GROUP-IB

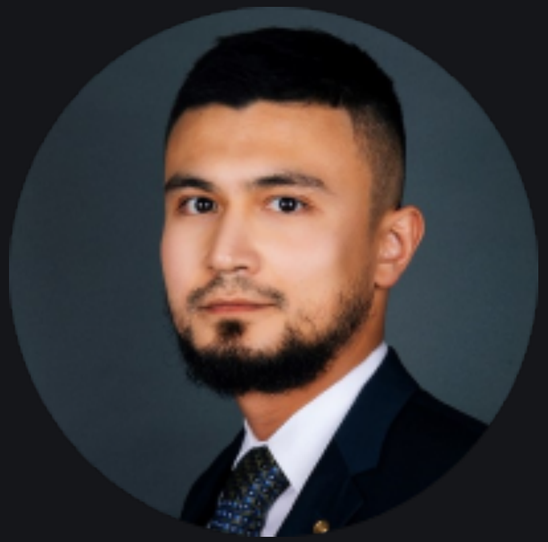
RTM: SINK-HOLING THE BOTNET

WHO WE ARE

Semyon Rogachev

Malware analyst

- 4+ years in malware analysis and incident response
- Strong skills in reverse engineering
- Author and co-author of Group-IB ransomware reports



Rustam Mirkasymov

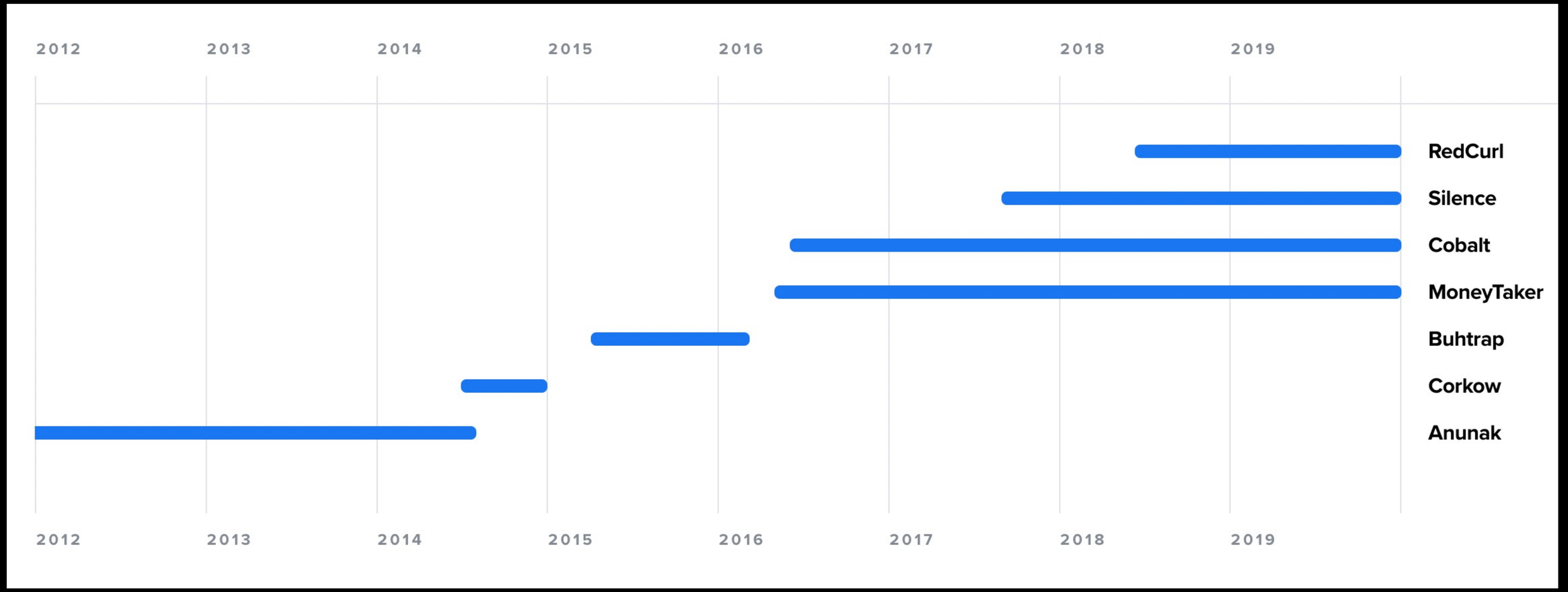
Threat Intelligence analyst

- 8 years in cyber threat research and threat intelligence
- Strong skills in reverse engineering, knowledge in exploit development and understanding software vulnerabilities mechanisms
- Author / co-author of numerous APT threat reports (including Lazarus, Silence, Cobalt, Moneytaker, RedCurl,)

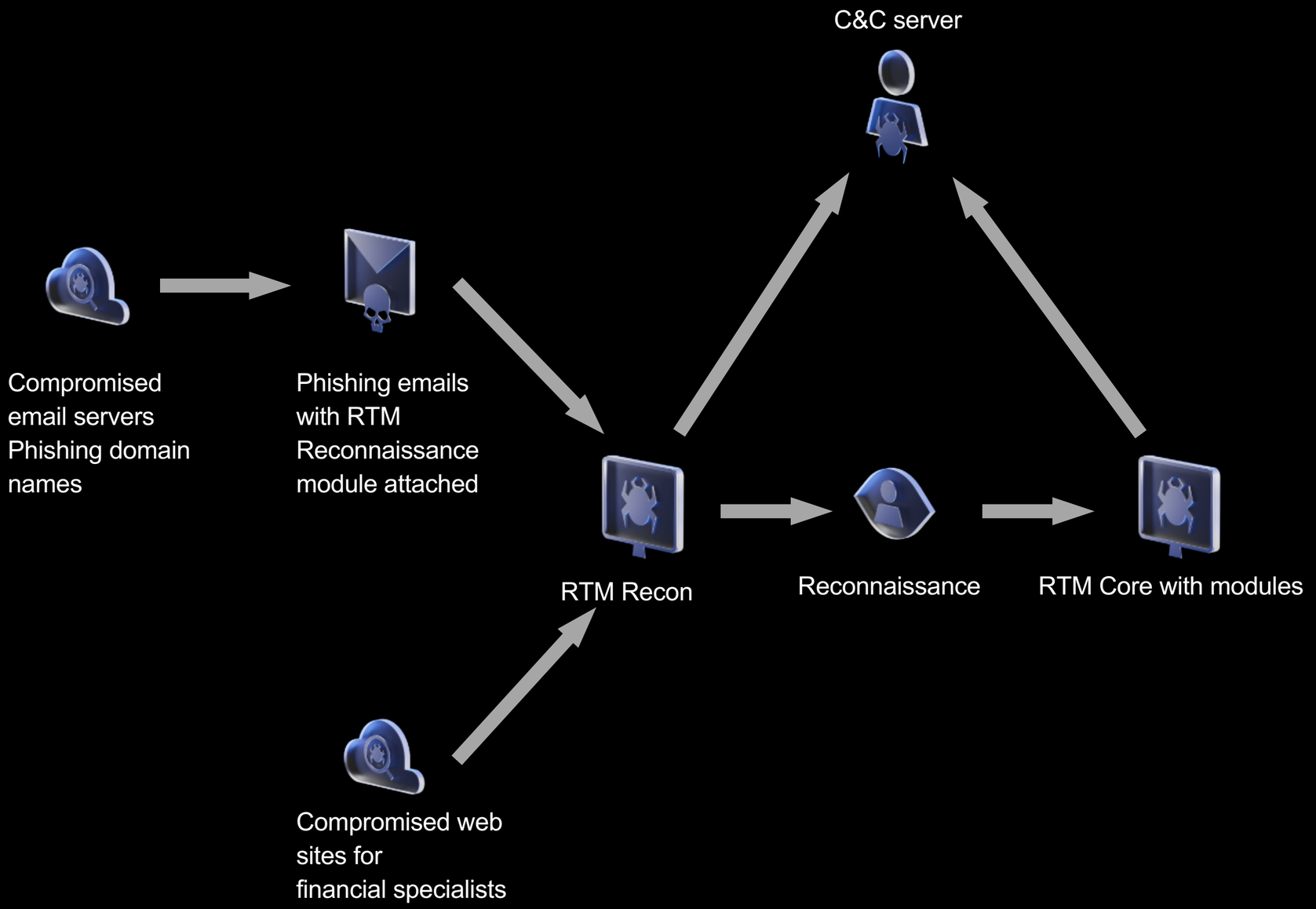
Twitter: @Ta1ien

GROUPS EVOLUTION

Most of banking hacking groups in 2008-2017 were Russian speaking.



RTM ATTACK OVERVIEW



FIRST STAGE – RECONNAISSANCE MODULE

Phishing emails contain a reconnaissance module, which checks if any indicators of financial activity is presented on the infected host.

SECOND STAGE – RTM CORE

If indicators of the financial activity were found, RTM core module is downloaded from the C2 server. In other case some common malware, like Pony Stealer is downloaded.

THIRD STAGE – RTM MODULES

RTM Core collects additional information about the infected host, downloads and executes modules, which are used for the network reconnaissance, lateral movement and money stealing.

FIRST STAGE - RECONNAISSANCE

RTM Reconnaissance module checks browser history to find traces of the following remote banking services:

SBERBANK

sbi.sberbank
online.sberbank
bps-sber

VTB

bco.vtb24.ru
dbo.vtb
bco.vtv.24

ALFABANK

link.alfabank
click.alfabank
ibank.alfa-bank.by

RAIFFEISEN

elba.raiffeisen

elbrus.Raiffeisen

BLOCKCHAIN.INFO

blockchain.info

WESTERNUNION

wupos.westernunion

FIRST STAGE - RECONNAISSANCE

RTM Reconnaissance module checks file system to find the traces of the following financial software:

1C

1cv7.exe
1cv7l.exe
1cv8.exe

SBERBANK

wclnt.exe

FTC GPK CRYPTOPROVIDER

_ftcgpk.exe

WEBMONEY

webmoney.exe

CRYPTO WALLETS

wallet.dat
wallet.dll

QIWI

qiwicashier.exe

SECOND STAGE – CORE MODULE

After reconnaissance module acquired and run RTM Core, it is capable of execution of the following commands:

Command	Description
del-module	Uninstalls module
find-files	Scans filesystem for the specified file
download	Uploads a specified file to the C&C
unload	Closes main window of the RTM
uninstall	Stops all activities and removes itself
uninstall-lock	Erases MBR, removes itself
shutdown	Shutdowns infected host
reboot	Reboots infected host
hosts-add	Adds records to the hosts file

Command	Description
hosts-clear	Removes all records from the hosts file
cfg-set-*	Commands to manipulate RTM settings
screenshot	Creates a screenshot every 5 seconds
dns	Gets/sets DNS servers via the WMI
auto-elevate	UAC bypass
reload	Restarts the RTM
cc	Sets new C&C address
get-cc	Sends the list of the C&C addresses
botnet-id	Sets new bot ID

SECOND STAGE – CORE MODULE

After reconnaissance module acquired and run RTM Core, it is capable of execution of the following commands:

Command	Description
prefix	Sets new bot prefix
connect-interval	Sets pause between C&C communications
dbo-scan	Scan for using banking services
kill-process	Terminates specified process
video-process	Starts a video recording thread
video-stop	Stops a video recording thread
msg	Shows a message box

THIRD STAGE – RTM MODULES

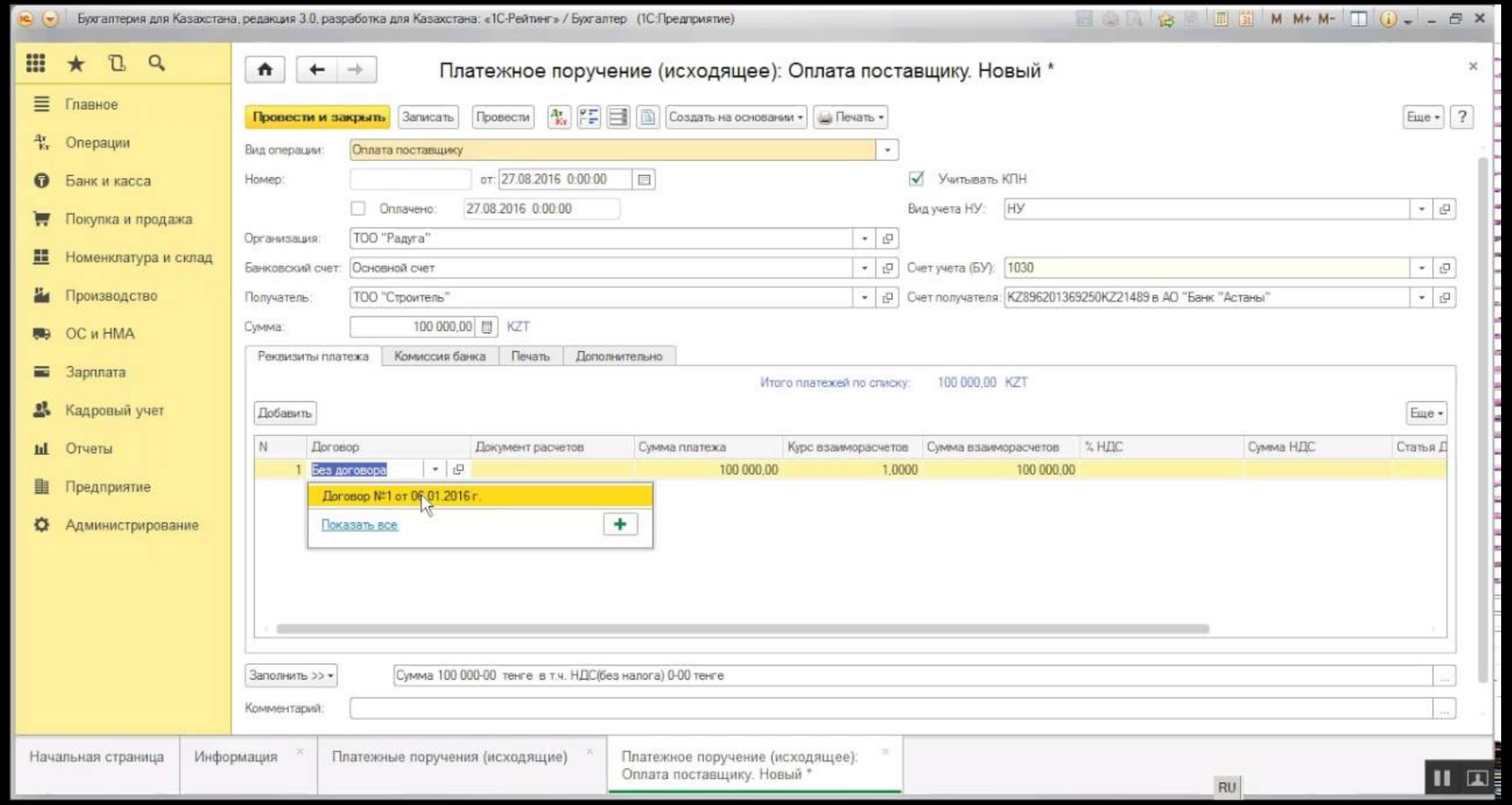
At the third stage additional modules are used. During the tracking of the RTM group, the following modules were detected:

Module name
1c_2_kl
445scan
alfa_scan
anti_mse
arp_scan
bdata
bss_hide
chrome_pw

Module name
chrome_hst
lpewnd
domain
ffa
flash_grab
lock_ie
inj_phone
mitm

Module name
persist
pony
proc_lock
rdp
lpe_evtvwr
mimi
prc_list
stealer

TYPICAL ATTACK SCHEMES



RDP OR TEAMVIEWER

The most frequently seen attack scheme nowadays. Modified version of the TeamViewer is uploaded to the infected host, which allows to transfer money directly, for example, via the browser.

1C_2_KL MODULE

Used to be frequently used method, but almost gone right now. Modifies the 1C banking software process to modify the 1c_2_kl.txt file, which stores the payment data.

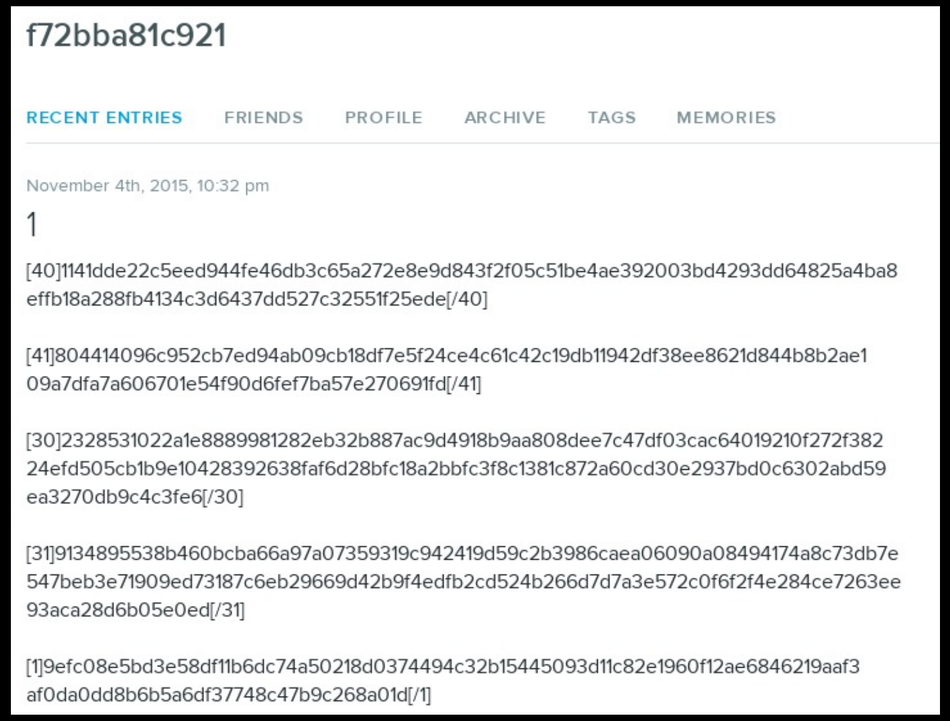
RANSOMWARE

Following the modern trends, RTM is capable of deploying a ransomware. During our monitoring of the RTM activity, at least 4 different were deployed on the compromised machines.

C&C ADDRESS COMPUTATION

LiveJournal «f72bba81c921.livejournal.com»

[<botnet-id>]<encrypted C&C address>[/<botnet-id>]



http://f72bba81c921.livejournal.com/data/rss

LIVEJOURNAL

First tracked versions of the RTM used LIVEJOURNAL blogs to store the C&C addresses.

.BIT DOMAINS

Newer versions of the RTM .bit domains as addresses of the C&C servers. .bit domain IP addresses are stored using the Namecoin technology.

BITCOIN WALLET

The newest versions of the RTM compute C&C server address using the transactions data of the exact Bitcoin wallet.

BITCOIN BASED ALGORITHM

RECON: 1BkeGqpo8M5KNVYXW3obmQt1R58zXAqLBQ

CORE: 1CeLgFDu917tgtunhJZ6BA2YdR559Boy9Y

BLOCKCYPHER API

Blockcypher API is leveraged to obtain an information about the transactions of the exact wallet.

C&C ADDRESS IN TRANSACTION VALUE

After obtaining of the transaction information, RTM extracts the value of the last 2 transactions and interprets it as an IP address octets.

NO ADDITIONAL TRANSACTIONS DATA CHECK

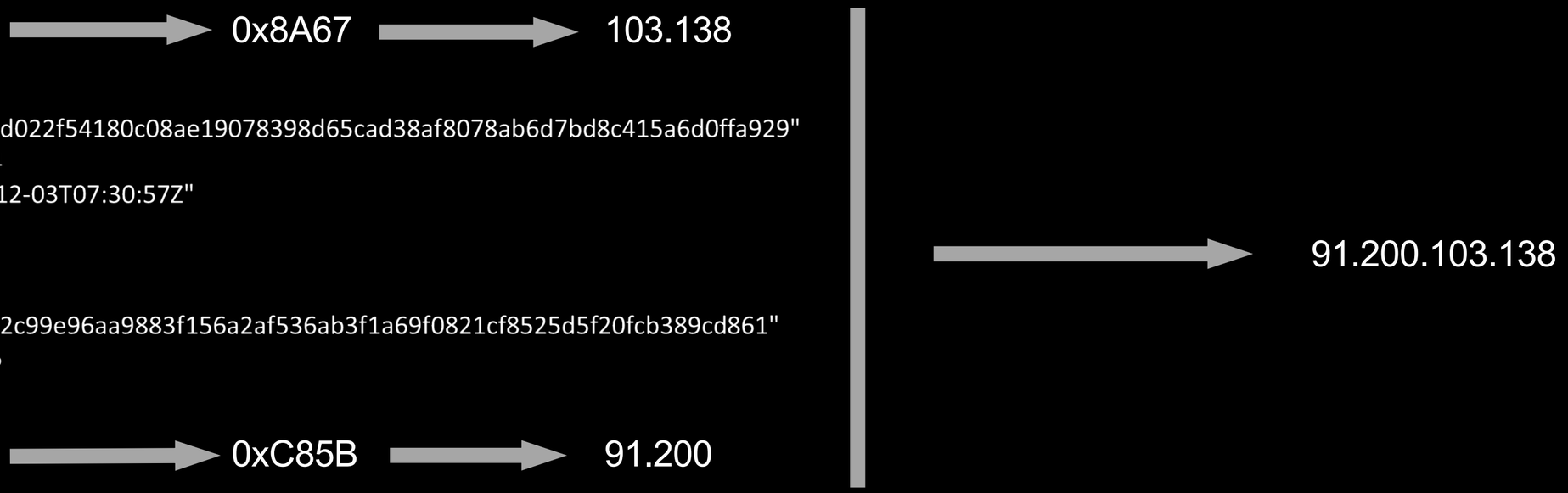
RTM hasn’t been checking which wallet the currency was received from, which made it possible to sinkhole the botnet.

BITCOIN BASED ALGORITHM

Examples for better understanding

```
53
tx_hash          "6bbfa5406e741cee44ac2c0b1ecfcf1f3f5bb1ed3ed3678a8a05df8532399820"
block_height     606417
tx_input_n       -1
tx_output_n      0
value            35431
ref_balance      86722
spent            true
spent_by         "ad881d022f54180c08ae19078398d65cad38af8078ab6d7bd8c415a6d0ffa929"
confirmations    126514
confirmed        "2019-12-03T07:30:57Z"
double_spend     false

54
tx_hash          "d27ae2c99e96aa9883f156a2af536ab3f1a69f0821cf8525d5f20fcb389cd861"
block_height     606415
tx_input_n       -1
tx_output_n      0
value            51291
ref_balance      51291
spent            true
spent_by         "ad881d022f54180c08ae19078398d65cad38af8078ab6d7bd8c415a6d0ffa929"
confirmations    126516
confirmed        "2019-12-03T07:15:26Z"
double_spend     false
```



BITCOIN BASED ALGORITHM

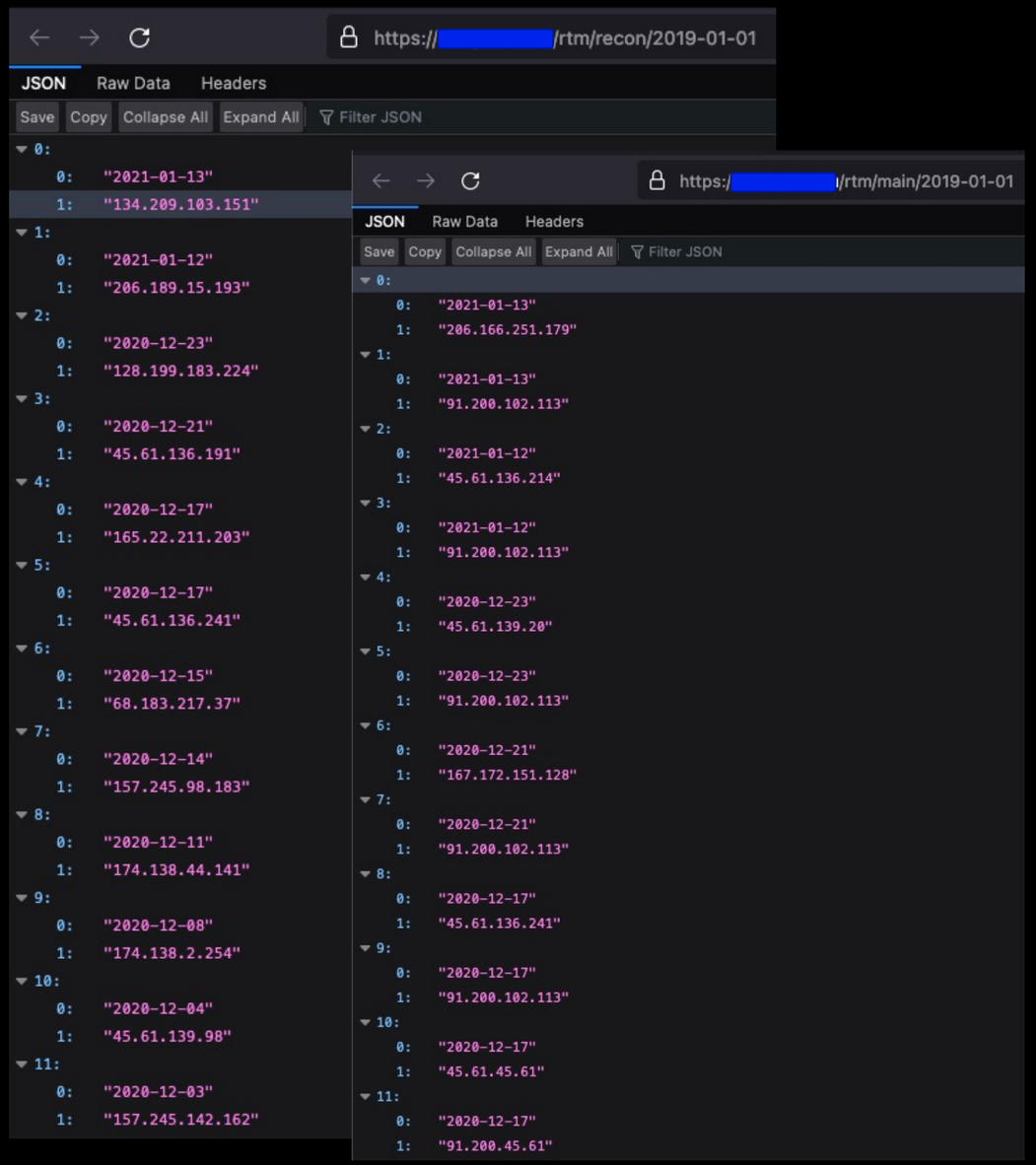
Examples for better understanding: 5th December

Fee	0.00002373 BTC (10.500 sat/B - 2.625 sat/WU - 226 bytes)			+0.00035431 BTC
Hash	8ced96b2dc00df9504830303bdd030e8497c97ffc661a7f90fa245b53819570b			2019-12-05 15:29
	1DmgXb4oESRsvz641nozM9tR76M1qrBffL	0.00413675 BTC	1CeLgFDu917tgtunhJZ6BA2YdR559Boy9Y	0.00035431 BTC
			1GnimRmFztsJKqeQW8PXCUfzA2ZoaGXq9v	0.00375871 BTC
Fee	0.00003300 BTC (14.602 sat/B - 3.650 sat/WU - 226 bytes)			+0.00051291 BTC
Hash	464a01786f783c8aaf9f5f360ee377f3f2f5dfef2261b86cb2c4ac2fe63c0b59			2019-12-05 15:27
	1MPJYJa1knZwVTnmurUnyWZkR7viPNbKjE	0.00468266 BTC	1CeLgFDu917tgtunhJZ6BA2YdR559Boy9Y	0.00051291 BTC
			1DmgXb4oESRsvz641nozM9tR76M1qrBffL	0.00413675 BTC

91.200.103.138

TRACKING C&CS

```
14
15  BLOCKCHAIN_INFO_URL = "https://api.blockcypher.com/v1/btc/main/addrs/"
16
17
18  def blockchain_info_request(blockchain_addr): ...
35
36
37  def get_incoming_transactions_list(transactions_json): ...
43
44
45  def get_transactions_till_specific(transactions_list, specific_tx_hash): ...
52
53
54  def parse_transactions_json_file(jsonfilename): ...
58
59
60  def get_ips_from_bchain(blockchain_addr):
61      return parse_incoming_trans_json_data(blockchain_info_request(blockchain_addr))
62
63
64  def parse_incoming_trans_json_data(json_data):
65      incoming_transactions = get_incoming_transactions_list(json.loads(json_data))
66      ips = []
67      IP_octet_count = 0
68      IP_octet_3 = 0
69      IP_octet_4 = 0
70      IP_octet_1 = 0
71      IP_octet_2 = 0
72      i = 1
73      for tx_idx, trans in enumerate(incoming_transactions):
74          tr_val = trans["value"]
75
76          tr_confirm_time_str = trans["confirmed"]
77          tr_confirm_time = datetime.datetime.strptime(tr_confirm_time_str, '%Y-%m-%dT%H:%M:%SZ')
78          odd_octet = tr_val & 0xff
79          even_octet = (tr_val >> 8) & 0xff
80          ip_half = str(odd_octet) + "." + str(even_octet)
81          # print(str(i) + "\t" + str(hex(tr_val)) + "\t" + ip_half + "\t" + str(tr_confirm_time) + "\t" + trans["tx_hash"])
82          i += 1
83          if IP_octet_count == 0:
84              IP_octet_3 = tr_val & 0xff
85              IP_octet_4 = (tr_val >> 8) & 0xff
86              IP_octet_count = 2
87              continue
88          if IP_octet_count == 2:
89              IP_octet_1 = tr_val & 0xff
90              IP_octet_2 = (tr_val >> 8) & 0xff
91              IP_octet_count = 0
92              #print("RTM IP address:\t`" + str(IP_octet_1) + "." + str(IP_octet_2) + "." + str(
93              #    IP_octet_3) + "." + str(IP_octet_4) + "`\t" + tr_confirm_time_str)
94              ips.append(("{0}.{1}.{2}.{3}".format(IP_octet_1,IP_octet_2,IP_octet_3,IP_octet_4),tr_confirm_time))
95      return ips
```



C&C IS JUST A PROXY

```
access_log /dev/null;
    error_log /dev/null;

server {
        listen          *:80;
bought
        location /index.php {
            proxy_set_header Accept-Encoding "";
            #proxy_set_header       Host      $http_host;
            #proxy_http_version 1.1;
            proxy_buffering off;

            proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header X-Forwarded-For $remote_addr;
```

```
proxy_connect_timeout       600;
            proxy_send_timeout          600;
            proxy_read_timeout          600;
            send_timeout                600;

            proxy_pass http://91.200.103.39/index.php;
          }

        location / { return 404; }

    }

##########################################################
```

SINK-HOLING

All C&Cs are proxies

New C&C is deployed every 2-3 days

No validation of a transaction source

Deploy a server which redirects data to the actual C&C server

Run a sniffer and collect all the traffic

Write a code that decrypts a traffic

Make transactions right after the adversary

Identify victims and notify them

Provide collected data to LE

TESTING ATTEMPT

Address	1CeLgFDu917tgtunhJZ6BA2YdR559Boy9Y
Format	BASE58 (P2PKH)
Transactions	229
Total Received	0.08036732 BTC
Total Sent	0.07801948 BTC
Final Balance	0.00234784 BTC

Address	bc1q0nw2g0dgm6shk0xazmaq2tzmwke7ypsz4upzps
Format	BECH32 (P2WPKH)
Transactions	6
Total Received	0.00100106 BTC
Total Sent	0.00100106 BTC
Final Balance	0.00000000 BTC

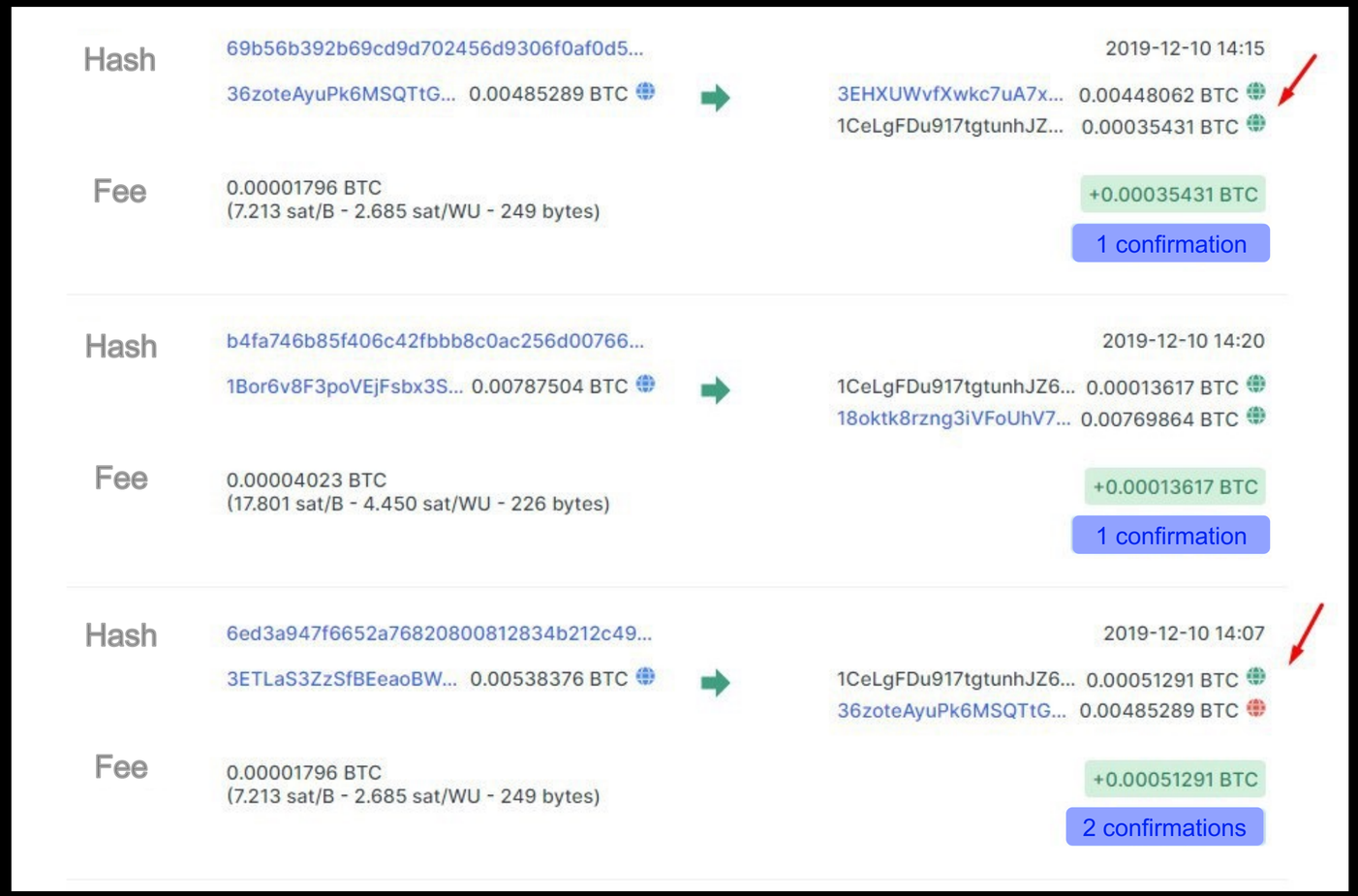
Transaction should be > 0.00000540

Higher fee, faster the confirmation

5.2.67.50

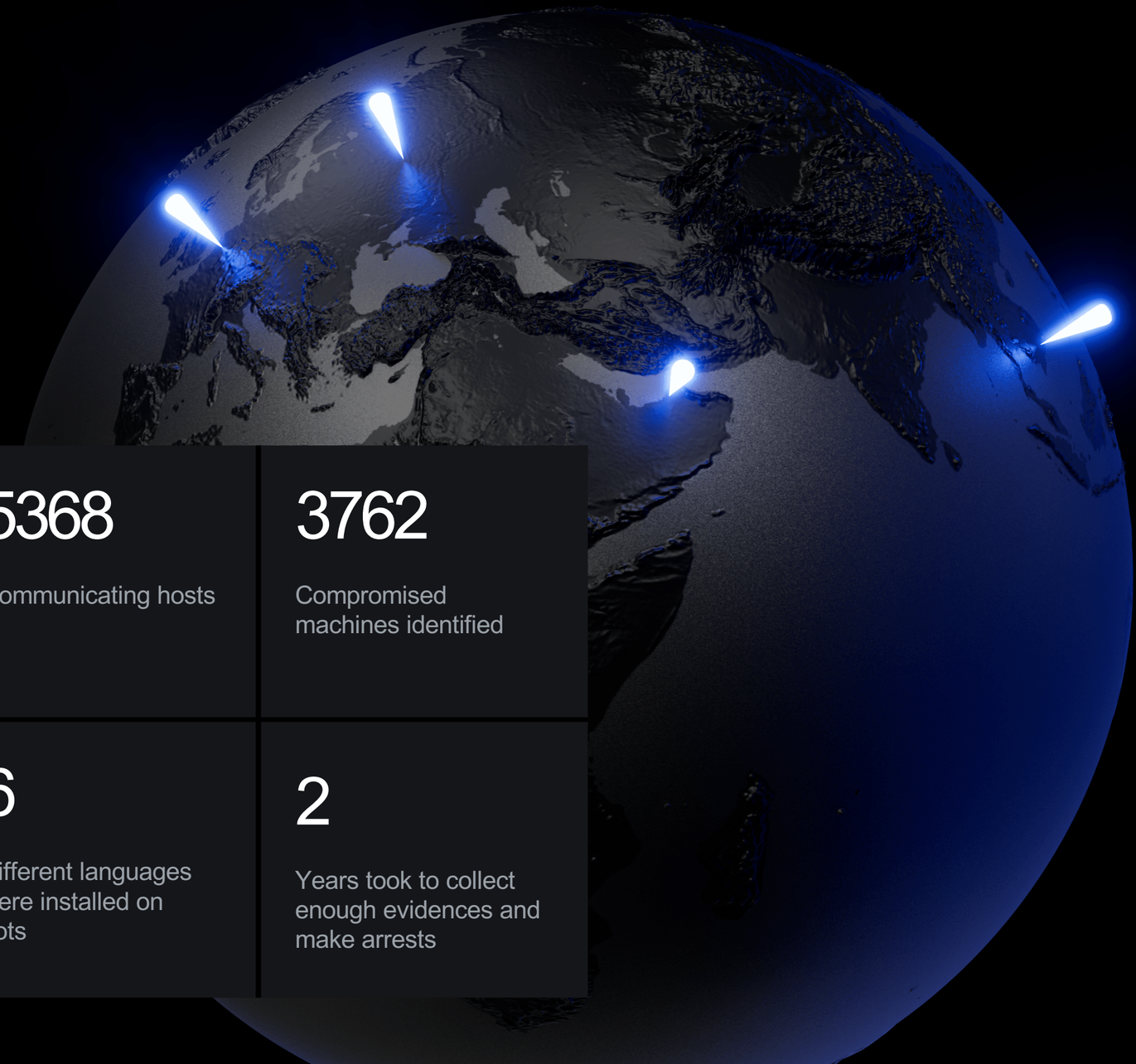
0x205 -> 0.00000517 < 0.00000540

DAY X



- We transferred money with the minimal fee.
- Every transaction should be confirmed by some amount of other members of the blockchain. Otherwise it will be unconfirmed and can not be observed.
- Our transferes came to the wallet in wrong sequence

OVERALL STATISTIC



5 LE of different countries were involved	**5368** Communicating hosts	**3762** Compromised machines identified
5 Years the botnet was alive	**6** Different languages were installed on bots	**2** Years took to collect enough evidences and make arrests

To fight effectively against cybercrime LE agencies should collaborate