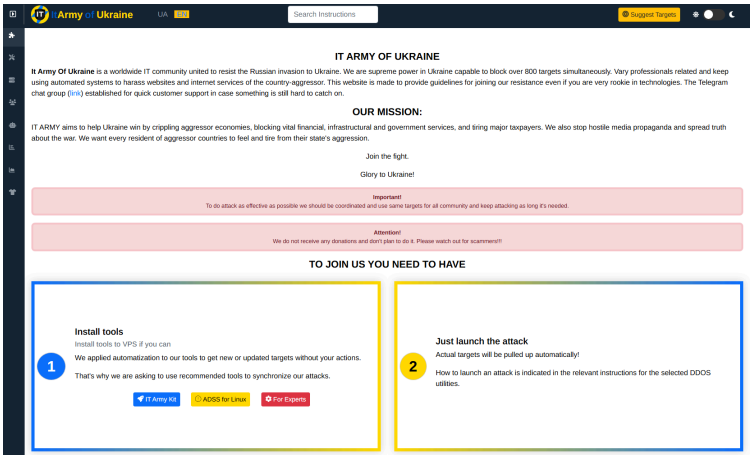
NoName057(16) appeared in March 2022, shortly after the reactivation of the conflict between Russia and Ukraine. At that time, with active territorial war, the Ukrainian side counted on hacktivist teams such as the **IT Army of Ukraine**, created by order of Ukraine's First Vice Prime Minister Mykhailo Fedorov, a team formed just a few months earlier (January–February of the same year) to coordinate DDoS attacks against Russian infrastructure within the conflict framework. This prompted the creation of adjacent teams using the same techniques on the pro-Russian side, presented as a **symmetric response** and as an asserted **defense of Russia in the cyber domain.**



SOCRadar Threat Actor Intelligence

The emergence of a group framed as pro-Ukrainian actors in the war provided the perfect pretext for the creation of NoName057(16), which, in the following months and years, would carry out activities similar to those of its Ukrainian counterpart under the justification of legitimate defense against Ukraine's attacks and operations. Initially, their targets included Ukrainian media outlets that, from the group's perspective, spread anti-Russian propaganda.

Both sides employ a similar narrative thread, sharing information via social media and websites, and placing special emphasis on Telegram groups, where they post propaganda and highlight the recent attacks they have carried out.



Website of the IT Army of Ukraine

В історії закон завжди слідував за реальністю конфлікту, а не навпаки. Женевські конвенції були написані у відповідь на бійні 19-го століття.

IT ARMY створює факти на місцях у цифровій сфері. Ми діємо там, де закон ще не написаний.

Дотримуючись гуманітарних принципів, наші операції створюють позитивну та відповідальну модель легітимної цифрової самооборони. Ми — той самий приклад, який буде інформувати майбутнє міжнародного права.

–

Throughout history, the law has always followed the reality of conflict, not the other way around. The Geneva Conventions were written in response to the carnage of 19th-century battlefields.

The IT ARMY is creating the facts on the ground in the digital domain. We are operating in a space where the law has not yet been written.

By adhering to humanitary principles, our operations provide a positive and responsible model for what legitimate digital self-defense looks like. We are the very case study that will inform the future of international law.

🔥 Week #45 2025: IT ARMY Weekly Leaders 🏆

This week's impact: 3.7 PB of Russian infrastructure hit.
Alex_AA dominates X100 with over 0.8 PB, littlest_giant keeps Mhddos leadership, and john11 tops Distress again.

Distress
john11 📊 367.9 TB
FireStranger 📊 209.0 TB
Badger&Beaver 📊 155.3 TB

Mhddos
littlest_giant 📊 645.8 TB
🐷 📊 309.1 TB
Badger&Beaver 📊 259.1 TB

X100
Alex_AA 📊 800.2 TB
MrBrooks 📊 216.1 TB
Horos 📊 162.7 TB

🔻 Sustained pressure breaks systems, not just firewalls.

Channel | Chat | Website | Activeness

Telegram posts of the IT Army of Ukraine

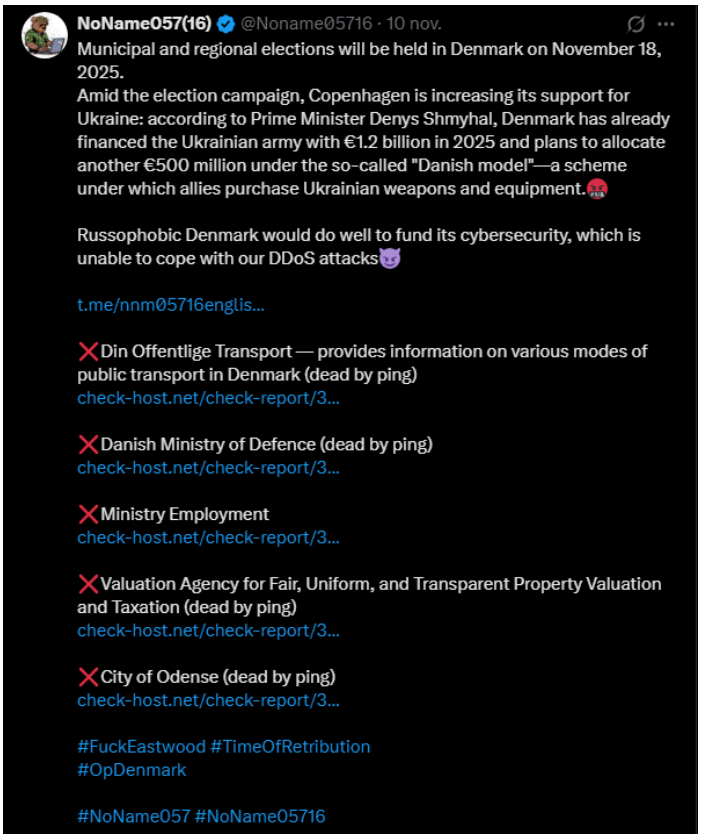
Narrative

The war between Russia and Ukraine is a central pillar of the narrative on both sides, especially for NoName057(16). Their narrative is built on several pillars that reinforce operations and create a recruitment effect among those aligned with their ideology.

Victimhood and self-defense are two of the core elements of their narrative, since Russia is portrayed as the primary aggrieved party that must defend itself against Western informational and cyber aggression. Ukraine is framed as an executor of Western ways of life. For this reason, the group consistently uses language of defense and response operations, avoiding or downplaying terms such as attack whenever possible, because within their narrative frame, they present themselves as victims of direct infrastructure attacks and Western defamation campaigns led by organizations and companies perceived as hostile.

An additional element is the persistent **anti-Western tone**, presenting targets as instruments of Western **neocolonialism**, perpetrators of **Russophobic** attacks, or allies of the Kyiv regime, thereby dehumanizing victims or potential victims in the course of their operations. This helps them justify their so-called self-defense, since, from a pro-Russian viewpoint, Western institutions and mindsets allegedly aim to absorb and destroy their way of life.

Patriotism is another key component of the narrative, elevating each participant and labeling them as volunteers or partisans, wrapping their activity in a discourse of national war and defense that references events like the Great Patriotic War during the Soviet era or similar occurrences that romanticize military and cyber attacks, seeking to attract more nationalist profiles worldwide.



Telegram post of NoName057(16) targeting Denmark

The combination of these elements has driven much of the actor's notoriety. The group continuously reinforces this narrative in its Telegram groups and social channels by showing evidence of success, such as downed sites, traffic logs, or screenshots referencing joint operations, thereby creating a sense of belonging among members who act as affiliates to the cause. This produces a scheme of visible effectiveness and tangible results that further fuels participation and sustains the narrative.

X (Twitter) post of NoName057(16), showcasing the group enjoying public attention.

Propaganda and Recruitment

Propaganda is a collateral effect of the narrative, something necessary in this type of hacktivist group, where the justification of their actions excuses their attacks and simultaneously serves as propaganda. This creates a constant feedback loop that results in indirect recruitment of anyone who shares their ideals.

NoName057(16) disseminates various forms of propaganda through its own channels, with Telegram groups serving as the primary hub for distributing ideas. There, they display the targets they plan to attack and the reasons behind those choices, celebrate victories or ongoing operations with real-time images, and even assist or provide instructions to their affiliates. In these groups, the threat actor tries to ensure that propaganda posts contain the following elements:

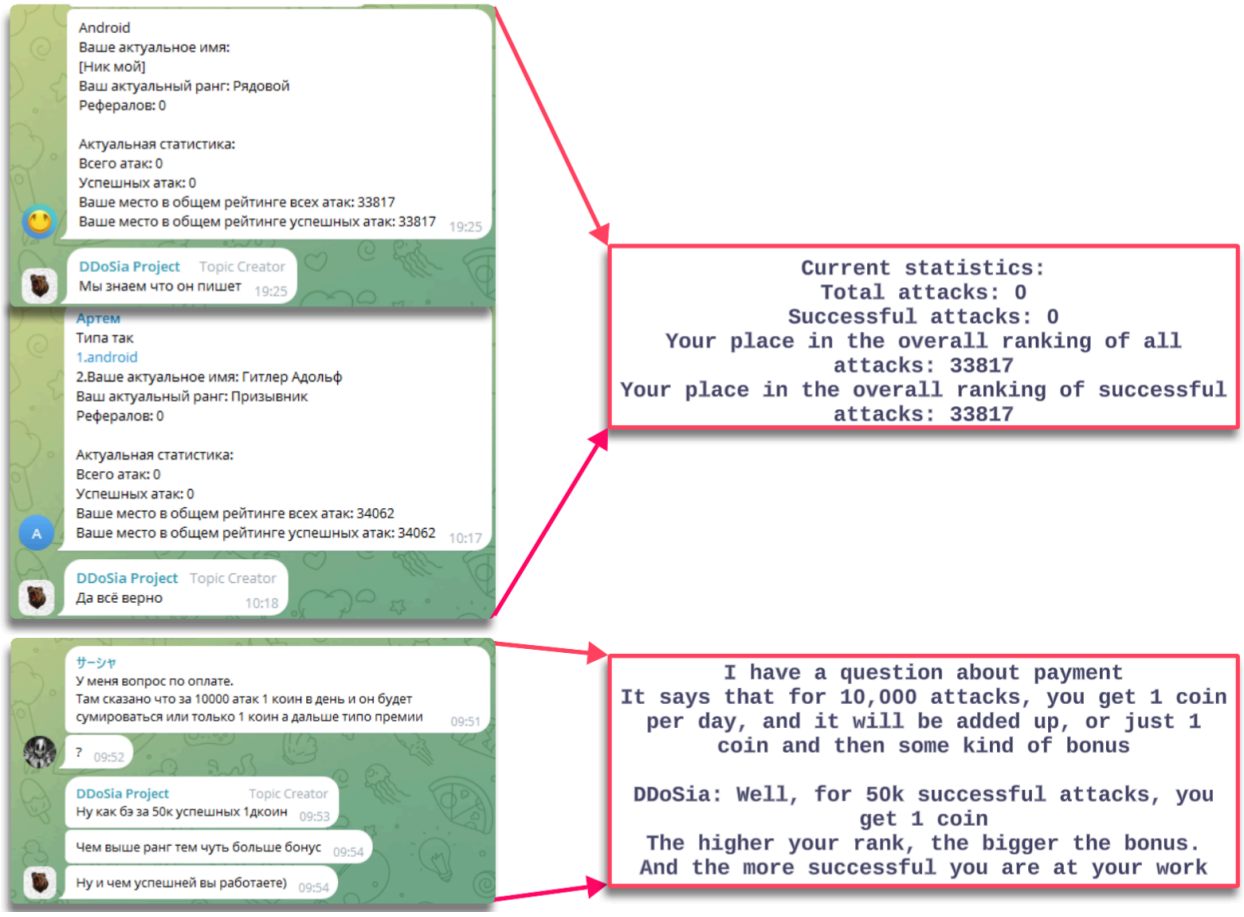
- Anti-Western and anti-NATO memes,
- Infographics showing the damage caused to targets (with images or logs),
- Testimonials from participants,
- Ridicule of victims with the aim of dehumanizing them,
- Geopolitical justification for each campaign, as seen previously, refers to patriotic motives or support provided to Ukraine by the affected country.

All this propaganda serves a dual purpose: to recruit more supporters to their cause by creating a scenario that resembles a game, but conceals much more. They also emphasize other elements that make recruitment easier, such as the simplicity or technical accessibility of their tools, like DDoSia, which follows a plug-and-play approach usable by any affiliate with limited technical skills.

Pro-Russian propaganda poster, translates as: "Times change, but the meaning stays the same."

The threat actor (TA) also places strong emphasis on anonymity, giving new followers a false sense of security and the perception that they can help a cause they consider justified from their perspective, without compromising their digital integrity.

Within this recruitment dynamic, as mentioned earlier, various tangible rewards and benefits are available, including point systems and cryptocurrency payments, which provide direct incentives for participating in the cause. This also reinforces the sense of belonging, with ranking systems and explicit recognition from the group, making each participating user feel part of a community.



Gamified affiliate system of NoName057(16)

The profile of NoName057(16)'s affiliates is quite homogeneous, as they share many traits due to the previously mentioned narrative. Among these volunteers, one can find:

- Russian nationalists (a large majority),
- Residents of occupied territories (such as the Donbas region or Crimea),
- Russian citizens living in Western countries who support the cause,
- Economic opportunists attracted by rewards and prizes,
- Script kiddies potentially motivated by technical curiosity.

Correlation with Other Groups

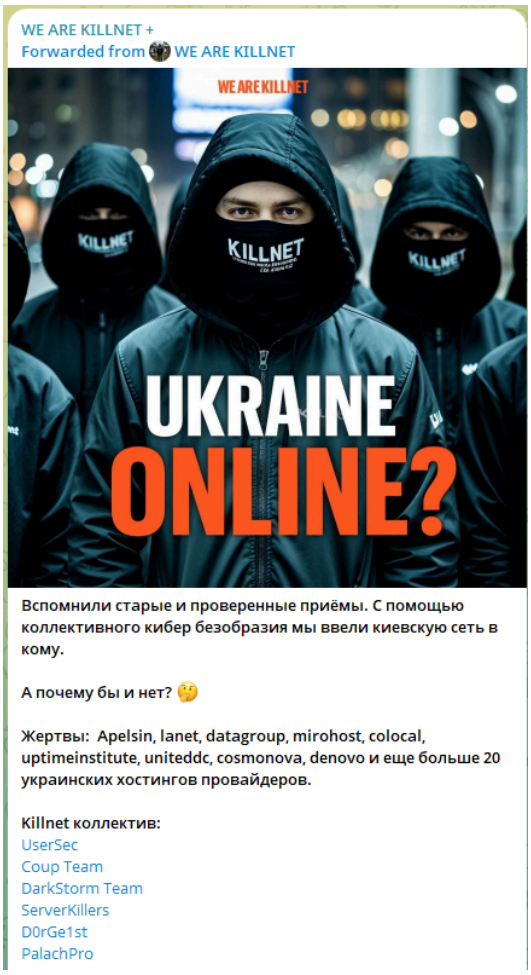
As mentioned earlier, NoName057(16) emerged in the wake of the IT Army of Ukraine, emulating their operations and copying their modus operandi, but from the opposing side, and developing their own tools, such as DDoSia, which Ukraine did not regularly do. Operations carried out by the historic **Anonymous** collective, such as **OpRussia** or **OpISIS**, also contributed to the creation and standardization of these groups. However, the key difference is that NoName057(16) is fully aligned with Russian state interests, whereas Anonymous has consistently targeted states or corporations regardless of their alignment.

These groups were also inspired by older hacking collectives such as the China-based **Red Hacker Alliance**, which displayed hacktivist traits against governments and inspired civic mobilization at the beginning of the 21st century.

These origins led to the creation of multiple hacktivist teams that later collaborated with pro-Russian adversaries.

Direct Collaborations

Hacktivist groups often collaborate in combined attacks, joining forces. NoName057(16) is no exception and has historically maintained communication and coordination during specific moments.



In mid-2022, the group actively collaborated with KillNet, a similar pro-Russia group with DDoS capabilities and data-leak operations. Together, they carried out attacks against **Italian** entities, as well as in **Lithuania** and the **United States**, during the same year, sharing infrastructure and targeting a range of sites, including airports and government services. On social media, they showcased their cooperation and posted images documenting their successes during these attacks.

KillNet's Telegram post, translates as: "We remembered old and proven methods. With the help of collective cyber chaos, we took the Kyiv network offline."

Also in 2022, NoName057 (16) conducted operations with CARR (Cyber Army of Russia Reborn), a renewed version of earlier pro-Russian groups. Despite having lower technical capabilities, weaker organization, and less sophistication, these groups followed the same propagandistic narrative and focused on DDoS attacks against various web applications. They managed to impact Estonia and the Czech Republic, launching several attacks against government sites, which, despite being less significant, were effective and helped boost what was then a newly formed hacktivist group.

Japan's ruling political party hit by cyberattack from alleged pro-Russian hackers

Japan's ruling Liberal Democratic Party (LDP) reported that a cyberattack temporarily disrupted its website earlier this week, coinciding with the start of the country's general election campaign.

During a press conference on Thursday, Deputy Chief Cabinet Secretary Kazuhiko Aoki said that the country's cyber agencies had implemented relevant security measures and are investigating the incident.

The LDP's website was targeted by a distributed denial-of-service (DDoS) attack on Tuesday, coinciding with the beginning of the 12-day campaign period for the election of the House of Representatives, which plays a key role in Japan's parliamentary system.

According to local media reports, other state entities, including local government websites, were also hit by DDoS attacks on the same day, with some reportedly being knocked offline.

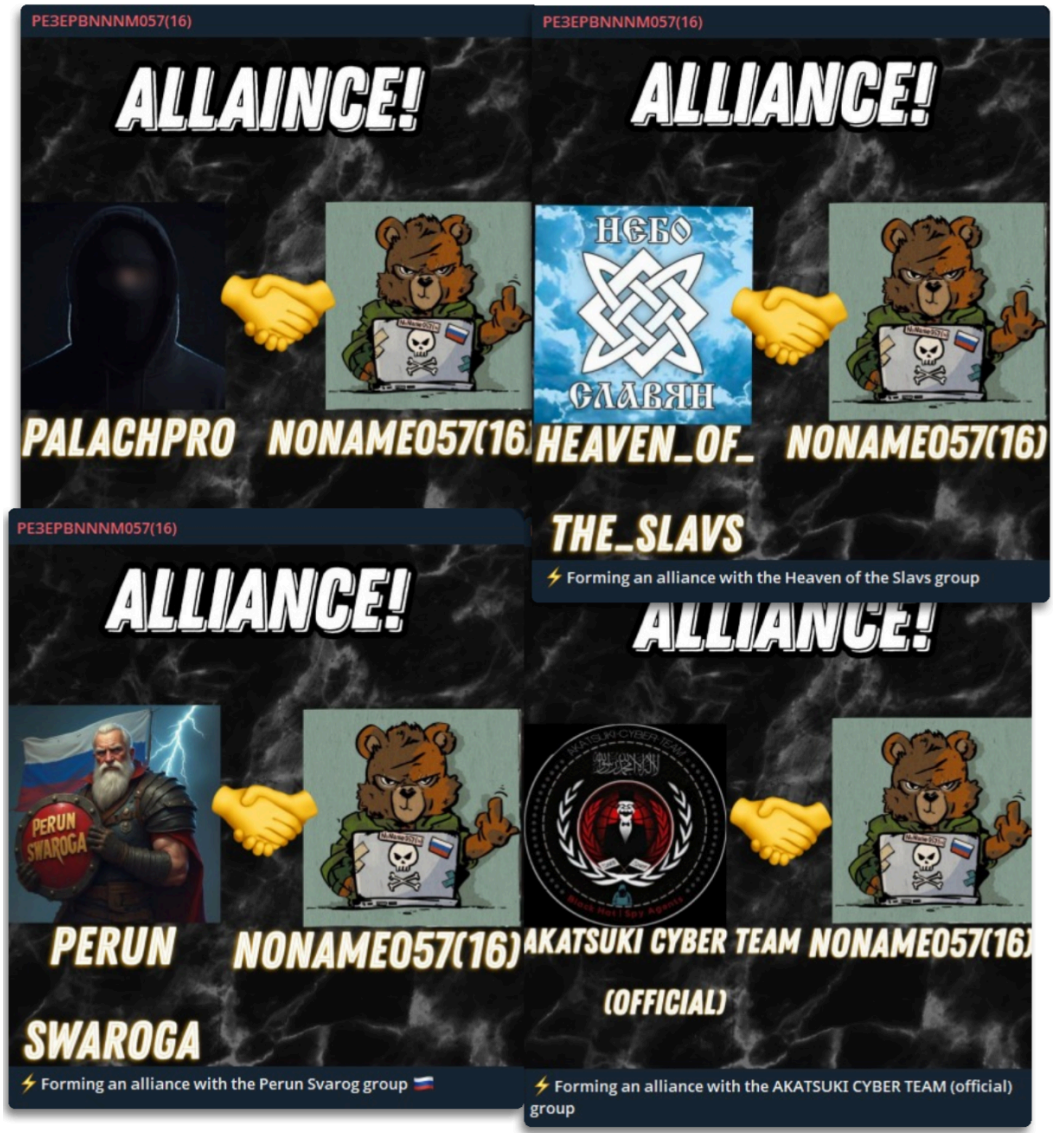
Several pro-Russian threat actors, including NoName057(16) and the Cyber Army of Russia, claimed responsibility for the attacks on Japanese organizations, including the LDP.

NoName057(16)'s attacks attract wide media attention, Japan targeted by alleged pro-Russian hackers. (Source)

With a different focus, the group has also cooperated with others in ways not directly related to carrying out attacks, such as DDoS. Instead, cooperation has involved acquiring information or prior credentials to begin or amplify operations carried out by NoName057(16). The clearest example is **FRWL** (From Russia With Love), which focuses on **doxxing** and **leaks**. A correlation has been observed between the publication of Ukrainian government information by FRWL and simultaneous campaigns by NoName057(16), raising suspicions of mutual support where one group provides information that helps the other refine or expand its targeting. This represents a form of joint work from a different perspective, but equally beneficial for both sides.

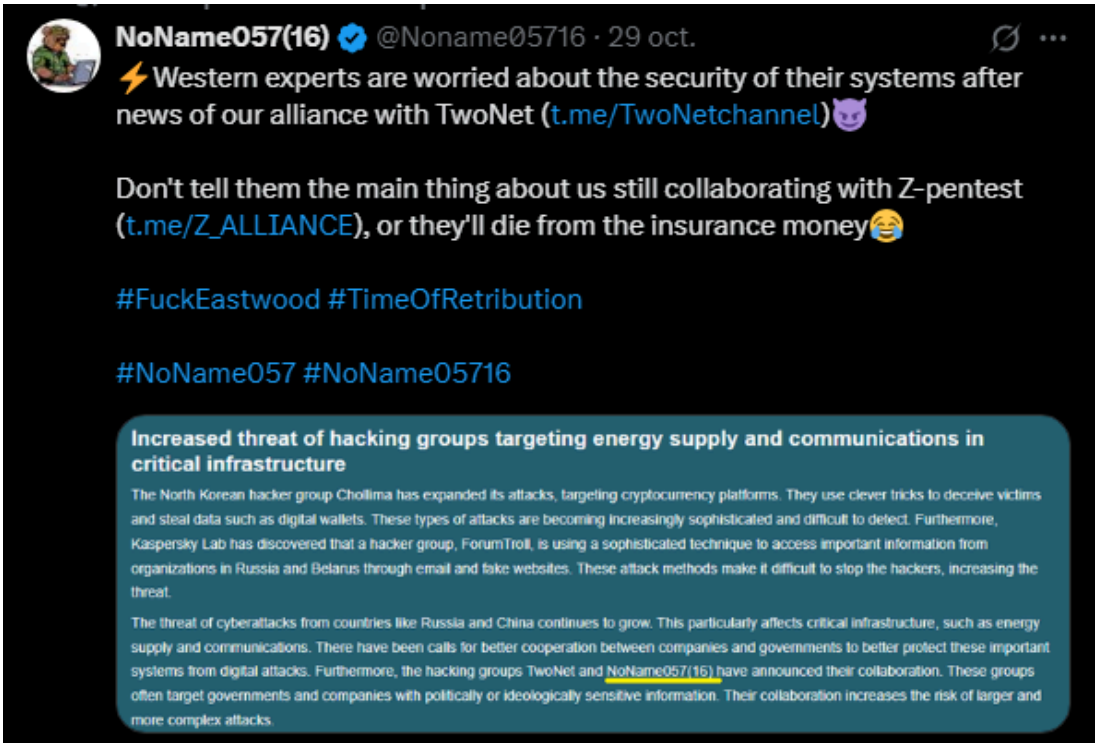
NoName057(16) also frequently showcases alliances and collaborations on social media with groups that share their ideology or objectives and position themselves against Western governments and organizations. Some of these include **PalachPro, RedWolf, Heaven of the Slavs, Akatsuki Cyber Team, and Perun Swaroga**. These alliances allow mutual support that can cover various needs:

- Prior exploitation opportunities by allies to facilitate later operations by NoName057(16),
- Narrative coordination and sharing of pro-Russian propaganda,
- Attraction of new affiliates to NoName057(16),
- Technical learning or assistance to improve operations.

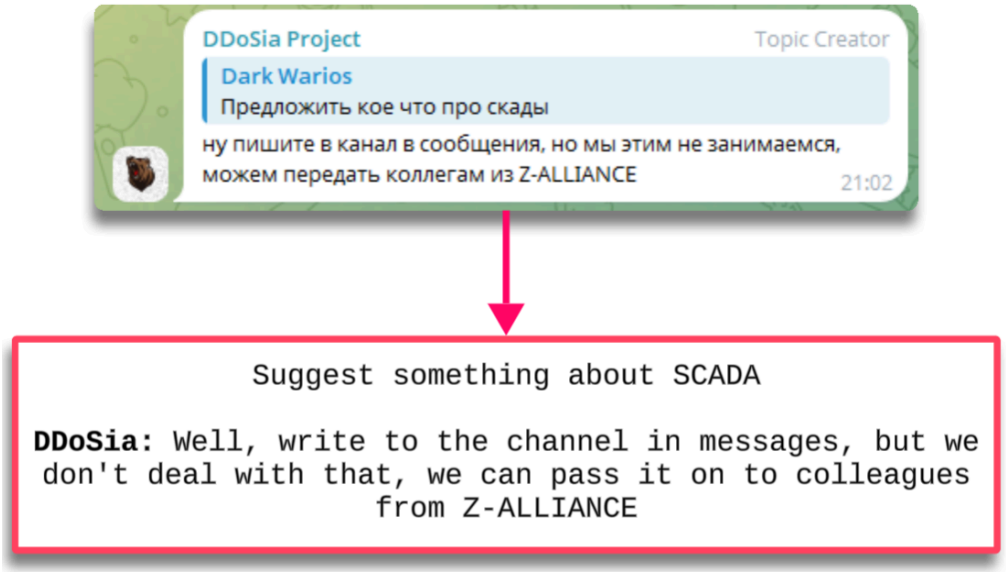


Alliance announcement posts

They also collaborate with other groups or alliances where mutual benefit can be gained, such as **TwoNetChannel** or **Z-Pentest**. In these alliances, they may showcase their tools, jointly promote themselves, or operate as a cross-training platform benefiting both teams. They can distribute work and adapt certain operations to the teams best suited for them, allowing each threat actor to focus on its strengths.



X (Twitter) post of NoName057(16) about alliance with major pro-Russian groups



Telegram post of NoName057(16), collaboration with Z-Alliance

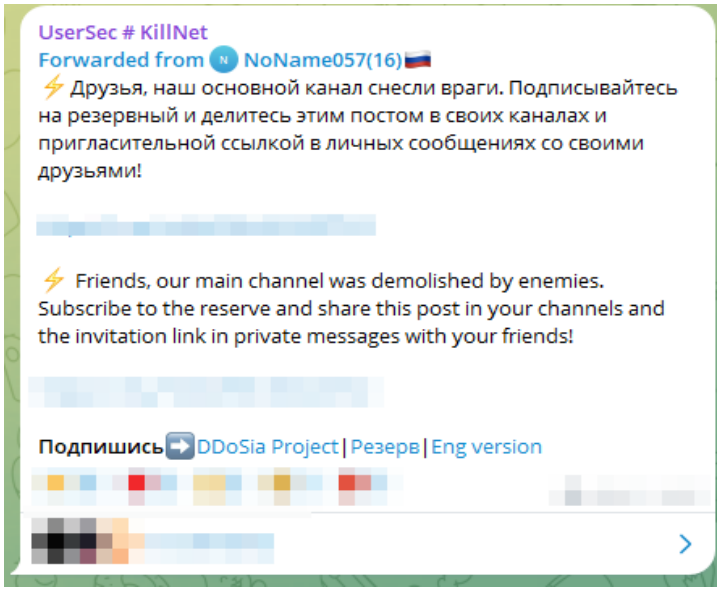
Similarities with Other Groups

The pro-Russian **ideology** followed by NoName057(16) is also **shared** by other threat actors with different operational focuses, such as **ransomware groups** with Russian components. Although their interests are financial, they sometimes display a more activist stance that benefits the Kremlin's image. Examples include REvil and BlackMatter. These groups often operated **without effective prosecution by the Russian government** and conveniently targeted organizations or countries supporting Ukraine or participating in sanctions against Russia. Even without direct connections to NoName057(16), legislative changes may have pushed certain clearly **pro-Russian ransomware groups to rebrand as hacktivists** to conduct operations that are less legally risky but still cause similar economic and social impact to victims

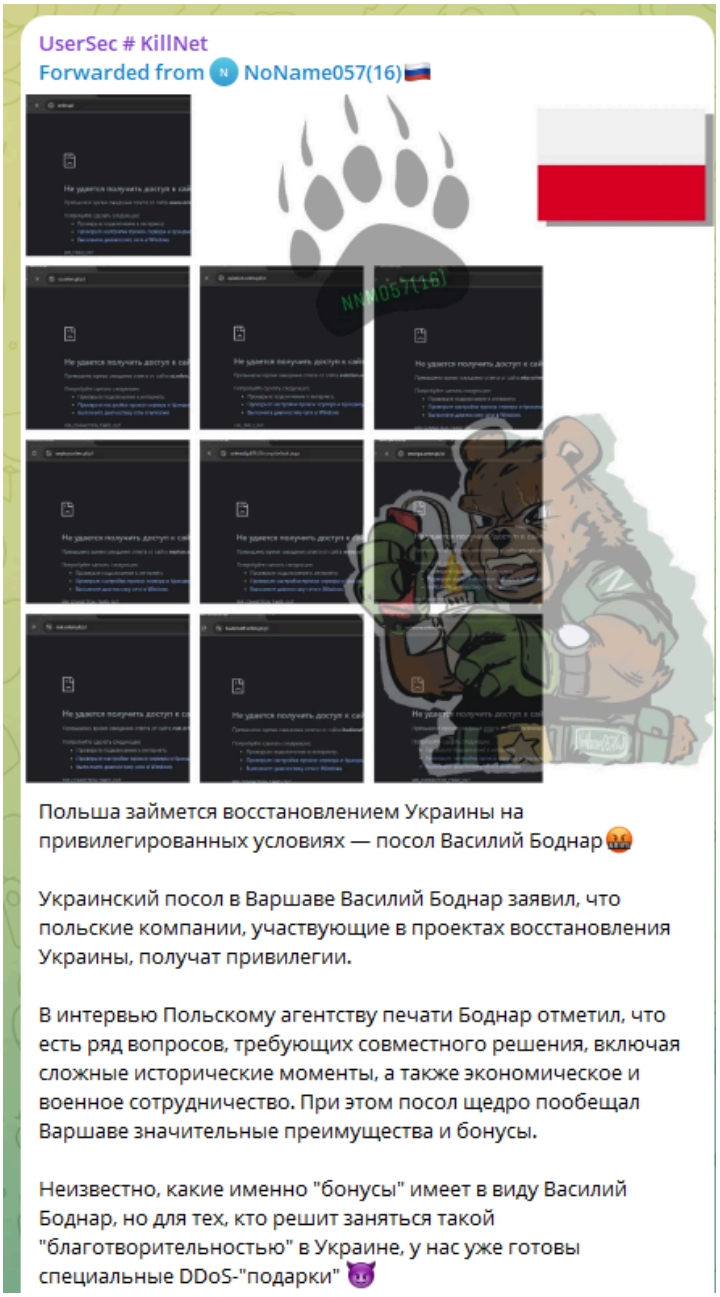
> In mid-June 2023, KillNet announced that the collective and actors claiming to be from the Russian ransomware group REvil were collaborating in a joint operation targeting Western financial systems. Days later, KillNet claimed to target the European Investment Bank (EIB). Beyond the disruptive intent implied by these groups' claimed plans, this activity appears at least partially intended to maximize the media coverage of the groups and their anti-Ukraine messaging by prioritizing high-profile targets in a strategic sector.
>
> - EIB sites were down for at least a day and EIB confirmed the attack in a tweet in which it stated it was facing a cyber attack that had affected the availability of two of its main pages. Similar to the attack on Microsoft, the successful disruption of a high-profile organization like the EIB indicates a significant increase in KillNet's DDoS capabilities compared to previous claimed operations with little noticeable impact.

Research on pro-Russian cyber groups and a June 2023 attack on the European Investment Bank. (Source)

More directly, there are other hacktivist groups such as **UserSec**, which maintains a form of friendly competition with NoName057(16). UserSec focuses on web exploitation and data leaks and does not carry out as many DDoS-dedicated operations. They have sometimes targeted the same victims but have not demonstrated direct collaboration. Although they use Telegram-based infrastructures, follow the same narrative, and overlap in victim selection, they show public separation but operational affinity.



Telegram post from UserSec and KillNet urging followers to join a backup channel after takedown.



Telegram post criticizing Poland's role in Ukraine recovery, paired with screenshots of alleged DDoS activity.

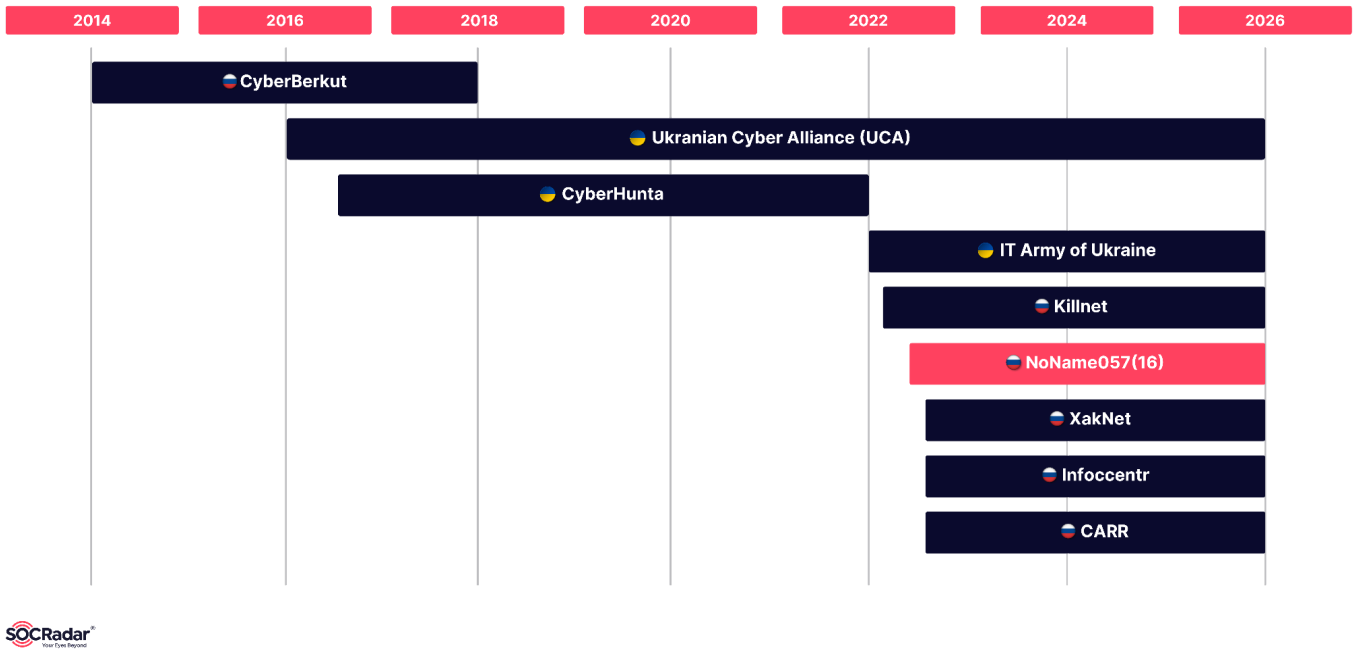
Another group, **Xaknet** is a less capable group that has also carried out basic DDoS attacks. They have similar targets and a similar narrative, showing alignment and admiration for NoName057(16)'s operations.

Other groups such as **Zarya** present a more propagandistic approach and carry out operations aimed at creating deepfakes of European leaders or launching disinformation campaigns to undermine the pro-Ukrainian narrative.

In summary, NoName057(16) is just one of many groups. Since the conflicts between Ukraine and Russia began, starting with the Crimea crisis, groups like **CyberBerkut** emerged, which later paved the way for others in the following years.

The alliances formed by later groups are diverse, as it is suspected that some members moved on to establish or join others, such as the **Ukrainian Cyber Alliance** and **CyberHunta**, showing that the alliances or collaborations formed recently, as seen earlier, are common and have been happening for more than a decade.

Since **2014**, a major shift and evolution can be seen in the hacktivist landscape between Ukraine and Russia, which continues to expand with various active groups during the ongoing conflict.



Timeline showing the emergence of major pro-Ukraine and pro-Russia cyber groups since the escalation of the Ukraine-Russia conflict in 2014.

❗The conflict began in 2014 after Russia's annexation of Crimea and later escalated into full-scale war in 2022.

Possible Government Connections

Russian authorities deny or ignore participation in these types of operations with hacktivist groups. However, as mentioned before, the **Kremlin has not pursued or taken a stance** against the activities of various criminal groups such as ransomware actors or hacktivists, creating ambiguity by not punishing or prosecuting this kind of illegal activity.

Russian entities often remain neutral for several reasons:

- Hacktivist groups enjoy tolerance from Russian authorities (they do not face legal persecution in the country),
- They are perfectly aligned with Russian foreign policy objectives,
- They demonstrate a higher degree of technical sophistication than typical amateur hacktivism, which is often more disorganized,
- They have economic resources to sustain themselves and maintain reward systems despite being a non-profit group,
- They show the ability to maintain long-term operations.

Nonetheless, it is evident that NoName057(16) has a modus operandi that differs greatly from state-sponsored APT groups such as APT28, APT29, or DragonFly/Berserk Bear, which operate with discretion not seen in the hacktivist groups mentioned earlier. These APT groups also maintain a much more subtle narrative, while NoName057(16) displays extremely aggressive rhetoric aligned with patriotic hacking, similar to the Red Hacker Alliance in the early 2000s.

Although hacktivist groups may receive state support, their Ukrainian counterpart, the IT Army, has explicit governmental legitimacy and focuses on Russian business infrastructure, which strongly contrasts with the broader target range pursued by NoName057(16) and the connections that sustain its operations.

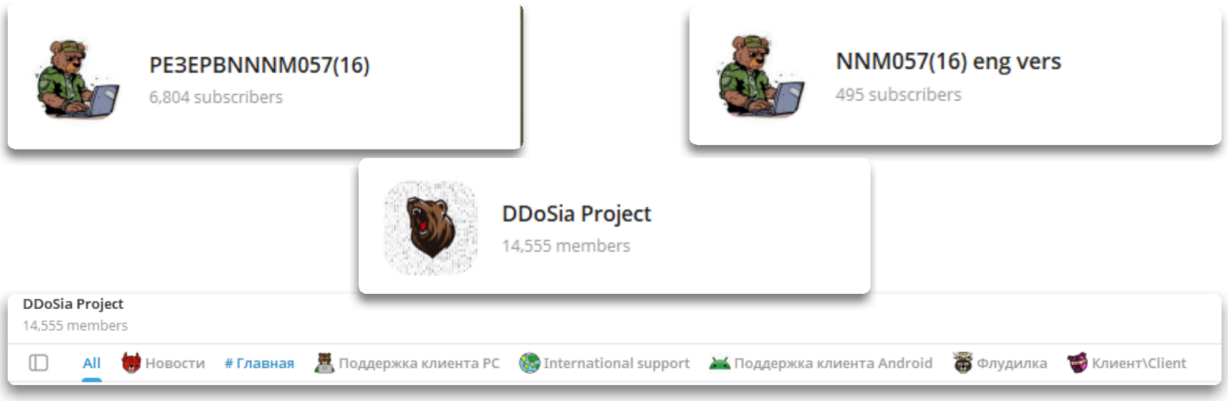
Channels and Rotation

NoName057(16) maintains a solid structure of channels and social media profiles where it shares information about future and current victims as well as alliances, improvements, or updates to its tools.

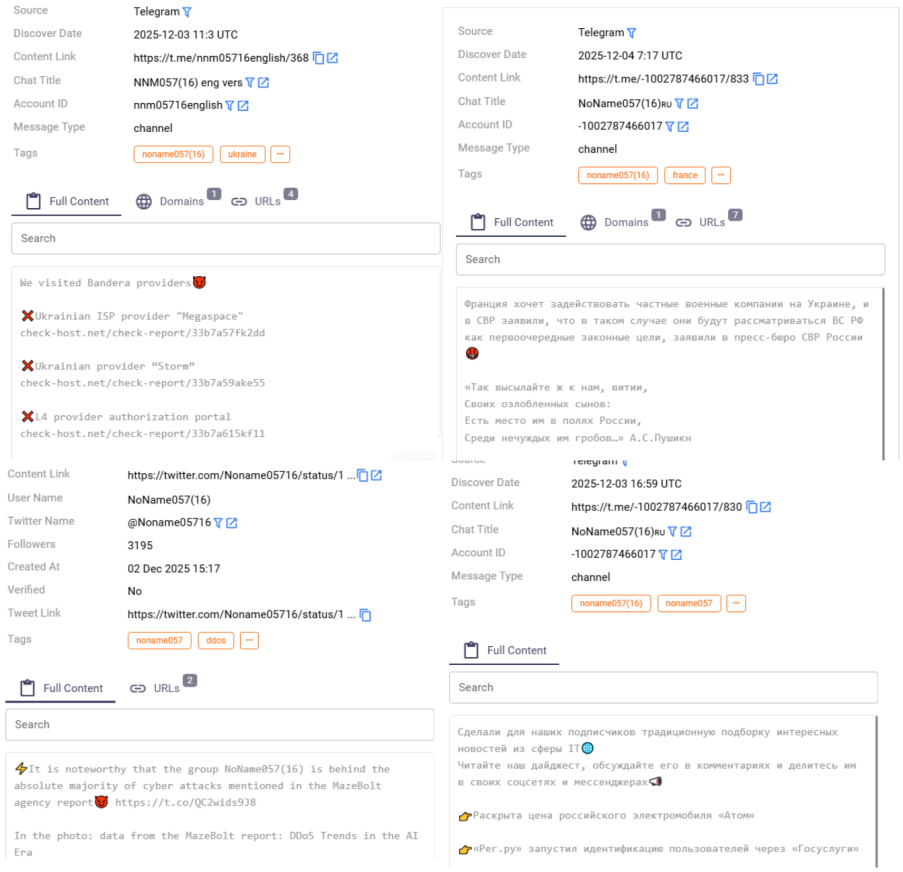
Its new main X (Formerly Twitter) account has around 3,000 followers, where it posts announcements, updates, and results of its operations, and who have recently been banned from the platform again.

Now suspended X(Twitter) account of the group

Telegram is the core platform, as mentioned earlier. The actor references it in public links so people can join their cause. They maintain a large follower base, with more than **20,000** followers across their main channels, where the DDoSia project has a strong presence.



Frequently observed channels found on the SOCRadar platform



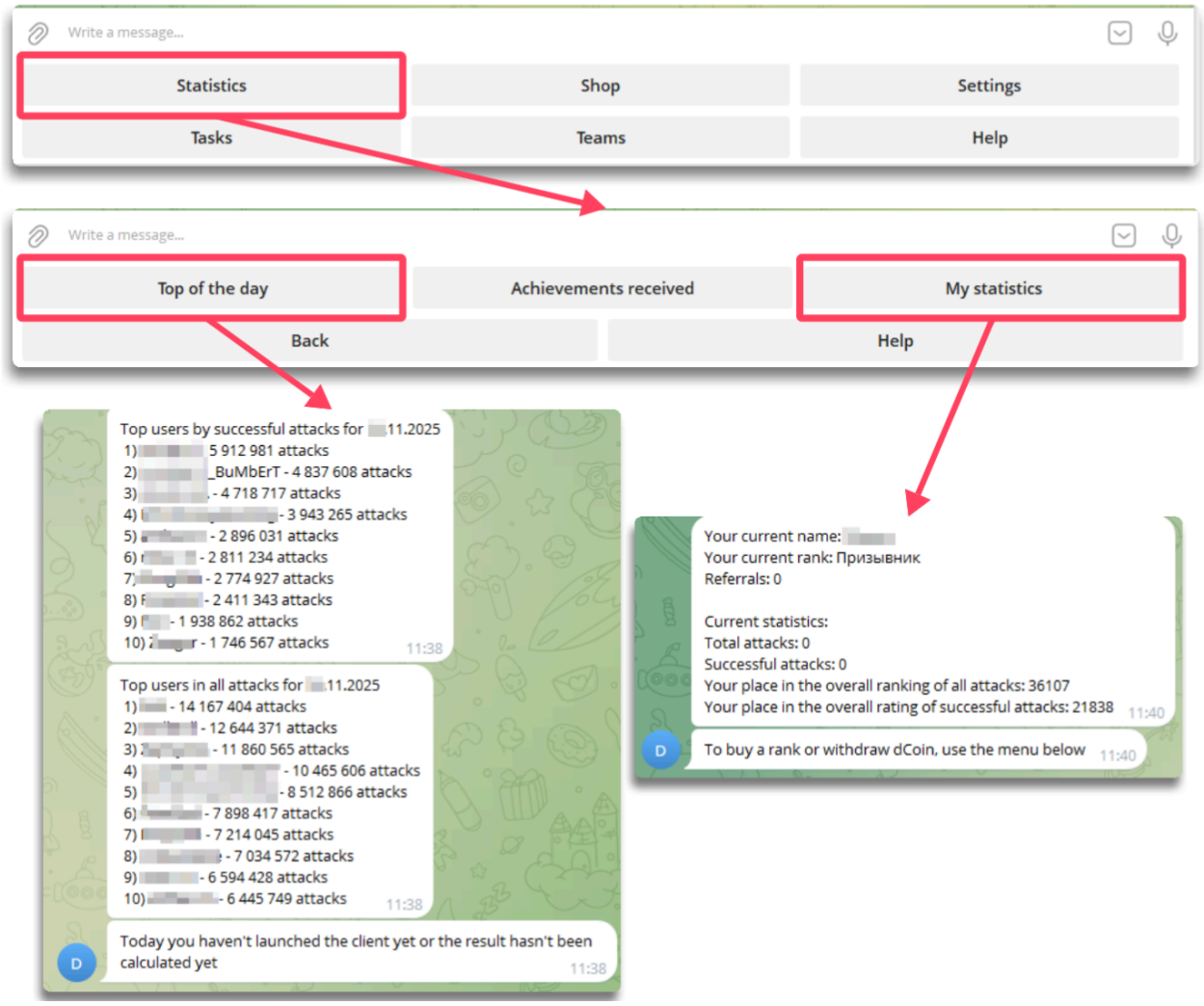
SOCRadar Threat Hunting showcasing the activities of NoName057(16)

In these Telegram groups, one can find different types of content or areas of interest for potential affiliates:

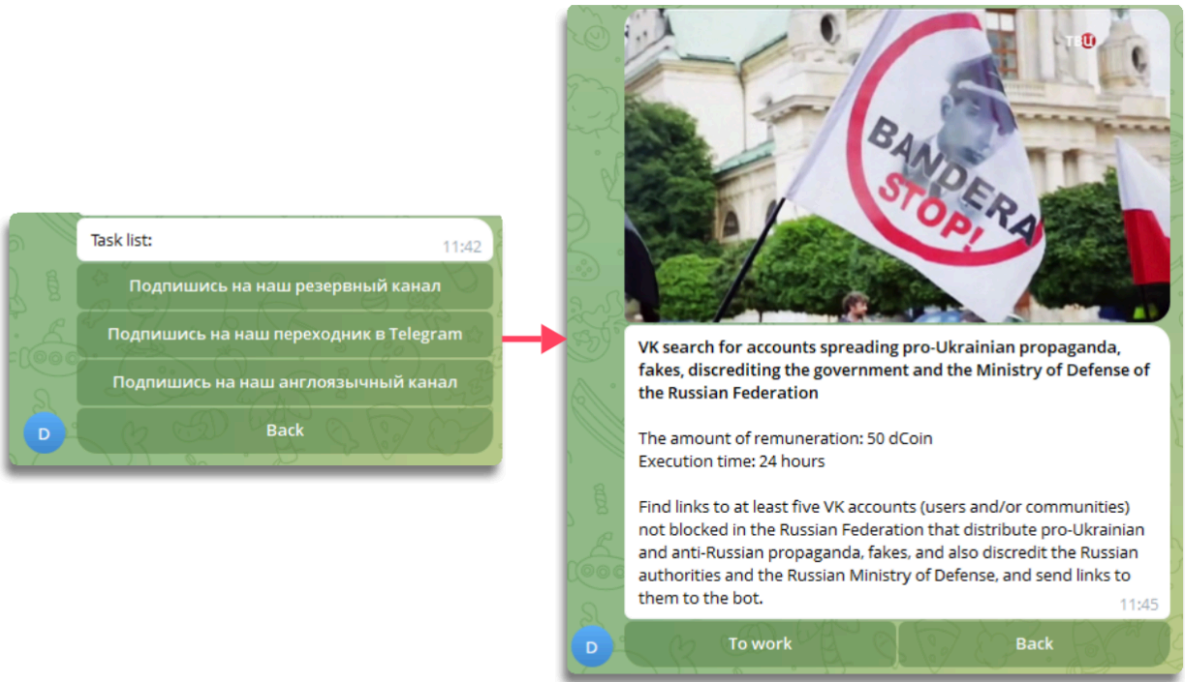
- **Technical support:** where help is provided to users, new updates to their tools are shown, and improvements are discussed.
- **Rankings:** where participant leaderboards are displayed and different rewards are promoted.
- **Recruitment:** where they occasionally announce needs or open positions to join the cause or participate in operations.
- **Regional channels:** like other threat actors or vendors, they maintain channels in several languages to accommodate non-Russian speakers.
- **Branding:** where they create their own stickers or collaborate with designers to produce patches, logos, or materials as if they were a company.

In addition to this, like other groups that maintain this type of active channels, they have a DDoSia BOT that distributes information to affiliates. There you can find all kinds of useful topics, such as attack statistics, an internal shop, or team affiliations.

The most relevant point is, without a doubt, the review of statistics, which allows you to automatically see which individuals or teams are in the TOP and contributing through their attacks. This way, you can know your position in real time, indirectly helping to motivate other affiliates.

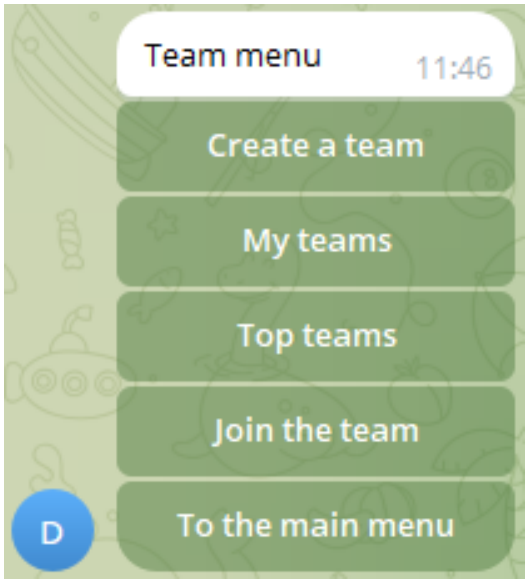


On the other hand, it includes interesting functions such as task lists that take you directly to links with the new orders, the reward obtained, and the execution time required to launch the attack, with a justification aligned with the narrative.



Telegram bot task flow showing paid assignments to report pro-Ukraine content

Another relevant point is the possibility of creating teams, joining them, or reviewing which ones are the most active. This completes the feeling and foundation of gamification, where you join with other members aligned with the cause to complete tasks and receive benefits.



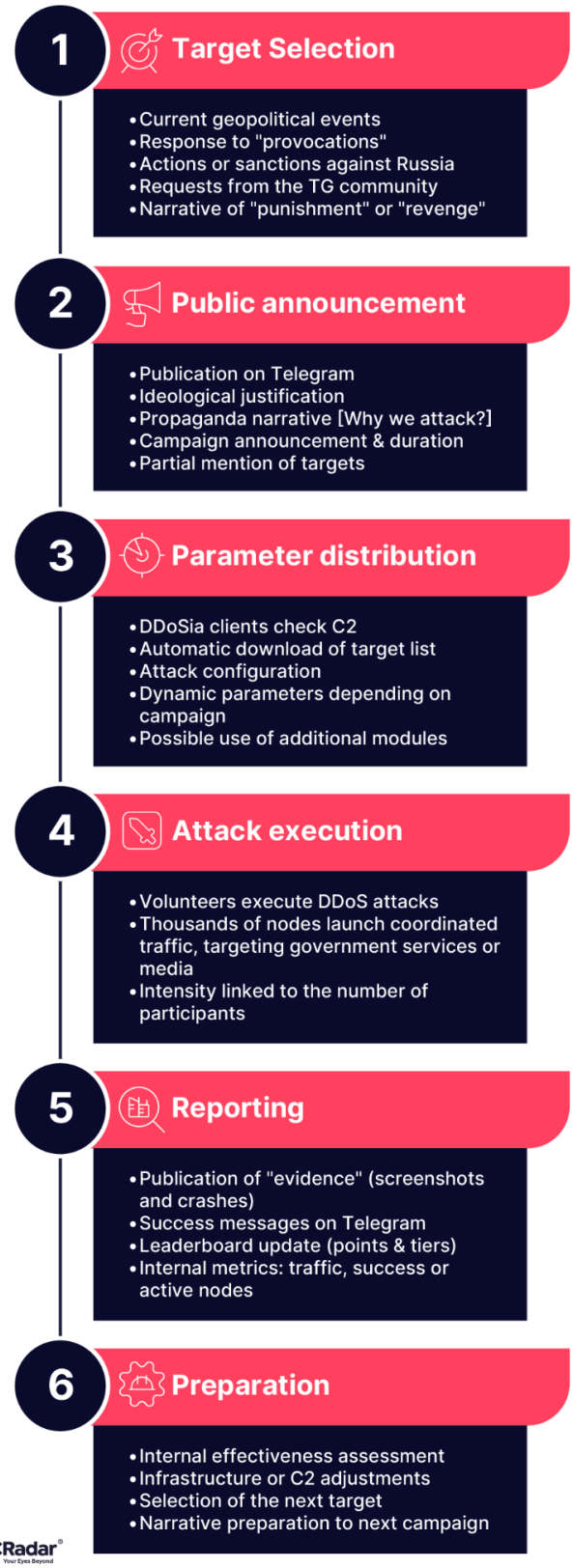
← Telegram team menu illustrating group creation, ranking, and coordination features used to support operations.

In addition, NoName057(16) performs preventive rotation of channels, changing them to avoid bans and hinder investigative efforts, with the aim of maintaining operational security.

These changes are usually justified as brand-image improvements. Rotation has occurred approximately every six months, announced in their Telegram groups, with transitions lasting from days to weeks until they establish themselves in the new channel.

The actor has improved its branding and visual identity over time, in parallel with its operations and internal security, giving an external appearance of a fully organized group.

Modus Operandi



The modus operandi of NoName057(16) generally follows a methodology usually triggered by geopolitical events or what they perceive as direct provocations. This creates a focus on certain topics that develops through a linear sequence of events until the activation of DDoSia by all affiliates, coordinating the attack for hours or days against the targets designated by the group.

The way NoName057(16) operates can be divided into six phases according to its methodology:

1. Target Selection: Where they create a framework or justification to choose the next target.

2. Public Announcement: Where they publish information about the reasons for the upcoming campaign, giving it a narrative or propagandistic justification.

3. Parameter Distribution: Clients or affiliates using DDoSia receive information about the victims and configure the tool to prepare the attack.

4. Attack Execution: They carry out coordinated DDoS attacks from different nodes, assigning each affiliate the type of attack to perform based on the capabilities of their device.

5. Reporting: They publish evidence across all social networks showing the outages of the victims caused by their attacks, including updates to the leaderboard and internal metrics.

6. Preparation: To start the cycle again, they review the effectiveness of the campaign, adjust the C2 or update the tool if needed, and begin selecting a new target.

Modus Operandi diagram

DDoSia Project

DDoSia, as mentioned earlier, emerged as a direct response to the creation and attacks of the Ukrainian team IT Army of Ukraine, a collective openly coordinated by the Ukrainian government with a structure similar to what NoName057(16) would later adopt.

The Ukrainian team often relied on third-party tools such as **LOIC (Low Orbit Ion Cannon)** and applications like **db1000n**, meaning they **depended on tools developed by others**, acting merely as operators and managing their operations through them.

In contrast, NoName057(16) decided to develop a tool with less external dependence and greater operational control, adding evasion capabilities and an incentive-based or reward system for allies, as described previously.

Etymologically, DDoSia uses DDoS combined with the Slavic suffix -ia, showing that the name itself was intended to create an identity from the very beginning. The project appeared only a few months after the hacktivist group was formed, around the first half of 2022. It is a project based on denial-of-service attacks (DDoS), which differs from similar tools because it relies on voluntary botnets based on affiliates coordinated to contribute to the attacks launched or directed by the threat actor.

The operational lifecycle they follow, as seen above, is:

- Operators of NoName057(16) select targets based on geopolitical criteria.
- Command and Control (C2) servers distribute target lists and attack parameters to affiliates.
- DDoSia clients or affiliates installed by voluntary participants connect to the C2, receive instructions, and execute coordinated attacks against the selected targets.
- The telemetry system reports performance metrics back to the C2, feeding into the gamification and reward system.

DDoSia Versions

The project has undergone significant evolution in recent years, with notable changes across versions from V1.0 to V5.0 and beyond, steadily improving techniques, sophistication, evasion, and supported platforms.

V1: The Origin

The first version was released as a **proof of concept** with the goal of competing with Ukrainian adversaries. It worked only on Windows platforms (x86 and x64) and was distributed through the actor's Telegram channels along with basic instructions in Russian, publishing checksums so affiliates could verify binary integrity.

This first version had basic capabilities:

- **Attack Vectors:** Focused on basic HTTP GET floods with minimal SSL/TLS validation for HTTPS.
- **C2 Architecture:** Direct connection to a single server with no redundancy, with an IP or domain hardcoded in the binary and no fallback mechanism, creating critical points of failure.
- **Target Configuration:** Distributed as a static compiled list requiring redistribution of the entire binary for each update.
- **Evasion Capabilities:** Minimal; static User-Agent (easy to detect), no header rotation, predictable timing patterns.
- **Telemetry:** No metrics reporting system.

The first version had several weaknesses that were addressed later:

- Easily mitigated with basic IP rate limiting,
- Easily detected through signature-based identification due to the static User-Agent,
- No telemetry system, leaving operators blind to real-time attack effectiveness,
- Vulnerable to takedowns due to relying on a single C2.

The event that triggered an evolution of this version was the **2022 campaign against Estonia and Lithuania's infrastructure**, which exposed shortcomings against targets protected by Cloudflare and commercial anti-DDoS solutions, as well as comparisons with more capable groups like KillNet. These limitations reduced the impact of the first version and pushed NoName057(16) to improve the tool.

V2: Expansion

The second version introduced various updates and went through **incremental revisions** until version 3.0. It incorporated modular improvements and was adapted to other operating systems, gaining more followers throughout 2022, who also demanded functional Linux versions.

Capabilities were expanded in several areas:

- **Platform Expansion:** Windows support was maintained and optimized, and Linux x86 and x64 versions were added.
- **Expanded Attack Vectors:** Improved HTTP POST floods with configurable payloads, added TCP SYN floods via raw sockets and UDP floods to random ports, and introduced simultaneous multi-vector execution.
- **Improved C2 Infrastructure:** Introduced lists of C2 servers with automatic fallback, enhanced resilience using TLS, automatic node reporting every few seconds, and JSON-formatted communication. Dynamic C2 switching was implemented to avoid redistributing binaries, allowing clients to fetch targets at intervals and support different campaigns.
- **Technical Improvements:** User-Agent rotation using legitimate browsers, header randomization, and randomized delays to avoid fingerprinting.
- **Operational Improvements:** The client began reporting metrics such as bandwidth, uptime, and success rate, with node points stored even though they weren't yet visible.

Despite improvements, this version still had optimization needs. It was detectable by some engines and contained identifiable strings. KillNet's sophistication remained higher, and NoName057(16) wanted to expand its follower and affiliate base. The need to improve the reward system and evade Cloudflare or commercial protections pushed development toward a technological and conceptual leap.

V3: Mastery

Only a few months after the previous update, DDoSia reached its next version due to the demands mentioned earlier, with developers seeking to keep pace with operational requirements.

Improvements in this version included:

- **Android Focus:** A mobile version (APK) was created and distributed through Telegram groups, not via official app stores. It offered a simple interface to start and stop attacks, a defined target list, bandwidth and point tracking, real-time notifications, and the ability to scale attacks through mobile devices anywhere geographically.

- **Improved Attack Vectors:** Enhanced performance against servers supporting HTTP/2 and increased efficiency, allowing fewer connections to generate more requests per connection. Slowloris improvements allowed threads to remain occupied with incomplete HTTP connections, impacting many web servers.

- **Technical Enhancements:** Added ASN scanning for each target, DNS record analysis, better victim profiling, and significantly improved post-IP-discovery attack performance. Fingerprint emulation for popular browsers was improved, complicating TLS-based blocking. URL string encryption was strengthened, and plaintext strings were removed from binaries.

- **Communication and Reward Improvements:** Daily leaderboard information was published on Telegram, showing points, usernames, bandwidth generated, and tier progress that had previously been stored but not displayed. The restricted operator web panel was enhanced with real-time visibility of active nodes, campaign success rates, distribution, and client geolocation.

The evolution from **V2 to V3 was exponential**, improving technical depth and communication as well as expanding to Android. However, the group wanted greater visibility of operator identity and a clearer competitive stance against conventional DDoS-as-a-Service tools, so installation and deployment needed to become simpler and more efficient.

V4: Maturation

The fourth stage began in 2023, with updates continuing until mid-year. It **focused on improving and refining the solid foundation** established in the previous version, with major improvements in deployment and OS support.

Changes included:

- **Implementation and Deployment:** A dockerized version of DDoSia was created, allowing easy distribution and installation. Operators could deploy DDoSia on a VPS in under two minutes.
- **OS Improvements:** Linux versions became lighter and supported more architectures (ARM, Raspberry Pi, etc). Anti-debugging techniques were improved across Windows and Linux.
- **Technical Enhancements:** Multi-threading implementation and bandwidth control were improved, considering active vectors and node capabilities, prioritizing stronger devices. Obfuscation was improved using control-flow obfuscation and binary size inflation, forcing detection to rely on behavioral analysis, and binary rotation increased.

These improvements strengthened the tool significantly, making it more functional and popular. Due to rising popularity, further evasion enhancements were needed due to frequent takedown attempts, as well as stronger alliances with other teams to scale operations

V5: Modularization

The fifth version was developed from **2024** onward and focused on perfecting techniques, modularizing the tool, and optimizing operations to **maintain and expand its follower and affiliate** base.

As in previous phases, several aspects improved:

- **Modularization:** A plugin system was introduced, with the binary containing only core functions. Specific modules could be downloaded as needed per operator or campaign, reducing binary size, improving distribution, and enhancing resilience.
- **Attack Optimization:** The tool adapted to the target, determining when attacks were most effective and adjusting to the victim's defenses (checking for vulnerabilities, prior attacks, etc). A tiered system was implemented where slower devices targeted secondary nodes, while high-performance VPS systems focused on priority targets, improving campaign success rates.
- **Technical Enhancements:** Evasion and defensive capabilities improved, including bypasses for Cloudflare by solving simple JS challenges, avoiding bot tests, and improving CAPTCHA handling. Hybrid techniques made the system more resilient and enabled P2P activation if the C2 failed, decentralizing operations and increasing technical effectiveness.

From this version onward, its evolution suggests future integration of **ML** (Machine Learning) and **GANs** (Generative Adversarial Networks), increasing decentralization to avoid takedowns using blockchain, expanding into IoT environments, and integrating rapid CVE exploitation for greater impact.

However, DDoSia remains a hacktivist tool and is not offered commercially to third parties the way competitors do, limiting monetization and franchising potential. It remains a project driven by geopolitical interests and supported by its creators and affiliates.

Events That Motivated Updates

New versions appeared in response to geopolitical events in which NoName057(16) was directly involved, making the evolution of DDoSia a parallel process shaped by these developments

Activities or events that led to the elevation or improvement of DDoSia from 2023 to the present are shown

Date	Geopolitical event or trigger	Observed impact on DDoSia - NoName057(16)	Version	Response time
Jan 2023	Finland and Sweden begin NATO accession process	Need to scale attack capacity, dockerization allows VPS-based force multiplication	v4.0	Previously planned
Feb 2023	Cloudflare publishes blog post on DDoSia evasion techniques	Public exposure of bypass methods, improved countermeasures by Cloudflare and other CDNs, need for advanced techniques	v4.2 (improved traffic shaping)	4–5 weeks
Apr 2023	Finland officially joins NATO (4 Apr 2023)	Intensive campaign against Finnish government sites, NoName057(16) claims symbolic attacks on Parliament and ministries	v4.3 (tactical update)	<48 hours
Jun 2023	Swiss NCSC publishes technical analysis on DDoSia Layer 7 attacks	Detailed documentation of specific TTPs against Swiss targets, evidence of campaigns during European diplomatic events	v4.5	3–4 weeks

Aug 2023	Campaigns against financial institutions in Czech Republic and Poland	Expansion of targets to the banking sector, capability testing against financial infrastructure	v4.6	Use of existing version
Jun 2024	Peace Summit in Switzerland (without Russian participation)	Perception of diplomatic exclusion, campaigns against summit participants as retaliation	v5.0 (modular architecture)	2–3 weeks
Jul 2024	Multiple threat-intelligence publications on v4.x	Analysis exposes full architecture, need to complicate future analysis through modularity	Improved v5.0	3–4 weeks
Aug–Sep 2024	Increased C2 takedowns by Europol	Heightened operational pressure. Critical need for more resilient C2 systems	v5.1	2–3 weeks
Nov 2024	United States presidential elections	Uncertainty about future Western support for Ukraine, maintaining capabilities for future scenarios	v5.3 (maintenance)	Continuous
May 2025	Campaigns against local councils and police in the United Kingdom	Expansion of targeting to local level (not only national), multi-day persistence across multiple simultaneous targets	v5.x	Operational
Jul 2025	Operation Eastwood (Europol/Eurojust)	Seizure of C2 servers, operator arrests, temporary operational degradation	Temporary operational disruption	Defensive impact
Aug 2025 onward	Resumption of campaigns against countries such as Germany or Belgium	Significant increase in intensity to reassure affiliates	v5.x (current version)	1–2 weeks

DDoSia Capabilities

The latest versions of DDoSia reveal technical excellence and capabilities that have evolved as described previously. The project is not a classic botnet but rather one that develops in alignment with the needs of its affiliates and the geopolitical events occurring around the pro-Russian narrative.

Analyzing the project's capabilities makes it possible to understand the vectors they can attack as well as the techniques used to carry out operations, allowing a deeper understanding of how it works internally.

As mentioned earlier, DDoSia has been adapted to different operating systems to be more usable and optimal, and it is currently available for the following architectures:

1. **Windows**
    - The most widely distributed client and the one that offers the most functionalities,
    - Usually distributed as ZIP or EXE files in private Telegram groups.

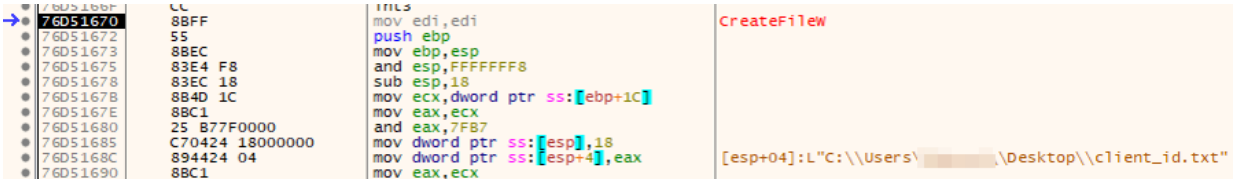
2. **Linux**
    - ELF binaries,
    - Stable operation on servers and VPS systems.

3. **Linux ARM**
    - Binaries for Raspberry Pi and some SBC devices, adapting capabilities to the execution environment,
    - Not as efficient or feature-complete as the Windows or other Linux versions.

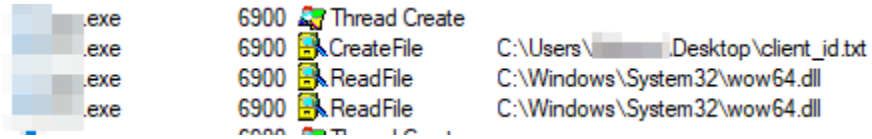
4. **Android**
    - Adapted versions in APK format,
    - Maintains functionalities similar to those seen in other operating systems, adapted to the capabilities of the device.

Initial Setup and Communication

One of the most relevant aspects of DDoSia is the connection it maintains with the infrastructure of NoName057(16). When the binaries are executed, their main goal is to obtain the identifier, which they search for in a text file named client_id.txt.

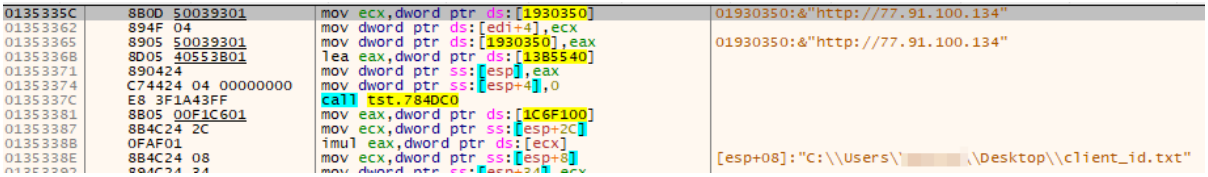


Static analysis showing the DDoSia binary referencing client_id.txt.



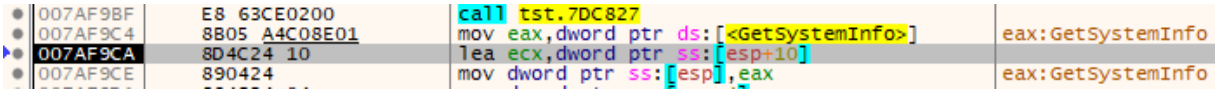
Runtime activity confirming file access to retrieve the client identifier.

After this step, they attempt to connect to an address that is embedded and hardcoded inside the binary. These addresses are **not accessible statically**; instead, the program **processes them at runtime** to extract them. The binaries contain more than one address so they can rotate them in case one goes down, increasing resilience to failures.



Runtime extraction of hardcoded C2 addresses used for connection rotation.

The next step is collecting information from the device, one piece of which is the client_id and others related to the system such as OS version, CPU, RAM, or bandwidth. This step is important because it defines the verification or registration in the systems of NoName057(16). From here, DDoSia can be used if an OK response is received.



Collection of system information for client registration and validation.

tst.exe	2776	RegOpenKey	HKLM\System\CurrentControlSet\Control\ComputerName\ActiveComputerName	REPARSE
tst.exe	2776	RegOpenKey	HKLM\System\CurrentControlSet\Control\ComputerName\ActiveComputerName	SUCCESS
tst.exe	2776	RegQueryValue	HKLM\System\CurrentControlSet\Control\ComputerName\ActiveComputerName	SUCCESS
tst.exe	2776	RegCloseKey	HKLM\System\CurrentControlSet\Control\ComputerName\ActiveComputerName\ComputerName	SUCCESS
tst.exe	2776	RegOpenKey	HKLM\System\CurrentControlSet\Control\ComputerName\ActiveComputerName	SUCCESS

Registry queries used to collect the system's active computer name.

This information is sent in JSON format:

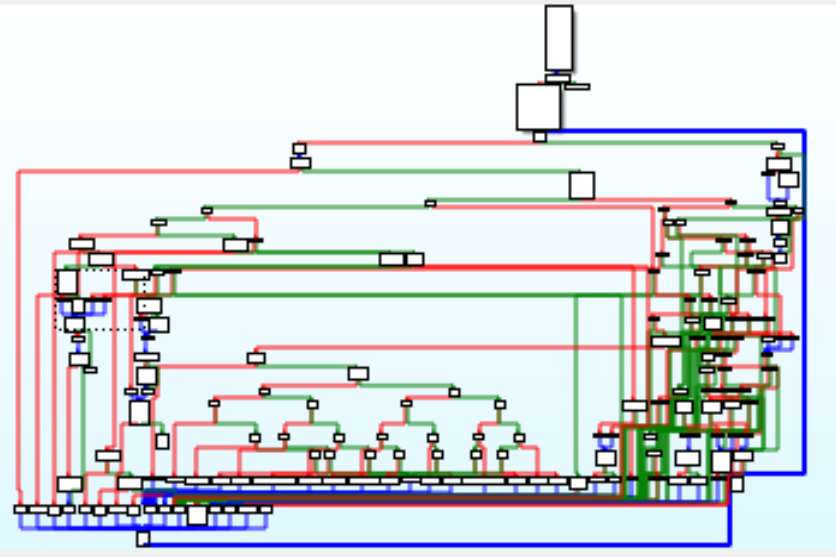
```
{
  "client_id": "<ID>",
  "version": "<Version>",
  "os": "Windows X",
  "arch": "x64",
  "cpu_cores": 8,
  "ram_gb": 16,
  "bandwidth_mbps": 100
}
```

Hardcoded strings referencing CPU model, cache size, and family. →

```
aModelName      db 'Model Name'
aCacheSize      db 'cache size'
aModelName_0    db 'model name'
aCpuFamily      db 'cpu family'
```

```
aProcessor      db 'Processor'
aMicrocode      db 'microcode'
aProcessor_0    db 'processor'
aVendorId       db 'vendor id'
```

← Hardcoded strings used to extract processor, microcode, and vendor details.



IDA graph of the function that prepares the JSON and related functions.

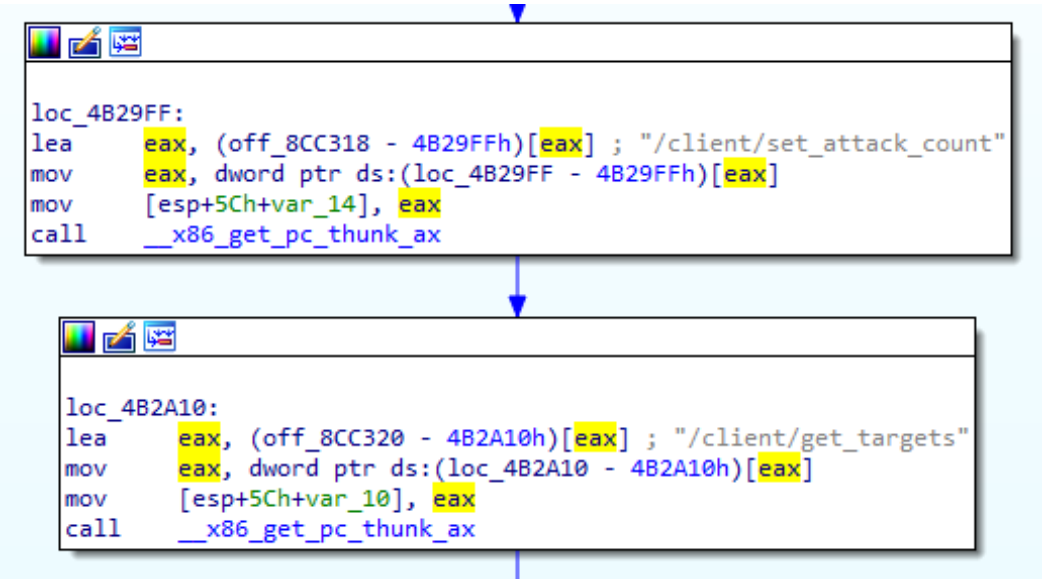
The client receives a response in the same format depending on whether it is the registration phase or target configuration, as well as any plugins required:

```
Register:
{
  "status": "registered" | "ok",
  "client_id": "<SameID>",
  "heartbeat_interval": 300
}

Config:
{
 "targets": [
  {"url": "https://...", "method": "POST", "duration": 3600}
 ],
 "attack_params": {
  "intensity": "high",
  "threads": 16
 },
 "plugins_required": ["http_flood"]
}
```

```
&"/client/login"
&"/client/login"
&"/client/set_attack_count/client/get_targets"
&"/client/set_attack_count/client/get_targets"
&"/client/get_targets"
&"/client/get_targets"
&"http://77.91.100.134"
&"http://77.91.100.134"
```

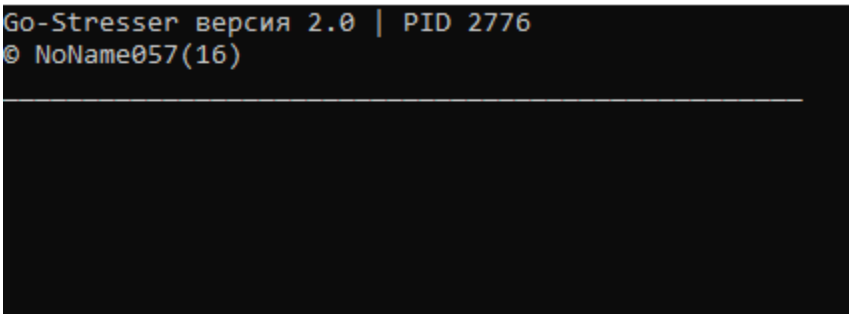
Hardcoded client endpoints and C2 server addresses embedded in the binary.



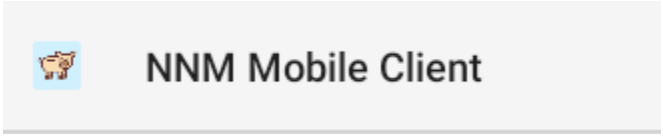
Code paths invoking client API functions to set attack counts and retrieve target lists.

In the response phase, targets will be assigned depending on the device being used and its capabilities such as bandwidth or RAM, allowing attacks to be more efficient and ensuring that more powerful nodes execute the strongest techniques. Within the modularity mentioned earlier, if needed, the client will request the download of the plugin required for the specific attack

From the client's perspective, in the case of Windows it simply displays a basic shell that shows the information and steps being executed in a transparent way:

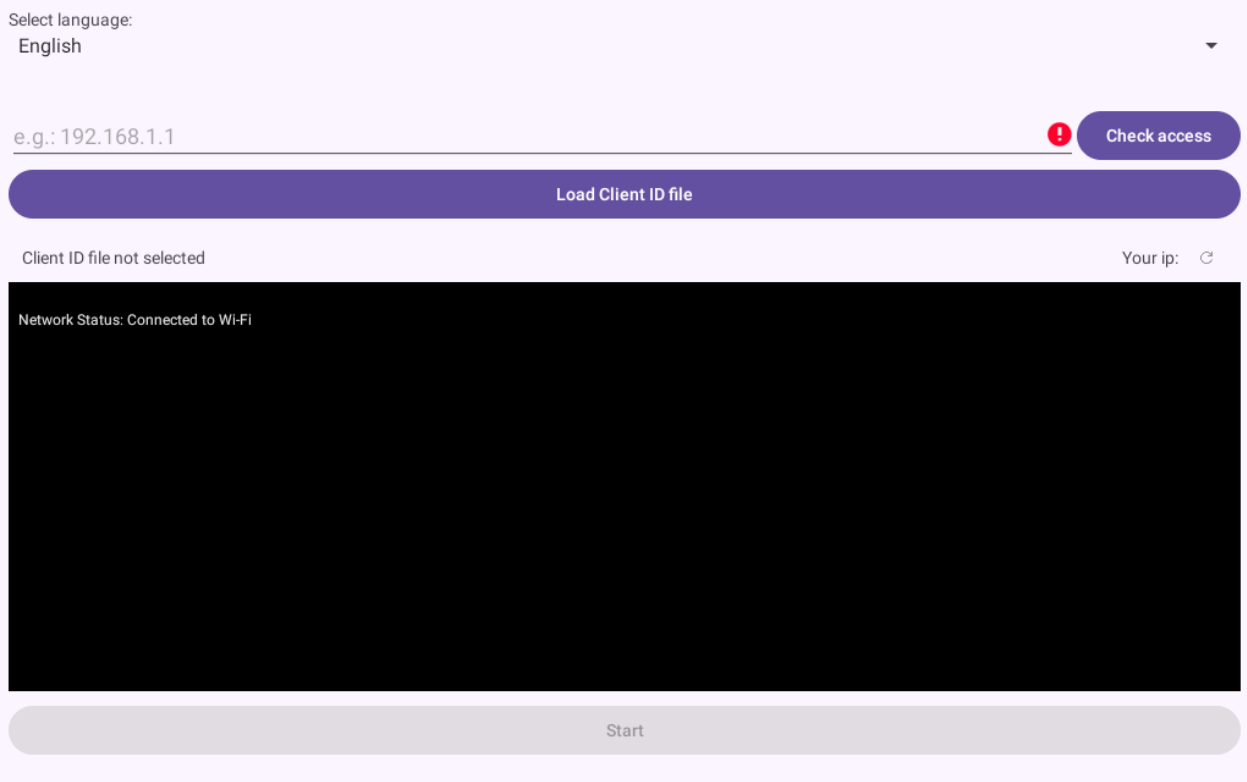


For the most recently published versions by the DDoSia team, an updated Android version is also available. Its functionalities and output are similar, and everything is preceded by the client ID which is used to display the connection to the server and receive the configuration.



Android version client

Available languages for the android tool



Android versions of DDoSia mirror desktop functionality and use the client ID to connect to the server and receive configuration.

Different requests can be executed and communication can be maintained for different functionalities:

1. POST /v1/register
   Initial authentication and agent registration.

2. GET /v1/config or GET /v1/targets
   Retrieve configuration data and assigned targets.

3. GET /v1/plugins/download?name=<plugin>
   Download required plugins if they are not available locally.

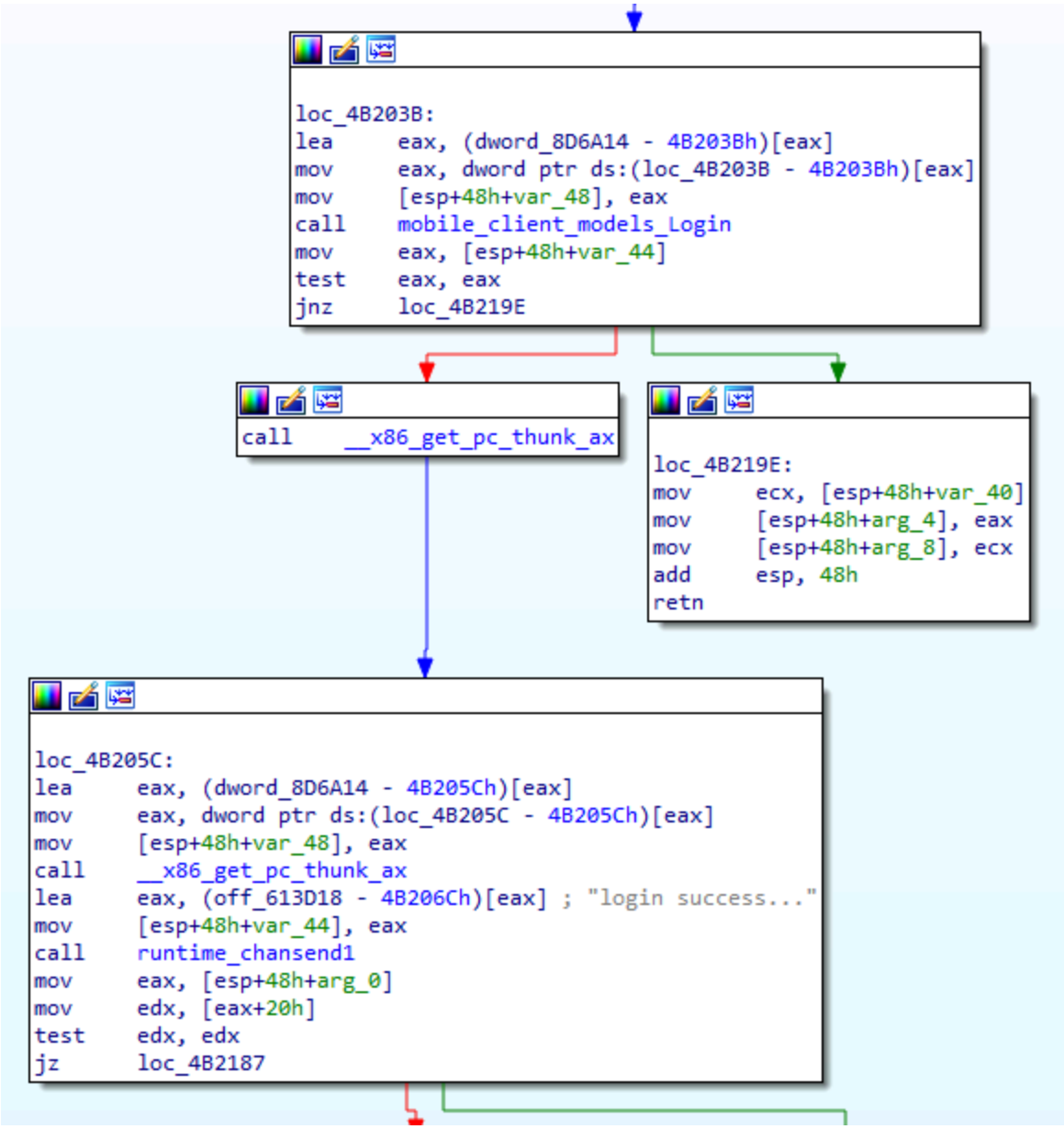
4. Execution start
   Agent starts task execution.

5. POST /v1/heartbeat
   Send periodic keep-alive messages.

6. POST /v1/report
   Submit execution metrics and results.

```
loc_4B1F50:
lea     eax, (dword_8D6A14 - 4B1F50h)[eax]
mov     eax, dword ptr ds:(loc_4B1F50 - 4B1F50h)[eax]
mov     [esp+48h+var_1C], eax
call    __x86_get_pc_thunk_ax
lea     eax, (aGoStresser - 4B1F61h)[eax] ; "Go-Stresser "
mov     [esp+48h+var_48], eax
mov     [esp+48h+var_44], 1Dh
mov     [esp+48h+var_40], 0
mov     [esp+48h+var_3C], 0
mov     [esp+48h+var_38], 0
call    fmt_Sprintf
mov     eax, [esp+48h+var_34]
mov     ecx, [esp+48h+var_30]
mov     [esp+48h+var_8], eax
mov     [esp+48h+var_4], ecx
mov     eax, [esp+48h+var_1C]
mov     [esp+48h+var_48], eax
lea     eax, [esp+48h+var_8]
mov     [esp+48h+var_44], eax
call    runtime_chansend1
mov     eax, [esp+48h+var_1C]
mov     [esp+48h+var_48], eax
call    __x86_get_pc_thunk_ax
lea     eax, (off_613D08 - 4B1FBFh)[eax] ; "© NoName057(16)\n"
mov     [esp+48h+var_44], eax
call    runtime_chansend1
mov     eax, [esp+48h+var_1C]
mov     [esp+48h+var_48], eax
call    __x86_get_pc_thunk_ax
lea     eax, (off_613D10 - 4B1FDAh)[eax]
mov     [esp+48h+var_44], eax
call    runtime_chansend1
call    __x86_get_pc_thunk_ax
```

Binary code displaying the "Go-Stresser" and a references to NoName057(16).



Code flow handling client login and success validation before continuing execution.

Once communication has been established and targets have been received, DDoSia begins operating with its different particular features, such as attack vectors depending on the originating node and target, a wide range of evasion techniques, and the configuration or customization options available in the tool.

Supported Attack Vectors

One of the most important aspects of DDoSia is undoubtedly its ability to execute different attacks depending on the capabilities of the node and the target. This allows it to adapt and remain efficient depending on the campaign and the device launching the offensive.

In the most recent versions of DDoSia, various techniques can be used to launch attacks from the tool.

HTTP

GET/POST flood

These techniques consist of massive requests to HTTP endpoints or repeated POSTs that generate computational cost on the victim's server. They are based on saturating web services with requests to slow them down or cause them to crash due to resource exhaustion.

Specific functions can be found that support this technique, where custom cookies are added to the request and later reused in order to imitate legitimate traffic. The tool uses different headers to avoid detection.

The tool sends a request with cookies and headers, receives cookies in return, reuses them and concatenates the next requests, making the traffic appear to come from a legitimate device, but in this case inside an iterative loop.

```
if ( (unsigned int)&retaddr <= *(_DWORD *)(__readgsdword(dword_874A88) + 8) )
  runtime_morestack_noctxt();
v2 = a1;
v3 = a1[3];
if ( v3 )
{
  v4 = ((int (__golang *)(void **, _DWORD))v3[3])(a1[4], a2[2]);
  v20 = v4;
  v5 = v15;
  v6 = 0;
  while ( v6 < v5 )
  {
    v19 = v6;
    net_http__Request_AddCookie(a2, *(_DWORD *)(v4 + 4 * v6));
    v6 = v19 + 1;
    v4 = v20;
    v5 = v15;
  }
  v2 = a1;
}
v7 = *v2;
if ( *v2 )
{
  v8 = (int)v2[1];
}
else
{
  v7 = off_8CC908;
  v8 = (int)*(&off_8CC908 + 1);
}
v9 = (int)v7;
v12 = v8;
sub_19702E();
net_http_send(a2, v9, v12);
if ( !v18 && a1[3] )
{
  SetCookies = net_http_readSetCookies(*(_DWORD *)(v17 + 28), v10, v13);
  if ( v14 )
    ((void (__golang *)(void **, _DWORD, int, int, int))a1[3][4])(a1[4], a2[2], v11, v14, SetCookies);
}
}
```

HTTP request handling code adding cookies and sending data to the server.

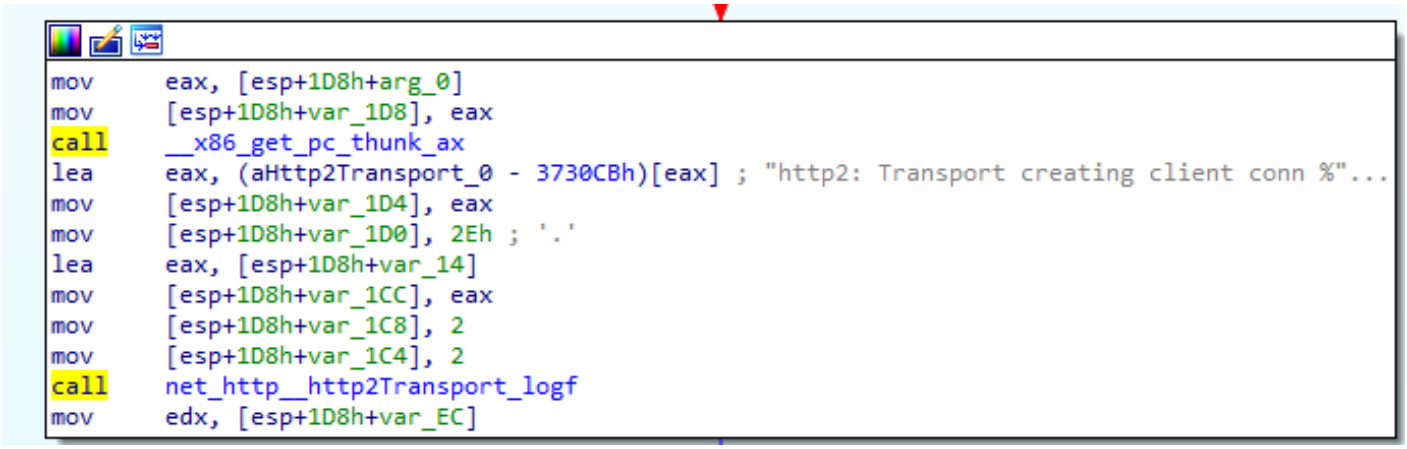
```
else
{
  v52 = runtime_newobject(&unk_601760);
  v7 = (_DWORD *)v52;
  v103 = (_DWORD *)v52;
  if ( dword_8E62F0[0] )
  {
    v64 = runtime_wbMove(&unk_601760, v52, a1);
    v7 = v103;
  }
  sub_196EDA();
  v51 = runtime_makemap_small(v48);
  v8 = (int)v103;
  v9 = v49;
  if ( dword_8E62F0[0] )
  {
    v119 = v49;
    v8 = runtime_gcWriteBarrier2();
    v9 = v119;
    *v7 = v119;
    v7[1] = v10;
  }
  *(_DWORD *)(v8 + 28) = v9;
  v6 = a1;
  v3 = (int)v103;
}
v11 = *(int **)(*(_DWORD *)(v3 + 8) + 16);
if ( v11 )
{
  v118 = v3;
  v93 = v11;
  v64 = net_textproto_MIMEHeader_Get(*(_DWORD *)(v3 + 28), &unk_609D4, 13);
  v3 = v118;
  v6 = a1;
  v12 = v68 == 0;
}
else
{
```

Code managing HTTP headers and internal data structures during network communication.

HTTP/2 framing and stream control

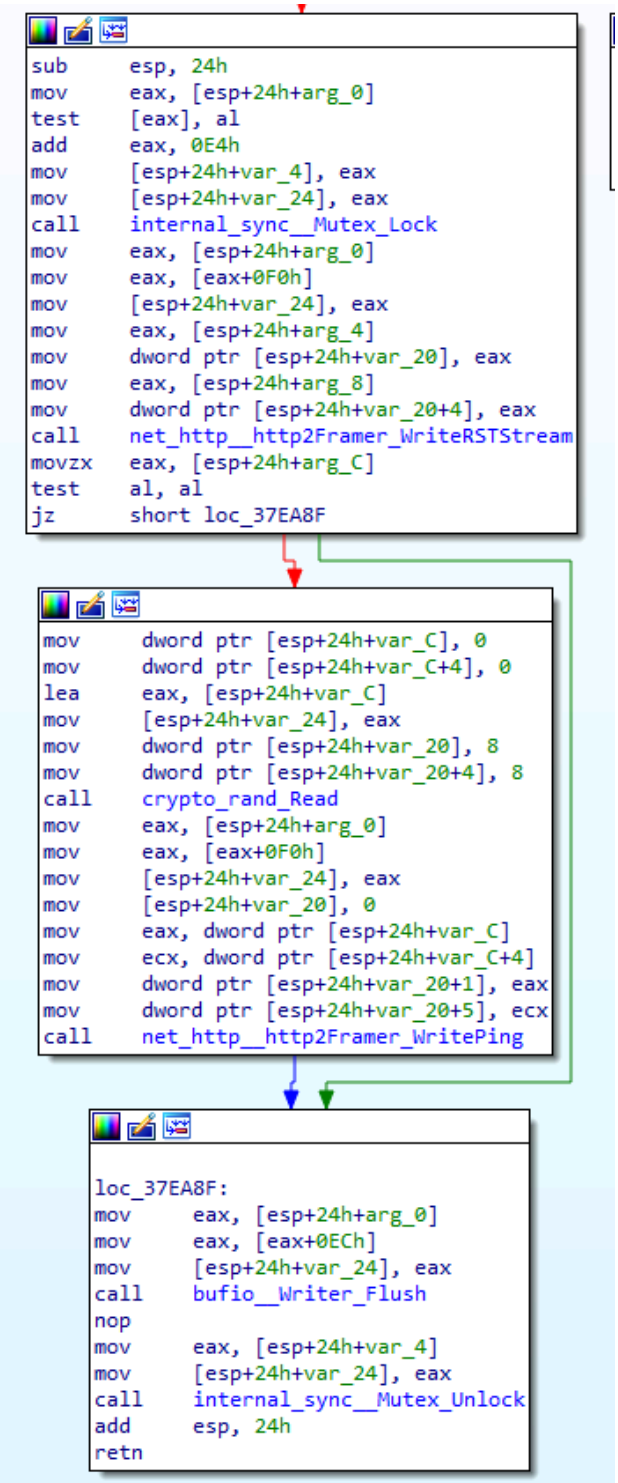
HTTP/2 is a stream multiplexing protocol that allows multiple simultaneous requests over a single TCP/TLS connection. This reduces latency, eliminates handshake overhead and makes detection more difficult, since traffic is normalized within a single encrypted tunnel.

Many functions can be found that form part of request creation, where streams are reset or pings are sent, reproducing typical HTTP/2 behaviours but executed in simultaneous loops.



```
mov     eax, [esp+1D8h+arg_0]
mov     [esp+1D8h+var_1D8], eax
call    __x86_get_pc_thunk_ax
lea     eax, (aHttp2Transport_0 - 3730CBh)[eax] ; "http2: Transport creating client conn %"...
mov     [esp+1D8h+var_1D4], eax
mov     [esp+1D8h+var_1D0], 2Eh ; '.'
lea     eax, [esp+1D8h+var_14]
mov     [esp+1D8h+var_1CC], eax
mov     [esp+1D8h+var_1C8], 2
mov     [esp+1D8h+var_1C4], 2
call    net_http__http2Transport_logf
mov     edx, [esp+1D8h+var_EC]
```

Code logging HTTP/2 transport creation for client network connections.



HTTP/2 request framing and ping handling within the network communication routine.

Within the HTTP techniques, there are also other functions covering widely used attack methods in DDoSia:

- **HTTP HEAD flood:** a lightweight variant to increase requests per second,
- **Slowloris:** HTTP connections that send headers slowly to keep threads busy indefinitely without consuming bandwidth,
- **Cache Busting:** based on adding random parameters to URLs to bypass CDNs so that requests reach the origin server, which in this case is the victim.

TCP

TCP Flood

TCP flood attacks consist of sending large numbers of TCP connections, usually incomplete ones, to a victim server in order to overload its ability to accept, process or close them. These techniques force the server to consume resources in handshakes, state maintenance or pending connection queues, causing latency or outages. They usually involve opening many sockets, parallel connection attempts and bursts of packets.

There are also various functions where TCP sockets are repeatedly opened and attempts are made to connect to the host. A loop is used to invoke dialers or standard connection routines, generating multiple attempts in a very short time. Sometimes a small initial payload may be sent, or the connection may simply remain incomplete, partially imitating a legitimate client. Repeating this process thousands of times per second causes saturation of the target service.

```
{
  void *retaddr; // [esp+0h] [ebp+0h] BYREF

  if ( (unsigned int)&retaddr <= *(_DWORD *)(__readgsdword(dword_874A88) + 8) )
    runtime_morestack_noctxt();
  return net__conn_Write(a1, a2, a3, a4);
}
```

```
{
  int (__golang **v5)(int, int, _DWORD, _DWORD, int, int); // edx
  void *retaddr; // [esp+24h] [ebp+0h] BYREF

  if ( (unsigned int)&retaddr <= *(_DWORD *)(__readgsdword(dword_874A88) + 8) )
    runtime_morestack_noctxt();
  v5 = (int (__golang **)(int, int, _DWORD, _DWORD, int, int))a1[29];
  if ( v5 )
    return (*v5)(a2, a3, a1[25], a1[26], a4, a5);
  if ( dword_8D6AC4 )
    return ((int (__golang *)(int, int, _DWORD, _DWORD, int, int))*dword_8D6AC4)(a2, a3, a1[25], a1[26], a4, a5);
  return net__sysDialer_doDialTCPProto(a1, a2, a3, a4, a5, 0);
}
```

Code handling network writes and protocol selection based on runtime conditions.

```
if ( (unsigned int)&retaddr <= *(_DWORD *)(__readgsdword(dword_874A88) + 8) )
  runtime_morestack_noctxt();
if ( !a1 || !*a1 )
  return (void **)&unk_FF4E0;
net__netFD_Write(*a1, a2, a3, a4);
result = v14;
if ( v14 )
{
  v6 = (_DWORD *)runtime_newobject(&unk_5DBA80);
  v6[1] = 5;
  *v6 = "write";
  v7 = a1;
  v8 = *(_DWORD *)(*a1 + 52);
  v6[3] = *(_DWORD *)(*a1 + 56);
  if ( dword_8E62F0[0] )
  {
    v6 = (_DWORD *)runtime_gcWriteBarrier1();
    *v4 = v8;
  }
  v6[2] = v8;
  v9 = *(_DWORD *)(*v7 + 64);
  v6[4] = *(_DWORD *)(*v7 + 60);
  if ( dword_8E62F0[0] )
  {
    v6 = (_DWORD *)runtime_gcWriteBarrier1();
    *v4 = v9;
  }
  v6[5] = v9;
  v10 = *v7;
  v11 = *(_DWORD *)(v10 + 68);
  v12 = *(_DWORD *)(v10 + 72);
  v6[6] = v11;
  if ( dword_8E62F0[0] )
  {
    v6 = (_DWORD *)runtime_gcWriteBarrier2();
    *v4 = v12;
    v13 = v15;
    v4[1] = v15;
  }
  else
  {
    v13 = v15;
  }
  v6[7] = v12;
  v6[8] = v14;
  v6[9] = v13;
  return &off_615960;
}
return result;
```

Function handling network write results and updating internal state based on response status.

Just as with HTTP, TCP includes several functionalities used as attack vectors:

- **TCP SYN Flood:** mass sending of SYN packets without completing the handshake to saturate server queues,
- **Connection Exhaustion:** opening hundreds or thousands of fully established TCP connections simultaneously to exhaust file descriptor or connection limits on the server.

UDP

UDP flood

UDP flood attacks are based on massive UDP packet sends to the victim server, generating high bandwidth consumption and load, similar to HTTP-based volumetric attacks. Since UDP does not require formal connection setup, extremely fast stateless traffic can be sent, saturating network capacity, system buffers or applications processing the protocol.

Multiple functions can be found for UDP-based techniques, where continuous sending loops construct packets repeatedly. Sockets are prepared and the remote address is resolved for both IPv6 and IPv4 to send datagrams in constant succession. The volumetric cost impacts the victim's service availability.

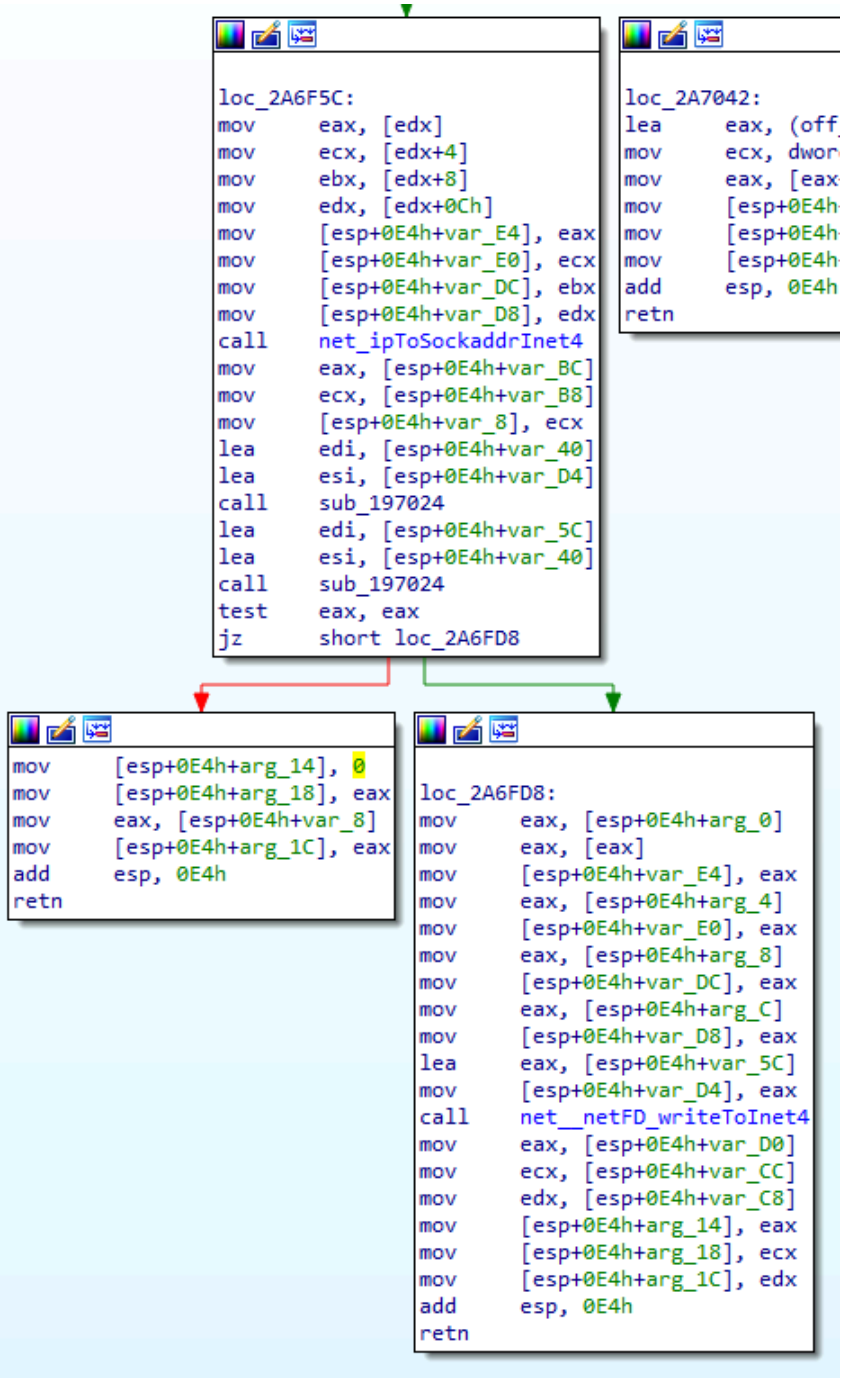
```
{
  void *retaddr; // [esp+0h] [ebp+0h] BYREF

  if ( (unsigned int)&retaddr <= *(_DWORD *)(__readgsdword(dword_874A88) + 8) )
    runtime_morestack_noctxt();
  return net__conn_Write(a1, a2, a3, a4);
}
```

```
{
  _DWORD *result; // eax
  void *retaddr; // [esp+2Ch] [ebp+0h] BYREF

  if ( (unsigned int)&retaddr <= *(_DWORD *)(__readgsdword(dword_874A88) + 8) )
    runtime_morestack_noctxt();
  result = a1;
  if ( a1 )
    return (_DWORD *)net_ipToSockaddr(a2, *a1, a1[1], a1[2], a1[3], a1[4], a1[5]);
  return result;
}
```

Code performing low-level network writes and IP-to-socket address handling.



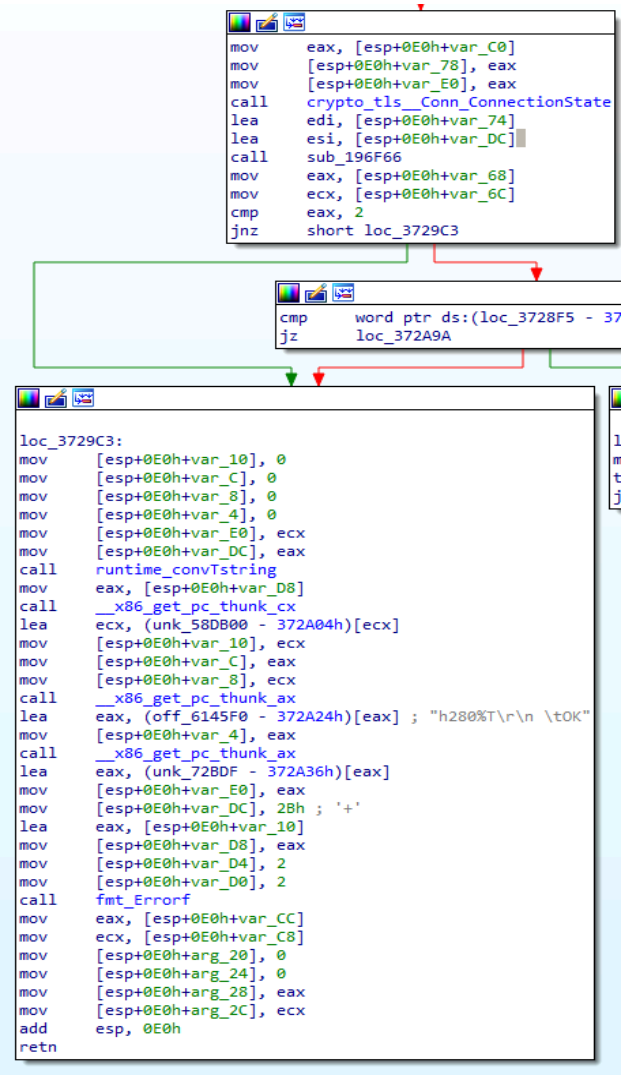
Control flow showing IP address conversion and packet transmission over the network.

TLS and SSL

TLS handshake

TLS provides an encrypted channel in which outgoing requests are encapsulated toward the target. The binary implements the handshake sequence, TLS configuration, encrypted channel setup and connection establishment. During handshake flood attacks, TLS connections are established iteratively, performing the full handshake but without sending application data or forcing immediate closure. This consumes CPU on the server due to the cryptographic operations required (signing, verification and more), without generating meaningful traffic.

TLS connection handling and response processing within the network communication flow.



Other techniques appear in this section, similar to previous ones:

- **SSL and TLS Renegotiation Attack:** forcing constant renegotiations to consume server CPU.

Evasion Techniques

Like other tools of this type, DDoSia uses several evasion techniques to avoid detection or hinder specialized software. Even though it is not extremely advanced in these areas, various techniques can be identified that complement and strengthen the broad set of attack vectors it offers

Obfuscation

In various DDoSia samples it is difficult to locate useful strings that could help with detection. Even though its methods are not as sophisticated as those of other tools, they complicate analysis and only become visible during runtime or when debugging after processing.

```
bL78aeuqRQm.(*mKTHtY4[go.shape.interface { CancelRead(C65_gMyieC.B0E8jstg); CancelWrite(C65_gMyieC.B0E8jstg)
bL78aeuqRQm.(*mKTHtY4[go.shape.interface { CancelRead(C65_gMyieC.B0E8jstg); CancelWrite(C65_gMyieC.B0E8jstg)
bL78aeuqRQm.(*mKTHtY4[go.shape.interface { CancelRead(C65_gMyieC.B0E8jstg); CancelWrite(C65_gMyieC.B0E8jstg)
bL78aeuqRQm.(*mKTHtY4[go.shape.interface { CancelRead(C65_gMyieC.B0E8jstg); CancelWrite(C65_gMyieC.B0E8jstg)
bL78aeuqRQm.(*mKTHtY4[go.shape.interface { CancelRead(C65_gMyieC.B0E8jstg); CancelWrite(C65_gMyieC.B0E8jstg)
bL78aeuqRQm.(*mKTHtY4[go.shape.interface { CancelRead(C65_gMyieC.B0E8jstg); CancelWrite(C65_gMyieC.B0E8jstg)
bL78aeuqRQm.(*mKTHtY4[go.shape.interface { CancelWrite(C65_gMyieC.B0E8jstg); Close() error; Context() lvCf7unvot
bL78aeuqRQm.(*mKTHtY4[go.shape.interface { CancelWrite(C65_gMyieC.B0E8jstg); Close() error; Context() lvCf7unvot
bL78aeuqRQm.(*mKTHtY4[go.shape.interface { CancelWrite(C65_gMyieC.B0E8jstg); Close() error; Context() lvCf7unvot
bL78aeuqRQm.(*mKTHtY4[go.shape.interface { CancelWrite(C65_gMyieC.B0E8jstg); Close() error; Context() lvCf7unvot
bL78aeuqRQm.(*mKTHtY4[go.shape.interface { CancelWrite(C65_gMyieC.B0E8jstg); Close() error; Context() lvCf7unvot
bL78aeuqRQm.(*mKTHtY4[go.shape.interface { CancelWrite(C65_gMyieC.B0E8jstg); Close() error; Context() lvCf7unvot
bL78aeuqRQm.(*bUZxNWGUyoj[go.shape.interface { CancelRead(C65_gMyieC.B0E8jstg); CancelWrite(C65_gMyieC.B0E
bL78aeuqRQm.(*bUZxNWGUyoj[go.shape.interface { CancelRead(C65_gMyieC.B0E8jstg); CancelWrite(C65_gMyieC.B0E
bL78aeuqRQm.(*bUZxNWGUyoj[go.shape.interface { CancelRead(C65_gMyieC.B0E8jstg); CancelWrite(C65_gMyieC.B0E
bL78aeuqRQm.(*bUZxNWGUyoj[go.shape.interface { CancelRead(C65_gMyieC.B0E8jstg); CancelWrite(C65_gMyieC.B0E
bL78aeuqRQm.(*bUZxNWGUyoj[go.shape.interface { CancelRead(C65_gMyieC.B0E8jstg); Read([]uint8) (int, error); SetR
bL78aeuqRQm.(*bUZxNWGUyoj[go.shape.interface { CancelRead(C65_gMyieC.B0E8jstg); Read([]uint8) (int, error); SetR
bL78aeuqRQm.(*bUZxNWGUyoj[go.shape.interface { CancelRead(C65_gMyieC.B0E8jstg); Read([]uint8) (int, error); SetR
bL78aeuqRQm.(*bUZxNWGUyoj[go.shape.interface { CancelRead(C65_gMyieC.B0E8jstg); Read([]uint8) (int, error); SetR
```

Obfuscated runtime strings that limit static detection and only appear during execution or debugging.

```
&"http://77.91.100.134"
&"http://77.91.100.134"
&"http://77.91.100.134"
&"/client/login"
&"/client/set_attack_count/client/get_targets"
&"/client/get_targets"
&"ownEthernetType"
&"/client/login"
&"/client/login"
&"/client/set_attack_count/client/get_targets"
&"/client/set_attack_count/client/get_targets"
&"/client/get_targets"
&"/client/get_targets"
&"http://77.91.100.134"
&"http://77.91.100.134"
```

Repeated hardcoded C2 URLs and client API paths embedded in the binary.

Rotation of user agents

Another technique seen in earlier versions and later improved is user agent rotation. Instead of using a static and easily recognizable one, the tool uses a large number of them and rotates between them to avoid detection, simulating a real device.

```
&"Mozilla/5.0 (Linux; Android 11; SM-A115M Build/RP1A.200720.012; wv) AppleWebKit/537.36 (
&"Mozilla/5.0 (Linux; Android 11; SM-A115M Build/RP1A.200720.012; wv) AppleWebKit/537.36 (
&"Mozilla/5.0 (iPhone; CPU iPhone OS 16_1_1 like Mac OS X) AppleWebKit/605.1.15 (KHTML, li
&"Mozilla/5.0 (iPhone; CPU iPhone OS 16_1_1 like Mac OS X) AppleWebKit/605.1.15 (KHTML, li
&"Mozilla/5.0 (Linux; Android 13; SAMSUNG SM-T220) AppleWebKit/537.36 (KHTML, like Gecko)
&"Mozilla/5.0 (Linux; Android 13; SAMSUNG SM-T220) AppleWebKit/537.36 (KHTML, like Gecko)
&"Mozilla/5.0 (Linux; Android 9) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome
&"Mozilla/5.0 (Linux; Android 9) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome
&"AppleCoreMedia/1.0.0.23A344 (Macintosh; U; Intel Mac OS X 14_0; da_dk)"
&"AppleCoreMedia/1.0.0.23A344 (Macintosh; U; Intel Mac OS X 14_0; da_dk)"
&"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/
&"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/
&"Mozilla/5.0 (iPhone; CPU iPhone OS 15_6_1 like Mac OS X) AppleWebKit/605.1.15 (KHTML, li
&"Mozilla/5.0 (iPhone; CPU iPhone OS 15_6_1 like Mac OS X) AppleWebKit/605.1.15 (KHTML, li
&"Mozilla/5.0 (Linux; Android 6.0.1; SM-G532MT Build/MMB29T; wv) AppleWebKit/537.36 (KHTML
&"Mozilla/5.0 (Linux; Android 6.0.1; SM-G532MT Build/MMB29T; wv) AppleWebKit/537.36 (KHTML
&"Dalvik/2.1.0 (Linux; U; Android 11; Tibuta_MasterPad-E100 Build/RP1A.201005.006)Mozilla/
&"Dalvik/2.1.0 (Linux; U; Android 11; Tibuta_MasterPad-E100 Build/RP1A.201005.006)Mozilla/
&"Mozilla/5.0 (Linux; Android 13; SM-F711U) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/
&"Mozilla/5.0 (Linux; Android 13; SM-F711U) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/
&"Mozilla/5.0 (X11; U; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Ubuntu Chromiu
&"Mozilla/5.0 (X11; U; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Ubuntu Chromiu
&"Mozilla/5.0 (X11; Linux x86_64; SMARTEMB Build/3.12.9076) AppleWebKit/537.36 (KHTML, lik
&"Mozilla/5.0 (X11; Linux x86_64; SMARTEMB Build/3.12.9076) AppleWebKit/537.36 (KHTML, lik
&"Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Ubuntu Chromium/1
&"Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Ubuntu Chromium/1
&"Mozilla/5.0 (Macintosh; U; PPC; en-US; rv:0.9.3) Gecko/20010802"
&"Mozilla/5.0 (Macintosh; U; PPC; en-US; rv:0.9.3) Gecko/20010802"
&"Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.2.1) Gecko/20021208 Debian/1.2.1-2"
&"Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.2.1) Gecko/20021208 Debian/1.2.1-2"
&"Mozilla/5.0 (Macintosh; U; PPC Mac OS X Mach-O; en-US; rv:1.7.6) Gecko/20050319"
&"Mozilla/5.0 (Macintosh; U; PPC Mac OS X Mach-O; en-US; rv:1.7.6) Gecko/20050319"
&"Mozilla/5.0 (X11; U; Linux i586; en-US; rv:1.0.0) Gecko/20020623 Debian/1.0.0-0.woody.1"
&"Mozilla/5.0 (X11; U; Linux i586; en-US; rv:1.0.0) Gecko/20020623 Debian/1.0.0-0.woody.1"
```

Large set of hardcoded User-Agent strings used for rotation to evade detection.

Internal proxy rotation

It is common for DDoSia to provide **SOCKS** or **HTTP proxy lists** to affiliates for request distribution. These may change between campaigns or nodes to avoid traceability. This increases anonymity for affiliates and complicates both analysis and mitigation.

Encrypted C2 channel

Communication with the **C2** is always **encrypted**, with functions that send information using **AES, RC4 or GCM**. This is widely used to transmit or receive JSON structures containing information required by the affiliate, such as the ID, host and other details.

```
  if ( (unsigned int)&retaddr <= *(_DWORD *)(__readgsdword(dword_874A88) + 8) )
LABEL_14:
    runtime_morestack_noctxt();
  if ( a5 != 12 )
  {
LABEL_13:
    runtime_gopanic(&unk_58DB00);
    goto LABEL_14;
  }
  v10 = crypto_aes_NewCipher(a1, a2, a3);
  v6 = v12;
  if ( v12 )
  {
LABEL_12:
    runtime_gopanic(*(_DWORD *)(v6 + 4));
    goto LABEL_13;
  }
  crypto_cipher_NewGCM(v10);
  if ( v11 )
  {
    runtime_gopanic(*(_DWORD *)(v11 + 4));
    goto LABEL_12;
  }
  result = runtime_newobject(&unk_5C0B40);
  *(_DWORD *)(result + 12) = v9;
  if ( dword_8E62F0[0] )
  {
    result = runtime_gcWriteBarrier1();
    v8 = v10;
    *v5 = v10;
  }
  else
  {
    v8 = v10;
  }
  *(_DWORD *)(result + 16) = v8;
  if ( a4 != result )
  {
    v13 = result;
    runtime_memmove(result, a4, 12);
    return v13;
  }
  return result;
}
```

AES-GCM cipher initialization and key handling used for encrypted communication.

Delays and temporal jitter

Attack routines do not execute at a constant interval. Instead of linear cadences, they vary execution frequency to evade detection tools that rely on identifying timing patterns. This makes the traffic and requests appear more human-like and less automated.

Samples contain many sleep functions and random calculations (using crypto/rand or math/rand) that are applied before executing attack routines.

00000012	C	math/randtlsrsakex
0000000A	C	\tmath/rand
0000000C	C	math/rand/v2
00000017	C	math/rand.(*Rand).Seed
00000019	C	math/rand.(*Rand).Int31n
00000018	C	math/rand.(*Rand).Int31
00000018	C	math/rand.(*Rand).Int63
00000013	C	math/rand.(*Rand).I
00000017	C	math/rand.(*Rand).Read
0000000F	C	math/rand.read
0000001D	C	math/rand.(*rngSource).Int63
0000001E	C	math/rand.(*rngSource).Uint64
00000015	C	math/rand.globalRand

References to Go math/rand functions used for random value generation.

Cookie or session handling

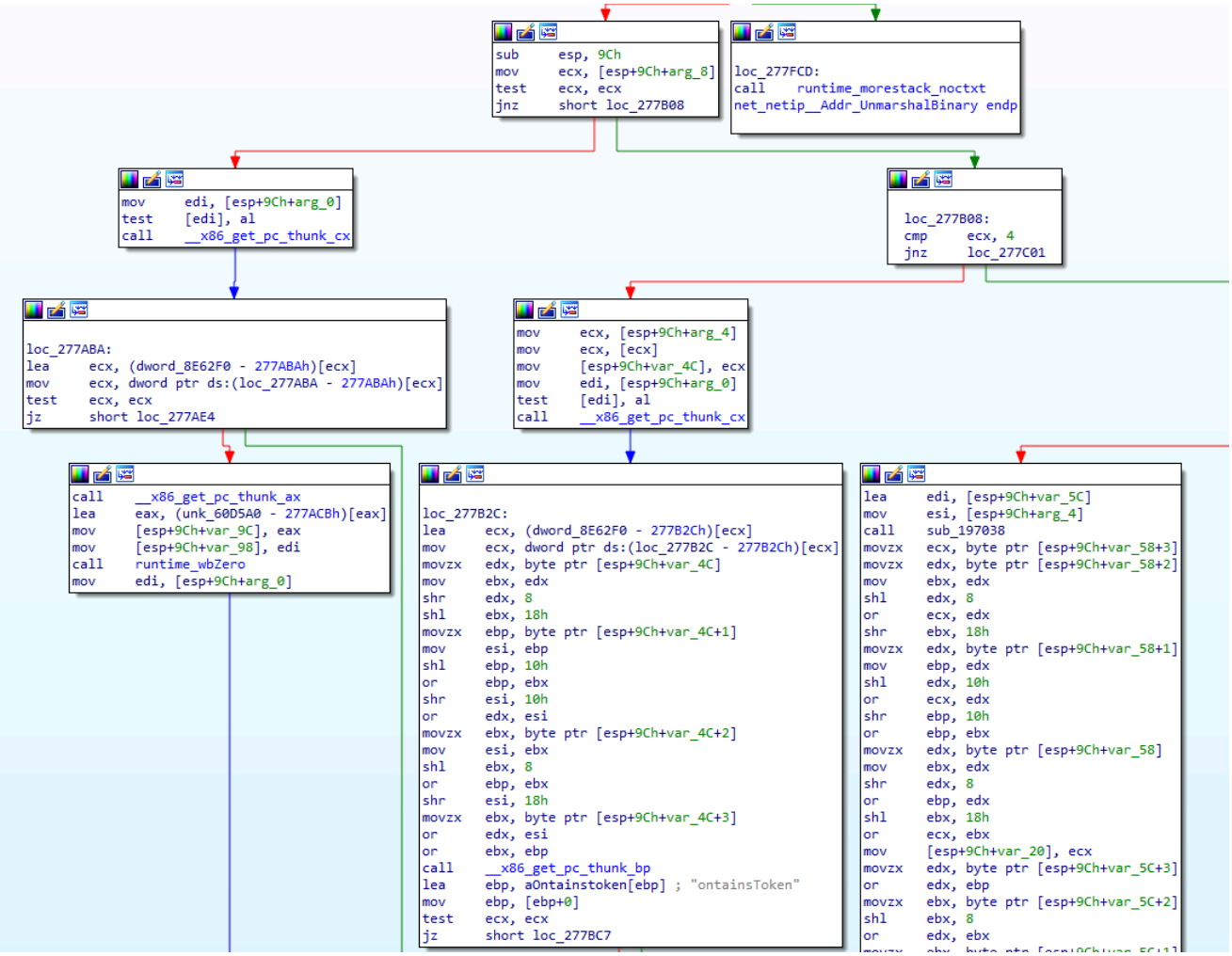
An important evasion factor is the ability to capture and reuse cookies to maintain sessions on the target server. During execution, cookies are extracted from HTTP responses, often used by anti-bot systems, and dynamically included in subsequent requests. This again makes the traffic resemble a real browser, since it reacts to the server's responses.

This allows bypassing protections that require basic human-like interactions and even solving simple JavaScript challenges. It cannot yet bypass complex visual CAPTCHAs or protections requiring genuine human action. Still, this capability allows chaining context between requests and bypassing basic anti-bot mechanisms like IUAM.

Dynamic C2 configuration

Another aspect marking a significant evolution in recent versions is the dynamic nature of its attacks. The victim is not embedded in the binary, but instead the sample can receive orders or configurations from the C2, as explained earlier.

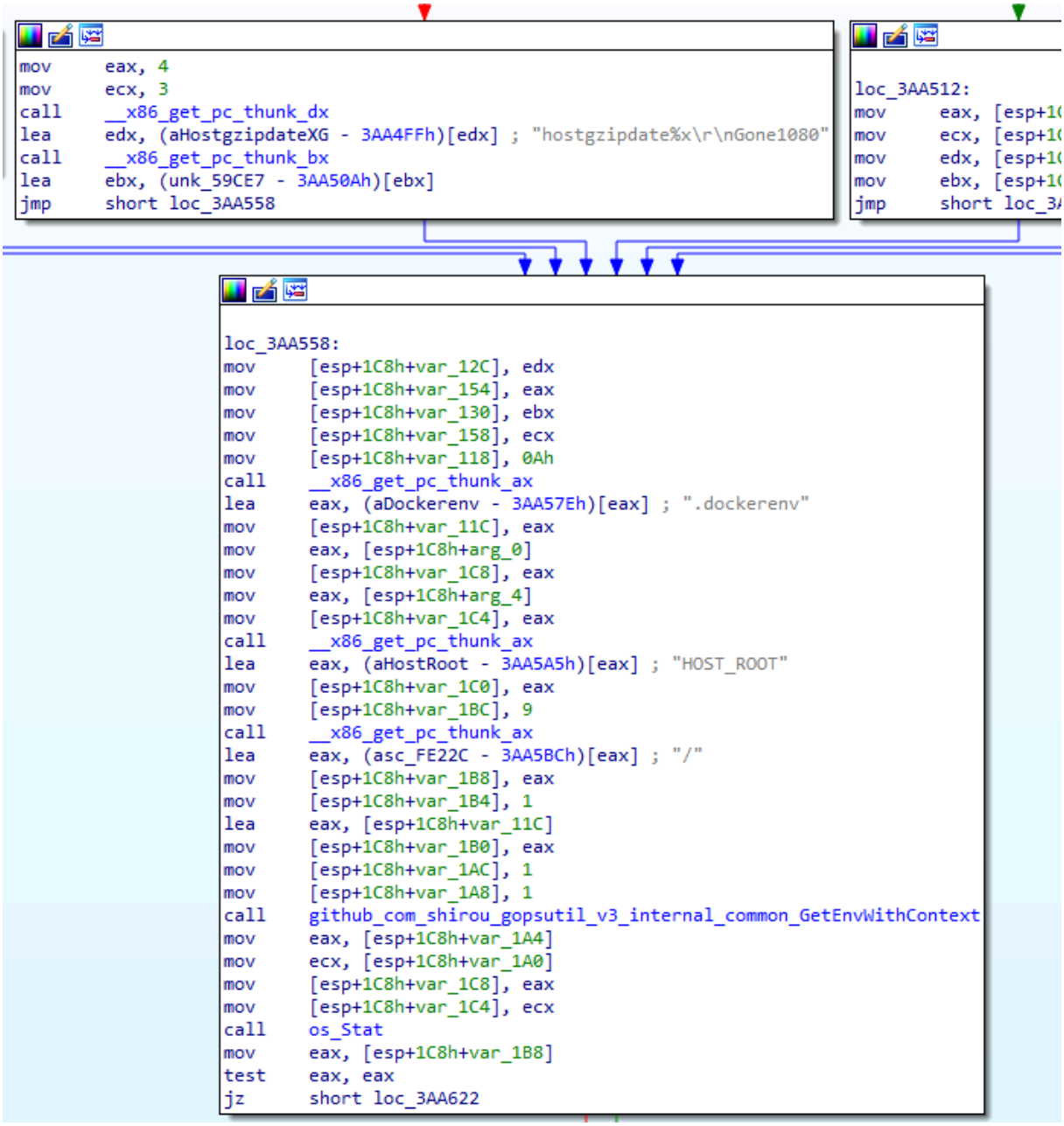
This allows DDoSia to update attack methods, target addresses, proxies or even entire target lists whenever the operator requires, avoiding the need to rotate binaries and providing longer lifespan and lower development effort.



Control flow for token parsing and binary address unmarshalling logic.

Anti-VM and anti-debug techniques

DDoSia, like other tools of this type, includes anti-analysis techniques. Beyond obfuscation or code packing, functions can be found that detect whether the process is being executed under a debugger, force breakpoints or attempt to identify virtualization software commonly used in analysis environments, such as vbox, KVM or QEMU. It is common to check registry keys linked to these tools or look for processes such as vboxservice or vmtoolsd.



```
mov     eax, 4
mov     ecx, 3
call    __x86_get_pc_thunk_dx
lea     edx, (aHostgzipdateXG - 3AA4FFh)[edx] ; "hostgzipdate%x\r\nGone1080"
call    __x86_get_pc_thunk_bx
lea     ebx, (unk_59CE7 - 3AA50Ah)[ebx]
jmp     short loc_3AA558
```

```
loc_3AA512:
mov     eax, [esp+1C
mov     ecx, [esp+1C
mov     edx, [esp+1C
mov     ebx, [esp+1C
jmp     short loc_3A
```

```
loc_3AA558:
mov     [esp+1C8h+var_12C], edx
mov     [esp+1C8h+var_154], eax
mov     [esp+1C8h+var_130], ebx
mov     [esp+1C8h+var_158], ecx
mov     [esp+1C8h+var_118], 0Ah
call    __x86_get_pc_thunk_ax
lea     eax, (aDockerenv - 3AA57Eh)[eax] ; ".dockerenv"
mov     [esp+1C8h+var_11C], eax
mov     eax, [esp+1C8h+arg_0]
mov     [esp+1C8h+var_1C8], eax
mov     eax, [esp+1C8h+arg_4]
mov     [esp+1C8h+var_1C4], eax
call    __x86_get_pc_thunk_ax
lea     eax, (aHostRoot - 3AA5A5h)[eax] ; "HOST_ROOT"
mov     [esp+1C8h+var_1C0], eax
mov     [esp+1C8h+var_1BC], 9
call    __x86_get_pc_thunk_ax
lea     eax, (asc_FE22C - 3AA5BCh)[eax] ; "/"
mov     [esp+1C8h+var_1B8], eax
mov     [esp+1C8h+var_1B4], 1
lea     eax, [esp+1C8h+var_11C]
mov     [esp+1C8h+var_1B0], eax
mov     [esp+1C8h+var_1AC], 1
mov     [esp+1C8h+var_1A8], 1
call    github_com_shirou_gopsutil_v3_internal_common_GetEnvWithContext
mov     eax, [esp+1C8h+var_1A4]
mov     ecx, [esp+1C8h+var_1A0]
mov     [esp+1C8h+var_1C8], eax
mov     [esp+1C8h+var_1C4], ecx
call    os_Stat
mov     eax, [esp+1C8h+var_1B8]
test    eax, eax
jz      short loc_3AA622
```

Code checking for containerized environments by accessing Docker and host system paths.

```
else
{
  v139 = 0;
  v125 = v84;
  v130 = v106;
  github_com_shirou_gopsutil_v3_internal_common_StringsContains(0, v84, v106, "kvm", 3, v84);
  if ( v85 )
  {
    v7 = 4;
    v8 = 3;
    v9 = "hostgzipdate%x\r\nGone1080";
    v10 = "kvm";
  }
  else
```

String checks for KVM indicators to detect virtualized environments.

```
else
{
  v138 = 0;
  v124 = v90;
  v129 = v107;
  github_com_shirou_gopsutil_v3_internal_common_StringsContains(0, v90, v107, "QEMU Virtual CPU", 16, v90);
  if ( v91 )
  {
    v15 = 1;
  }
  else
  {
    github_com_shirou_gopsutil_v3_internal_common_StringsContains(v138, v124, v129, &unk_651AA, 20, 0);
    v15 = v91;
  }
  if ( v15
    || (github_com_shirou_gopsutil_v3_internal_common_StringsContains(v138, v124, v129, &unk_69540, 27, v91), v92) )
  {
```

Detection logic targeting QEMU virtual CPU artifacts.

```
else
{
  internal_stringslite_Index(v31, v30, &unk_60C6B, 13, v73);
  if ( v73 < 0 )
  {
    v25 = v132;
    v26 = v131;
    v27 = v142;
    v28 = v141;
  }
  else
  {
    v25 = 5;
    v26 = 3;
    v27 = (const char *)&unk_5AF6E;
    v28 = "lxc";
  }
 }
}
v142 = v27;
```

Additional string matching to identify sandbox or VM execution.

```
else
{
  github_com_shirou_gopsutil_v3_internal_common_StringsContains(v139, v125, v130, &MEMORY[0x5D04A], 7, 0);
  if ( v87 )
  {
    v7 = 4;
    v8 = 4;
    v9 = "hostgzipdate%x\r\nGone1080";
    v10 = "vbox,#=:cap  -> failsha";
  }
  else
  {
    github_com_shirou_gopsutil_v3_internal_common_StringsContains(v139, v125, v130, "vboxguest", 9, 0);
    if ( v88 )
    {
      v7 = 5;
      v8 = 4;
      v9 = (const char *)&unk_5AF6E;
      v10 = "vbox,#=:cap  -> failsha";
    }
    else
    {
      github_com_shirou_gopsutil_v3_internal_common_StringsContains(v139, v125, v130, &MEMORY[0x5BF5D], 6, 0);
      if ( v89 )
      {
        v7 = 5;
        v8 = 6;
        v9 = (const char *)&unk_5AF6E;
        v10 = (const char *)&MEMORY[0x5BF5D];
      }
      else
      {
```

String-based checks for VirtualBox and virtualization artifacts to detect VM environments.

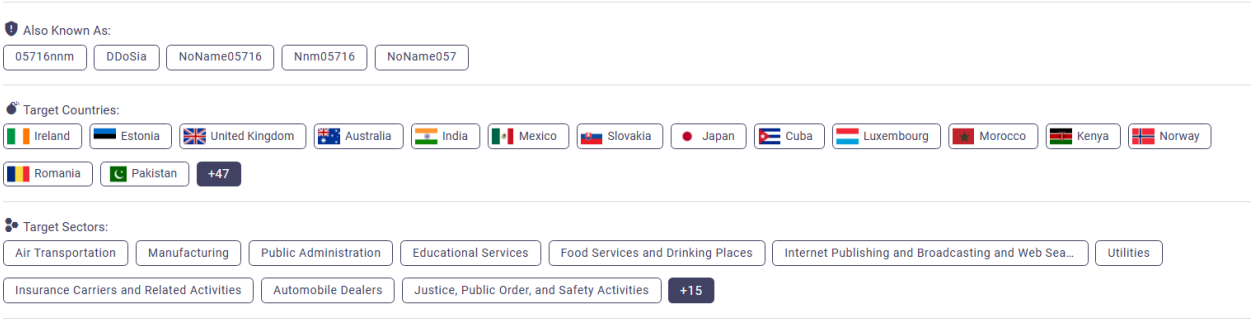
Configuration, Customization and Adaptability of the Tool

DDoSia in its latest versions allows adaptability to both the victim being attacked and the origin device. This gives the affiliate some degree of control, even though instructions are ultimately received from the C2. Certain settings can still be modified by the user executing the tool.

- **Limit of threads:** Samples typically use multiple threads to perform simultaneous tasks more efficiently. In this case, the number of threads for attacks can be adjusted.
- **Speed or intensity**: In parallel to threads, attack speed can be modified either by the C2 automatically or by the user in some samples, adjusting delays or workers to control intensity or workload on the node and therefore the rate of outgoing requests.
- **Proxy rotation:** As discussed earlier, the C2 provides proxies, but the affiliate can enable or disable their use. Although presented as optional, proxy rotation is recommended since it prevents blocking and tracking.
- **Failover:** The tool can continue receiving configurations or instructions even if the main C2 fails. Samples usually contain multiple hardcoded addresses that rotate if the primary one is not reachable, prioritizing functionality and maintaining active attacks.
- **Debug mode:** Some versions allow viewing in real time the requests being executed simultaneously, as well as error codes presented as logs, providing traceability for the affiliate using the tool.

Victimology

It is important to understand the terms in which NoName057(16) operates in order to choose its victims and understand its attack trends. This allows us to predict or understand what its motivations have been in justifying its actions, adopting a broader understanding of its operations.



SOCRadar Threat Actor Intelligence

Target Profile & Geopolitical Coordination

As shown earlier, NoName057(16) is a group that selects its targets carefully, following strategic or symbolic political logic, using technical opportunism and the media impact produced by its operations. They try to coordinate or blend their attacks with geopolitical events that legitimize them.

Since its creation, the group has relied on several criteria to choose its next victims.

Symbolism and geopolitical narrative

As mentioned previously, narrative is one of the main elements for NoName057(16). Any geopolitical event the group perceives as an aggression or hostile act against Russia can serve as a reason to select a new target.

Events around them have not only contributed to the evolution of DDoSia but also enabled them to respond quickly to geopolitical situations, showing an immediate reaction to any event that may affect their perspective. Some clear examples include:

- **NATO accessions and processes:** The entry of Finland on April 4, 2023 triggered a campaign against Finnish institutions that lasted three weeks. This also occurred with Sweden during its accession process between 2023 and 2024, with notable increases in attack volume in the following weeks, showing immediate reaction to what the group views as a provocation and a violation of treaties from a pro-Russian perspective.

- **Aid to Ukraine:** Events such as weapon shipments or financial support have repeatedly triggered DDoSia campaigns within 24 to 72 hours against the country providing the aid. Nations like Germany or Poland have suffered continuous attacks by the group after offering assistance to Ukraine.

- **Economic sanctions:** When the EU or its member states announce sanctions, the group responds with DDoS campaigns. Targets vary, but these have included ministries of economy, financial regulatory institutions and trade portals of the country applying or supporting the sanctions.

- **Events excluding Russia:** Since the start of the conflict, Russia has frequently been excluded from summits, conferences or international events, which has become another reason for the group to attack the countries responsible. This happened during the peace summit in Switzerland, various sports events and general assemblies.

Technical ease or notable vulnerability

A basic criterion for any emerging group is to choose easier targets. Therefore, continuous attacks often focus on poorly defended systems to maximize the likelihood of success, since this information helps reinforce the group's narrative.

Some targets historically exploited due to their technical weakness include:

- **Institutions with limited DDoS protections:** Public entities related to local governments or municipalities and others with similar profiles are common victims. In 2024, NoName057(16) attacked dozens of UK local councils because they lacked effective protection.

- **Sites without CDN or with misconfigurations:** The group often performs reconnaissance, accepting suggestions sent through their Telegram channels mentioned earlier, to identify sites not using Cloudflare, Akamai or similar services, as well as testing for basic DDoS protections or poor CDN configurations, allowing them to select the most vulnerable targets.

- **Symbolism and visibility:** Symbolic targets that are easy to disrupt are ideal for the group. They have repeatedly attacked regional news portals, public information systems or digital citizen services, which are easy to interrupt and generate significant public attention.

Media impact and psychological effect

Related to the previous point, one of the key elements for the group is media visibility and the psychological impact their attacks may cause. This indirectly amplifies their operations and increases the perceived success and reach of the group.

NoName057(16) operates under a principle of “propaganda by the deed”, ensuring their operations generate:

- **Media coverage:** Based on attacks causing interruptions on news portals or recognizable institutions, making incidents more likely to reach the press and amplify the group’s message.
- **Frustration:** Public interruptions to digital services, such as tax payments, appointments or administrative procedures, generate discomfort among the population and criticism toward governments. Political opposition often exploits these outages, increasing the spread and impact of the incident.
- **Demonstrations of strength and capability:** Attacks against high-value targets such as parliaments, ministries or large companies address the need to display operational and technical capability. This also helps with recruiting more participants and building a reputation.

The combination of all these factors reveals how they choose targets and what motivates them. Their main goal is to reach as many people as possible using the easiest available path, maximizing visibility to reinforce their legacy and aligning attacks with geopolitical events used as justification for every operation.

Temporal Coordination with Geopolitical Events

As noted earlier, geopolitical events are one of the main motivations for the Threat Actor to begin attacks. Although the operations appear deliberate, all of them follow strategic planning with immediate responses to particular events.

They respond quickly, usually within 24 to 72 hours, driven by various motivations already discussed. Their affiliates mobilize rapidly toward chosen targets, showing significant response capability.

They also plan campaigns weeks or even months in advance for predictable events such as summits, elections in certain countries or symbolic anniversaries, often related to NATO and its member states. These operations are prepared beforehand, selecting targets carefully to align their actions.

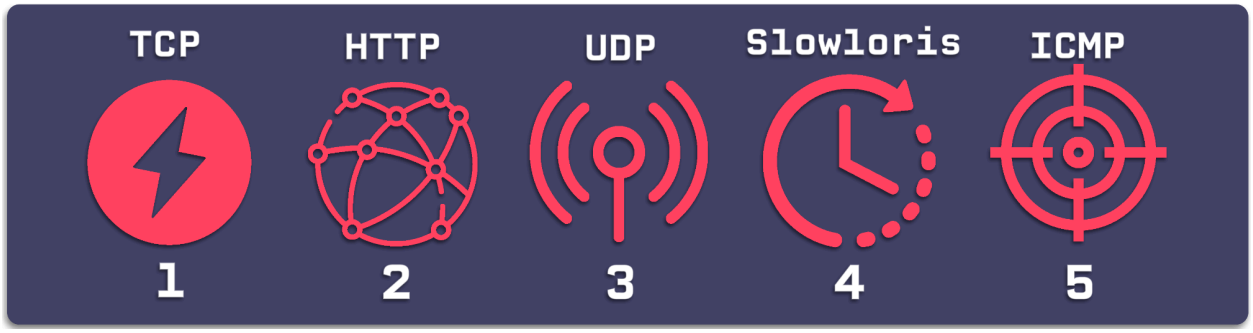
Continuing with the conflict, NoName057(16) does not stop its constant operations against persistent targets such as Ukrainian institutions or those providing support to Ukraine. This creates continuous pressure unrelated to any specific event.

Date / Event	Geopolitical context	Link to NoName057(16) activity / Response in 24–72h
June 2024 – Ukraine Peace Summit	International events supporting Ukraine, diplomatic actions condemning Russia, and intensified political messaging	Multiple European countries reported DDoS attacks by NoName057(16) during critical weeks of June 2024, aligned with the summit
Late February 2025 – 9th consecutive day of attacks in Italy	Diplomatic tensions and Italian political movements related to sanctions, defense actions, or support for Ukraine	On 25 February 2025 another wave of DDoS attacks against Italian institutions was attributed to the group
February–March 2025 – "OpSpain" campaign	Spain issued new sanctions and political gestures supporting Ukraine, increasing friction with pro-Russian actors	Between 2 and 4 March 2025, several Spanish municipal websites were attacked and attributed to NoName057(16)
May 2025 – UK announces new military support for Ukraine	Official British statements of military support, sanctions packages, and diplomatic actions opposing Russia	Shortly after, in May 2025, the group claimed responsibility for DDoS attacks on UK municipal and public-sector websites
Late June 2025 – NATO Summit in the Netherlands	NATO meeting with high-level officials discussing strategic decisions on Ukraine and collective military support	NoName057(16) launched DDoS attacks on websites linked to the summit around 23–24 June 2025
June 2024 – Ukraine Peace Summit	International events supporting Ukraine, diplomatic actions condemning Russia, and intensified political messaging	Multiple European countries reported DDoS attacks by NoName057(16) during critical weeks of June 2024, aligned with the summit

Most Used Attack Methods

Based on DDoSia's capabilities and the strategy followed by the group, numerous operations have been conducted. Attack methods change depending on the target, yet a clear trend has emerged in recent months, with consistent use of certain techniques.

In the attacks carried out by the group, there is a clear tendency toward TCP and HTTP attack vectors, which account for the vast majority of operations. These two methods have shown strong consistency over time.



Most used attack methods of NoName057(16)

The floods in these protocols have been dominant for the past six months, representing 90 percent of all attack methods used across multiple sectors and countries.

Method/Vector	Usage frequency
TCP L4 Flood	53.3%
HTTP L7 Flood	37.5%
UDP Flood	7.4%
Slowloris	1.3%
ICMP	0-1%

These figures differ from the evolution observed from the previous year to late 2025, where there was slightly more balance among attack vectors. TCP and HTTP remained dominant, but UDP and Slowloris had a more noticeable presence.

Method/Vector	Usage frequency
TCP Flood	45-55%
HTTP/HTTP/2/3 GET/POST Flood	35-45%
UDP Flood	5-10%
Slowloris	1-5%
ICMP	0-2%

This trend is common because these floods require less bandwidth and less technical complexity, relying on volumetric output. Slowloris or other techniques are less used because they focus on very specific targets, reducing operational volume. Meanwhile, UDP is reserved for more critical infrastructures and often supports or complements the main attack methods.

Most Targeted Sectors

NoName057(16) does not attack targets randomly; rather, they focus on ones that are more feasible or capable of generating greater visibility depending on the situation.

From last year through 2025, the group has continuously targeted various sectors. When compared with the target profiles previously discussed, the trend aligns clearly with governmental and public-sector institutions as well as media outlets, representing 50 to 65 percent of all attacks.

Private companies such as financial institutions or other industries, and even critical infrastructure, make up the remaining percentage of NoName057(16)'s attacks.

Sector	Attack frequency
Governmental or public	40-50%
Financial	17-23%
Media	12-18%
Critical infrastructure or portals	7-12%
Other companies	4-8%

The symbolic value and motivations for focusing on these sectors are clear: maintaining their supporter base while balancing easy targets (public sector environments) with more impactful ones (technology firms, financial institutions or critical infrastructure).

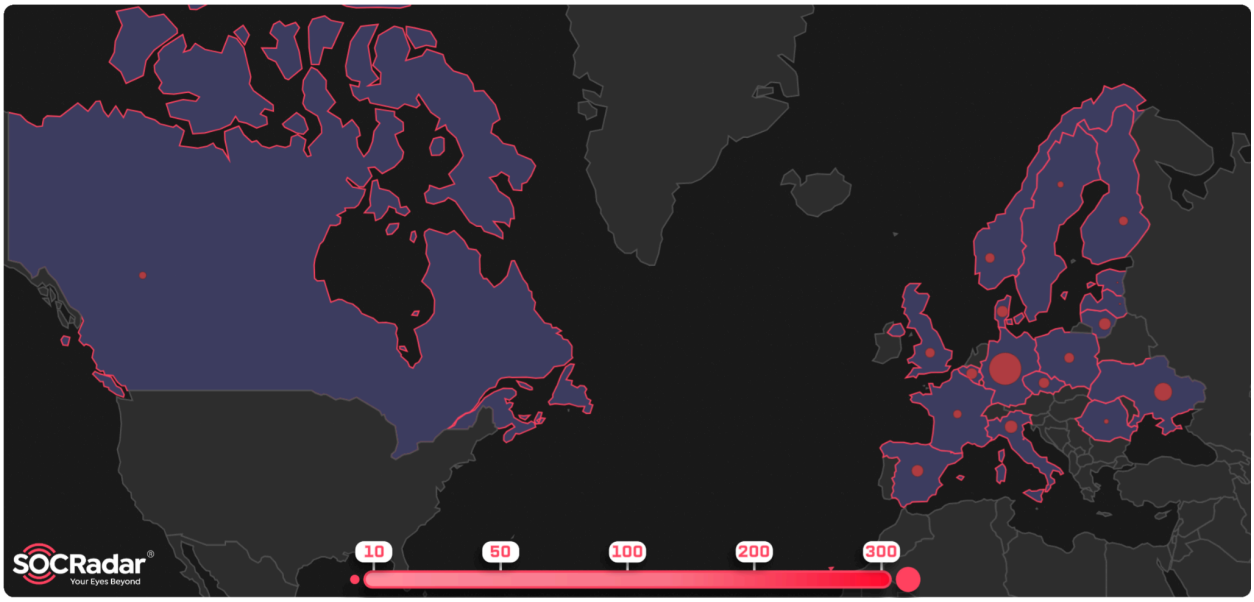
Most Targeted Countries

Just as some sectors are more frequently attacked for clear motivations, the same applies to countries. NoName057(16) must maintain its pro-Russian narrative, which influences both the types of organizations attacked and their geographical location. Targets are typically in countries perceived as hostile or offending Russia's interests.

Since the group's emergence until early 2025, there is a clear trend toward attacking countries near the Russia-Ukraine conflict. Ukraine is the epicenter, with attacks also extending to neighboring countries such as Lithuania and Poland.

Country	Attack frequency
Ukraine	+38%
Lithuania	+14%
Poland	+12%
Estonia	+9%
Czech Republic	+8%
Finland	+7%
Latvia	+6%
Germany	+5%

In the last six months, this trend has persisted, with a strong increase in attacks against central, northern and southern European countries including Germany, Denmark, Belgium, Italy and Spain, as detailed in the Spain and Belgium threat intelligence articles. These countries have faced between 100 and 350 operations after offering support to Ukraine.



Heatmap of main targets of NoName057(16)

The combination of geopolitical factors and the victim profiles selected by NoName057(16) is clearly reflected in the countries targeted, just as it is in the sectors. Any nation supporting Ukraine or acting against Russian interests becomes a primary target for the threat actor.

Future Trends

The future is uncertain, but years of activity by NoName057(16) are known, as well as by other adjacent pro-Russian groups previously mentioned, with similar motivations and modus operandi. This makes it possible to understand the idiosyncrasy of this type of group and to project a plausible line of action in the near future as long as the current trend continues.

As long as the same armed conflicts remain active, such as Russia–Ukraine, related sanctions, and so on, NoName057(16) may continue reaffirming its stance toward those who oppose its position. However, it is a reactive group that is also sensitive to changes and external geopolitical contexts.

Conflicts and geopolitical events

Regarding conflicts, tensions may arise that lead the group to act. These include the wars already underway, but also latent situations such as Russia's absolute opposition to NATO admitting more members, or tensions in Moldova or the South Caucasus.

- Ukraine VS Russia: It is evident that the TA may continue supporting the Russian side in the Ukraine–Russia conflict in 2026, concentrating a large portion of its attacks as it has done until now, maintaining constant pressure on the Baltic countries, Poland, the United Kingdom, Germany, and others. In the case of possible negotiations, which are currently active, it is likely that overall activity would be reduced, as the group usually shows itself receptive to change, seeking not to harm Russia in a potential agreement, adopting a more watchful and less offensive profile.

- Tensions with NATO: NATO and Russia remain in constant tension that could continue to deteriorate next year, potentially adding more targets to the TA's list, especially due to active alliances such as Japan or South Korea. The group may also intensify, as it already has this year, its attacks against Nordic countries due to their stronger ties and accessions to NATO by Finland and Sweden.

- New NATO members: NATO's pressure to keep adding members, through relationships with countries such as Moldova or Georgia, could be taken by NoName057(16) as direct provocations, prompting automatic actions. This is especially relevant considering there are already tense scenarios with Moldova regarding the reintegration of Transnistria, where Russia is completely opposed. Similarly, the group may take reprisals for Armenia's rapprochement with NATO during its tensions with Azerbaijan. In essence, any news bringing another country closer to NATO, or involving NATO members in external conflicts or tensions, can trigger active responses from the TA.

- Events: On the other hand, recent campaigns could be expected around events such as the anniversary of the invasion in February 2026 or upcoming NATO or EU summits, all scenarios where the group has repeatedly deployed its attacks before.

Conclusion

The use of narrative by NoName057(16) to justify its acts or operations is clear, particularly considering that the Russian government does not criminalize them and appears as a possible supporter of these actions. This raises suspicions of some form of relationship with these hacktivist groups, which help from other angles to reinforce their ideals and contribute to pro-Russian tensions.

The group has demonstrated resilience and technical capability, matching or surpassing other previous or current groups with more experience in DDoS attacks. As long as Russia has enemies or ongoing tensions, if NoName057(16) remains active, it will continue gaining followers and improving DDoSia, focusing on countries and institutions in the Western world.

MITRE ATT&CK TTPs

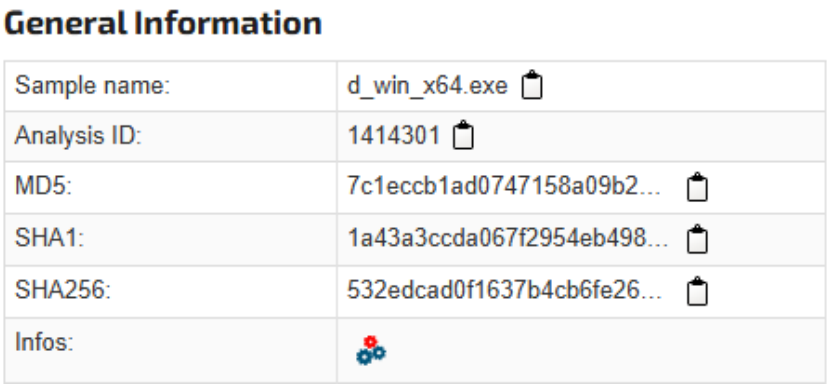
Tactic	Technique	Description
TA0043 Reconnaissance	T1595: Active Scanning	Operators uses HTTP/HTTPS/UDP services to confirm target availability before coordinated campaigns
TA0043 Reconnaissance	T1589: Gather Victim Identity Information	Public institutional and political information is collected to select targets aligned with geopolitical objectives
TA0001 Initial Access	T1190: Exploit Public-Facing Application	Occasionally used to validate reachable or weak endpoints before distributing them as DDoS targets
TA0001 Initial Access	T1078: Valid Accounts	Affiliates or operators abuse weak/default services for basic enumeration, though not tied to infection
TA0002 Execution	T1059: Command and Scripting Interpreter	The Go binary of DDoSia is executed with CLI parameters controlling threads, speed and debug output
TA0002 Execution	T1106: Native API	The malware uses native Go networking syscalls (TCP/UDP sockets, deadlines, write loops) for high-rate attacks
TA0002 Execution	T1047: Windows Management Instrumentation	Some operators automate execution on Windows using simple scripts or system automation tools

TA0005 Defense Evasion	T1027: Obfuscated Files or Information	Strings in DDoSia are obfuscated or decrypted at runtime, complicating static analysis
TA0005 Defense Evasion	T1132: Data Encoding	AES,RC4 or GCM encoding protects JSON C2 tasking from inspection and fingerprinting
TA0005 Defense Evasion	T1090: Proxy Use	Use of rotating internal proxies hides affiliate identity and complicates traffic attribution
TA0005 Defense Evasion	T1036: Masquerading	Rotating user-agents mimics legitimate browsers and reduces detectability in floods
TA0011 Command and Control	T1071.001: Web Protocols	Encrypted JSON tasking and results are sent over HTTPS to the C2 server
TA0011 Command and Control	T1573: Encrypted Channel	Communication uses AES-GCM or RC4 variants to protect C2 traffic
TA0011 Command and Control	T1095: Non-Application Layer Protocol	Some variants use raw TCP sockets for lightweight heartbeat communication
TA0040 Impact	T1498: Network Denial of Service	Primary purpose using HTTP/2, HTTP flood, TCP flood, UDP flood, and mixed volumetric attacks
TA0040 Impact	T1498.002: Reflection/Amplification	Proxy-driven amplification effects increase volume and obfuscate origins
TA0040 Impact	T1499: Endpoint Denial of Service	Randomized payloads or heavy HTTP options cause CPU exhaustion on web servers

Indicators

Reviewing the indicators to understand what infrastructure the adversary uses is a key point for understanding how they operate their tools and how they plan their attacks. Likewise, collecting these indicators allows early mitigation or detection of possible uses of any of the tools employed by the group.

During the reverse-engineering phase, several indicators were identified that are useful for discovering others. The first of them is the IP address **77[.]91[.]100[.]134**, which appears referenced in some reports containing binaries that follow the same structure seen in DDoSia-related groups.

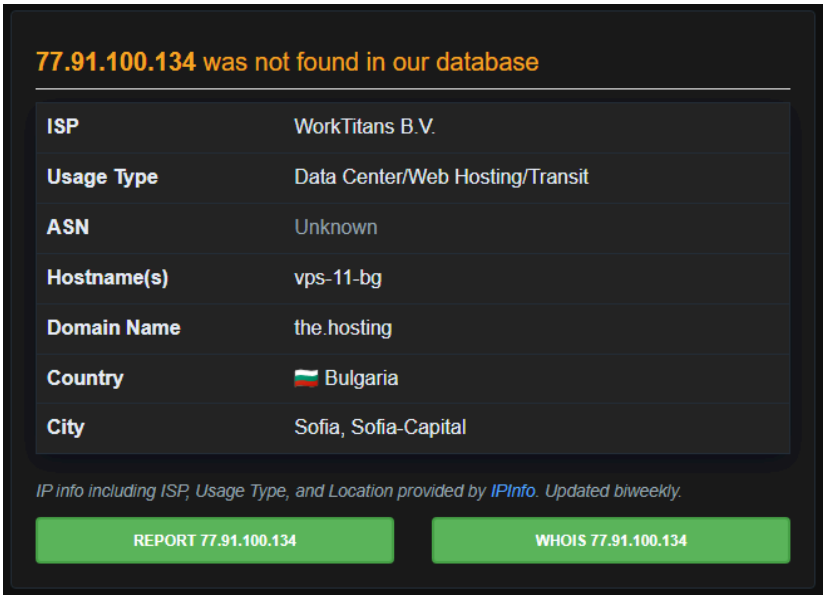


General Information

Sample name:	d_win_x64.exe
Analysis ID:	1414301
MD5:	7c1eccb1ad0747158a09b2...
SHA1:	1a43a3ccda067f2954eb498...
SHA256:	532edcad0f1637b4cb6fe26...
Infos:	

VirusTotal Sample information

However, **no report** is directly associated with **this IP,** although it does show some hits on VirusTotal.



77.91.100.134 was not found in our database

ISP	WorkTitans B.V.
Usage Type	Data Center/Web Hosting/Transit
ASN	Unknown
Hostname(s)	vps-11-bg
Domain Name	the.hosting
Country	Bulgaria
City	Sofia, Sofia-Capital

IP info including ISP, Usage Type, and Location provided by IPInfo. Updated biweekly.

REPORT 77.91.100.134 | WHOIS 77.91.100.134

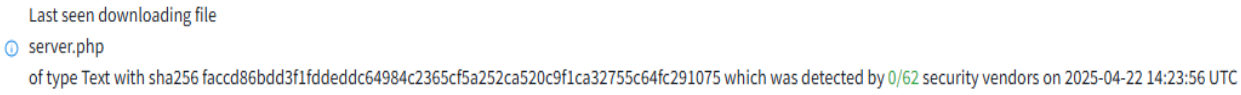
IP lookup showing the C2 server hosted on a Bulgarian data center infrastructure.

Low detection rate for the C2 IP, with only one security vendor flagging it as malicious.

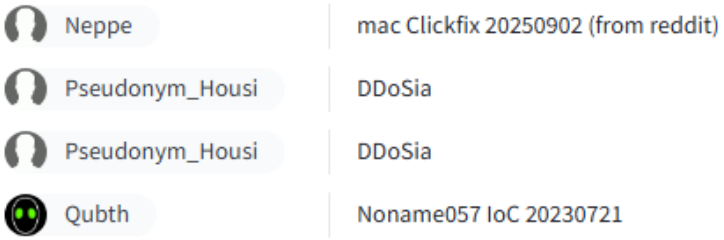
It is possible to observe various paths and addresses linked to that IP with reports that directly lead to DDoSia indicators.

Scanned	Detections	Status	URL
2025-10-10	1 / 98	-	http://77.91.100.134/basic
2025-09-09	1 / 98	-	http://77.91.100.134/client/login
2024-12-19	2 / 96	-	http://77.91.100.134/
2024-11-16	2 / 96	-	https://77.91.100.134/

Scans show consistently low detection rates for C2 URLs over time.

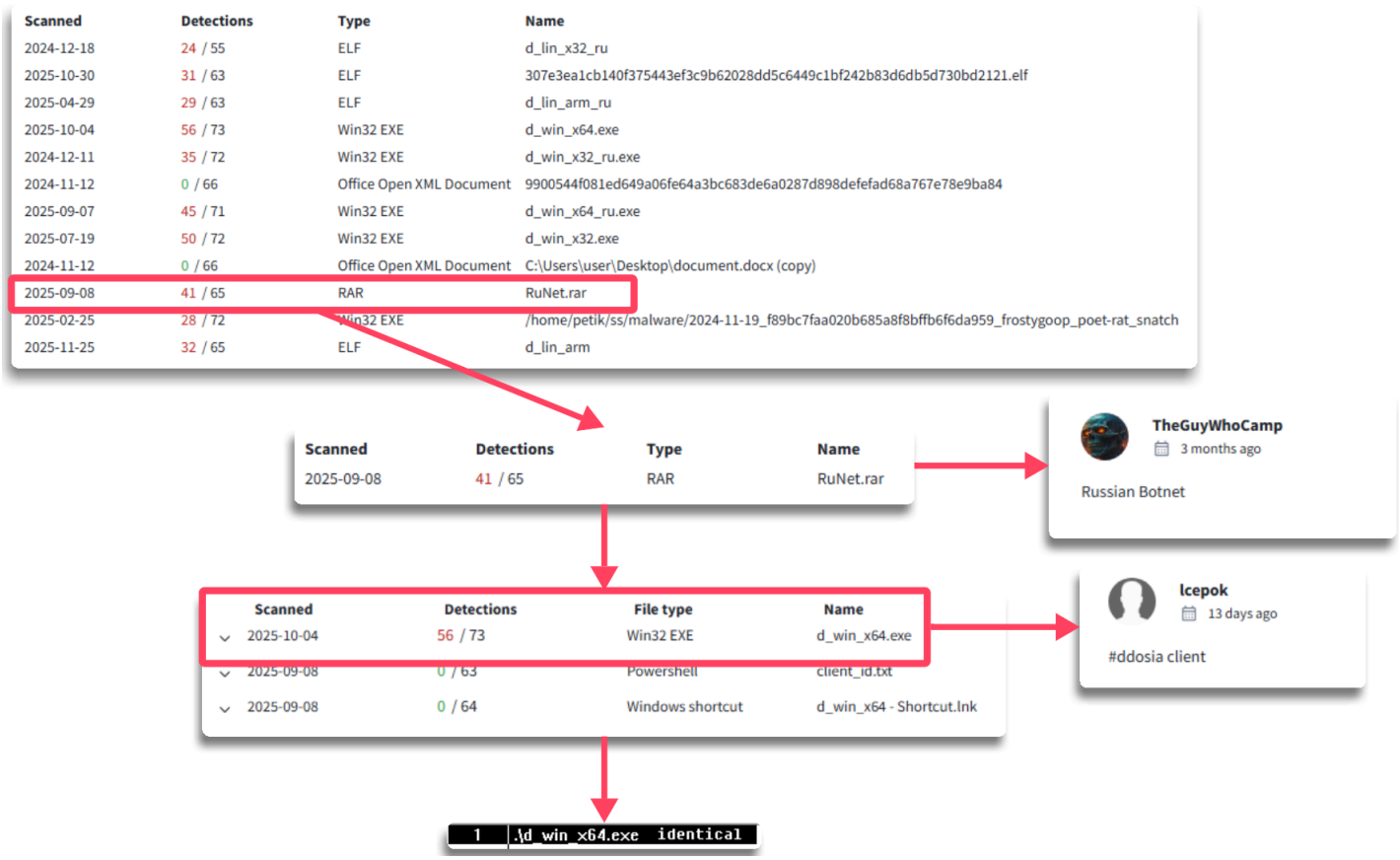


C2 server last observed hosting a PHP file.



Community submissions linking the IP to DDoSia and NoName057(16) activity.

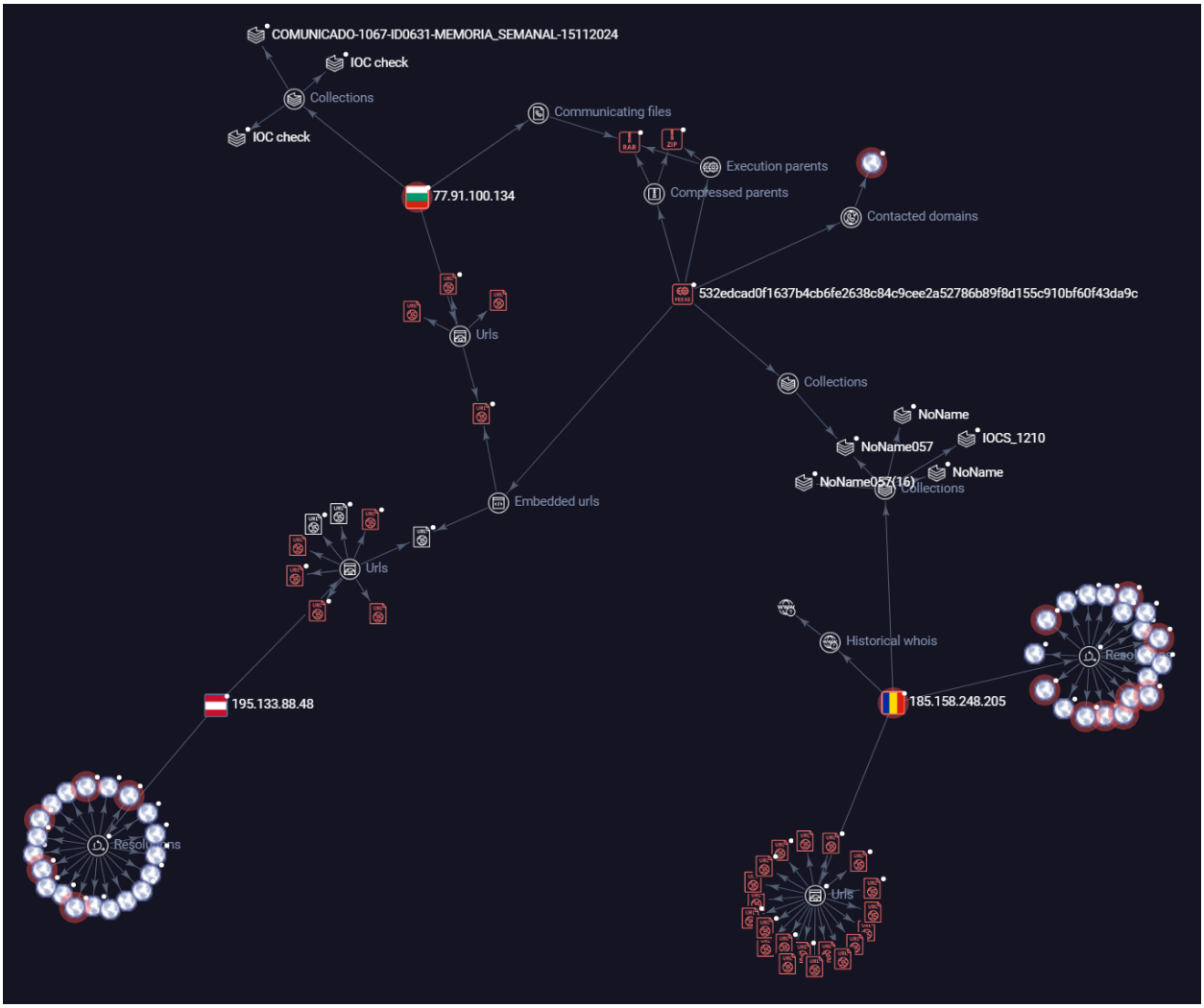
At the same address, other binaries can be found following the same pattern described earlier, as well as compressed files that point both to the mentioned IP and to others that also lack reports but contain the same relationships with binaries identical to those previously analyzed.



Threat intelligence records linking DDoSia samples, client files, and related infrastructure to botnet activity.

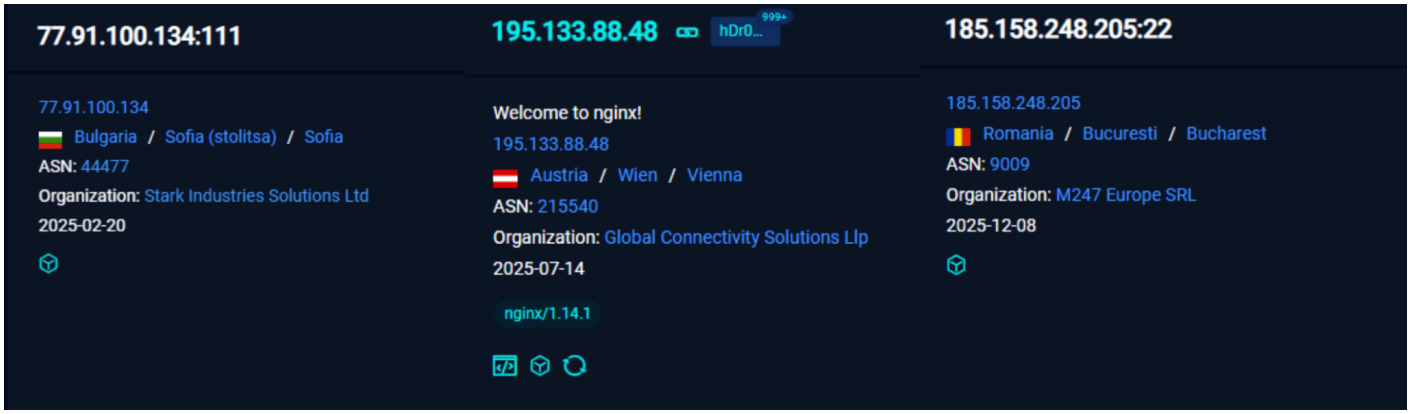
← Relationship graph linking NoName057(16), DDoSia samples, and shared C2 infrastructure.

Within the use of these addresses, there is a direct relationship with NoName057(16) and DDoSia binaries, which are also correlated with each other and are practically identical versions to those already examined.



Correlation view showing identical binaries and overlapping addresses across related campaigns.

Despite the three addresses being very different from one another, with different organizations, ASNs, and other attributes, they show direct correlations to the mentioned DDoSia binaries, which contact these IPs. However, NoName usually rotates these communication points frequently during its operations, in addition to using proxies as mentioned earlier.



IP addresses correlated with the analyzed DDoSia binaries.

Connections can be observed with some lists and direct references to other binaries that are part of DDoSia:

Scanned	Detections	Type	Name
2024-12-18	24 / 55	ELF	d_lin_x32_ru
2025-10-30	31 / 63	ELF	307e3ea1cb140f375443ef3c9b62028dd5c6449c1bf242b83d6db5d730bd2121.elf
2025-04-29	29 / 63	ELF	d_lin_arm_ru
2025-10-04	56 / 73	Win32 EXE	d_win_x64.exe
2024-12-11	35 / 72	Win32 EXE	d_win_x32_ru.exe
2025-09-07	45 / 71	Win32 EXE	d_win_x64_ru.exe
2025-07-19	50 / 72	Win32 EXE	d_win_x32.exe
2025-09-08	41 / 65	RAR	RuNet.rar
2025-02-25	28 / 72	Win32 EXE	/home/petik/ss/malware/2024-11-19_f89bc7faa020b685a8f8bffb6f6da959_frostygoop_poet-rat_snatch
2025-11-25	32 / 65	ELF	d_lin_arm

Detection results for multiple DDoSia binaries across Linux and Windows platforms.

The relationship between these indicators is also visible on the SOCRadar platform.

SOCRadar Platform correlations

Indicators of Compromise (IoCs)

Hashes

- 532edcad0f1637b4cb6fe2638c84c9cee2a52786b89f8d155c910bf60f43da9c
- 9a4e37b900ae3ed3d8d5adff39f462a3607626f4f8c56687b4ef0f0295ed5318
- b81734717f36d3cea59e5690b984333c5a6908a15883a0463d77cb20dadcec0c
- 87cd40fbf9f363c212a8402cc8350f624fd6760799c013a0cdd301707a5bd083
- 7ee3574b0693e78060d863a5794437960aec0614af6c1909dd075daec0bcaf92
- 0eae66824c65efe6b69937bf8427b7f28df591f2788b8088fbe9a05e8c26e077
- 943bceda805a54b72bc26dc660ff1d4b7bc49b801aa45db8e9f70b0d02aff9cc

IP Addresses

- 77[.]91[.]100[.]134
- 195[.]133[.]88[.]48
- 185[.]158[.]248[.]205

References

- https://socradar.io/blog/operation-eastwood-targets-noname05716/
- https://socradar.io/blog/weekly-ddosia-threat-intelligence-spain/
- https://socradar.io/blog/coordinated-hacktivist-threat-activity-target-belgium/
- https://blog.sekoia.io/following-noname05716-ddosia-projects-targets/
- https://assets.recordedfuture.com/insikt-report-pdfs/2025/cta-2025-0722.pdf
- https://blog.sekoia.io/Noname05716-Ddosia-project-2024-updates-and-behavioural-shifts/