Back to the ‘root’ of Incident Response

Boston Park Plaza Hotel | June 22-27, 2014
The art of sinkhooling

Tomasz Bukowski
CERT Polska / NASK
About

WHOIS

• Name: Tomasz Bukowski (tomasz.bukowski@cert.pl)
• Works in CERT Polska/NASK
• 5 years in IRT
• Fight malware && monitor botnets
• Linux user and sysadadmin
• Programmer
• Member of Dragon Sector CTF team 😊
Introduction
So, you want fight botnets?

Botnet lifecycle:
1. write/buy malware
2. write/buy exploit pack
3. buy/hack VPS/hosting for (1) and (2)
4. buy domain for (3)
5. spread malware using exploit pack
6. $ profit $
So, you want fight botnets?

Life of security researcher:
1. monitor spam/social media/internets
2. see malware spreading using exploit pack
3. gather samples
4. monitoring / analysis / incubation
5. locate CnC domains
6. locate rest of infrastructure
7. << action required! >>
Fighting botnets ...

Malware domain takedown:
+ cut off botmaster from his flock of sheep
- devices still infected, no one get noticed

Malware domain takeover:
+ cut off botmaster from his flock of sheep
+ malware will keep talking to CnC
+ can gather and share information on infections!
  → make cyberspace better place
Sinkholing
Sinkholing

Sinkholing – let me google it for you ...

Sinkholing is a technique that researchers use to redirect the identification of the malicious command-and-control (C&C) server to their own analysis server. This way, the malicious traffic that comes from each client goes straight to the research box, ready to be analyzed.

source: the internet
Sinkholing

Scope: global

- Take over CnC domain
  - Point to researcher box (directly or by nameserver)
  - Doable
  - Need to provide evidence
  - Good will from domain operator (TLD)

- Take over CnC IP:
  - Hard to do - need persuade IP owner (ISP/Hosting)

- Take over CnC infrastructure (server)
  - Physically takeover
  - Often can be done only by law enforcements
Sinkholing

Scope: global

- **Take over CnC domain**
  - Point to researcher box (directly or by nameserver)
  - Doable
  - Need to provide evidence
  - Good will from domain operator (TLD)

- **Take over CnC IP**:
  - Hard to do - need persuade IP owner (ISP/Hosting)

- **Take over CnC infrastructure (server)**
  - Physically takeover
  - Often can be done only by law enforcements

a lot of legislation problems
Sinkholing

„Local” sinkholing (LAN) - redirect CnC traffic:
  • By DNS: local DNS redirection
  • By destination IP: traffic redirection

  • Provide useful information on infected workstations
    • Especially when you run multi-layered big internal company network 😊
Sinkholing

DNS perspective

1

IN A

??

domena-1.tld

IN A

1.2.3.4

IN A

IN CNAME

sinkholed

sinkhole.cert.pl
Sinkholing

DNS perspective

1. IN A ?? → domena-1.tld
   IN A 1.2.3.4
   IN A ??
   IN NS

2. IN A ?? → domena-1.tld
   IN A 1.2.3.4
   IN NS
Sinkholing

DNS perspective

1. IN A ?? → domena-1.tld → domena-1.tld
   IN A 1.2.3.4
   IN A sinkhole.cert.pl

2. IN A ?? → domena-1.tld → domena-1.tld
   IN A 1.2.3.4
   IN A sinkhole.cert.pl

3. IN A ?? → domena-1.tld → domena-2.tld → sinkhole.cert.pl
   IN A 1.2.3.4

accidently sinkholed
Sinkholing

„the goal”

- Allow malware to connect to your box
- Keep malware connected to your sinkhole as long as possible
- Prevent malware from using alternative/bacup communication channels
CnC Types

- CnC
  - TCP
  - UDP
CnC Types

- TCP
- UDP
- DNS
- other...
CnC Types

- TCP
- UDP
- DNS
- other ...
- TXT record
- IP calculation
- tunelling ?
- p2p
CnC Types

- TCP
- +SSL?
- fancy encryption FTW
- p2p
- other ...
- DNS

CnC

The art of sinkholing :: Tomasz Bukowski :: June 2014
CnC Types

- TCP
- +SSL?
- fancy encryption FTW
- HTTP
- IRC
- 90+ % ?
- UDP
- other ...
- DNS
- p2p

CnC
CnC Types

- TCP
- UDP
- DNS
- +SSL?
- HTTP
- IRC
- other ...
- fancy encryption FTW
- p2p
- other ...
- CnC
CnC Types

- TCP
  - +SSL?
  - fancy encryption FTW
  - HTTP
  - IRC
  - other ...
- UDP
- DNS
- p2p

POST/GET params encrypted
response content encrypted
Extra content (i.e. jpg file)
Hidden in „legit” content
CnC Types

- CnC
  - TCP
    - +SSL?
    - HTTP
    - IRC
    - other ...
    - fancy encryption FTW
  - UDP
  - DNS
  - p2p

POST/GET params encrypted
response content encrypted
Extra content (i.e. jpg file)
Hidden in „legit”content

- Facebook
- Twitter
- <!-- ? -->
CERT .PL

story (1)
Timeline

end of 2012 – dorkbot

• Yet another malware using .pl domain as CnC
• Yet did not have TLD sinkhole procedure (in progress)
• Registrar decided to help (after abuse report)
• Am... but we do not have sinkhole !?
CnC – example: Dorkbot

TCP → SSL → IRC → IRC server
Timeline

end of 2012 – dorkbot

- Take this old unused server and do somethink
  (4 GB RAM, 2x 3.0 Ghz CPU, 160 GB HDD, decend 1U !)

- We need TLS IRC
  - Take charybdis irc server, remove 80 % functions
Timeline

end of 2012 – dorkbot

begin of 2013 – virut

- Really long-living malware still sitting on .pl domains 😞
- TLD sinkhole procedure in progress
- Promising results from sinkholing dorkbot 😊
- Decision: we need to do this!
Timeline

end of 2012 – dorkbot
begin of 2013 – virut

• We already got hardware (+)
• We need (a lot) more software
CnC – example: Virut
(expectations)

TCP
IRC (crippled)
IRC server
TCP
ENCRYPT
IRC (crippled)
IRC server

random key

Not working?
CnC – example: Virut - reality

(reality)

TCP PORT 80

(GET|POST)_

HTTP

PLAIN

ENCRIPT

(crippled)

traffic

brute the key

random key

else?

IRC

random key

BOSTON 26th annual FIRST conference
Timeline

end of 2012 – dorkbot
begin of 2013 – virut

• Write python script
  ➢ peek first 5 bytes (decision: irc/http/crypted)
  ➢ keep TCP connection as long as possible
Timeline

end of 2012 – dorkbot
begin of 2013 – virut

200 K connections in „established” state !?
Timeline

Encountered problems: TCP timeouts

• close timeout = 10s
• close-wait timeout = 60s
• established timeout = 5 days
• fin-wait timeout = 120s
• last-ack timeout = 30s
• syn-received timeout = 60s
• syn-sent timeout = 120s
• time-wait timeout = 120s

srsly! It is just waiting for RST
Timeline

Encountered problems: software
(you know them when you hit the limit 😞)

Somewhere in code you need to „select()” over opened file descriptors. It uses limited size bit-fields!

Hint: on Linux use poll!
Timeline

Encountered problems: default OS limits
(you know them when you hit the limit 😞)

• max opened file descriptors (each tcp connection=new FD)
  ➢ can be easily fixed: `ulimit -n 999999 😊`

• max entries in contract table
  ➢ requires kernel param tweak, fixable 😊
Timeline

Conclusion (1)

Establishing TCP connection and leaving it with default settings is **bad idea**!

Use SO_KEEPALIVE socket option 😊

(organic ?)
Timeline

Conclusion (2)

`SELECT()`

`POLL()`
Timeline

end of 2012 – dorkbot
begin of 2013 – virut

30K simultaneous connections 😊
Timeline

end of 2012 – dorkbot
begin of 2013 – virut
spring 2013 – few ZueS domains

• Write python script that will understand HTTP and decode incoming zeus data ...
CnC – example: ZeuS

TCP -> SSL -> HTTP -> POST /gate.php

- RC4 visual-encrypt
- binstorage pack
- records

key

- RC4 visual-encrypt
- binstorage extract
- records

CNC
Timeline

d end of 2012 – dorkbot
begin of 2013 – virut
spring 2013 – few ZueS domains
summer 2013 – domainsilver takedown

• A LOT of various malware domains

• Write python scripts .... .... ?
Encountered problems:
we already got numerous different python scripts running different CnC
Conclusion (3)

Need decent sinkhole software 😊
(obvious ?)
CERT.PL
sink-soft 😊
Requirements

- Build consistent framework for sinkholing
- Make event logging/sharing easy
- Identify common content processing functions
- Handle undret of TCP connections from
- Be as elastic as possible
- Implement any fancy encryption/encoding anywhere

Allow to sinkhole new malware with lowest possible effort
Design

Build your sinkhole out of blocks (modules) 😊

TCP → SSL → HTTP → POST /gate.php

RC4 → visual-encrypt → binstorage pack → records

RC4 → visual-encrypt → binstorage extract → records

CNC
CnC Types

- TCP
- +SSL?
- HTTP
- IRC
- POST/GET params encrypted
- response content encrypted
- fancy encryption FTW
- visual-encrypt
- b64
- rot13
- XOR
- gzip
- RC4
- AES
- ....

CnC

The art of sinkholing :: Tomasz Bukowski :: June 2014
Design

- Provide TCP connectivity layer
- Use MQ – ZeroMQ (fast && simple)
- PUB-SUB messaging pattern
- Deployed as standalone package with lowest possible requirements (msgpack && zmq-python)
- Easy configuration (chose ip,port and modules chain)
- unpack & config & run
Sink-soft
Sinkholing > 200 active malware domains

root@sinkhole:~# w
14:20:38 up 561 days, 19:05, 4 users, load average: 0.00, 0.00, 0.00
Sink-soft

demo :)

CERT.PL
Fin.

Tomasz Bukowski
tomasz.bukowski@cert.pl