P2P Botnets: Review - Status - Continuous Monitoring

blog.netlab.360.com/p2p-botnets-review-status-continuous-monitoring/

360Netlab

November 3, 2022

Origins

P2P networks are more scalable and robust than traditional C/S structures, and these advantages were recognized by the botnet authors early on and used in their botnets. In terms of time, Storm, which appeared in 2007, can be considered the progenitor of this area, when botnet threats were first known to the public. after Storm, there have been Karen, ZeroAccess, GameOver, Hijime, mozi and other kinds of P2P botnes, P2P botnets come and go, and some, for example, Mozi keep on going even though the author had been caught.

The early P2P botnets mainly targeted Windows machines, such as Storm, ZeroAccess and GameOver, which were infected with the Windows operating system. after Mirai appeared in 2016, Linux IoT devices, which exist in large numbers on the network and lack some basic security defense, started to become the target of many botnets. For example, Hijime, mozi, pink all target Linux devices.

Because of the "centerless" nature of P2P networks, it is somewhat difficult to assess their size using traditional means. To try to solve this problem, security researchers have invented P2P crawler technology, which can track a P2P botnet and obtain node IPs, download links and configuration information for scale assessment and targeted removal.

Our team(360 netlab) has been focusing on discovering and tracking active botnets for a long time, and P2P botnets are surely on our radar. For example, we first disclosed the mozi botnet in 2019. In order to gain better visibility, we built an industrial-level tracking system for P2P botnets, with the goal of covering major active P2P botnets, This blog article will briefly analyze the current status of the following 5 families based on the tracking data generated by this system.

(In addition to the 5 families mentioned in this article, readers are also welcomed to leave comments below about new|active families of interest, and we will look into them accordingly)

Overview of tracking Strategy

This section will briefly introduce the main tracking strategies used in our tracking system.

Tracking Goals

The main goal of the system is to record the IPs of all the p2p nodes, we “create” a node by simulating the communication protocol, so it can join the corresponding P2P network to participate in the data message exchange. Every time a message exchange is successfully completed, the IP of the other party is recorded, this goes on and on and finally the majority of the nodes from the target P2P network would be recorded.

Methods

The protocol design of each P2P family varies, but following are some common strategies, normally at least one of these would be selected as the tracking method.

Active Probe: This strategy is somewhat similar to a public network scanner in terms of working principle. It first feeds probe messages to the target node, then parses the received reply messages, and identifies the peer as a peer node when the returned message format matches the family characteristics. In practice, we will first delineate a probe range and then probe the nodes within this range (where the probe range may consist of a suspicious network segment or suspicious nodes generated from other policies).

Recent communications：Common P2P families maintain a "recent communications list" of recent peers in each node's memory. In some families, this list is also available to other nodes via specific commands, and would commonly be used as a seed list when the node “boots up”, so that they can quickly join the P2P network. In this case, we can discover more peer nodes by traversing this "recent communication list".

Node heartbeat: When a node maintains a "recent communication list", it will send heartbeat messages to the nodes on the list periodically to declare its online status. Based on this, we can add the "fake node" to the other node's active list to get the active status of the corresponding node at any time. In some cases, we also send heartbeat messages to ensure that we don’t get kicked out by the network.

Wait and see: "Hajime" and "Mozi", for example, use "Distributed Hash Table" to implement their P2P network structure. This technique is designed to speed up data lookup by adding a rule of information-to-node distance and prioritizing the information to be stored on those nodes that are closer. Based on this rule, we can forge a “we-are-more-closer” node then wait for the arrival of other nodes. When other nodes try to obtain the information of the corresponding family from the forged node, we can directly record the other IP as the tracking result.

How to read the Data

Tracking family selection basis

We consider the following two dimensions to screen the appropriate families for tracking to ensure the relative objectivity of the final results.

Based on size: When selecting families, the most priority indicator is that the size of the botnet has to be large, or once historically large enough, in this case "Hajime"/"Mozi"/ “pink" all stand out.

Recent Disclosures: The next choice is the newly emerged ones that have been active at lease for a little while, based on this, we have chosen "panchan" and "frizefrog" as they are newly discovered this year.

Size of the infected bot nodes

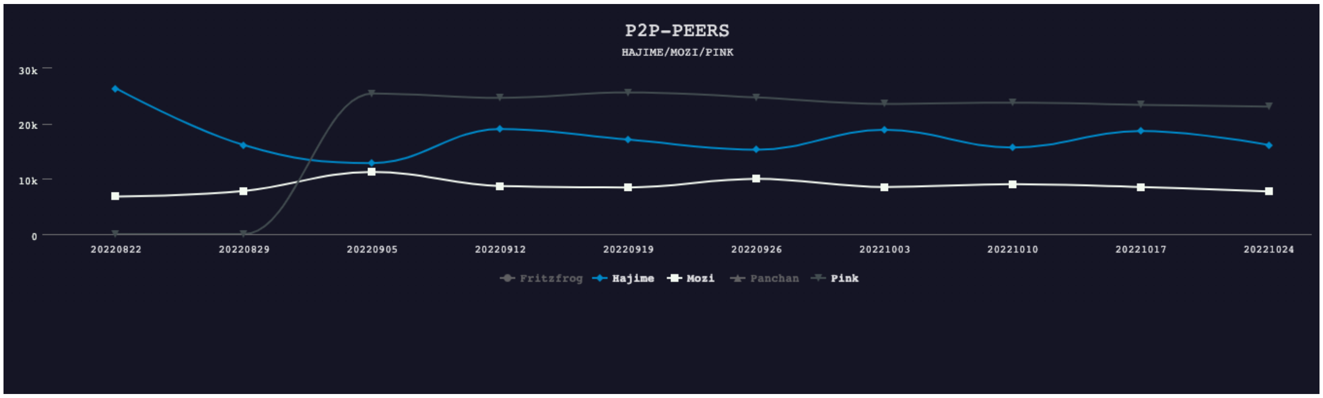
Depending on the type of the host device, the number of bot IPs does not necessarily reflect the true number of infected devices.

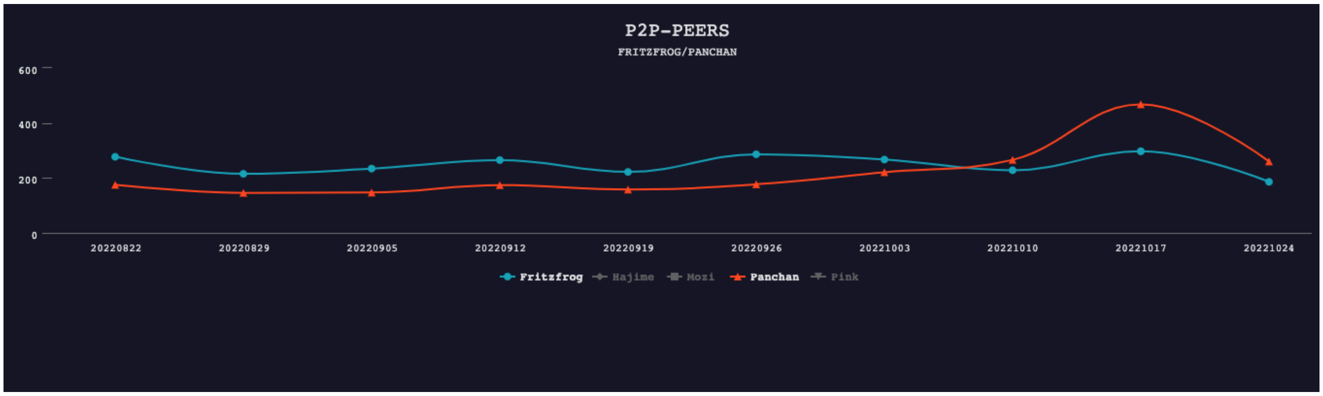
Bots are servers: In order to provide stable services, the public IPs of servers usually do not change, so the numbers are more accurate.

Bots are IoT devices: These devices are usually found in the residential network. We all know that the IP addresses of residents devices change frequently. This can lead to a large uncertainty in the mapping relationship between the public IP and the device. Multiple devices might share a common public IP (NAT scenario), and devices can switch to different IPs multiple times within a time window (dial-up Internet scenario).

Daily activity of each family

As a comparison, if we take the daily activity of each Monday since August as a sample to plot the medium- and long-term tracking graph, the following is shown.





We can clearly see the order of the family size:

Pink > Hajime > Mozi >> FritzFrog <> Panchan

We can see that the daily activity data of each family has not changed much over the three months period (see the discussion below for Pink's fluctuations in August)

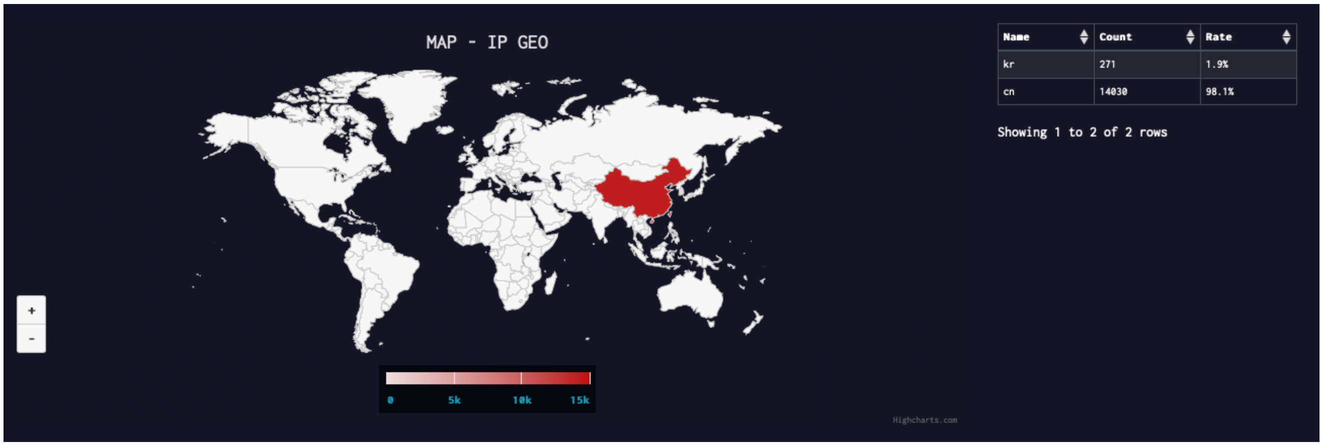
Statistics by family

Pink

At one point, the Pink family had infected more than one million devices in China, it has some cleverly designed command and control protocols. For time sensitive instructions, the control commands are pushed to the bots through a centralized mechanism, On the other hand, if the instructions are not urgent, P2P would be used. For more information, please refer to our previous report.

《Pink, a botnet that competed with the vendor to control the massive infected devices》

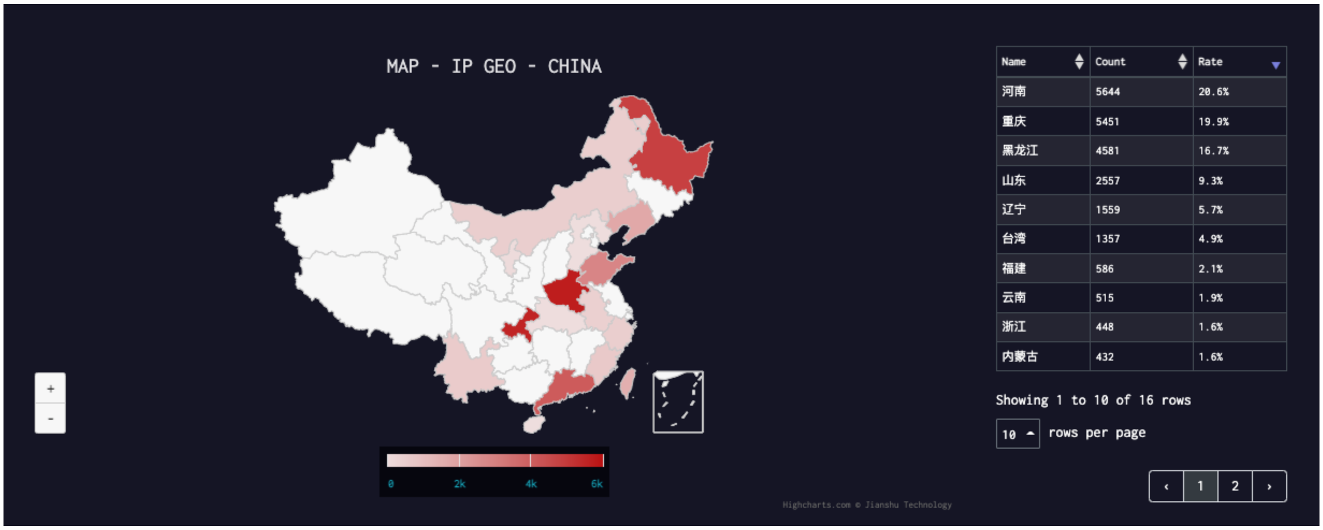
Geographical distribution



Name	Count	Rate
kr	271	1.9%
cn	14030	98.1%

Showing 1 to 2 of 2 rows

As shown in the figure, Pink's scope of influence is mainly domestic IoT devices, and the following is its distribution in China.



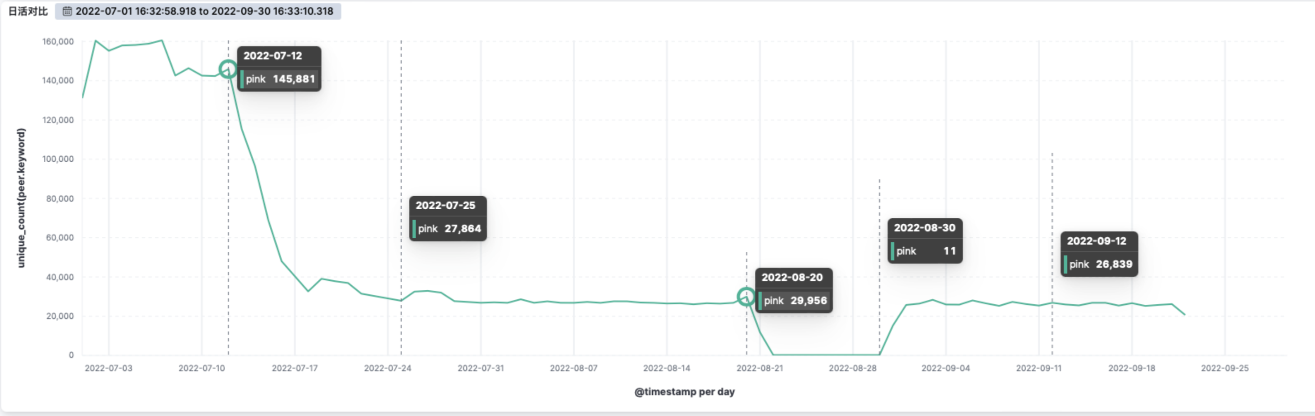
Name	Count	Rate
河南	5644	20.6%
重庆	5451	19.9%
黑龙江	4581	16.7%
山东	2557	9.3%
辽宁	1559	5.7%
台湾	1357	4.9%
福建	586	2.1%
云南	515	1.9%
浙江	448	1.6%
内蒙古	432	1.6%

Showing 1 to 10 of 16 rows

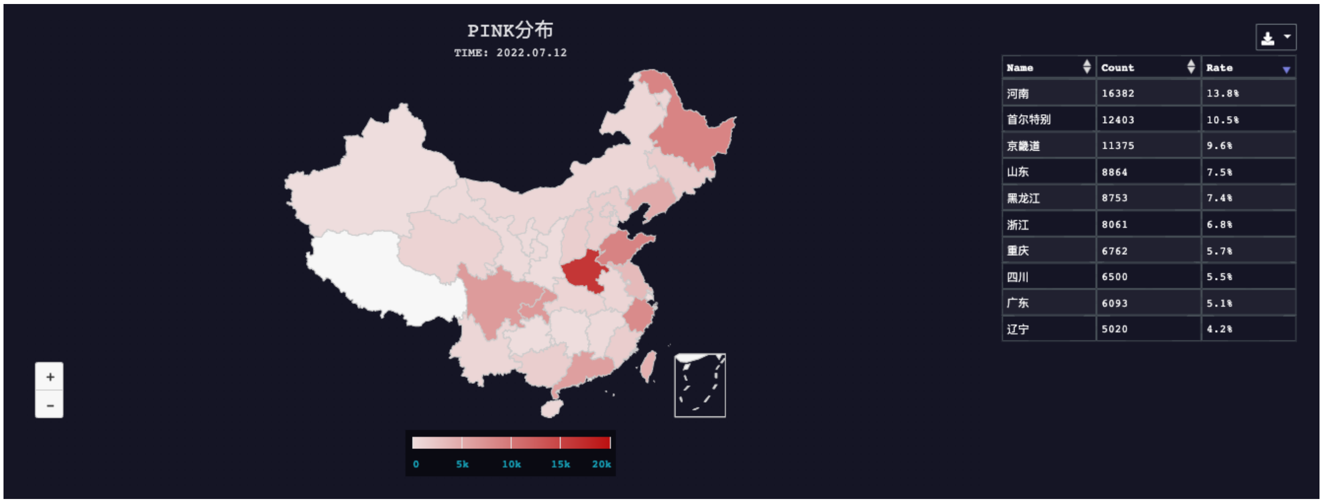
10 rows per page

Daily activity fluctuation

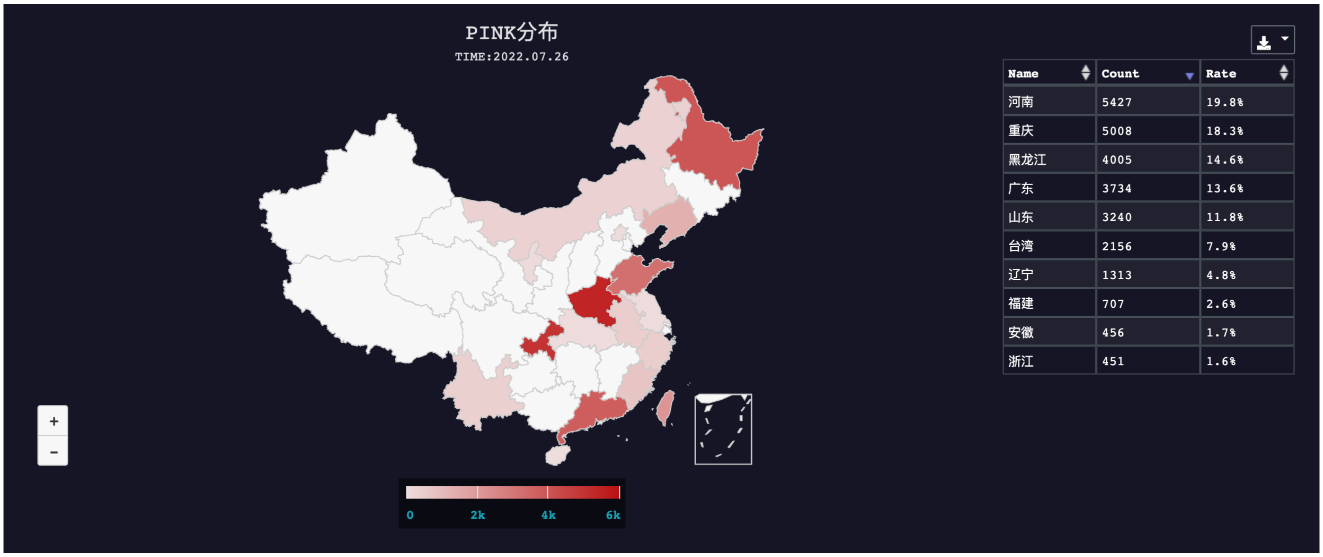
It is worth mentioning in particular that the daily activity data of the family has fluctuated greatly since July, first dropping by an order of magnitude in a week starting on July 12, reaching a daily activity of about 20,000, then returning to zero for about 10 days after August 20, and then returning to 20,000 in September. The fluctuation of daily activity can be seen in the following chart.



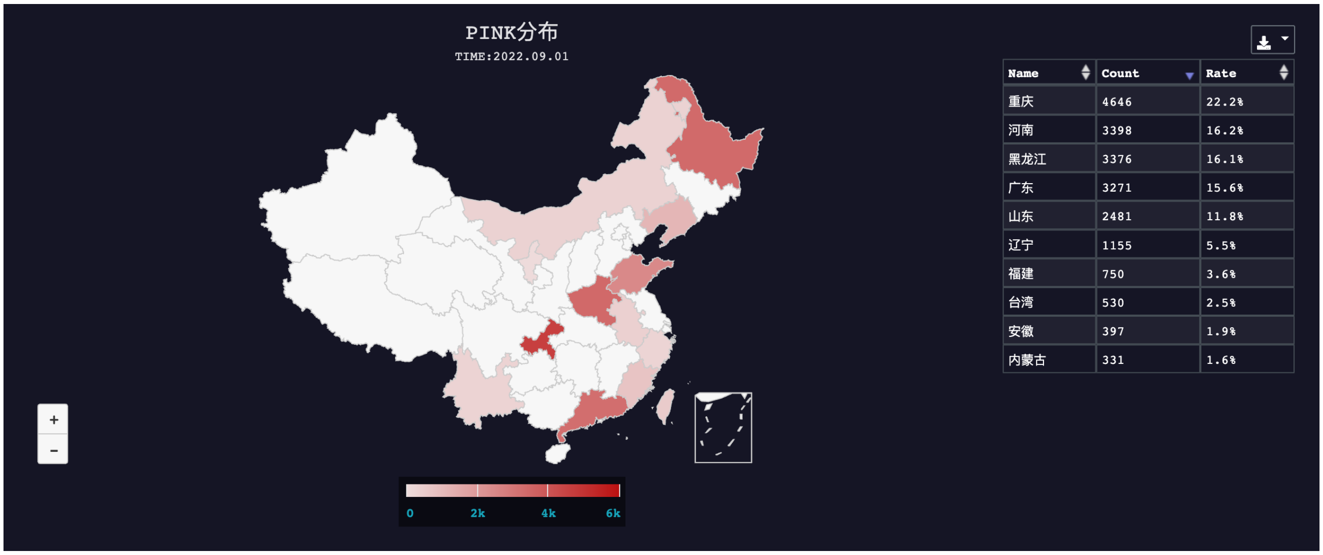
Now let’s take a look at daily activity data of July 12, July 26 and September 1. It is easy to tell that the number of daily activities in most provinces decreased significantly with time goes by.



Name	Count	Rate
河南	16382	13.8%
首尔特别	12403	10.5%
京畿道	11375	9.6%
山东	8864	7.5%
黑龙江	8753	7.4%
浙江	8061	6.8%
重庆	6762	5.7%
四川	6500	5.5%
广东	6093	5.1%
辽宁	5020	4.2%



Name	Count	Rate
河南	5427	19.8%
重庆	5008	18.3%
黑龙江	4005	14.6%
广东	3734	13.6%
山东	3240	11.8%
台湾	2156	7.9%
辽宁	1313	4.8%
福建	707	2.6%
安徽	456	1.7%
浙江	451	1.6%



Name	Count	Rate
重庆	4646	22.2%
河南	3398	16.2%
黑龙江	3376	16.1%
广东	3271	15.6%
山东	2481	11.8%
辽宁	1155	5.5%
福建	750	3.6%
台湾	530	2.5%
安徽	397	1.9%
内蒙古	331	1.6%

So, it is very likely that starting from July, the major device vendor carried out a national wide clean up effort, resulting in a significant decrease in the number of infected devices.

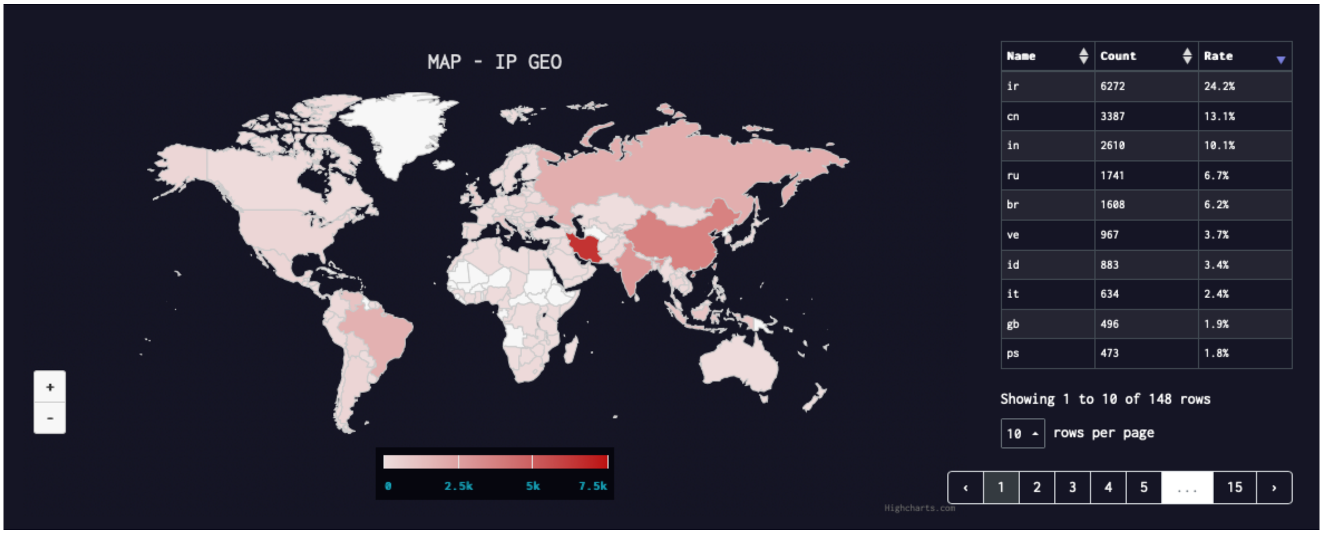
The fluctuations in late August, on the other hand, are likely due to a national wide C2 blocking action in place.

Hajime

Hajime, which emerged in the same year as MIRAI, less than a few months apart, has been claimed to be run by "white hats" in its alert messages, and its components function with the primary goal of self-propagation. The communication and management between its components makes extensive use of asymmetric encryption and decryption algorithms, making it an extremely classic P2P botnet family. We have covered this botnet in our previous blog.

《Is Hajime botnet dead?》

Geographical distribution



Name	Count	Rate
ir	6272	24.2%
cn	3387	13.1%
in	2610	10.1%
ru	1741	6.7%
br	1608	6.2%
ve	967	3.7%
id	883	3.4%
it	634	2.4%
gb	496	1.9%
ps	473	1.8%

Showing 1 to 10 of 148 rows

10 rows per page

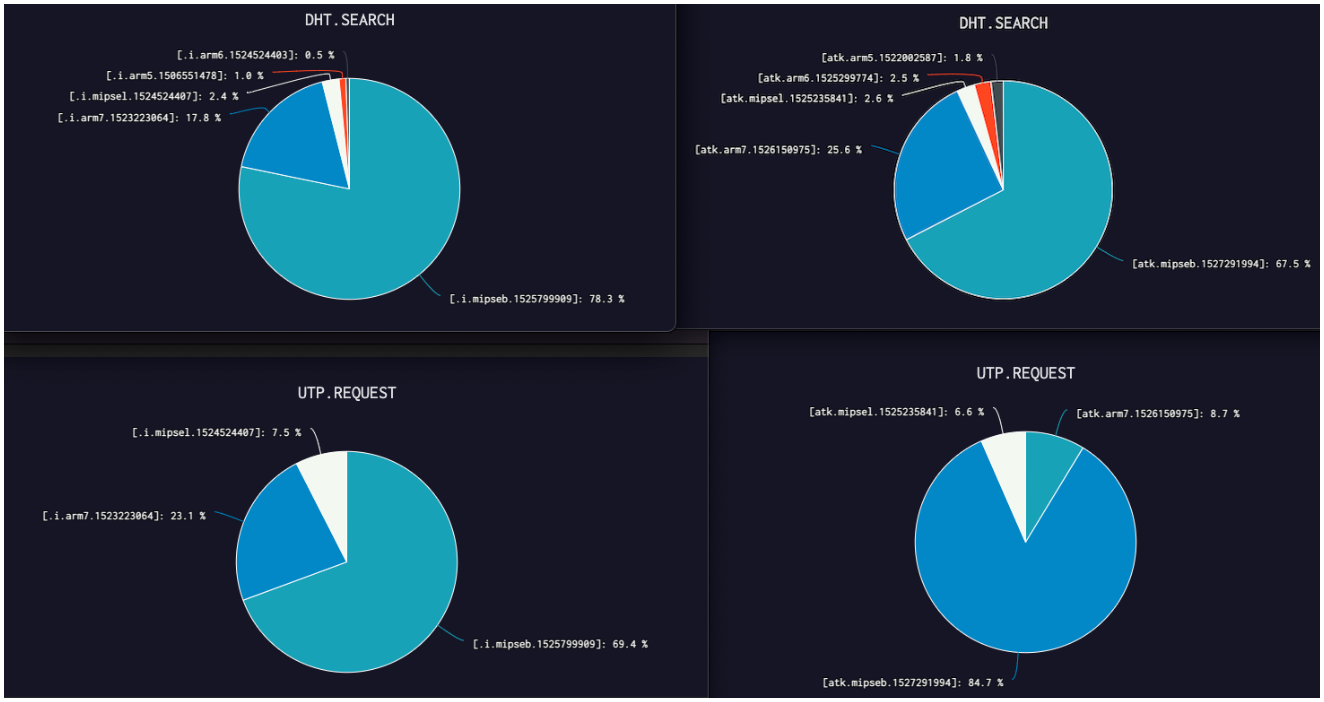
‹ 1 2 3 4 5 ... 15 ›

Iran tops the list

We normally don’t see Iran on our security event list, but with Hajime infection, Iran is leading the pack, which is pretty interesting. Please leave comments below if you have more insights.

CPU distribution in Hajime

Hajime is a P2P network built on file exchanging and each Hajime bot tries to find the latest version of .i.xxx and atk.xxx files (e.g. atk.arm7/.i.arm7) when it runs. This gives us an opportunity to evaluate the CPU distribution in the "Hajime network". When the Hajime node asks us which nodes contain the corresponding files, we gets a DHT.search count. When a Hajime node asks us to download the corresponding file, we gets a uTP.Request count. Putting these two types of files and two types of counts together, we would have the following four pie charts:



Based on the above pie charts, we can see that MIPS based bots are the majority in the Hajime network, far exceeding the sum of the other types of hosts, while MIPSEL has the fewest host nodes.

If we consider that Hajime had integrated a large number of vulnerabilities for propagation, this data can even reflect to some extent the distribution of each type of CPU in smart devices.

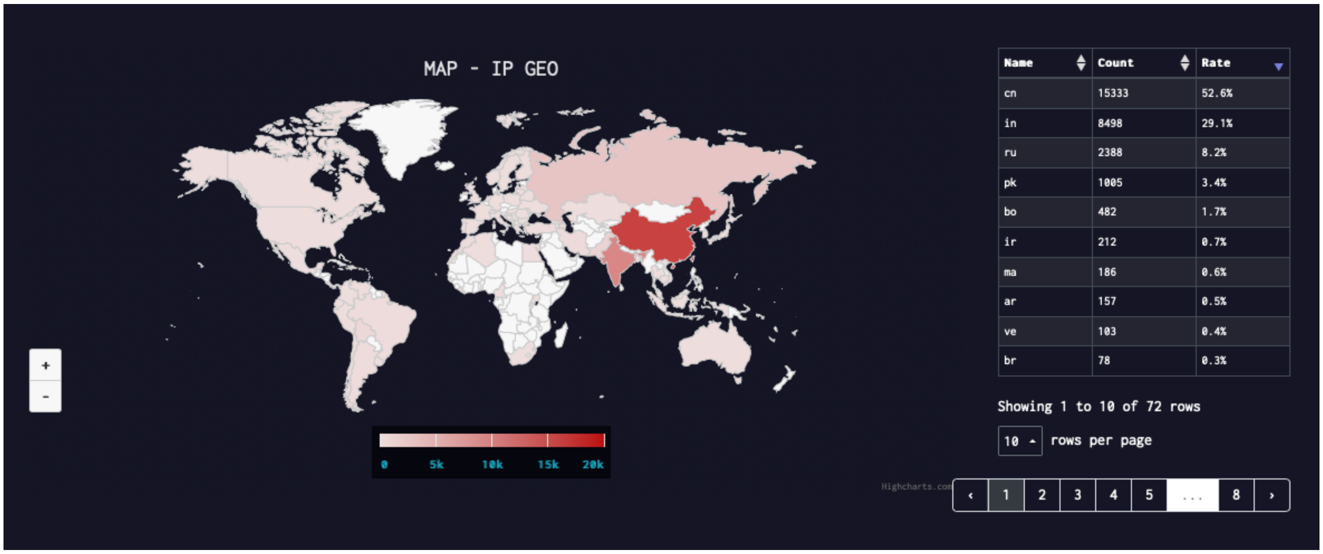
Mozi

Mozi started out as a P2P family performing DDoS attacks for profits, and later added a mining component. Its network topology is built on the basis of the DHT protocol. More information can be found in our previously published reports.

《Mozi, Another Botnet Using DHT》

《The Mostly Dead Mozi and Its’ Lingering Bots》

Geographical Distribution



Name	Count	Rate
cn	15333	52.6%
in	8498	29.1%
ru	2388	8.2%
pk	1005	3.4%
bo	482	1.7%
ir	212	0.7%
ma	186	0.6%
ar	157	0.5%
ve	103	0.4%
br	78	0.3%

Showing 1 to 10 of 72 rows

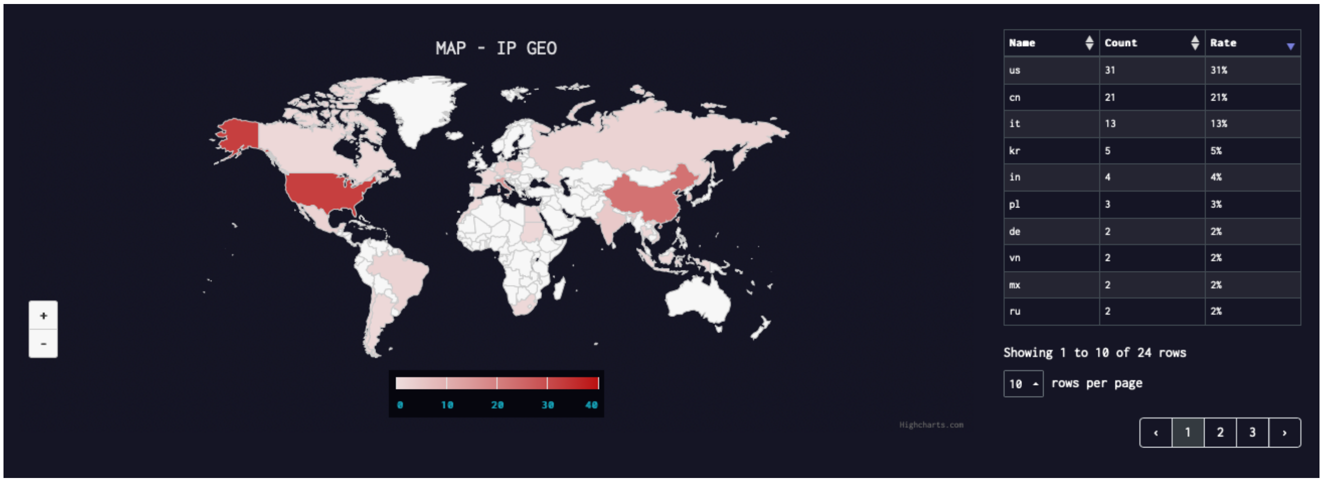
10 rows per page

FritzFrog

FritzFrog is a mining P2P family which relies on SSH services to build P2P networks. It was first disclosed by akamai. More details can be found in the following report (interestingly, FritzFrog wallet address is related to Mozi).

《FritzFrog: P2P Botnet Hops Back on the Scene》

Geographical Distribution



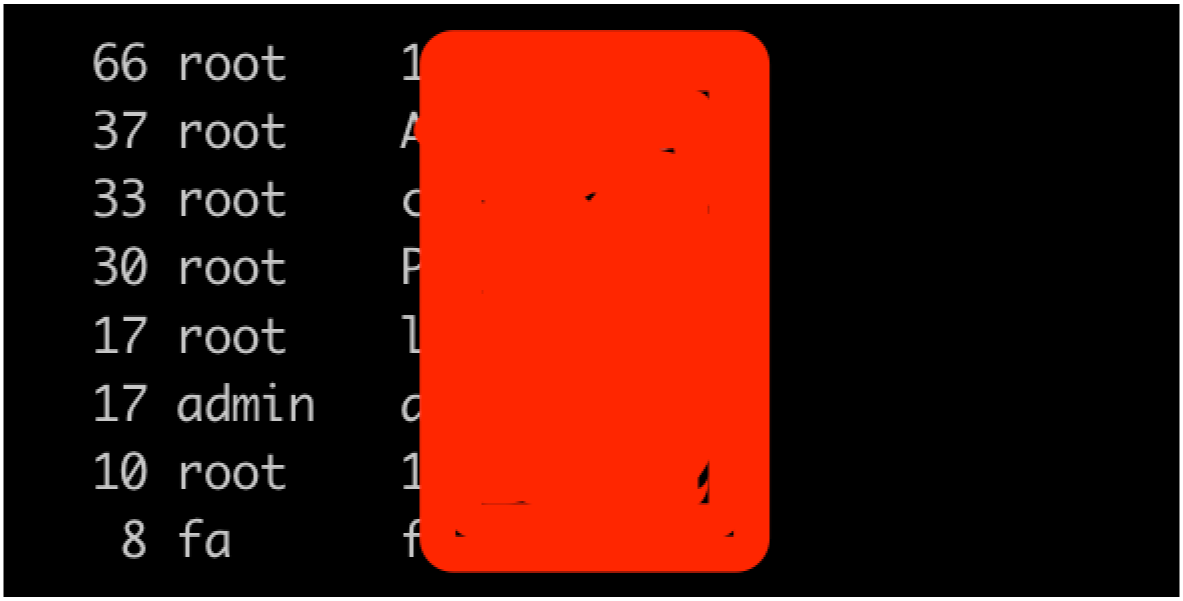
Name	Count	Rate
us	31	31%
cn	21	21%
it	13	13%
kr	5	5%
in	4	4%
pl	3	3%
de	2	2%
vn	2	2%
mx	2	2%
ru	2	2%

Showing 1 to 10 of 24 rows

10 rows per page

Account Passwords in FritzFrog

Since FritzFrog's P2P is based on SSH implementation, the password of the infected nodes are reflected in the crawled data, the following are the top passwords.



```
 66 root
 37 root
 33 root
 30 root
 17 root
 17 admin
 10 root
  8 fa
```

[redacted]

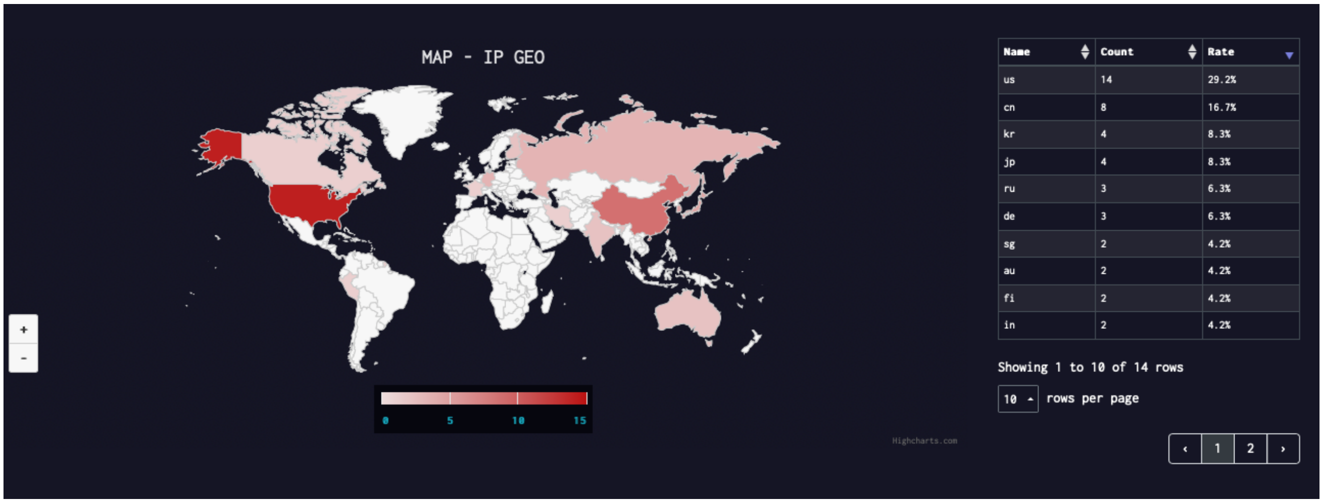
The No.1 weak password starts with `1`, can anyone take a guess what it is?

Panchan

Panchan is a mining P2P botnet developed in Go language, it also uses SSH weak passwords for propagation. Its code contains a lot of Japanese katakana, which suggests that Panchan's developers are fluent in Japanese. Another interesting point is that it implements an interactive console on the listening port, using the idea of protocol reuse, allowing administrators to perform some simple queries and management of the nodes from the network. More detailed information can be found in the following Akamai report.

《Panchan’s Mining Rig: New Golang Peer-to-Peer Botnet Says “Hi!”》

Geographic Distribution



Name	Count	Rate
us	14	29.2%
cn	8	16.7%
kr	4	8.3%
jp	4	8.3%
ru	3	6.3%
de	3	6.3%
sg	2	4.2%
au	2	4.2%
fi	2	4.2%
in	2	4.2%

Showing 1 to 10 of 14 rows

10 rows per page

Conclusion

Normally we end our blog with some conclusion, do we have one here? Not really, we just want to shout out to our readers again: if you have seen some interesting p2p botnet, leave a comment, shoot us an email(**netlab[at]360.cn**) or on **twitter**.