ZeuS meets VM

Botconf 2014

CERT.PL

Nancy
@maciekkotowicz

~~ZeuS meets VM~~

Botconf 2014

CERT.PL

Nancy
@maciekkotowicz

Zeusology

Botconf 2014

CERT.PL

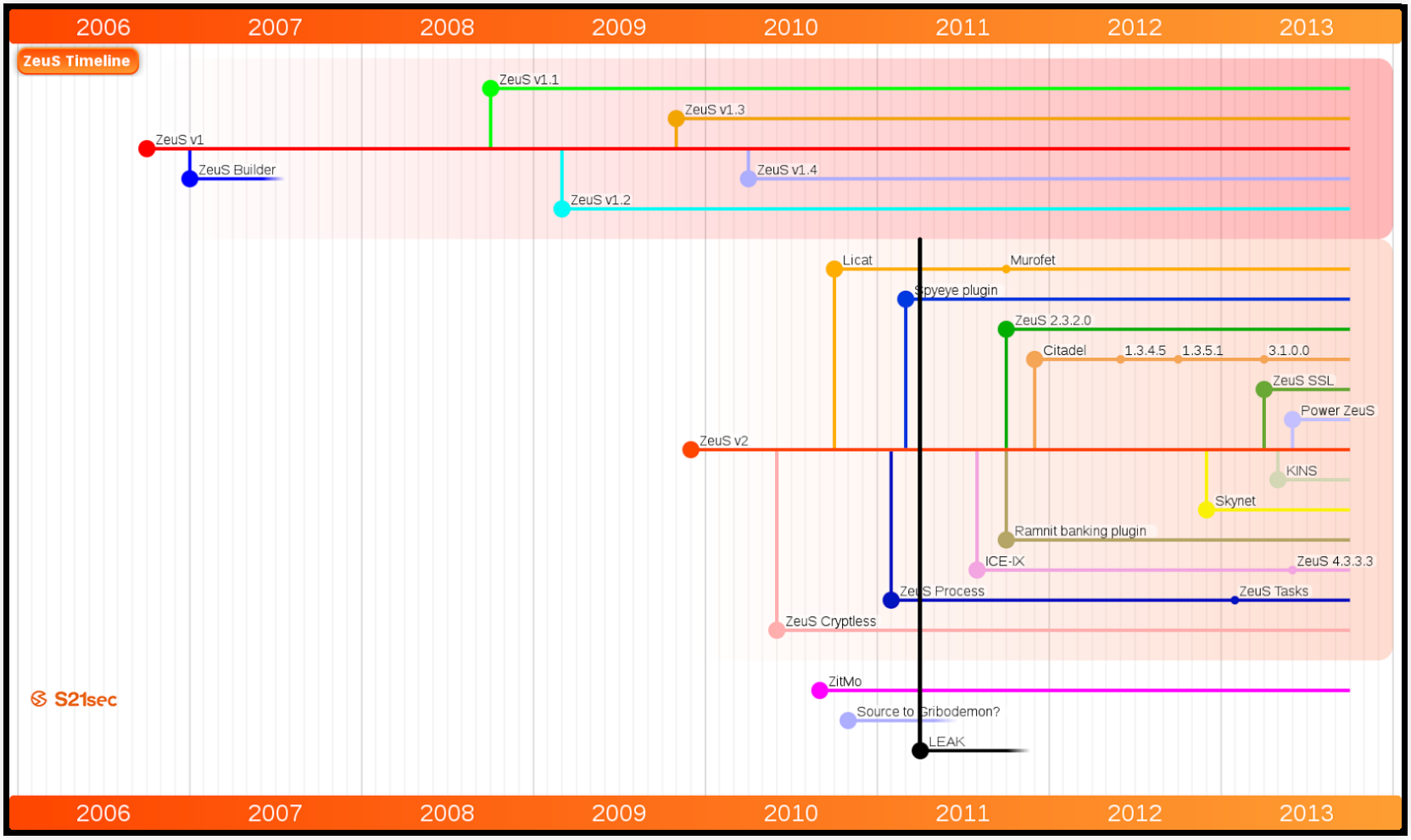
Nancy
@maciekkotowicz

$ whoami

- ~~student @ ii~~
- ~~tutor @ ii~~
- IRT@CERT.pl
- DragonSector CTF
- RE/Exploit dev
- Formal methods


DISCLAIMER

Happy family



(from **http://securityblog.s21sec.com/**)

ZeuS v2

- Socks proxy
- Grabber / Stealer
- Semi-hidden dirs
- RtlDecompressBuffer
- RC4 + VisualCrypt

- Anti-Tracking
- RC4 + shift + VisualCrypt

Citadel

- Video mode
- Keylogger
- RC4/AES + salt + login key + VisualCrypt

PowerZeus

- PowerLoader
- ZeuS as module
- Spyeye style

- Shell_TrayWnd inj
- Stelthy dirs
- Custom VM + AES/RC4

VMZeuS / ZeuSVM

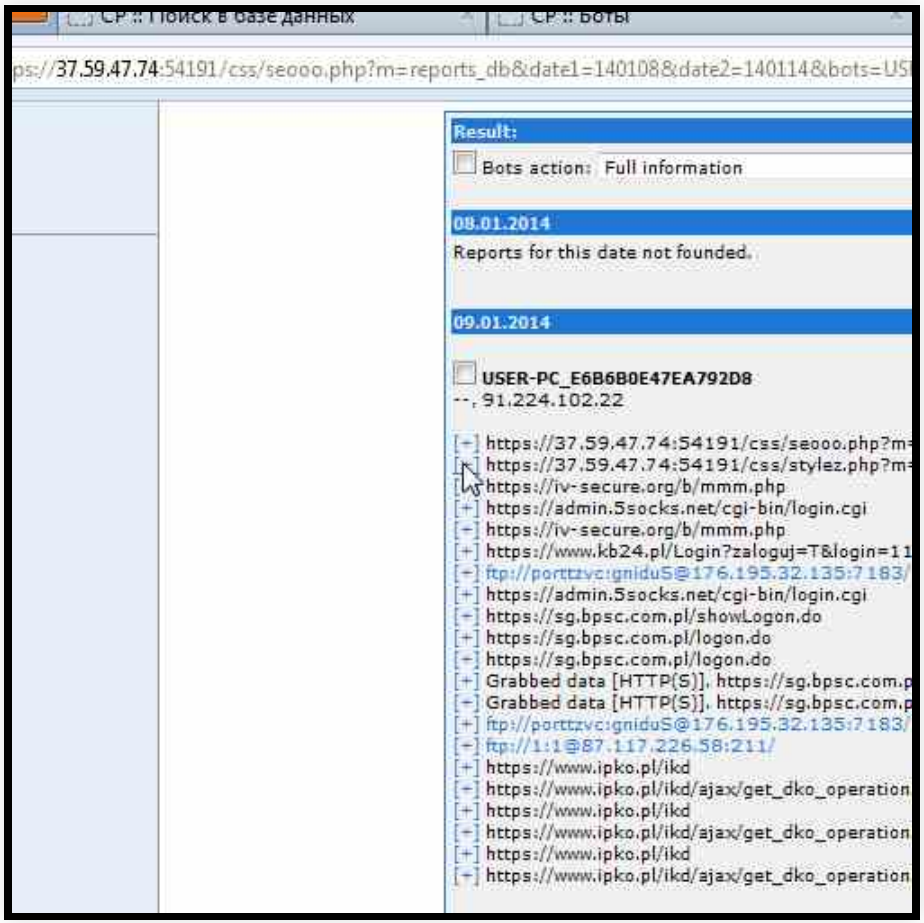
- pseudo-DGA
- vnc module
- steganography / base64 + jpg-comment
- Custom VM + RC4

strings -100 | base64 -d

```
0000000: 416a 4aa4 229e 4f76 c2b1 c10c a33a 6f58  AjJ.".Ov.....:oX
0000010: f3f9 ca64 546e fe59 f7f9 cf1f 5e4e 9c69  ...dTn.Y....^N.i
0000020: 2a29 85cc 7cb4 7b40 62f3 4a83 0619 ed4f  *)..|.{@b.J....O
0000030: 405d df5f f9a5 5824 052b dd22 f46f 0c70  @]._..X$.+.".o.p
0000040: 89c1 4086 cf0c f6ce f505 3736 47a9 f664  ..@.......76G..d
0000050: b5ad 4356 6c3e 0b60 2de0 34f8 0847 1bbe  ..CVl>.`-.4..G..
0000060: 2195 4bca 2e4e 4dd0 5b11 d518 41af be07  !.K..NM.[...A...
0000070: 85bb c838 8a5d aae2 6768 c9bb 077d ccc3  ...8.]..gh...}..
0000080: fcb1 759f 0395 a7a0 88bc c67f 5fcb 4bc4  ..u........._.K.
0000090: 2212 7bae bcab 0930 5ebf 77c9 095e e514  ".{....0^.w
```

strings -100 | base64 -d | decode-parse

```
VMZeus: 01.00.00.02

{{SERVER_URLS}}
https://connauzzzzzuuurt308.com/adms/gate.php
{{END_SERVER_URLS}}
{{ADV_SERVER_URLS}}
https://carbonerast500.com/adms/config.jpg

{{ADV_END_SERVER_URLS}}
{{NOTIFY_SERVERS}}
https://localhost/notifygate.php
{{END_NOTIFY_SERVERS}}
{{CAPTCHA_SERVERS}}
https://localhost/captchaupload.php
{{END_CAPTCHA_SERVERS}}
{{CAPTCHA_LIST}}
URL:  MASK:
{{END_CAPTCHA_LIST}}
{{WEBFILTERS}}
{{DONT-REPORT}} *.microsoft.com/*
{{DONT-REPORT}} http://*
{{SCREENSHOT}} */login.osmp.ru/*
{{SCREENSHOT}} */atl.osmp.ru/*
{{NOTIFY}} http://www.apple.com/mac/
{{NOTIFY}} http://digg.com/news*
{{END_WEBFILTERS}}
```


CP :: Поиск в базе данных
CP :: Боты
ps://37.59.47.74:54191/css/seooo.php?m=reports_db&date1=140108&date2=140114&bots=US
Result:
Bots action: Full information
08.01.2014
Reports for this date not founded.
09.01.2014
USER-PC_E6B6B0E47EA792D8
--, 91.224.102.22
[+] https://37.59.47.74:54191/css/seooo.php?m=
[+] https://37.59.47.74:54191/css/stylez.php?m=
[+] https://iv-secure.org/b/mmm.php
[+] https://admin.5socks.net/cgi-bin/login.cgi
[+] https://iv-secure.org/b/mmm.php
[+] https://www.kb24.pl/Login?zaloguj=T&login=11
[+] ftp://porttzvc:gniduS@176.195.32.135:7183/
[+] https://admin.5socks.net/cgi-bin/login.cgi
[+] https://sg.bpsc.com.pl/showLogon.do
[+] https://sg.bpsc.com.pl/logon.do
[+] https://sg.bpsc.com.pl/logon.do
[+] Grabbed data [HTTP(S)], https://sg.bpsc.com.p
[+] Grabbed data [HTTP(S)], https://sg.bpsc.com.p
[+] ftp://porttzvc:gniduS@176.195.32.135:7183/
[+] ftp://1:1@87.117.226.58:211/
[+] https://www.ipko.pl/ikd
[+] https://www.ipko.pl/ikd/ajax/get_dko_operation
[+] https://www.ipko.pl/ikd
[+] https://www.ipko.pl/ikd/ajax/get_dko_operation
[+] https://www.ipko.pl/ikd
[+] https://www.ipko.pl/ikd/ajax/get_dko_operation

Other S**t

- MMBB
- evo
- Zeus_1134 / Header
- Tasks
- Skynet
- etc...

Important families

(via zeus-tracker)

- Zeus
- Citadel
- ICEX
- KINS / (VMZeus acctualy)

How they differ?

- Injection methods
- Code organisation
- **CRYPTO**

ZeuS configuration

- Local / PESettings
- BaseConfig
- DynamicConfig

PESettings

```
typedef struct
{
  DWORD size;
  WCHAR compId[60];
  GUID guid;
  Crypt::RC4KEY rc4Key;
  struct
  {
    char coreFile[20];
    char reportFile[20];
    char regKey[CORE_REGISTRY_VALUE_BUFFER_SIZE];
    char regDynamicConfig[CORE_REGISTRY_VALUE_BUFFER_SIZE];
    char regLocalConfig[CORE_REGISTRY_VALUE_BUFFER_SIZE];
    char regLocalSettings[CORE_REGISTRY_VALUE_BUFFER_SIZE];
  }userPaths;

  DWORD processInfecionId;
  DWORD storageArrayKey;
}PESETTINGS;
```

BaseConfig

```
typedef struct
{
  BYTE  padding0[80];
  DWORD flags;
  BYTE  padding1[58];
  DWORD delayStats;
  BYTE  padding2[91];
  char  defaultConfig[100 + 1];
  BYTE  padding3[7];
  Crypt::RC4KEY baseKey;
  BYTE  padding4[77];
  WCHAR defaultBotnet[BOTNET_MAX_CHARS + 1];
  BYTE  padding5[38];
  DWORD delayReport;
  BYTE  padding6[14];
  DWORD delayConfig;
  BYTE  padding7[21];
}BASECONFIG;
```

DynamicConfig

```
typedef struct {
    CHAR junk[20];
    DWORD size;
    DWORD flags;
    DWORD count;
    CHAR md5[0x10];
} BinStruct_HEADER;


typedef struct {
    DWORD id;
    DWORD flags;
    DWORD size;
    DWORD realSize;
    if(realSize == 4) {
        DWORD data;
    } else {
        CHAR data[size];
    }
} BinStruct_ITEM;
```

Begin of ZBOT

```
void WINAPI Core::_entryPoint(void)
{
  bool ok = false;

  if(init(INITF_NORMAL_START))
  {
    ...
#   if(BO_DEBUG == 0)
    Core::disableErrorMessages();
#   endif
    {
      int argsCount;
      LPWSTR *args = CWA(shell32, CommandLineToArgvW)(CWA(kernel32, GetComma
```

```
DWORD Core::disableErrorMessages(void)
{
  return CWA(kernel32, SetErrorMode)(SEM_FAILCRITICALERRORS | SEM_NOALIGNMEN

}
```

Unpacked

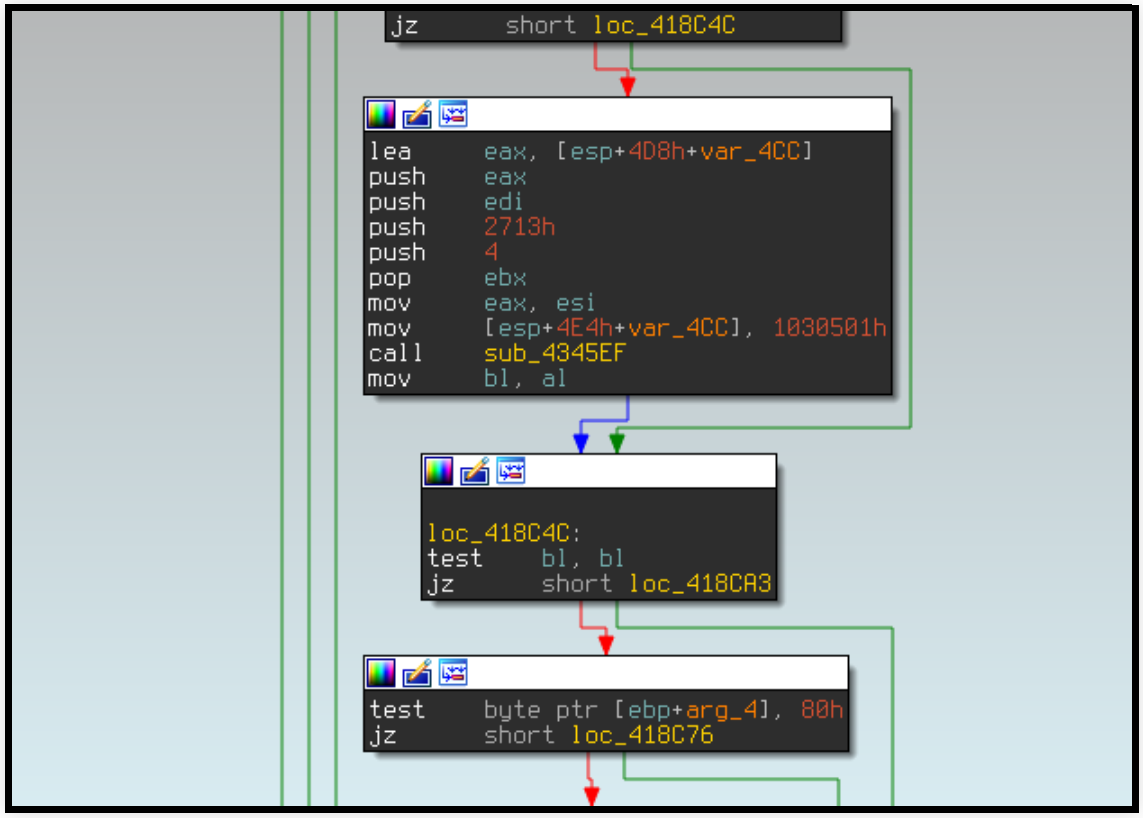
```
class EntryPointBP(vtrace.Breakpoint):
   APIS = ['kernel32.SetErrorMode','kernel32.GetCommandLineW']

   def notify(self,ev,tr):
        ret = getRET(tr)
        _,_,perm,name = tr.getMemoryMap(ret)
        print '[*] SetError mode - ret %x (%s)' % (ret,name)

        dis = filter(lambda c: c.mnem == 'call' and c.opers[0].isDeref(),dis
        if len(dis) == 3 and all(map(lambda x: readDword(tr,x[0].opers[0].im
            _map=tr.getMemoryMap(ret)
            print _map
            hsh = '-'
            if _map[-1]:
                hsh=get_hash(_map[-1])

            print '[+] Unpacked - SetErrorMode in entrypoint @ %X (%s) - %s
            code = tr.readMemory(_map[0],_map[1])
            hh=md5(code).hexdigest()+'.bin'
            with open(hh,'wb') as f:f.write(code)
            print '[+] Saved unpacked payload to %s' % hh
            return run(0,tr)

        if perm == perm_lookup[PAGE_EXECUTE_READWRITE]:
            print '[+] Unpacked(RWX) - SetErrorMode in entrypoint @ %X (%s)
            return run(0,tr)

        tr.runAgain()
```

BinStruct - heart of bot

```
$_COMMON_DEFINE['SBCID_BOT_ID']                 = '10001'; //ID Bot.
$_COMMON_DEFINE['SBCID_BOTNET']                 = '10002'; //Botnet.
$_COMMON_DEFINE['SBCID_BOT_VERSION']            = '10003'; //Version.
$_COMMON_DEFINE['SBCID_NET_LATENCY']            = '10005'; //Lag network.
$_COMMON_DEFINE['SBCID_TCPPORT_S1']             = '10006'; //Port for Socks.
$_COMMON_DEFINE['SBCID_PATH_SOURCE']            = '10007'; //Source file path
$_COMMON_DEFINE['SBCID_PATH_DEST']              = '10008'; //The final path o
```

```
$_COMMON_DEFINE['CFGID_LAST_VERSION']           = '20001'; //The latest avail
$_COMMON_DEFINE['CFGID_LAST_VERSION_URL']       = '20002'; //URL, where you c
$_COMMON_DEFINE['CFGID_URL_SERVER_0']           = '20003'; //URL server-side
$_COMMON_DEFINE['CFGID_URL_ADV_SERVERS']        = '20004'; //Multi-line backu
$_COMMON_DEFINE['CFGID_HTTP_FILTER']            = '20005'; //Multi-line filte
$_COMMON_DEFINE['CFGID_HTTP_POSTDATA_FILTER']   = '20006'; //Multi-line for p
$_COMMON_DEFINE['CFGID_HTTP_INJECTS_LIST']      = '20007'; //List HTTP-inzhek
```

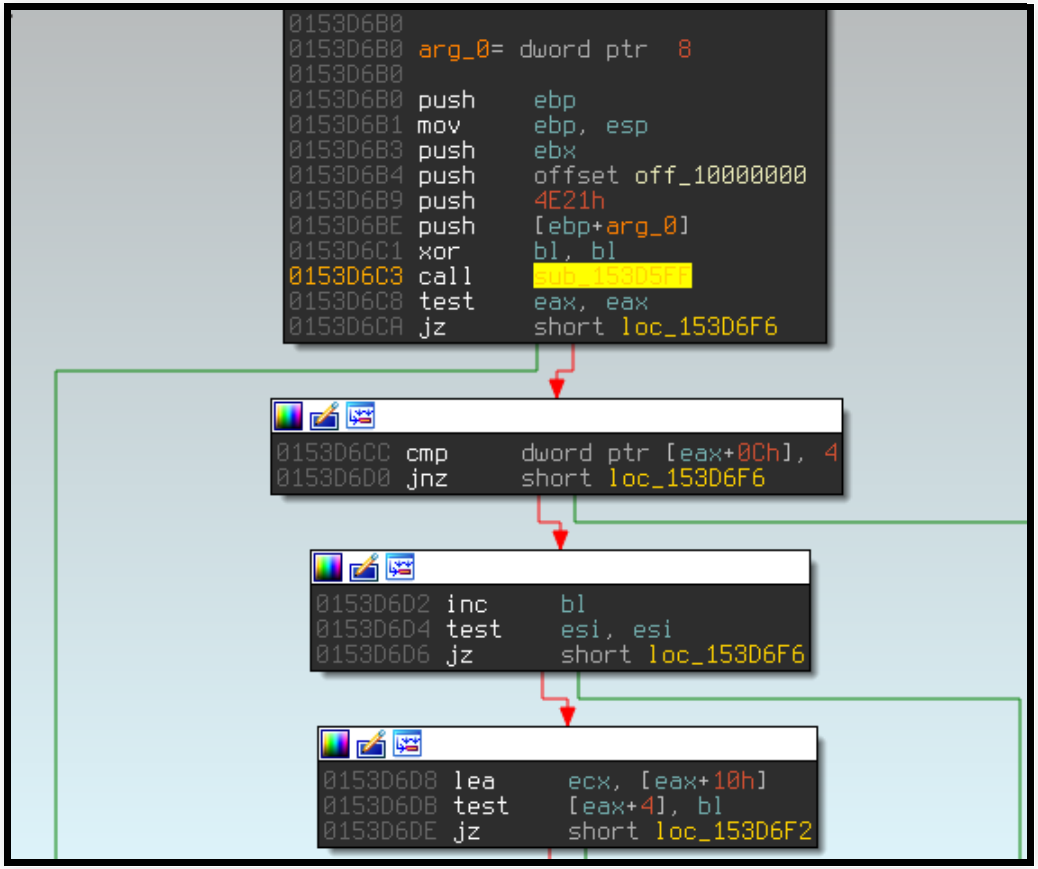
ZBOT - version



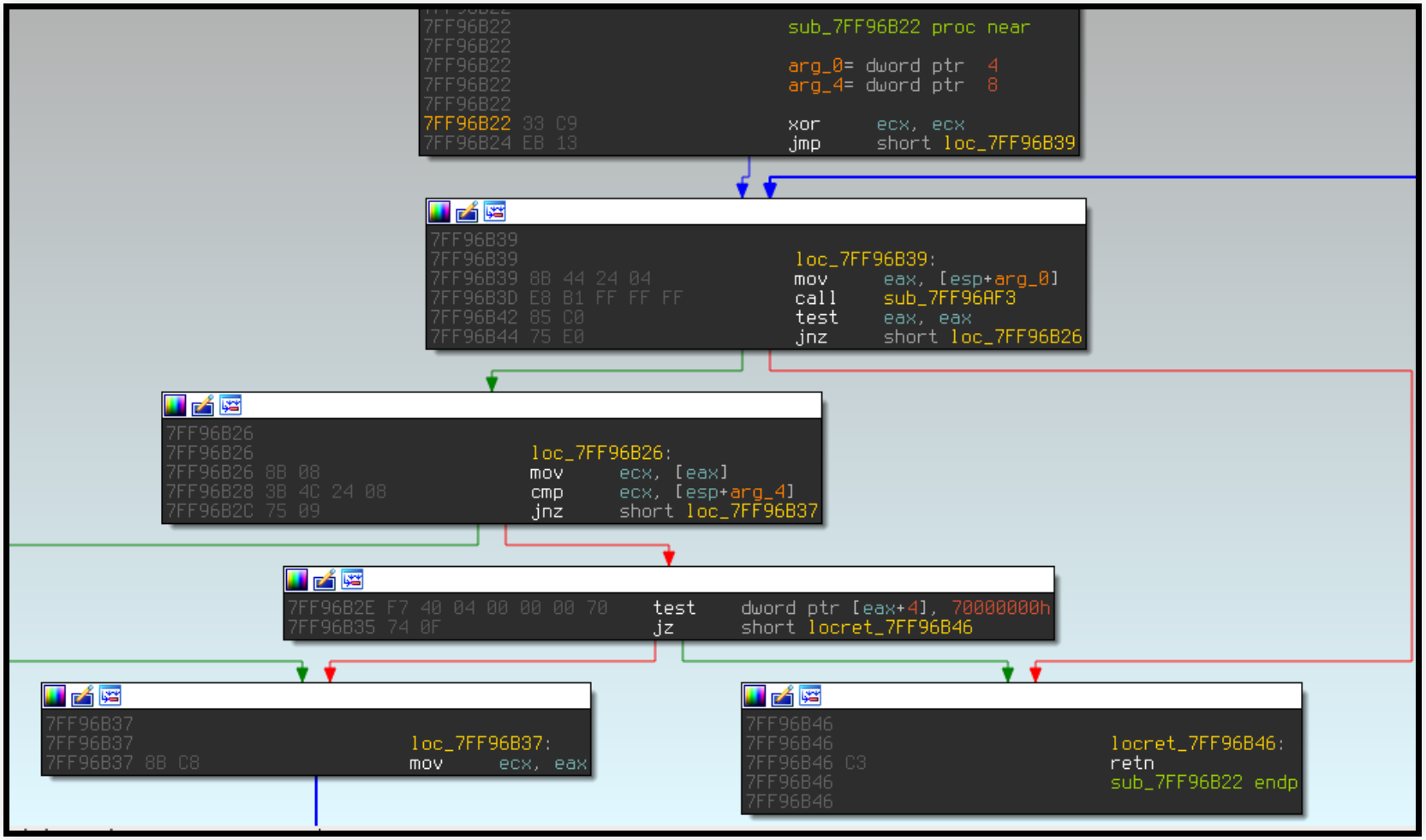
ZBOT - call home

- c&c from baseconfig
- c&c from dynamic config

ZBOT - get dynamic cfg



ZBOT - get dynamic cfg



ZBOT - get dynamic cfg

```
for a,l,p,n in T.getMemoryMaps():
    if p == perm_lookup[PAGE_EXECUTE_READWRITE]:
       print '[*]Found RWX segment @ %X - %X (%s)' % (a,a+l,n if n else '-
      ## dynamic config
       for hit in T.searchMemoryRange("\x68\x21\x4e\x00\x00",a,l):
           print "[*].Found `push 20001` @ %X"  % hit
           Find_GetItem(T,hit)
       for hit in T.searchMemoryRange("\x68\x13\x27\x00\x00",a,l):
           ver = get_version(hit)
           print '[*].Found Version: %X' % ver
           CFG['version']=ver
```

ZBOT - get dynamic config

```
def analCode(t,addr):
    code = disasm(t,addr,64,is_function=True)
    ncorrect = True
    for c in code:
        if c.mnem in ('and','test') and c.getOperValue(1) == 0x70000000:
            ncorrect = False
            break
    if ncorrect:
        for c in code:
            if c.mnem == 'call':
                return analCode(t,c.getOperValue(0))
    return addr

def Find_GetItem(t,hit):
    global BPADDRESSES,HITS

    for op in disasm(t,hit,20):
        if op.mnem == 'call':
            a  = analCode(t,op.getOperValue(0))
            if a not in BPADDRESSES:
                print '[*]..Found BinStorage::_getItem @ %X ' %  a
                bp = CfgDumpBP(a)
                t.addBreakpoint(bp)
                BPADDRESSES.add(a)
```

ZBOT - get dynamic config

```
class CfgDumpBP(vtrace.Breakpoint):
    def notify(self,ev,tr):
        ptr = unpack('I',tr.readMemory(tr.getRegisterByName('esp')+4,4))[0]
        print '[*]...BinStorage @ %X ' % ptr
        self.dumpSections(tr,ptr)

    def dumpSections(self,tr,addr):
        global CFG
        with open('c:\\xcfg','wb') as f:
            hdr = tr.readMemory(addr,0x30)
            f.write(hdr); f.flush()
            count = unpack('I',tr.readMemory(addr + 20 + 8,4))[0]
            addr += 0x30; off  = 0x30
            CFG['cfg'] = hdr
            print "attempt to dump %d sections" % count
            for i in range(count):
                size = unpack('I',tr.readMemory(addr+8,4))[0]
                print "Section size: %d - off: %x" % (size,off)
                sec = tr.readMemory(addr,0x10 + size)
                CFG['cfg'] += sec
                f.write(sec);f.flush()
                addr += 0x10 + size
                off += 0x10 + size
            CFG['cfg'] = base64.b64encode(CFG['cfg'])
            self.exit_dbg(tr)
```

ZBOT - parse BinStruct

libzpy

- python
- modular
- more-or-less complete;]

libzpy

```
├── bins.py
├── fmt
│   ├── citadel.py
│   ├── kins.py
│   ├── powerzeus.py
│   ├── vmzeus20.py
│   └── zeus.py
├── libs
│   ├── basecfg.py
│   ├── crypto.py
│   ├── kdNRV2b.py
│   ├── libucl.so
│   ├── structure.py
│   └── UCL.py
├── modules
│   ├── citadel.py
│   ├── kins.py
│   ├── powerzeus.py
│   ├── template.py
│   ├── vmzeus.py
│   └── zeus.py
├── README
└── structs
    ├── citadel.py
    ├── kins.py
    ├── powerzeus.py
    ├── vmzeus20.py
    └── zeus.py
```

DEMO TIME

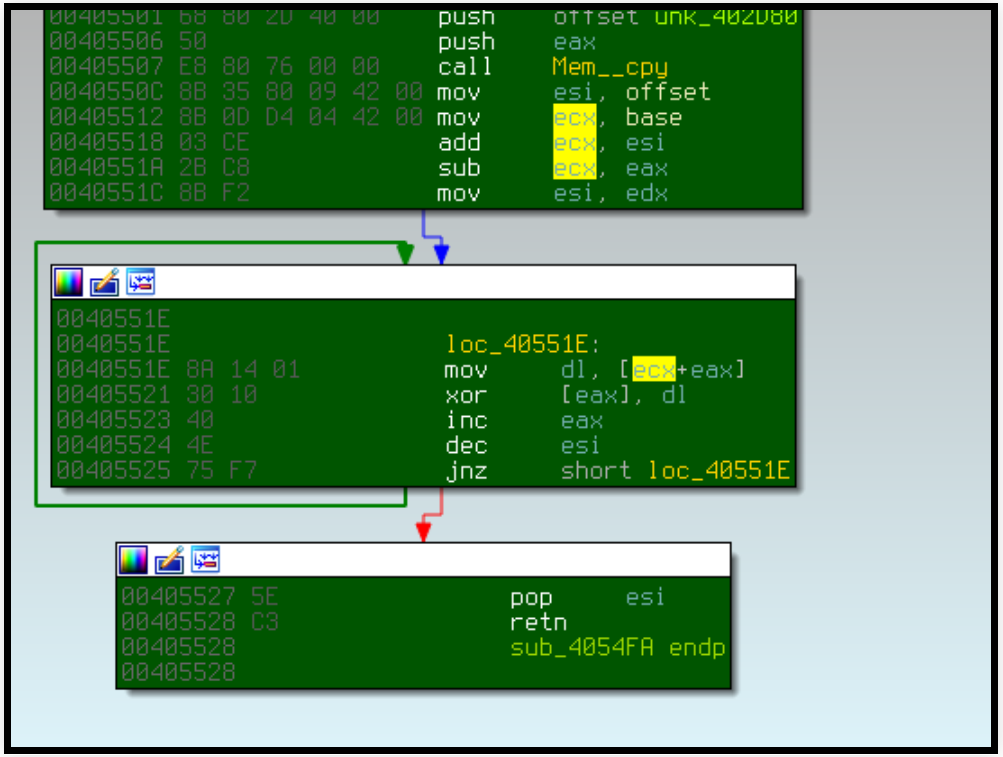
Crack ZeuS


```
{
    "_id" : ObjectId("547c573afa3bd948bbb91baf"),
    "binary" : "da5200f971cf63153cddd22d275309c5",
    "family" : "vmzeus2",
    "rc4sbox" : "6e5d860fb3eb8b93d040c463479b31844f5076327e16039acf5f49b
    "cfg-key" : "rc6sbox",
    "cfg" : "https://marjorieccoates.com/florence/florencec.jpg",
    "alive" : true,
    "strings" : [
        "&Qkhttp://ibhxqaxdh.com/ajsgcxrz/cfg.bin"
    ],
    "datahash" : "a9fcd8104d10da3410d25a6e97bd6db5",
    "version" : "02.00.00.00",
    "timestamp" : ISODate("2014-12-01T12:55:38.435Z"),
    "urls" : [
        "http://gesupfx.net/k39284/mod_vnc.bin",
        "https://marjorieccoates.com/florence/florencec.jpg"
    ],
    "fakeurl" : "http://ibhxqaxdh.com/ajsgcxrz/cfg.bin",
    "rc6sbox" : "4414985ed48f4b456f9af6f1c1b29f2cccbb290626ad6bdc365e5c4
    "botname" : "US"
}
```

BaseConfig

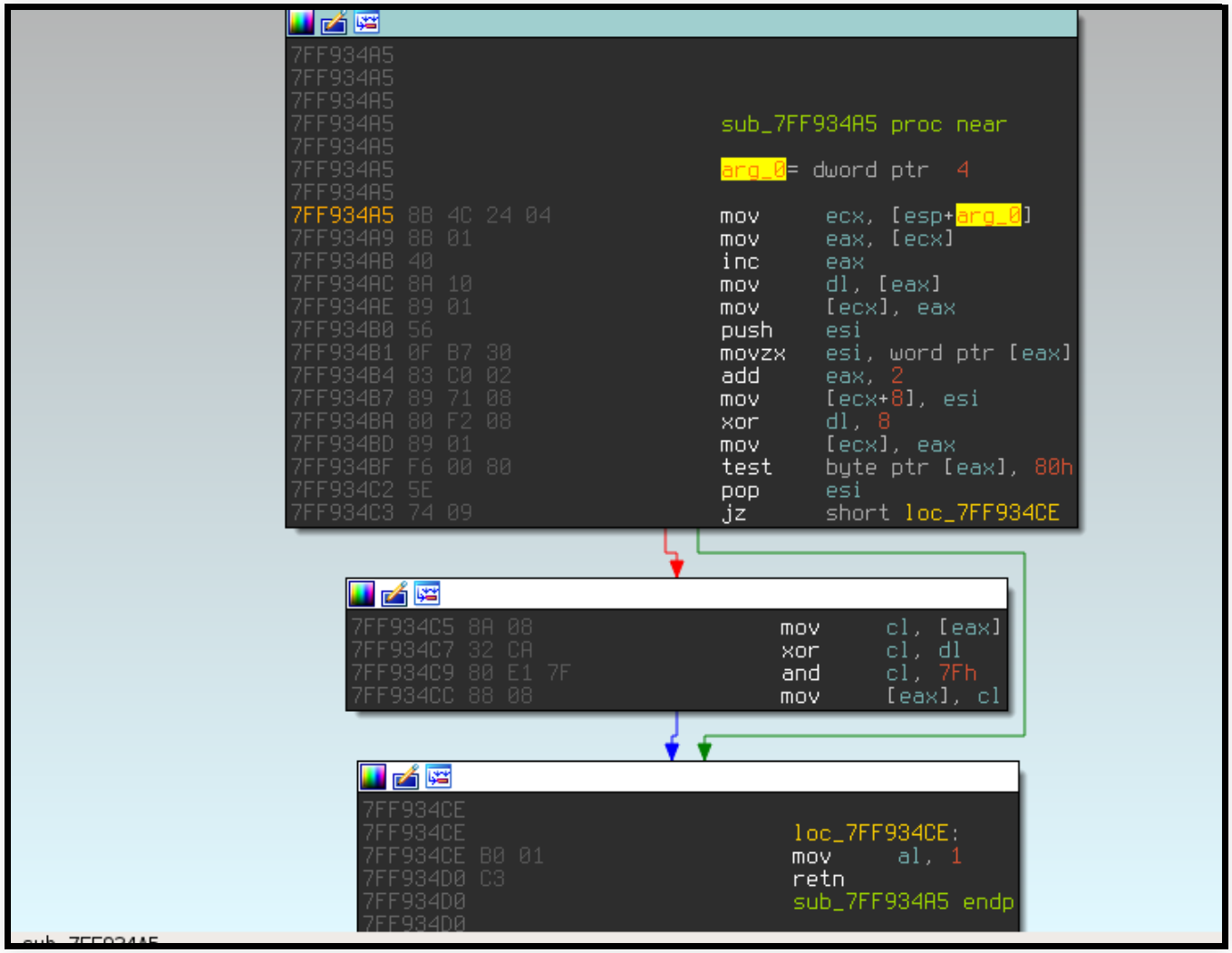
```
typedef struct
{
  BYTE  padding0[80];
  DWORD flags;
  BYTE  padding1[58];
  DWORD delayStats;
  BYTE  padding2[91];
  char  defaultConfig[100 + 1];
  BYTE  padding3[7];
  Crypt::RC4KEY baseKey;
  BYTE  padding4[77];
  WCHAR defaultBotnet[BOTNET_MAX_CHARS + 1];
  BYTE  padding5[38];
  DWORD delayReport;
  BYTE  padding6[14];
  DWORD delayConfig;
  BYTE  padding7[21];
}BASECONFIG;
```

ZBOT


00405501 68 80 2D 40 00 push offset unk_402D80
00405506 50 push eax
00405507 E8 80 76 00 00 call Mem__cpy
0040550C 8B 35 80 09 42 00 mov esi, offset
00405512 8B 0D D4 04 42 00 mov ecx, base
00405518 03 CE add ecx, esi
0040551A 2B C8 sub ecx, eax
0040551C 8B F2 mov esi, edx
0040551E
0040551E loc_40551E:
0040551E 8A 14 01 mov dl, [ecx+eax]
00405521 30 10 xor [eax], dl
00405523 40 inc eax
00405524 4E dec esi
00405525 75 F7 jnz short loc_40551E
00405527 5E pop esi
00405528 C3 retn
00405528 sub_4054FA endp
00405528

VMZeus -- deep look

VMZeus -- deep look



VMZeus -- deep look

```
# Copyright (c) 2014
# Daniel Plohmann <daniel.plohmann<at>gmail<dot>com>
# All rights reserved.
self.magics = {"nop": {1: 0x32, 2: 0x26, 4: 0xF3},
                       "xor": {1: 0xF1, 2: 0xE9, 4: 0x6A},
                       "add": {1: 0x73, 2: 0xD9},
                       "sub": {1: 0xEB},
                       "rol": {1: 0x85},
                       "ror": {1: 0xF3, 2: 0xA6},
                       "not": {1: 0xE7, 2: 0x8B, 4: 0x24},
                       "reorder": 0xBA,
                       "rc4": 0xE9,
                       "setecx": {2: 0xDA, 4: 0x70},
                       "setedi": 0xC8,
                       "loop": {1: 0xD0, 2: 0xEC},
                       "mov_r_const": {1: 0x0B, 2: 0x0F, 4: 0x70},
                       "mov_r_r": {1: 0x46, 2: 0x39, 4: 0x9E},
                       "add_r_r": {1: 0x72, 2: 0x5F, 4: 0xF9},
                       "sub_r_r": {1: 0x1E, 2: 0xD1, 4: 0xA6},
                       "xor_r_r": {1: 0xDB, 2: 0xAC, 4: 0xB7},
                       "add_r_const": {1: 0x9D, 2: 0x8f, 4: 0x28},
                       "sub_r_const": {1: 0x52, 2: 0x85, 4: 0xD8},
                       "xor_r_const": {1: 0xD5, 2: 0x24, 4: 0x2E},
                       "stos_add": {1: 0x65, 2: 0x89, 4: 0xDC},
                       "stos_sub": {1: 0x1D, 4: 0xA0},
                       "stos_xor": {1: 0x41, 2: 0x46, 4: 0x29},
                       "lods": {1: 0xB8, 2: 0x68, 4: 0xFD},
                       "stos": {1: 0x16, 2: 0xBB, 4: 0xF0}
```

VMZeus < 01.00.01.00

- fake url
- encrypted cnc url
- rc4sbox
- botnet name
- dga suffix

```
def vmzeus_dga(key,suff,idx=0):
    key = key[::-1][2:]
    key = chr(ord(key[0]) + idx) + key[1:] + "\x00\x00"
    return 'http://' + md5(key).hexdigest()[:11] + suff
```

VMZeus >= 01.00.01.00

- fake url
- encrypted cnc url
- rc4sbox
- rc6sbox
- botnet name
- ~~dga suffix~~

VMZeus2 -- sigs

```
rule vmzeus : zeus
{
    meta:
         author     = "mak"

    strings:
        $VMZEUS_GetRC6_Key = { 81 ec b0 00 00 00 }
        $VMZEUS_GetBaseConfig = { 8b 45 ?? 0f b6 00 8d 4d ?? ff 14 85 ?? ??
        $config_decrypt = { 81 78 02 3F 10 00 00}

    condition:
        any of them
}
```

DEMO TIME

malwarelibs

open source / web-service

Cheers

- pnX
- KjellChr
- kafeine
- shadowserver
- emerging threats

Q & A

@maciekkotowicz lokalhost.pl/talks/botconf2014
CERT Polska @CERT_Polska_en

CERT.PL >_