CATCHING MALWARE EN MASSE: DNS AND IP STYLE

Dhia Mahjoub (@DhiaLite) dhia@opendns.com
Thibault Reuille (@ThibaultReuille) thibault@opendns.com
Andree Toonk (@atoonk) andree@opendns.com

Abstract

The Internet is constantly growing, providing a myriad of new services, both legitimate and malicious. Criminals take advantage of the scalable, distributed, and rather easily accessible naming, hosting and routing infrastructures of the Internet. As a result, the battle against malware is raging on multiple fronts: the endpoint, the network perimeter, and the application layer. The need for innovative measures to gain ground against the enemy has never been greater.

In this paper, we present novel strategies to catch malware at the DNS and IP level, as well as our unique 3D visualization engine. We will describe the detection systems we built, and share several successful war stories about hunting down malware domains and associated rogue IP space.

At the DNS level, we describe efficient methods for tracking fast flux botnets and describe a study we carried for several months of the Zeus fast flux proxy network. At the IP level, classical reputation methods assign "maliciousness" scores to IPs, BGP prefixes, or ASNs by merely counting domains and IPs. Our system takes an unconventional approach that combines two opposite, yet complementary views and leads to more effective predictive detections.

On one hand, we abstract away from the ASN view. We build the AS graph and investigate its topology to uncover hotspots of malicious or suspicious activities and monitor our DNS traffic for new domains hosted on these malicious IP ranges. We will also describe a unique method of identifying seemingly autonomous networks that are actually operated by one organization, which helps further identify potentially malicious areas of the Internet. On the other hand, we examine a granularity finer than the BGP prefix. For this, we zero in on IP ranges re-allocated or re-assigned to bad customers within large prefixes to host Exploit kit domains, browlock, and other attack types.

We will present various techniques we devised to efficiently discover suspicious reserved ranges and sweep en masse for candidate suspicious IPs. Our system provides actionable intelligence and preemptively detects and blocks malicious IP infrastructures prior to, or immediately after some of them are used to wage malware campaigns, therefore decisively closing the detection gap.

The discussion of these detection systems and "war stories" wouldn't be complete without a visualization engine that adequately displays the use cases and offers a graph navigation and investigation tool. Therefore, in this paper, we will also discuss our own 3D visualization engine, demonstrating the full process which transforms raw data into stunning 3D visuals. We will also present different techniques used to build and render large graph datasets: Force Directed algorithms accelerated on the GPU using OpenCL, 3D rendering and navigation using OpenGL ES, and GLSL Shaders.

Part 1: Catching Malware DNS-style

In today’s cybercrime world, bad actors strive to keep their operations (spam, phishing, malware distribution, botnets, etc.) online at all times, and for that the hosting network infrastructure plays a crucial role. The domain name system (DNS) and IP hosting infrastructures are the foundations of the Internet, and they are equally used for legitimate and criminal activities alike.

Botnets as proxy networks

Botnets require a notable hosting and attack delivery infrastructure. As it is composed of a large collection of compromised machines that receive instructions from a command & control server, a botnet can be quite versatile. The bots can perform numerous malicious activities on demand, and therefore represent an all-purpose weapon for criminals.

Botnets can be used as proxy networks to shield the identity and location of malware CnCs during the communication between an infected host and the CnC. In this case, the proxy network can take the form of a fast flux service network [1]. Fast-flux service networks take advantage of DNS to redirect C&C connection attempts to a set of proxy nodes that are constantly shifting. These proxy nodes serve as intermediaries to relay information between infected hosts and the C&C component. A fast-flux service network is created by setting up a selection of domains to resolve to the IP addresses of a subset of available proxy nodes (bots). These IP addresses are then frequently changed to the IP addresses of new proxy nodes which is known as IP fluxing. This way, the proxy network provides an extra layer of evasion and protection for the actual malware CnCs. The communication between the infected hosts always goes through the fast flux proxy network to reach the malware backend CnCs.

In general, botnet-based proxy networks can be used to serve malware pushed from CnCs down to infected hosts via Exploit kit attacks or spam attachements, or forward communication from infected clients to backend CnCs like in the case of Kelihos (fast flux botnet wuth TTL=0) and zbot (fast flux botnet with a TTL=150).

Zeus Crimeware

The Zeus Crimeware consists of the following main components: a control panel, config files (which hold URLs for drop zones, extra payload, extra configs, and target websites of web injects), binary files, and a builder. The crimeware’s main characteristics are to steal financial data such as online bank account and credit card information, steal sensitive credentials, and perform web injects.

Zeus CnCs

A Zeus command and control domain can be hosted on different platforms: it can be a compromised site, a host that is part of a fast flux botnet, or hosted on bulletproof or free hosting providers [3]. Additionally, Zeus CnC domains can be used for three types of purposes: they can serve configuration files, serve binary files, or serve as drop zones.

In this study, we focus on the zbot proxy network that hosts fast flux domains with a characteristic TTL of 150, where these domains share the same hosting infrastructure of infected hosts. We use two methods to detect the fast flux domains hosted on this network: periodic Hadoop Pig jobs, and IP harvesting combined with filtering heuristics applied on our streaming authoritative DNS traffic.

For the first method, we run a periodic Pig job on our authoritative logs stored on HDFS where we retrieve domains with a TTL<=150. We filter out unwanted domains such as spam or legitimate domains, then build the bipartite graph of domains-IPs, and take the largest connected component. This heuristic extracts the zbot fast flux domains along with their resolving IPs from the portion of logs we processed. The IPs are appended to the pool of zbot IPs that is also used in the second method.

For the second method, we incrementally build the list of confirmed zbot fast flux domains, continuously resolve them, and append the IPs to the pool of zbot IPs. At the same time, we tap into our streaming authoritative DNS traffic and extract any domain whose IP or name server IP is in the zbot IP pool.

The combination of these two methods allows us to catch the bulk of new zbot fast flux domains that we see in our traffic. Using the authoritative DNS stream is faster than a DNSDB stored on Hadoop, as the traffic is coming live from a selection of our resolvers at a rate of hundreds to thousands of entries per second. A typical entry looks like:
ASN, domain, 2LD, IP, NS_IP, timestamp, TTL, type.

Zbot proxy network usage

There are a few different uses of the fast flux domains riding on the zbot proxy network we are studying. One recorded use is as zbot CnCs after infection via Magnitude EK. A second recorded use is as Kuluoz CnCs post-infection. In this case, various exploit kits or malicious attachments lead to the dropping of the Kuluoz/Dofoil malware. The infected host becomes part of the Asprox botnet and phones to one or several of the fast flux domains.

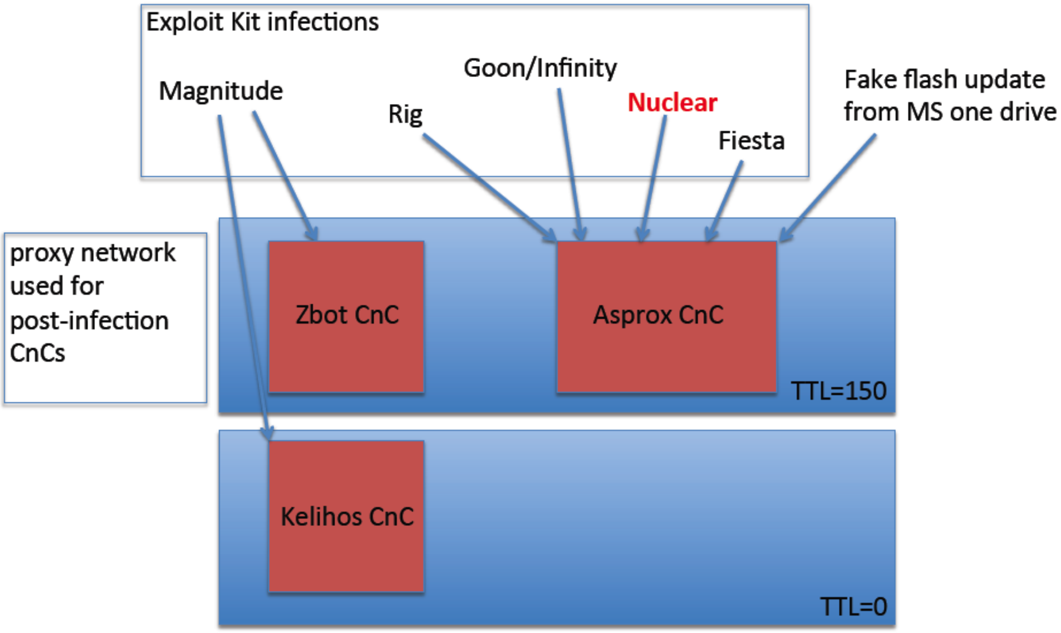


Figure 1. Infection vectors followed by beaconing to fast flux proxy networks.

In the figure above, we show the different documented scenarios of Exploit kit attacks or malicious attachments leading to beaconing to generic zbot CnCs, Asprox CnCs, or Kelihos CnCs. These scenarios are differentiated with ET signatures. There are two main infection vectors: Exploit kit attack via drive-by downloads or spam emails with embedded links leading to malware, or the malware dropper coming as attachment (fake Flash update) [4].

HTTP Traffic URL Patterns

We monitored and studied the HTTP traffic to the zbot CnCs by temporarily sinkholing the domains and querying VirusTotal's database, as outlined below.

Zeus CnC traffic

A Zeus CnC can serve 3 types of URLs: Config, Binary, or drop zone [3]. In the Table below, we show examples of Zeus CnC URLs that were observed in traffic.

Table 1. Zeus fast flux CnC domains, URL patterns and malware variants.

Zeus CnC domain	HTTP method	Variant	Url type
seorubl.in	GET /forum/popap1.jpg	Zeus	ConfigURL
reznormakro.su	GET /winconf/kernl.bin	ICE IX	ConfigURL
orbitmanes.ru	GET /01.exe	KINS	BinaryURL
reportonh.com	GET /pack32/sysconf.exe	Zeus	BinaryURL
sytemnr.com	GET /pack32/sysconf.exe	Zeus	BinaryURL

Asprox CnC traffic

The zbot fastflux domains are also used as CnCs for the Asprox botnet. After a machine is infected via Exploit kit attacks or malicious spam attachments (Figure 1), it performs multiple Asprox type callbacks to some of the fast flux domains then follows with clickfraud traffic. In the Table below, we show example domains used as CnCs (live at the moment of this writing), the HTTP methods and URL patterns and the Emerging Threats alerts they trigger.

Table 2. zbot fast flux domains with ET alerts and URL pattern.

ET alert	Domain	HTTP method
ET TROJAN W32/Asprox.ClickFraudBot CnC Beacon	vision-vaper.su joye-luck.com	GET /b/eve/
	grade-well.com vision-vaper.su joye-luck.com	GET /b/letr/
	gummiringes.com carbon-flx.su	GET /b/shoe/
ET TROJAN W32/Asprox.ClickFraudBot POST CnC Beacon	grade-well.com hefu-juder.com jogurt-jetr.com joye-luck.com juice-from.com	POST /b/opt/

	ray-green.ru shark-yope.su tundra-red.com vision-vaper.su	
	joye-luck.com vision-vaper.su	POST /b/req/

Notice that in the case of the URL patterns: /b/eve/, /b/letr/, /b/opt/ and /b/req/, they are followed by a 12 2-byte hexadecimal string. For example:
http://vision-vaper.su/b/eve/1f40f89ea1eebf47748490eb
http://grade-well.com/b/letr/7165F757DAA94FE2CD116CC4
http://hefu-juder.com/b/opt/17D2BCDADA720FE35C6F00F1
http://joye-luck.com/b/req/648EEF9F6EDBDEE2E5E7F800

Other traffic patterns

We list observed URL patterns used for other purposes:

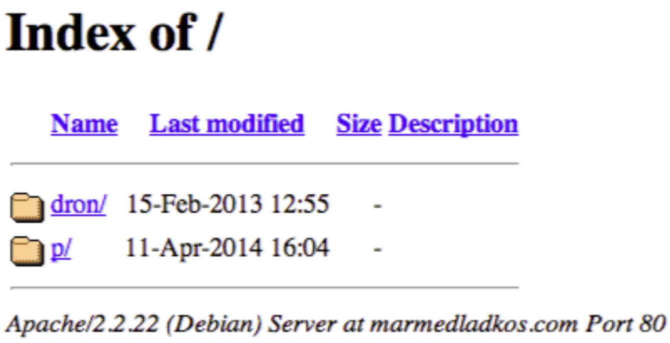
Beaconing and announcing version, make, OS
GET /1/?uid=17428742&ver=1.14&mk=bb3b62&os=WinXP&rs=adm&c=1&rq=0
with several occurring OS versions:
os=S2000
os=Win07
os=Win_V
os=WinXP
os=Win08

Getting binaries and configs
azg.su, GET /coivze7aip/modules/bot.exe
tundra-tennes.com, GET /infodata/soft32.dll
tundra-tennes.com, GET /info-data/soft32.dll
bee-pass.com, GET /info/soft32.dll

quarante-ml.com, GET /nivoslider/jquery/
quarante-ml.com, GET /nivoslider98.45/ajax/
quarante-ml.com, GET /nivoslider98.45/jquery/
tundra-tennes.com, GET /nivoslider/ajax/

Pony panel hosted on zbot proxy network

While investigating the zbot fast flux domains, we came across one that was hosting a panel for the Pony malware (botnet). As Figure 2 shows, the panel was hosted on the domain marmedladkos.com:



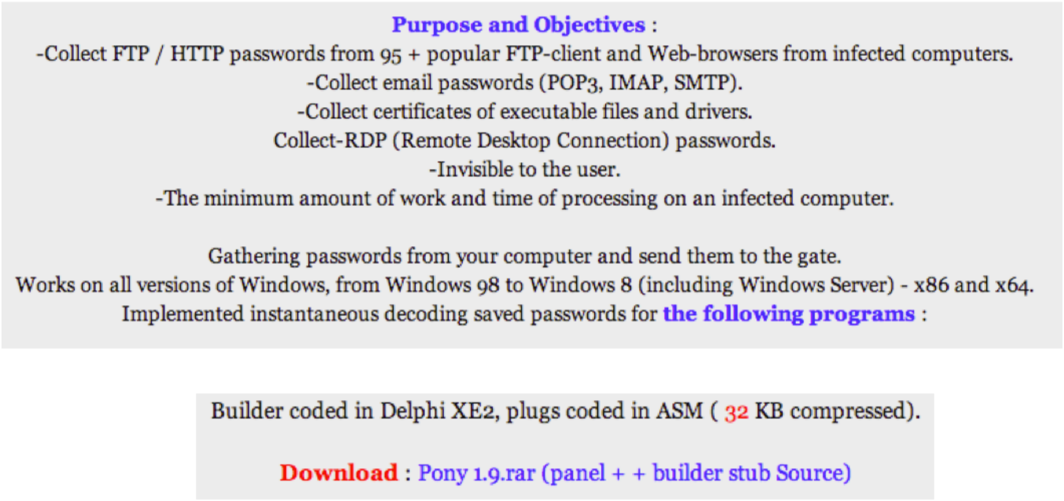
Index of /

Name	Last modified	Size	Description
dron/	15-Feb-2013 12:55	-	
p/	11-Apr-2014 16:04	-	

Apache/2.2.22 (Debian) Server at marmedladkos.com Port 80

Figure 2. Pony panel on marmedladkos.com.

Pony 1.9 was leaked in late 2012 to Trojan Forge. The malware is an infostealer identified by AV vendors as Win32/Fareit. It consists of a botnet controller via a panel, has features for user management, logging and statistics via a database.



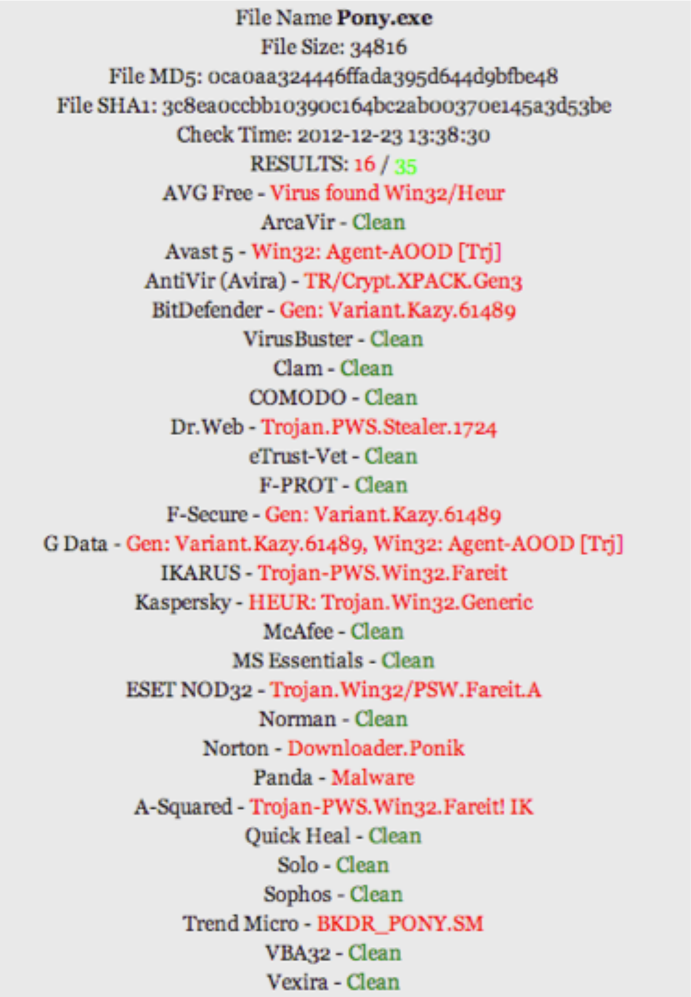
Purpose and Objectives :
-Collect FTP / HTTP passwords from 95 + popular FTP-client and Web-browsers from infected computers.
-Collect email passwords (POP3, IMAP, SMTP).
-Collect certificates of executable files and drivers.
Collect-RDP (Remote Desktop Connection) passwords.
-Invisible to the user.
-The minimum amount of work and time of processing on an infected computer.

Gathering passwords from your computer and send them to the gate.
Works on all versions of Windows, from Windows 98 to Windows 8 (including Windows Server) - x86 and x64.
Implemented instantaneous decoding saved passwords for **the following programs** :

Builder coded in Delphi XE2, plugs coded in ASM (32 KB compressed).

Download : Pony 1.9.rar (panel + + builder stub Source)

Figure 3. Pony features as advertised on Forums.



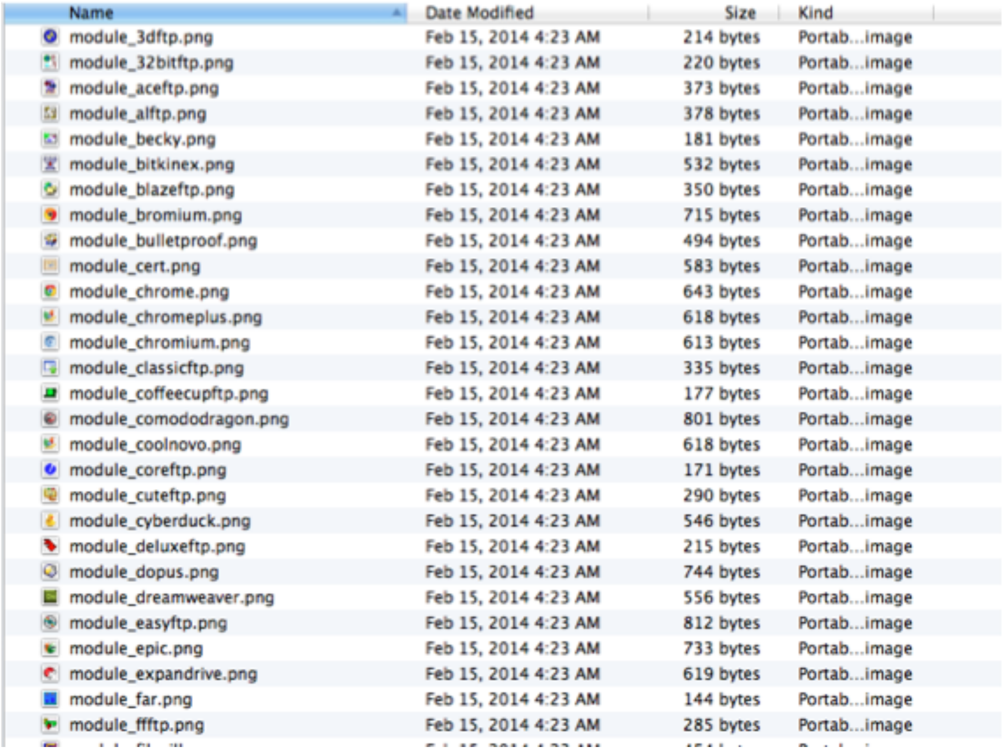
File Name **Pony.exe**
File Size: 34816
File MD5: 0ca0aa324446ffada395d644d9bfbe48
File SHA1: 3c8ea0ccbb10390c164bc2ab00370e145a3d53be
Check Time: 2012-12-23 13:38:30
RESULTS: 16 / 35
AVG Free - Virus found Win32/Heur
ArcaVir - Clean
Avast 5 - Win32: Agent-AOOD [Trj]
AntiVir (Avira) - TR/Crypt.XPACK.Gen3
BitDefender - Gen: Variant.Kazy.61489
VirusBuster - Clean
Clam - Clean
COMODO - Clean
Dr.Web - Trojan.PWS.Stealer.1724
eTrust-Vet - Clean
F-PROT - Clean
F-Secure - Gen: Variant.Kazy.61489
G Data - Gen: Variant.Kazy.61489, Win32: Agent-AOOD [Trj]
IKARUS - Trojan-PWS.Win32.Fareit
Kaspersky - HEUR: Trojan.Win32.Generic
McAfee - Clean
MS Essentials - Clean
ESET NOD32 - Trojan.Win32/PSW.Fareit.A
Norman - Clean
Norton - Downloader.Ponik
Panda - Malware
A-Squared - Trojan-PWS.Win32.Fareit! IK
Quick Heal - Clean
Solo - Clean
Sophos - Clean
Trend Micro - BKDR_PONY.SM
VBA32 - Clean
Vexira - Clean

Figure 4. Detection of Pony by AV vendors as advertised in Forums.

The Pony payload is commonly delivered via Exploit kit attacks or attachments in spam emails. In the table below, we show a few folders on the panel site and their functionality. It is notable that the character set cp1251 characteristic of Cyrillic scripts is used everywhere on the site and in config.php, the variable date_default_timezone_set is set to ('Europe/Moscow') which would be an indication about the origin country of the authors or users of the panel.

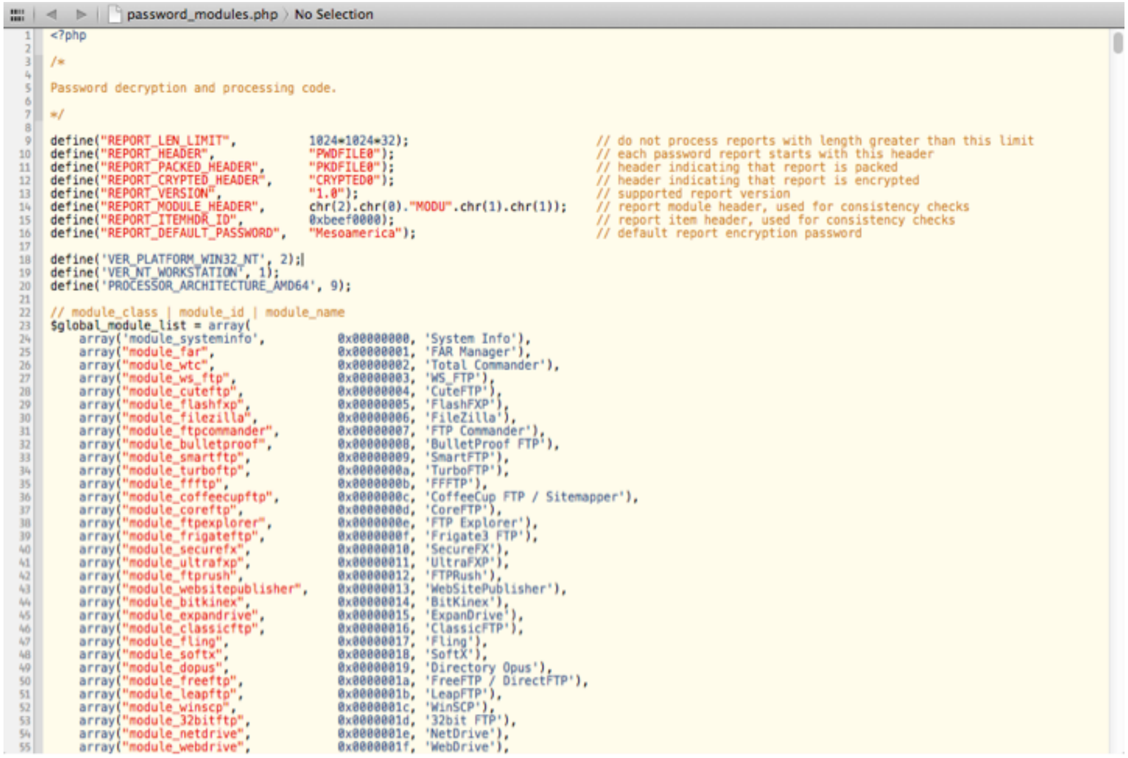
Table 3. Folders on panel site and functionalities.

Path on site	**Function**
p/Panel.zip	controlling php scripts
p/includes/design/images/modules/*	images for each zeus plugin supported/tracked (Figure ww)
p/includes/password_modules.php	contains array with all software it tries to steal credentials for (Figure ee)
p/includes/database.php	contains db schema and accessors



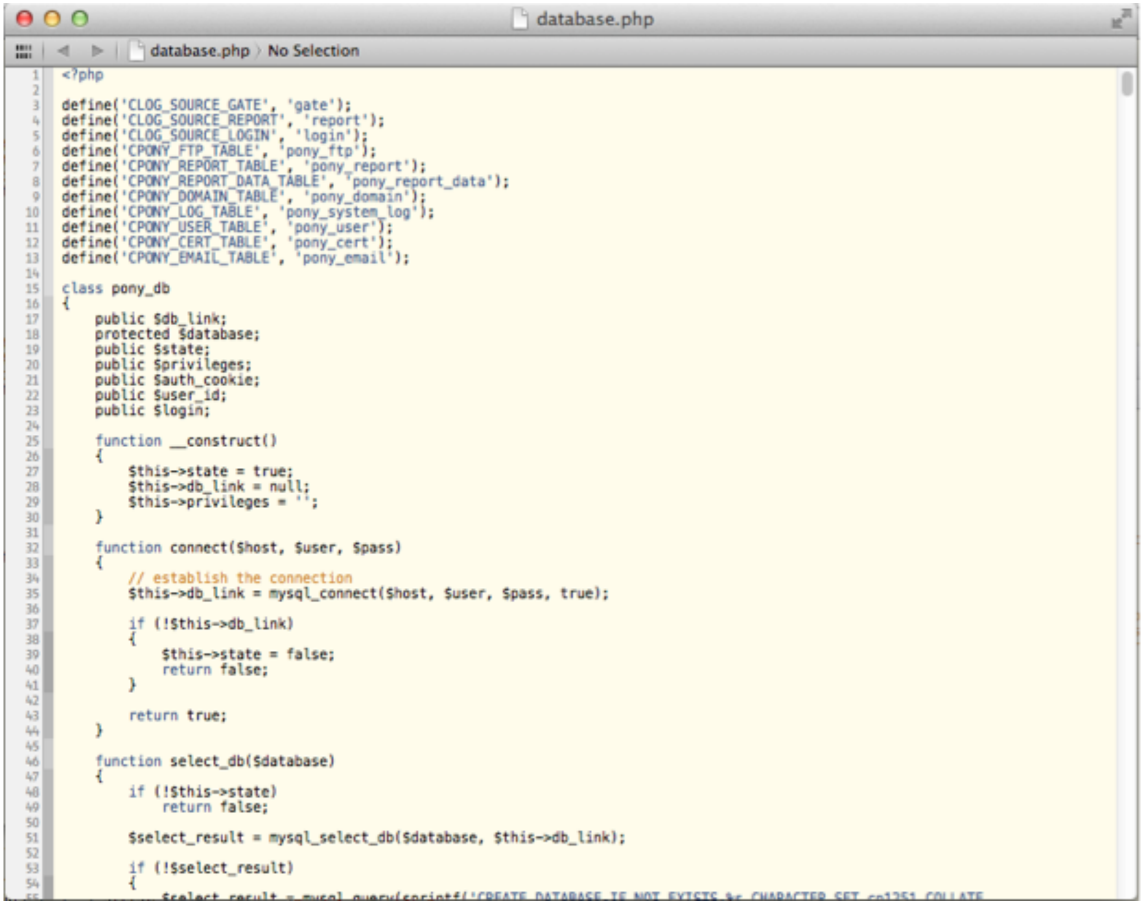
Name	Date Modified	Size	Kind
module_3dftp.png	Feb 15, 2014 4:23 AM	214 bytes	Portab...image
module_32bitftp.png	Feb 15, 2014 4:23 AM	220 bytes	Portab...image
module_aceftp.png	Feb 15, 2014 4:23 AM	373 bytes	Portab...image
module_alftp.png	Feb 15, 2014 4:23 AM	378 bytes	Portab...image
module_becky.png	Feb 15, 2014 4:23 AM	181 bytes	Portab...image
module_bitkinex.png	Feb 15, 2014 4:23 AM	532 bytes	Portab...image
module_blazeftp.png	Feb 15, 2014 4:23 AM	350 bytes	Portab...image
module_bromium.png	Feb 15, 2014 4:23 AM	715 bytes	Portab...image
module_bulletproof.png	Feb 15, 2014 4:23 AM	494 bytes	Portab...image
module_cert.png	Feb 15, 2014 4:23 AM	583 bytes	Portab...image
module_chrome.png	Feb 15, 2014 4:23 AM	643 bytes	Portab...image
module_chromeplus.png	Feb 15, 2014 4:23 AM	618 bytes	Portab...image
module_chromium.png	Feb 15, 2014 4:23 AM	613 bytes	Portab...image
module_classicftp.png	Feb 15, 2014 4:23 AM	335 bytes	Portab...image
module_coffeecupftp.png	Feb 15, 2014 4:23 AM	177 bytes	Portab...image
module_comododragon.png	Feb 15, 2014 4:23 AM	801 bytes	Portab...image
module_coolnovo.png	Feb 15, 2014 4:23 AM	618 bytes	Portab...image
module_coreftp.png	Feb 15, 2014 4:23 AM	171 bytes	Portab...image
module_cuteftp.png	Feb 15, 2014 4:23 AM	290 bytes	Portab...image
module_cyberduck.png	Feb 15, 2014 4:23 AM	546 bytes	Portab...image
module_deluxeftp.png	Feb 15, 2014 4:23 AM	215 bytes	Portab...image
module_dopus.png	Feb 15, 2014 4:23 AM	744 bytes	Portab...image
module_dreamweaver.png	Feb 15, 2014 4:23 AM	556 bytes	Portab...image
module_easyftp.png	Feb 15, 2014 4:23 AM	812 bytes	Portab...image
module_epic.png	Feb 15, 2014 4:23 AM	733 bytes	Portab...image
module_expandrive.png	Feb 15, 2014 4:23 AM	619 bytes	Portab...image
module_far.png	Feb 15, 2014 4:23 AM	144 bytes	Portab...image
module_ffftp.png	Feb 15, 2014 4:23 AM	285 bytes	Portab...image

Figure 5. Images for supported plugins.



```php
<?php

/*

Password decryption and processing code.

*/

define("REPORT_LEN_LIMIT",         1024*1024*32);                           // do not process reports with length greater than this limit
define("REPORT_HEADER",            "PWDFILE0");                             // each password report starts with this header
define("REPORT_PACKED_HEADER",     "PKDFILE0");                             // header indicating that report is packed
define("REPORT_CRYPTED_HEADER",    "CRYPTED0");                             // header indicating that report is encrypted
define("REPORT_VERSION",           "1.0");                                  // supported report version
define("REPORT_MODULE_HEADER",     chr(2).chr(0)."MODU".chr(1).chr(1));     // report module header, used for consistency checks
define("REPORT_ITEMHDR_ID",        0xbeef0000);                             // report item header, used for consistency checks
define("REPORT_DEFAULT_PASSWORD",  "Mesoamerica");                          // default report encryption password

define('VER_PLATFORM_WIN32_NT', 2);
define('VER_NT_WORKSTATION', 1);
define('PROCESSOR_ARCHITECTURE_AMD64', 9);

// module_class | module_id | module_name
$global_module_list = array(
    array('module_systeminfo',          0x00000000, 'System Info'),
    array("module_far",                 0x00000001, 'FAR Manager'),
    array("module_wtc",                 0x00000002, 'Total Commander'),
    array("module_ws_ftp",              0x00000003, 'WS_FTP'),
    array("module_cuteftp",             0x00000004, 'CuteFTP'),
    array("module_flashfxp",            0x00000005, 'FlashFXP'),
    array("module_filezilla",           0x00000006, 'FileZilla'),
    array("module_ftpcommander",        0x00000007, 'FTP Commander'),
    array("module_bulletproof",         0x00000008, 'BulletProof FTP'),
    array("module_smartftp",            0x00000009, 'SmartFTP'),
    array("module_turboftp",            0x0000000a, 'TurboFTP'),
    array("module_ffftp",               0x0000000b, 'FFFTP'),
    array("module_coffeecupftp",        0x0000000c, 'CoffeeCup FTP / Sitemapper'),
    array("module_coreftp",             0x0000000d, 'CoreFTP'),
    array("module_ftpexplorer",         0x0000000e, 'FTP Explorer'),
    array("module_frigateftp",          0x0000000f, 'Frigate3 FTP'),
    array("module_securefx",            0x00000010, 'SecureFX'),
    array("module_ultrafxp",            0x00000011, 'UltraFXP'),
    array("module_ftprush",             0x00000012, 'FTPRush'),
    array("module_websitepublisher",    0x00000013, 'WebSitePublisher'),
    array("module_bitkinex",            0x00000014, 'BitKinex'),
    array("module_expandrive",          0x00000015, 'ExpanDrive'),
    array("module_classicftp",          0x00000016, 'ClassicFTP'),
    array("module_fling",               0x00000017, 'Fling'),
    array("module_softx",               0x00000018, 'SoftX'),
    array("module_dopus",               0x00000019, 'Directory Opus'),
    array("module_freeftp",             0x0000001a, 'FreeFTP / DirectFTP'),
    array("module_leapftp",             0x0000001b, 'LeapFTP'),
    array("module_winscp",              0x0000001c, 'WinSCP'),
    array("module_32bitftp",            0x0000001d, '32bit FTP'),
    array("module_netdrive",            0x0000001e, 'NetDrive'),
    array("module_webdrive",            0x0000001f, 'WebDrive'),
```

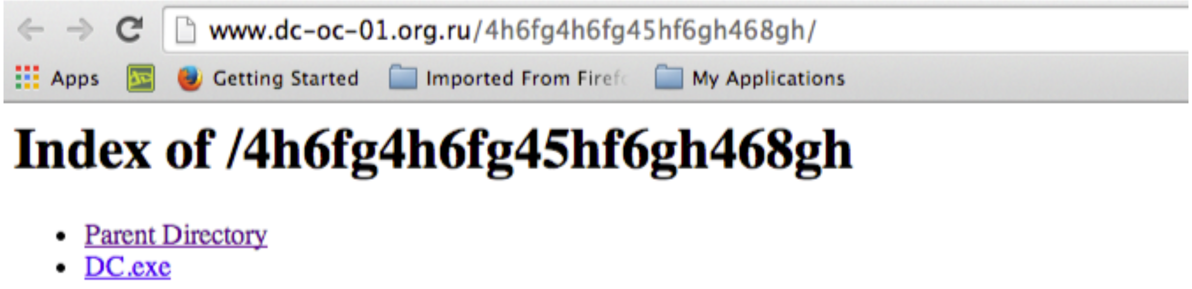
Figure 6. List of applications stored in the Panel db of which Pony steals the passwords.



```php
<?php

define('CLOG_SOURCE_GATE', 'gate');
define('CLOG_SOURCE_REPORT', 'report');
define('CLOG_SOURCE_LOGIN', 'login');
define('CPONY_FTP_TABLE', 'pony_ftp');
define('CPONY_REPORT_TABLE', 'pony_report');
define('CPONY_REPORT_DATA_TABLE', 'pony_report_data');
define('CPONY_DOMAIN_TABLE', 'pony_domain');
define('CPONY_LOG_TABLE', 'pony_system_log');
define('CPONY_USER_TABLE', 'pony_user');
define('CPONY_CERT_TABLE', 'pony_cert');
define('CPONY_EMAIL_TABLE', 'pony_email');

class pony_db
{
    public $db_link;
    protected $database;
    public $state;
    public $privileges;
    public $auth_cookie;
    public $user_id;
    public $login;

    function __construct()
    {
        $this->state = true;
        $this->db_link = null;
        $this->privileges = '';
    }

    function connect($host, $user, $pass)
    {
        // establish the connection
        $this->db_link = mysql_connect($host, $user, $pass, true);

        if (!$this->db_link)
        {
            $this->state = false;
            return false;
        }

        return true;
    }

    function select_db($database)
    {
        if (!$this->state)
            return false;

        $select_result = mysql_select_db($database, $this->db_link);

        if (!$select_result)
        {
            $select_result = mysql_query(sprintf('CREATE DATABASE IF NOT EXISTS %s CHARACTER SET cp1251 COLLATE
```

Figure 7. db schema in database.php.

Furthermore, we searched on Google for certain strings from the Pony panel website and we found several more sites with open panels with some sites hosting other malware payload. One Example of such site is shown in the figure below:



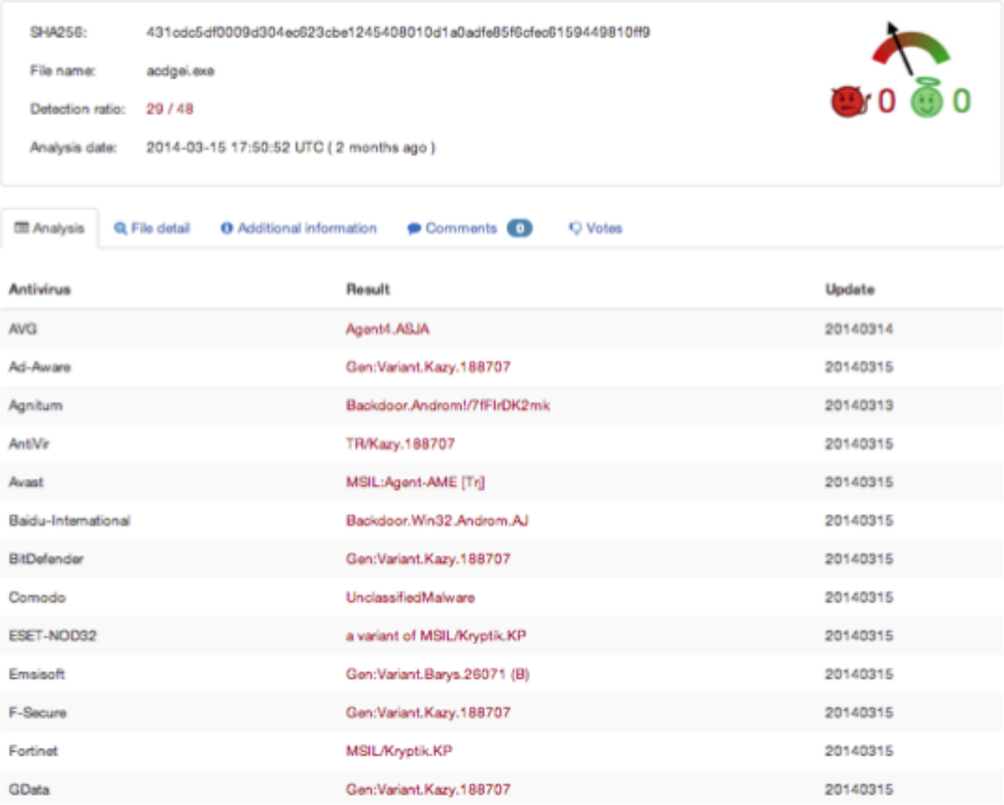
www.dc-oc-01.org.ru/4h6fg4h6fg45hf6gh468gh/

Index of /4h6fg4h6fg45hf6gh468gh

- Parent Directory
- DC.exe

Figure 8. Open panel with malware payload.

The file DC.exe is an Andromeda malware sample as the below VT report shows:



SHA256: 431cdc5df0009d304ec623cbe1245408010d1a0adfe85f6cfec6159449810ff9

File name: acdgei.exe

Detection ratio: 29 / 48

Analysis date: 2014-03-15 17:50:52 UTC (2 months ago)

Antivirus	Result	Update
AVG	Agent4.ASJA	20140314
Ad-Aware	Gen:Variant.Kazy.188707	20140315
Agnitum	Backdoor.Androm!/7fFIrDK2mk	20140313
AntiVir	TR/Kazy.188707	20140315
Avast	MSIL:Agent-AME [Trj]	20140315
Baidu-International	Backdoor.Win32.Androm.AJ	20140315
BitDefender	Gen:Variant.Kazy.188707	20140315
Comodo	UnclassifiedMalware	20140315
ESET-NOD32	a variant of MSIL/Kryptik.KP	20140315
Emsisoft	Gen:Variant.Barys.26071 (B)	20140315
F-Secure	Gen:Variant.Kazy.188707	20140315
Fortinet	MSIL/Kryptik.KP	20140315
GData	Gen:Variant.Kazy.188707	20140315

Figure 9. VirusTotal's detection results of the sample discovered on the panel site.

In the table below, we show some of the open Pony panels we discovered from the initial one. These are not hosted on the zbot fast flux domains though.

Table 4. Additional discovered panels.

Open Panels
epvpcash.net16.net/Panel/temp/ hgfhgfhgfhfg.net/pony/temp/ http://pantamati.com/dream/Panel/temp/ http://pantamati.com/wall/Panel/temp/ mastermetr.ru/steal/Panel/temp/ microsoft.blg.lt/q/temp/ santeol.su/p/temp/ terra-araucania.cl/pooo/temp/ thinswares.com/panel/temp/ www.broomeron.com/pn2/temp/ www.kimclo.com/cli/temp/ www.sumdfase2.net/adm/temp/ www.tripplem2.com/images/money/temp/

Top Level Domain distribution of the zbot fast flux domains

In the Figure below, we show the TLD distribution of a 900+ sample of zbot fast flux CnC domains. It is clear that .su and .ru are the most abused ccTLDs followed by the generic .com, .net.

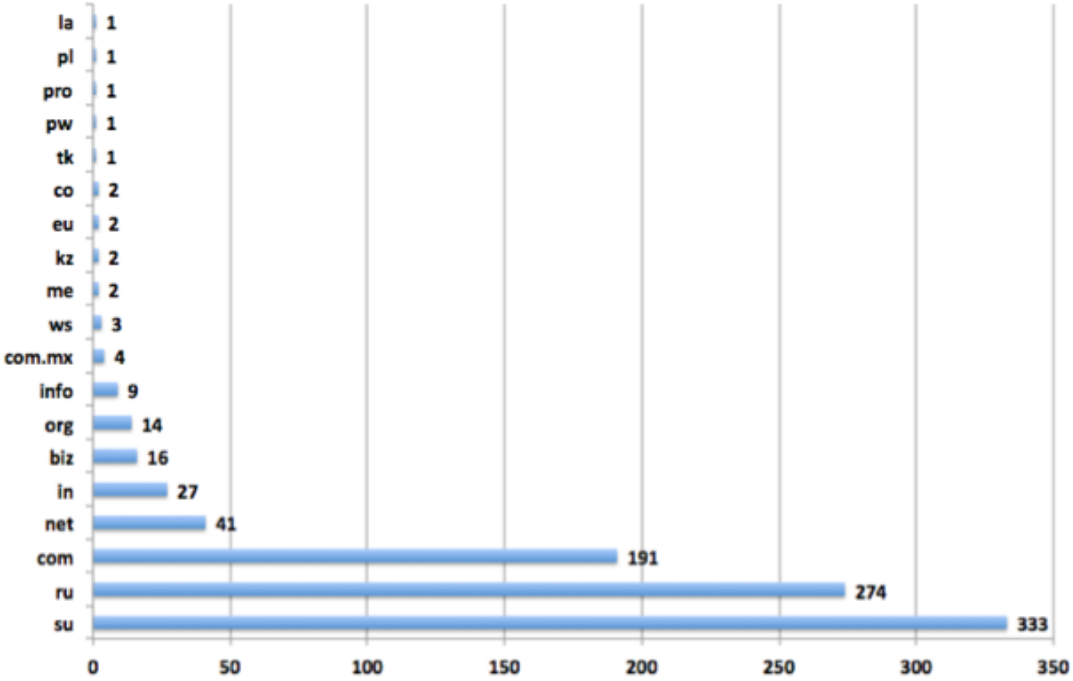


Figure 10. TLD distribution of zbot fast flux CnC domains.

Proxy network country distribution

We take a sample of 170,000+ IPs of the zbot proxy network and show in the figures below the top hosting countries. We see a high presence of infected machines acting as hosting bots in Russia, Ukraine and Turkey.

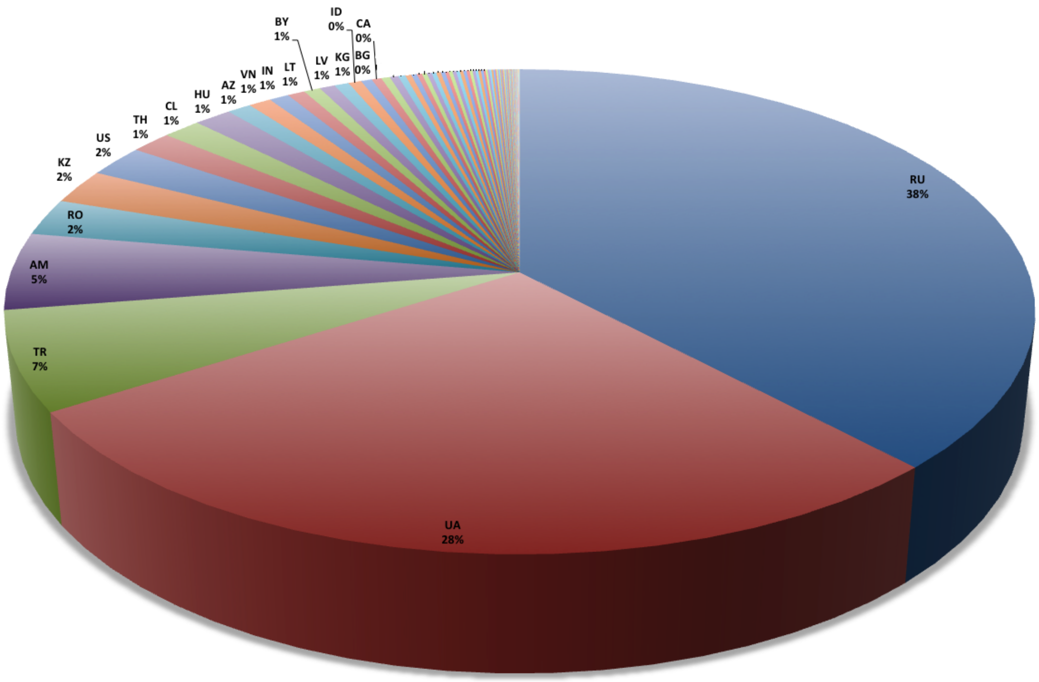


Figure 11. Top hosting countries of zbot proxy network IPs.

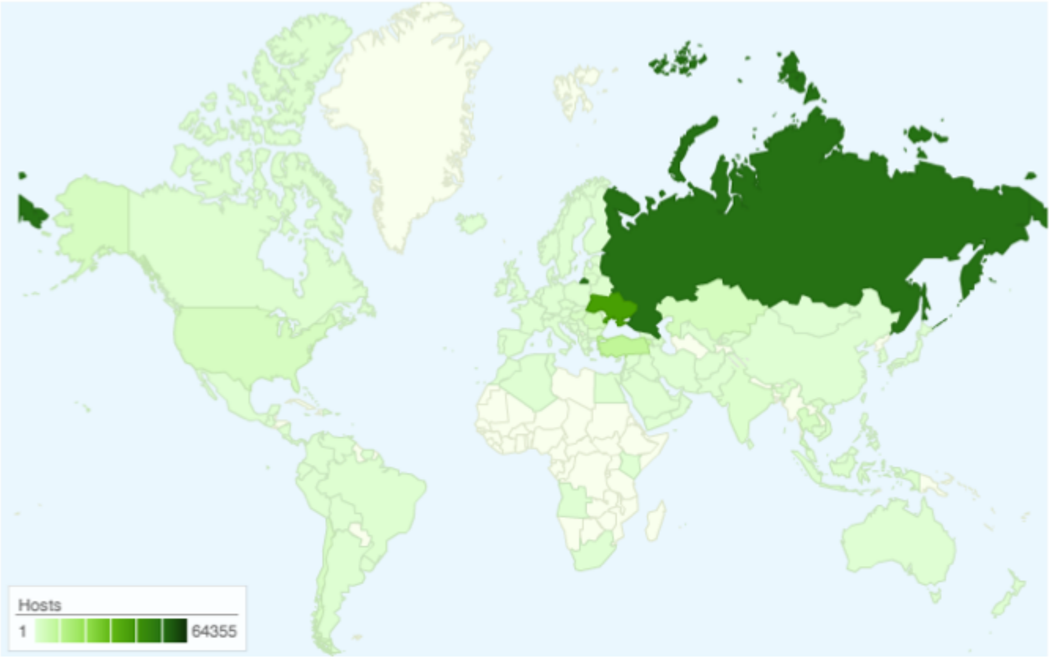


Figure 12. Country distribution of zbot proxy network IPs on world map.

Country distribution of clients beaconing to CnCs

We collected the client IPs that looked up the fast flux CnC domains for a period of 24 hours. The figure below shows the country distribution. A high volume of lookups comes from the US, which can be caused by targeted US victims.

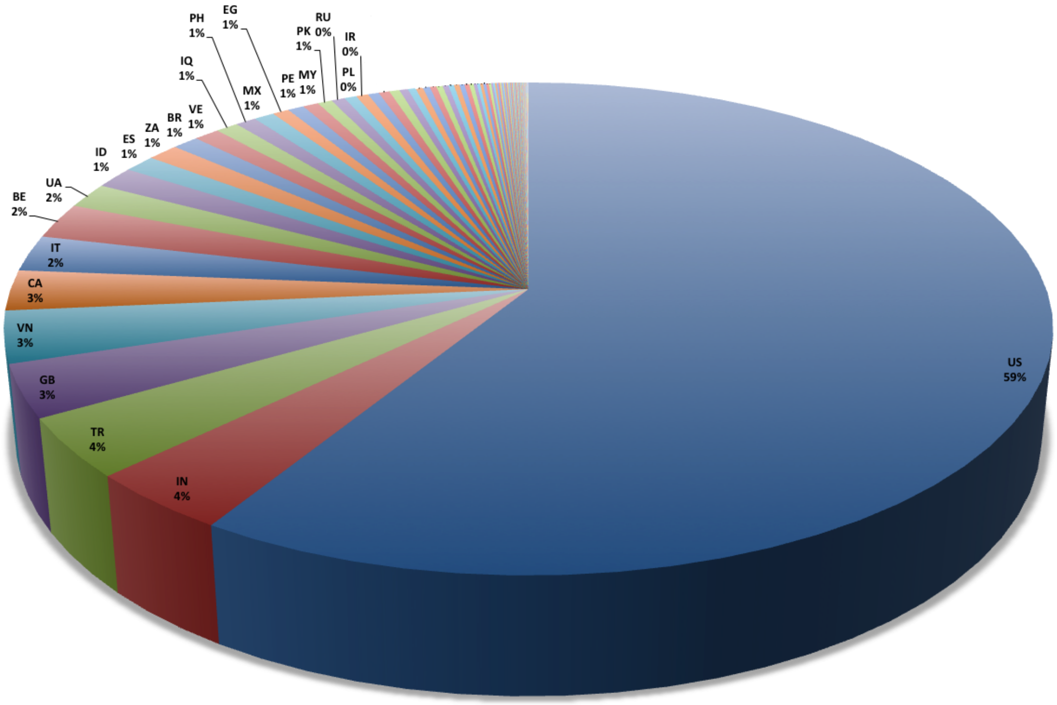


Figure 13. Top countries of client IPs looking up zbot fast flux domains for 24 hours.

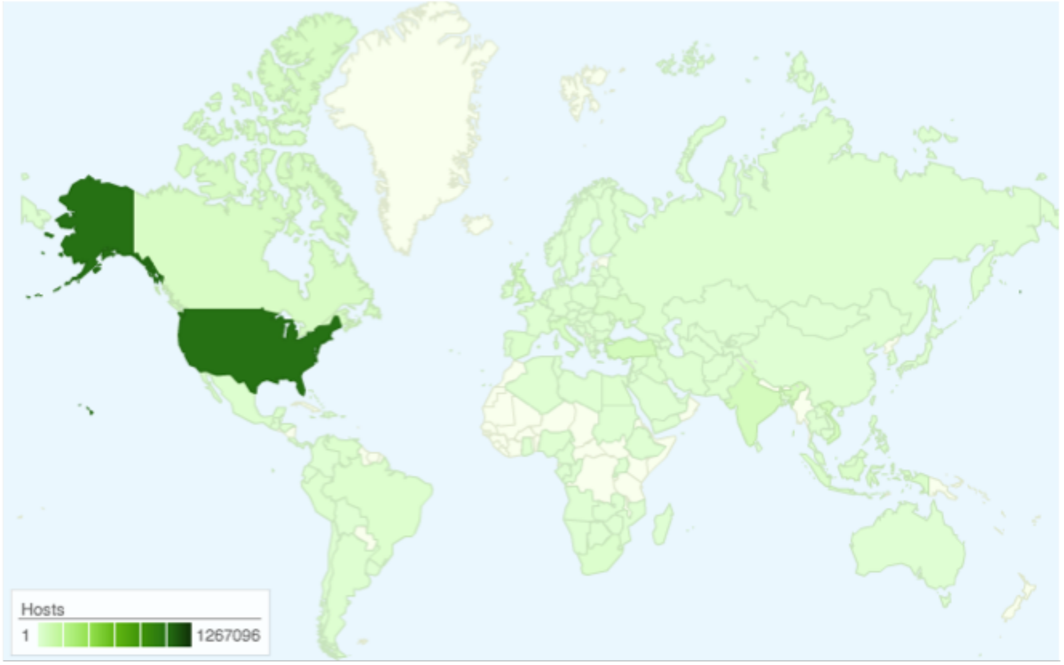


Figure 14. Country distribution of client IPs looking up zbot fast flux domains.

CnC domains and related malware samples

We took a sample of 337 fast flux domains hosted on the zbot proxy network and identified 208 different samples (unique sha256) that have communicated with the CnCs. The notable top observed samples communicating with the CnC domains are:
Trojan[Spy]/Win32.Zbot
TrojanDownloader:Win32/Upatre

Notice that Upatre has served as a downloader for Zeus GameOver and has been recorded as being sent as attachment in spam emails delivered by the Cutwail botnet.

Part 2: Catching Malware IP-style

Classical reputation systems used for network-level threat detection assign scores to IPs, BGP prefixes and ASNs based on counting the volume of hosted malicious domains or IPs. In this study, our goal is to assess malicious IP ranges in certain ASNs from a new perspective. We look beyond the simple counting of the number of bad domains and IPs hosted on prefixes of an ASN, by exploring the topology of the AS graph, and looking at a smaller granularity than the BGP prefix (sub-allocated ranges within BGP prefixes). We will also present a unique approach for finding similar prefixes and autonomous systems that are related to the network hosting malicious software. This will help identify more networks and hosts that are already high risk or will be in the near future.

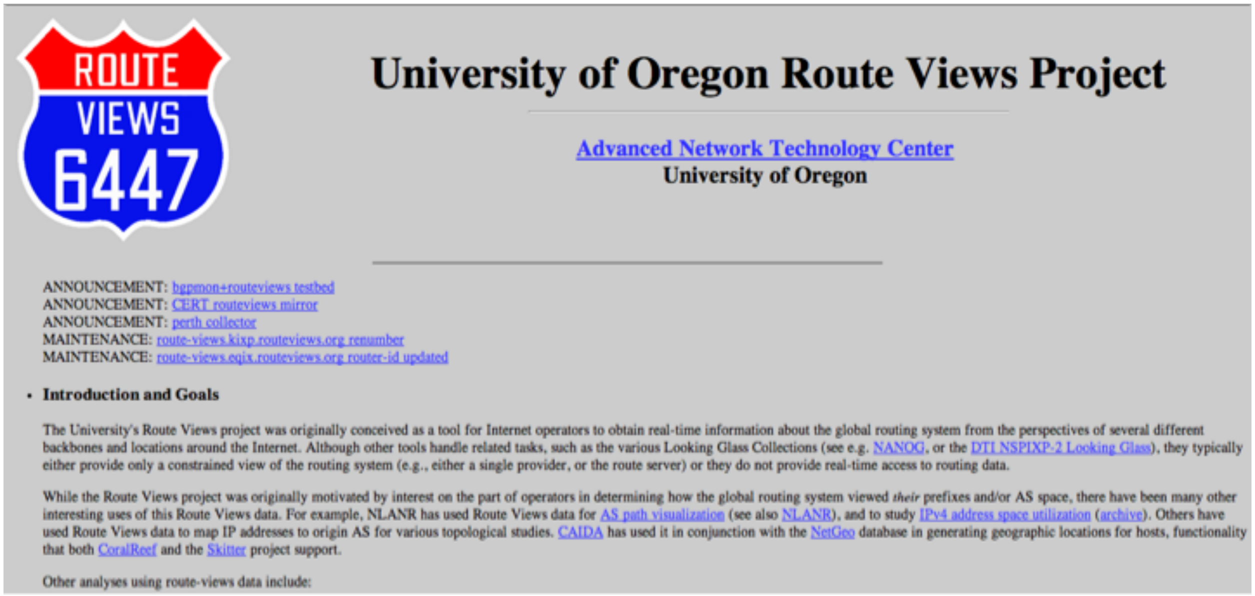
Previous research has been conducted on exploring malicious ASNs. For example, in [5], B. Stone-Gross et al. assign scores to rogue ASNs based on the amount of events involving hosts engaged in phishing, spamming, hosting drive-by download malware, or botnet traffic. In [6] and [7], the authors use visualization to track security incidents and malware events drawn from blacklist databases, and [8] explores ASNs providing transit for malware ASNs.

ASN GRAPH

An Autonomous System Number (ASN) identifies every globally routable network on the Internet. An Autonomous System (AS) represents a collection of IPv4 and IPv6 network prefixes administered by the same entity and sharing a common routing policy. In practice, an AS announces the prefixes under its authority or on behalf of their customers to its upstream providers and peer ASNs. The routers that receive these updates will use them to update their routing tables, which are used to make routing decisions. Depending on their policy will propagate the announcement to their peers and providers.

The BGP table is the accumulation of all announced prefixes with their reachability information (AS paths). An AS path is a sequence of ASNs through which an announced prefix can be reached [9]. The BGP table is not only important for packet forwarding and loop detection on every Internet router, it is also very useful to study the evolution of the Internet from a topology and security threat perspective. For that, we need to build an AS graph which represents the interconnections between peering ASNs.

In this study, we built an AS graph using publicly available data sources. Our primary source is the RouteViews data from University of Oregon [10], which provides a global BGP data by collecting BGP, updates and dumps from hundreds of Autonomous Systems worldwide. For our measurements we augmented our dataset with BGP data from all the routers we operate in the OpenDNS global network of 23 data centers.



University of Oregon Route Views Project

Advanced Network Technology Center
University of Oregon

ANNOUNCEMENT: bgpmon+routeviews testbed
ANNOUNCEMENT: CERT routeviews mirror
ANNOUNCEMENT: perth collector
MAINTENANCE: route-views.kixp.routeviews.org renumber
MAINTENANCE: route-views.eqix.routeviews.org router-id updated

- **Introduction and Goals**

The University's Route Views project was originally conceived as a tool for Internet operators to obtain real-time information about the global routing system from the perspectives of several different backbones and locations around the Internet. Although other tools handle related tasks, such as the various Looking Glass Collections (see e.g. NANOG, or the DTI NSPIXP-2 Looking Glass), they typically either provide only a constrained view of the routing system (e.g., either a single provider, or the route server) or they do not provide real-time access to routing data.

While the Route Views project was originally motivated by interest on the part of operators in determining how the global routing system viewed *their* prefixes and/or AS space, there have been many other interesting uses of this Route Views data. For example, NLANR has used Route Views data for AS path visualization (see also NLANR), and to study IPv4 address space utilization (archive). Others have used Route Views data to map IP addresses to origin AS for various topological studies. CAIDA has used it in conjunction with the NetGeo database in generating geographic locations for hosts, functionality that both CoralReef and the Skitter project support.

Other analyses using route-views data include:

Figure 15. RouteViews website.

Other valuable data sources that are useful for studying the IP, BGP prefix and ASN landscapes are the CIDR report [11] and Hurricane Electric Internet Services website [12].

BUILDING THE ASN GRAPH

The BGP data is collected in MRT format as described in rfc6396. An example MRT entry in text format looks like this:

TABLE_DUMP2|1392422403|B|96.4.0.55|11686|**67.215.94.0/24**|**11686 4436 2914 36692**|IGP|96.4.0.55|0|0||NAG||

We mark the fields that are of interest to us in red. In this entry, 67.215.94.0/24 is an example network prefix, and 11686 4436 2914 36692 is the associated AS path. The ASN that appears at the end of the AS is the origin ASN of the prefix. In this case AS 36692 is originating the prefix, the origin AS is typically the owner of the prefix or announcing it on behalf of their customer.

The AS path reveals how the most right AS reaches the prefix announced by the most left, the origin, AS. In this example it shows that AS 11686 relies on AS 4436, who then relies on 2914 to reach 36692. Not only does the AS path reveal useful topology information, it can also be used to determine business relationship between each of the ASNs. For example in this case it’s likely that 36692 (OpenDNS) is a customer of 2914 (NTT America).

As described above, we will use the AS path data to build a directed graph, where an ASN is denoted by a node and there is a directed edge between an ASN and every one of its upstream ASNs. For example, in the BGP table entry above, 36692 is the origin ASN for 67.215.94.0/24, and 2914 is an upstream ASN of 36692 (the last ASN before reaching the origin ASN when packets are traveling towards an IP in the origin ASN), therefore that entry can be graphically represented as follows:

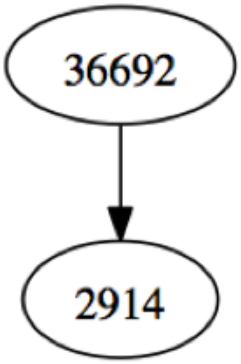


Figure 16. Graph representation of an entry of the BGP table.

An alternative method to build the AS graph is to use the entire AS path on every prefix entry of the BGP table. In this case, from the example above, we can build the following edges in the graph: **36692->2914, 2914->4436, 4436->11686.** The AS graph is built by parsing the BGP table line by line. Since many ASNs announce more than one prefix and we have data from hundreds of viewpoints on the Internet, this provides us with hundreds of paths to a single Origin AS. By assigning weights to each edge we can predict the usage of edge.

In the directed AS graph, an AS node can have incoming and/or outgoing edges. The outgoing edges point to upstream ASNs and incoming edges originate from downstream ASNs. Below, we define a few terms describing the AS graph nodes from a directed graph topological perspective [13].

A *source ASN* is an ASN that has only outgoing edges and no incoming edges, i.e. the ASN has only upstream ASNs that it relies upon for connectivity and for propagating its prefix announcements. A *leaf ASN* is a special case where an ASN has a single outgoing edge and no incoming edge. This is often described as *"stub"* ASN in the BGP routing terminology.

We define a set of ASNs that are *source* ASNs (or *leaves*) and who share the same parent(s) (upstream ASNs) as *sibling ASNs*. For clarity, we will use the more intuitive term "*peripheral*" ASNs to denote *source* ASNs for the remainder of this paper.

The BGP table/ASN graph is a dynamic entity and always changing as new prefixes are announced, old prefixes are withdrawn, new ASNs are introduced and start advertising prefixes, while others cease to exist and withdraw all their prefixes. Most common changes are probably new AS relations, new peers or previously unseen relations.
This dynamic state can be the result of multiple factors: intentional technical and business decisions, human errors, hardware faults, route hijacking, etc. By parsing the entries of the BGP table, we can extract two types of useful data: the upstream and downstream ASNs of every ASN, and IP to ASN maps (via prefix to ASN mapping). For this, we can load the prefix and the origin ASN data into a radix tree. With the radix tree, (given an IP as input) we can quickly find the best matching prefix, and consequently, matching ASN.

Alternatives are to use services like BGPmon.net (e.g whois -h whois.bgpmon.net 8.8.8.8), Team Cymru IP to ASN mapping [14], GeoIPASNum.dat from maxmind [15], or http://ipinfo.io/ (e.g. curl ipinfo.io/8.8.8.8/org returns the AS number and AS name of Google Inc.). In this study, we discuss interesting patterns in the AS graph topology – typically, suspicious peripheral ASNs that are siblings, i.e., they share common parents (upstream ASNs) in the AS graph. By clustering peripheral nodes in the AS graph by country, we found that certain peripheral sibling ASNs in a few countries have been delivering similar suspicious campaigns.

USE CASE 1: SUSPICIOUS SIBLING PERIPHERAL ASNs

During manual investigations of suspicious domains and IPs that we detected in our traffic, we observed several cases of sibling peripheral ASNs that are hosting similar malware payloads. In this section, we will describe one such use case.

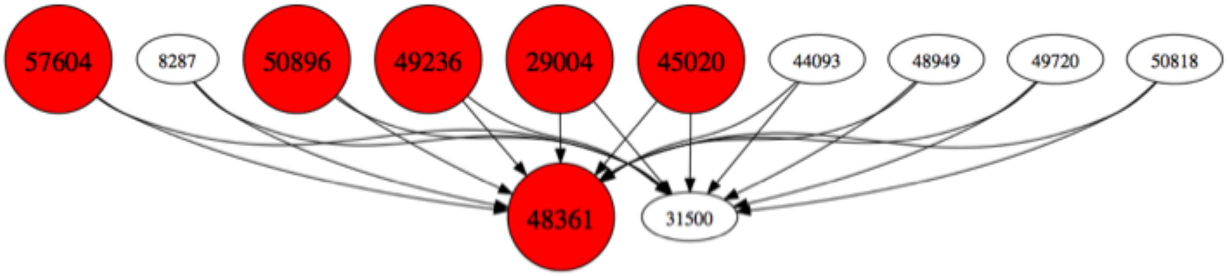


Figure 17. Malicious ASN subgraph.

In the Figure above, we show the snapshot of a suspicious ASN subgraph taken on January 8th, 2014, consisting of 10 sibling peripheral ASNs (57604, 8287, 50896, 49236, 29004, 45020, 44093, 48949, 49720, 50818) sharing 2 upstream ASNs (48361 and 31500). We color the ASNs that were hosting malicious payloads in red. The malicious payload is identified by some AVs as Trojan-Downloader.Win32.Ldmon.A [16][17] and described as a Trickler [18]. Notice that most of these peripheral ASNs are small scale with one single prefix as Table 5 shows:

Table 5. Sibling peripheral ASNs prefixes.

ASN	No of prefixes	Prefixes
57604	1	91.233.89.0/24
8287	3	91.213.72.0/24 91.213.93.0/24 91.217.162.0/24
50896	5	195.78.108.0/23 91.198.127.0/24 91.200.164.0/22 91.201.124.0/22 91.216.3.0/24
49236	1	62.122.72.0/23
29004	1	195.39.252.0/23
45020	1	194.29.185.0/24
44093	1	193.243.166.0/24
48949	1	95.215.140.0/22
49720	1	194.242.2.0/23
50818	1	194.126.251.0/24

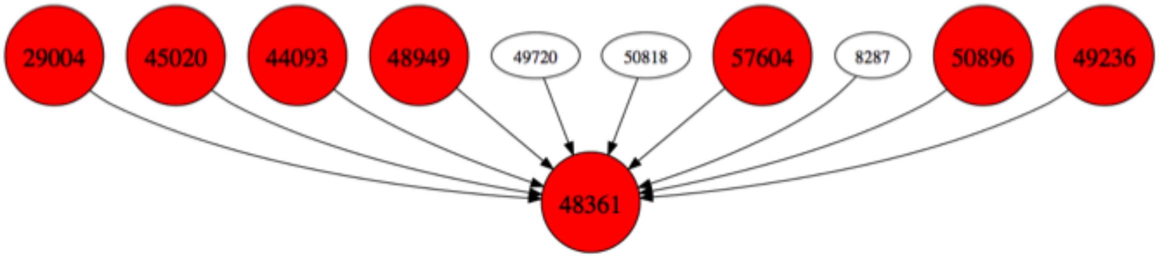


Figure 18. Malicious ASN subgraph after it evolved 6 weeks later.

In Figure 18, we show the same subgraph 6 weeks later, on February 21st. Notice the change in subgraph topology: more leaves started hosting the same suspicious payloads (via new resolving domains or directly on the IPs). Additionally, AS31500 detached itself from the leaves by ceasing to forward their prefix announcements.

We observed that a large pool of contiguous IPs in /23 or /24 prefixes of these ASNs were hosting the same aforementioned type of payload. In most cases, the payload URLs were live on the entire range of IPs before any domains were hosted on them. Furthermore, the IPs were set up with the same server infrastructure. For instance, we took a random sample of 160 live IPs in this subgraph.

In this sample, 50 IPs had a similar nmap fingerprint:

```
22/tcp   open  ssh        OpenSSH 6.2_hpn13v11 (FreeBSD 20130515; protocol 2.0)
8080/tcp open  http-proxy 3Proxy http proxy
Service Info: OS: FreeBSD
```

and 108 IPs shared the following fingerprint:

```
22/tcp open  ssh     OpenSSH 5.3 (protocol 1.99)
80/tcp open  http?
```

In total, this subgraph featured 3100+ malware domains on 1020+ malware hosting IPs, and it is clear this IP infrastructure across multiple ASNs was set up in bulk and in advance to deliver the same rogue campaign [17].

USE CASE 2: Detecting sibling Autonomous systems by looking at BGP outages

In the previous section we described sibling autonomous systems are one ore more Autonomous systems that are under control by the same organization and possibly share the same infrastructure. One way to find these sibling Autonomous systems is to look at the upstream relations as typically all siblings share the same upstream provider(s). The problem with this approach is that it will not always work as expected. For example some of the larger service providers such as Level3, NTT, GTT, etc., have many customers located throughout the world so by looking only at the common upstream providers won’t provide us with enough granular information to correctly determine siblings.

Since the sibling networks we are interested in are those that are under the control of one entity and often colocate in the same facilities and could even share hardware, we could at least search for risk sharing properties. The next section describes a new novel approach of detecting sibling ASNs with a high degree of certainty.

Using BGP outages to detect sibling Autonomous systems.

BGP, the routing protocol used on the Internet, uses primarily two types of message to advertise network reachability information: update messages to announce a new path for one or more prefixes, and withdrawal messages to inform BGP speakers that a certain prefix can no longer be reached.

When looking for BGP withdrawal messages, and the frequency of these withdrawals for a certain prefix, we can detect global outages for the prefix. For example, when a large number of BGP speakers see a BGP withdrawal message for 208.67.222.0/24 we can assume that the prefix is no longer reachable, which means there would be an outage and the hosts in this network would be unreachable. The next step is to look for new BGP update messages that provide a new path for 208.67.222.0/24, indicating the prefix is reachable again. With this data, we know exactly how long a prefix was unreachable.

For this research project, our hypothesis was that sibling autonomous systems are very closely related and often share the same servers, hardware, collocation facilities and Internet service providers. To test this hypothesis we look at the outage pattern for Autonomous System, and find Autonomous Systems that have the exact same outage pattern, i.e. the exact same outage start and stop time for one or more of prefixes in that AS.

To test this hypothesis we partnered with BGPmon.net, a BGP monitoring service. Using their BGP outage detection system and the historical outage data collected over the last few years, we compared outages for different Autonomous systems and tested our hypothesis.

This approach is unique as it provides more granular insight into the relationships between Autonomous systems as compared to looking at just peering relationships. This approach results in a set of Autonomous systems that share, with a high degree of certainty, shared risk–which means they are likely located in close proximity of each other. The same could in theory be achieved by active ping or traceroute measurements, however in practice it’s impossible to scale this to every possible network (500,000 prefixes) with the same amount of time granularity. So by leveraging the routing control protocol for the Internet, we can scale this much more effectively and operate in stealth mode for both IPv4 and Ipv6.

Illustrative Example

Using the method described above we start searching for prefixes and their corresponding autonomous systems with similar outages. We looked at outages where the start and end time of the outage was the same as the outage for AS we provided as input. As a second heuristic, we only looked at Autonomous systems that had 3 or more similar outages.

Just as the example in the previous sections, we will focus on the potential siblings for the following Autonomous systems: 57604, 8287, 50896, 49236, 29004, 45020, 44093, 48949, 49720, and 50818. We looked at the outage data between June 1st, 2013 and June 1st, 2014.

In this example we will focus on AS57604 (PE Ivanova Yuliya Geraldovna), which as can be seen in Figure 18, receives transit exclusively from 48361 and has a number of sibling Autonomous Systems. We’ll look at one of the siblings, AS29004 and compare the outages detected for both siblings as well as the upstream provider AS48361.
The table below compares a subset of the outages for the three networks between June 1st, 2013 and October 1st, 2013. The outage table shows there are eighteen unique events where AS57604 became unreachable for at least 60 seconds. Its sibling AS29004 has the exact same outage pattern. The upstream network for both of these networks, AS48361, had only one outage on August 31st - this outage also affected both of the downstream networks.

Table 6. List of outages.

ISP 48361	AS57604 91.233.89.0/24	AS29004 195.39.252.0/23
no outage	down for 35 minutes 2013-07-12 18:53 - 2013-07-12 19:28	down for 36 minutes 2013-07-12 18:53 - 2013-07-12 19:29
no outage	down for 497 minutes 2013-07-12 21:33 - 2013-07-13 05:50	down for 497 minutes 2013-07-12 21:33 - 2013-07-13 05:50
no outage	down for 479 minutes 2013-07-22 21:57 - 2013-07-23 05:56	down for 479 minutes 2013-07-22 21:57 - 2013-07-23 05:56
no outage	down for 33 minutes 2013-07-23 18:51 - 2013-07-23 19:24	down for 33 minutes 2013-07-23 18:51 - 2013-07-23 19:24
no outage	down for 63 minutes 2013-07-29 04:54 - 2013-07-29 05:57	down for 63 minutes 2013-07-29 04:54 - 2013-07-29 05:57
no outage	down for 155 minutes 2013-07-31 22:37 - 2013-08-01 01:12	down for 155 minutes 2013-07-31 22:37 - 2013-08-01 01:12
no outage	down for 6 minutes 2013-08-01 03:00 - 2013-08-01 03:06	down for 6 minutes 2013-08-01 03:00 - 2013-08-01 03:06
no outage	down for 7 minutes 2013-08-05 18:51 - 2013-08-05 18:58	own for 7 minutes 2013-08-05 18:51 - 2013-08-05 18:58
no outage	down for 8 minutes 2013-08-09 21:01 - 2013-08-09 21:09	down for 8 minutes 2013-08-09 21:01 - 2013-08-09 21:09
no outage	down for 13 minutes 2013-08-12 08:05 - 2013-08-12 08:18	down for 13 minutes 2013-08-12 08:05 - 2013-08-12 08:18
no outage	down for 237 minutes 2013-08-15 10:15 - 2013-08-15 14:12	down for 237 minutes 2013-08-15 10:15 - 2013-08-15 14:12
no outage	down for 520 minutes 2013-08-19 21:26 - 2013-08-20 06:06	down for 520 minutes 2013-08-19 21:26 - 2013-08-20 06:06
down for 11 minutes 2013-08-31 18:39 - 2013-08-31 18:50	down for 11 minutes 2013-08-31 18:39 - 2013-08-31 18:50	down for 11 minutes 2013-08-31 18:39 - 2013-08-31 18:50
no outage	down for 11 minutes 2013-09-12 04:33 - 2013-09-12 04:44	down for 11 minutes 2013-09-12 04:33 - 2013-09-12 04:44
no outage	down for 12 minutes 2013-09-12 10:33 - 2013-09-12 10:45	down for 12 minutes 2013-09-12 10:33 - 2013-09-12 10:45
no outage	down for 86 minutes 2013-09-24 08:02 - 2013-09-24 09:28	down for 86 minutes 2013-09-24 08:02 - 2013-09-24 09:28
no outage	down for 46 minutes 2013-09-26 12:36 - 2013-09-26 13:22	down for 46 minutes 2013-09-26 12:36 - 2013-09-26 13:22
no outage	down for 46 minutes 2013-09-28 16:19 - 2013-09-28 17:05	down for 46 minutes 2013-09-28 16:19 - 2013-09-28 17:05

The fact that there are so many overlapping outages between AS57604 and AS29004 is unique and further proves there is a close relationship between these two sibling autonomous systems listed earlier. The table below lists all overlapping outages between each of the sibling autonomous systems as well as the upstream provider AS48361.

Table 7. Overlapping outages between sibling ASNs.

	57604	8287	50896	49236	29004	45020	44093	48949	49720	50818	48361
57604	x	20	17	12	22	16	11	24	20	13	5
8287	20	x	41	15	17	17	15	18	18	15	5
50896	17	41	x	17	16	17	18	19	16	18	7
49236	12	15	17	x	8	15	13	8	12	17	3
29004	22	17	16	8	x	12	22	28	18	9	6
45020	16	17	17	15	12	x	12	12	12	15	4
44093	11	15	18	13	22	12	x	16	10	13	6
48949	24	18	19	8	28	12	16	x	20	9	8
49720	20	18	16	12	18	12	10	20	x	10	4
50818	13	15	18	17	9	15	13	9	10	x	4
48361	5	5	7	3	6	4	6	8	4	4	x

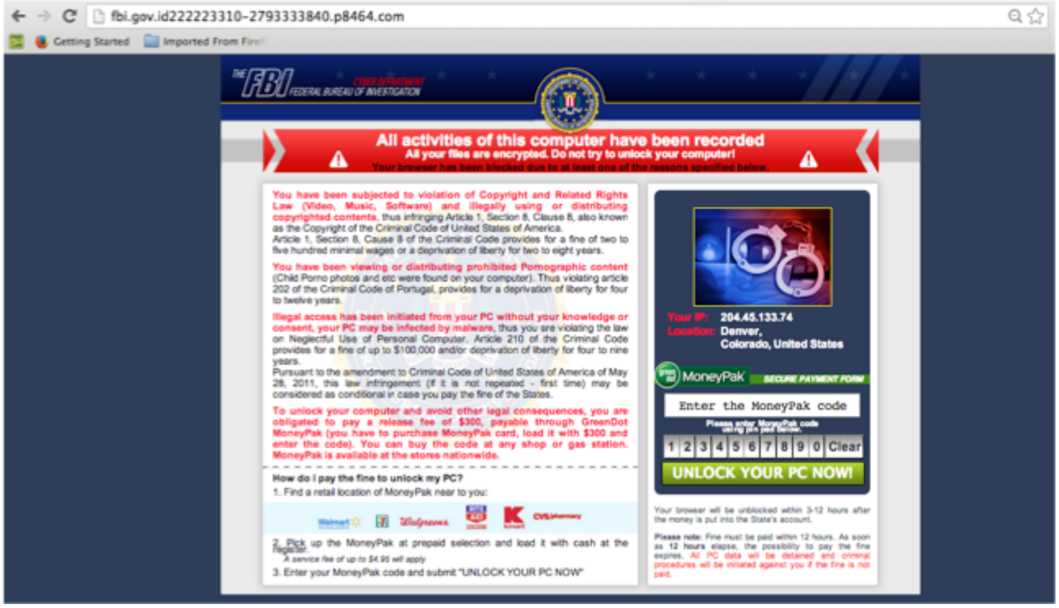
Looking at the results in the table above it is clear that there is a high degree of duplicate outages between all sibling Autonomous Systems and to some degree between the sibling Networks hosting the malware and the upstream AS48361.

When correlating the outage results with the earlier results in use case 1, we can see that there is indeed a strong relationship between all of the sibling Autonomous systems we found earlier. Not only do we now know they share a common upstream provider, we also know that there is a high degree of risk sharing between the networks. It is to be expected that when an outage affects the upstream ASN, one or more of the downstream networks will be affected as well, especially if the downstream provider is single-homed and relies solely on this upstream provider for network connectivity. However in our data there are many cases where there is no outage for the upstream provider, while there are sometimes hour-long outages for the downstream networks of which the timelines overlap exactly up to the minute.

The conclusions we can draw from this are that the set of autonomous systems we looked at most likely rely on the same infrastructure for connectivity. Normally an outage by a service provider may cause an outage for some customers, but typically only for those in a specific geographic location. The fact that there is so much correlation between the Autonomous systems we looked at is a strong indicator that they could be operated by the same organization, are in the same physical location, and could even share a joint routing infrastructure. So even though typically each Autonomous systems has its own infrastructure in terms of routing, switching and compute, our data indicates there are strong indicators that these autonomous systems could well be operated by the same organization or even run on the same hardware.

USE CASE 3: ROGUE ASN DE-PEERED OR GONE STEALTH

In this section, we discuss one case among many we observed of rogue peripheral ASNs that serve various malware content. In this example, it is AS48031, XSERVER-IP-NETWORK-AS PE Ivanov Vitaliy Sergeevich 86400 that had a single upstream provider AS15626. AS48031 has been hosting browser-based ransomware, porn sites, spam, and radical forums.


fbi.gov.id222223310-2793333840.p8464.com
Getting Started
Imported From Fire
THE FBI FEDERAL BUREAU OF INVESTIGATION
CYBER DEPARTMENT

All activities of this computer have been recorded
All your files are encrypted. Do not try to unlock your computer!
Your browser has been blocked due to at least one of the reasons specified below.

You have been subjected to violation of Copyright and Related Rights Law (Video, Music, Software) and illegally using or distributing copyrighted contents, thus infringing Article 1, Section 8, Clause 8, also known as the Copyright of the Criminal Code of United States of America. Article 1, Section 8, Cause 8 of the Criminal Code provides for a fine of two to five hundred minimal wages or a deprivation of liberty for two to eight years.

You have been viewing or distributing prohibited Pornographic content (Child Porno photos and etc were found on your computer). Thus violating article 202 of the Criminal Code of Portugal, provides for a deprivation of liberty for four to twelve years.

Illegal access has been initiated from your PC without your knowledge or consent, your PC may be infected by malware, thus you are violating the law on Neglectful Use of Personal Computer. Article 210 of the Criminal Code provides for a fine of up to $100,000 and/or deprivation of liberty for four to nine years. Pursuant to the amendment to Criminal Code of United States of America of May 28, 2011, this law infringement (if it is not repeated - first time) may be considered as conditional in case you pay the fine of the States.

To unlock your computer and avoid other legal consequences, you are obligated to pay a release fee of $300, payable through GreenDot MoneyPak (you have to purchase MoneyPak card, load it with $300 and enter the code). You can buy the code at any shop or gas station. MoneyPak is available at the stores nationwide.

How do I pay the fine to unlock my PC?

1. Find a retail location of MoneyPak near to you:

2. Pick up the MoneyPak at prepaid selection and load it with cash at the register.
   A service fee of up to $4.95 will apply
3. Enter your MoneyPak code and submit "UNLOCK YOUR PC NOW"

Your IP: 204.45.133.74
Location: Denver, Colorado, United States

MoneyPak
SECURE PAYMENT FORM
Enter the MoneyPak code
Please enter MoneyPak code using pin pad below.
1 2 3 4 5 6 7 8 9 0 Clear
UNLOCK YOUR PC NOW!

Your browser will be unblocked within 3-12 hours after the money is put into the State's account.

Please note: Fine must be paid within 12 hours. As soon as 12 hours elapse, the possibility to pay the fine expires. All PC data will be detained and criminal procedures will be initiated against you if the fine is not paid.

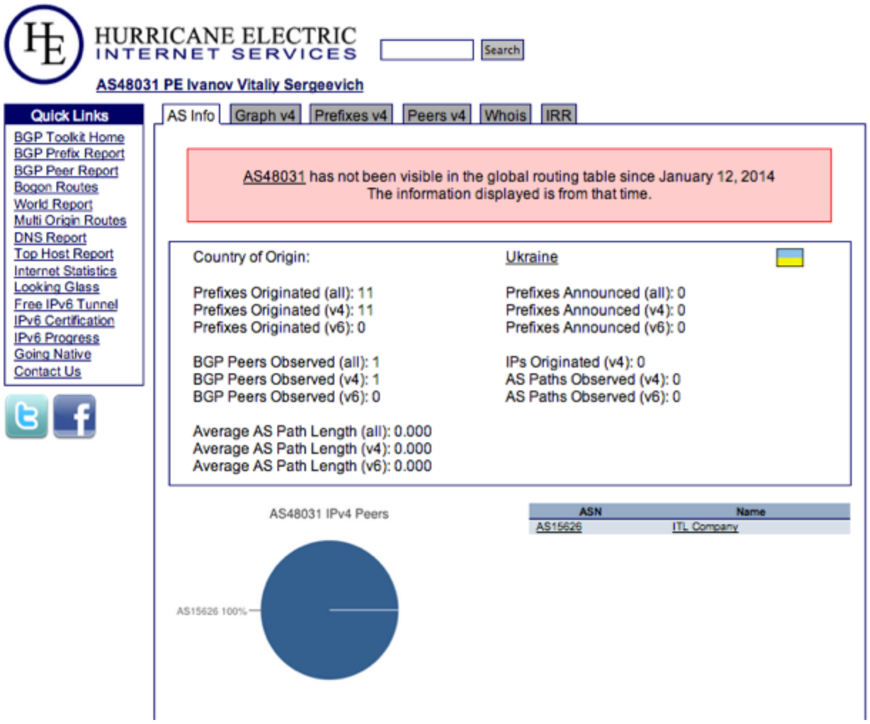
Figure 19. Browlock web page.

Browser-based ransomware or “browlock” is a rudimentary ransomware that consists of an HTML page that loads when the user visits the browlock domain. It locks the browser screen (through HTML or JavaScript code) and demands payment, supposedly for either possession of illegal material or usage of illegal software [19]. This is more of a scam than real ransomware (which corrupts or encrypts user’s data), because the browlock alert can be neutralized by simply killing the browser task. Despite its simplicity, browlock has been around for a few years targeting several countries, and seems to be generating profit for the criminals. Browlock has been delivered by dedicated (domains specifically registered for malicious intent) as well as compromised domains.

Table 8. Prefixes announced by AS48031 in Fall 2013.

Prefixes
176.103.48.0/20
193.169.86.0/23
193.203.48.0/22
193.30.244.0/22
194.15.112.0/22
196.47.100.0/24
91.207.60.0/23
91.213.8.0/24
91.217.90.0/23
91.226.212.0/23
91.228.68.0/22
93.170.48.0/22
94.154.112.0/20

In Table 8., we show the prefixes announced by AS48031 in the fall of 2013. A few months later, in January 2014, AS48031 stopped advertising prefixes and disappeared from the global routing table as Figure 20 shows.



AS48031 PE Ivanov Vitaliy Sergeevich

AS48031 has not been visible in the global routing table since January 12, 2014
The information displayed is from that time.

Country of Origin: Ukraine

Prefixes Originated (all): 11
Prefixes Originated (v4): 11
Prefixes Originated (v6): 0

Prefixes Announced (all): 0
Prefixes Announced (v4): 0
Prefixes Announced (v6): 0

BGP Peers Observed (all): 1
BGP Peers Observed (v4): 1
BGP Peers Observed (v6): 0

IPs Originated (v4): 0
AS Paths Observed (v4): 0
AS Paths Observed (v6): 0

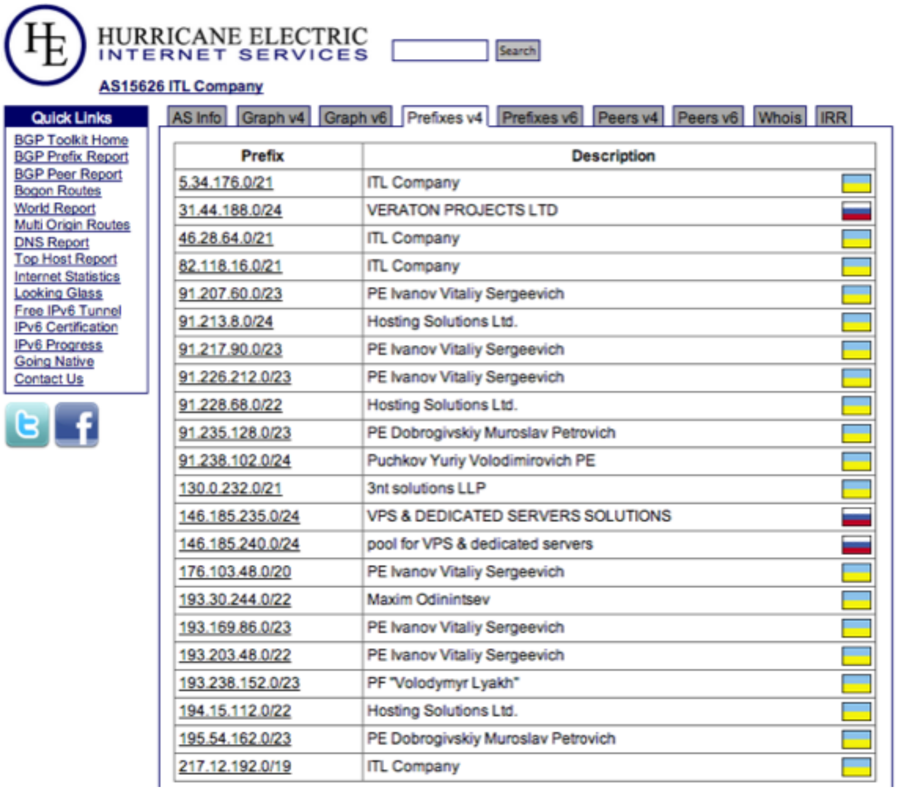
Average AS Path Length (all): 0.000
Average AS Path Length (v4): 0.000
Average AS Path Length (v6): 0.000

ASN	Name
AS15626	ITL Company

Figure 20. AS48031 disappears off the global BGP routing table.

However, those prefixes did not actually disappear, and AS48031's only parent in the AS graph, its upstream peer AS15626, took over announcing them as Figure 21 shows. The rogue IPs in those prefixes continued to host malware content.

The question remains whether AS15626 had been abused by its downstream client AS48031 to host malware, and if it acted responsibly by ceasing to announce those prefixes when it took notice of the malicious content on AS48031 prefixes, *or* if both AS48031 and AS15626 are complicit in hosting malware, and AS15626 is simply being evasive by hiding AS48031 from the global routing table and yet keeping connectivity to the rogue IPs by announcing their prefixes. There are several such suspicious cases occurring on the BGP routing space.



Prefix	Description	
5.34.176.0/21	ITL Company	🇺🇦
31.44.188.0/24	VERATON PROJECTS LTD	🇷🇺
46.28.64.0/21	ITL Company	🇺🇦
82.118.16.0/21	ITL Company	🇺🇦
91.207.60.0/23	PE Ivanov Vitaliy Sergeevich	🇺🇦
91.213.8.0/24	Hosting Solutions Ltd.	🇺🇦
91.217.90.0/23	PE Ivanov Vitaliy Sergeevich	🇺🇦
91.226.212.0/23	PE Ivanov Vitaliy Sergeevich	🇺🇦
91.228.68.0/22	Hosting Solutions Ltd.	🇺🇦
91.235.128.0/23	PE Dobrogivskiy Muroslav Petrovich	🇺🇦
91.238.102.0/24	Puchkov Yuriy Volodimirovich PE	🇺🇦
130.0.232.0/21	3nt solutions LLP	🇺🇦
146.185.235.0/24	VPS & DEDICATED SERVERS SOLUTIONS	🇷🇺
146.185.240.0/24	pool for VPS & dedicated servers	🇷🇺
176.103.48.0/20	PE Ivanov Vitaliy Sergeevich	🇺🇦
193.30.244.0/22	Maxim Odinintsev	🇺🇦
193.169.86.0/23	PE Ivanov Vitaliy Sergeevich	🇺🇦
193.203.48.0/22	PE Ivanov Vitaliy Sergeevich	🇺🇦
193.238.152.0/23	PF "Volodymyr Lyakh"	🇺🇦
194.15.112.0/22	Hosting Solutions Ltd.	🇺🇦
195.54.162.0/23	PE Dobrogivskiy Muroslav Petrovich	🇺🇦
217.12.192.0/19	ITL Company	🇺🇦

Figure 21. Former prefixes of AS48031 now announced by upstream AS15626.

USE CASE 4: MALICIOUS SUB-ALLOCATED RANGES

In this section, we summarize a study we conducted for 5 months between Oct 2013 and Feb 2014 that consisted of monitoring rogue sub-allocated ranges on OVH [20] IP space, where these ranges are reserved by recurring suspicious customers and used to serve Nuclear Exploit kit domains. In this type of infection, visitors are lead to the Exploit landing sites through malvertising campaigns, and then malware is dropped on victims' machines (e.g. zbot). Results of the study were published in [21].

For several months, OVH IP ranges had been abused. Notably, the IPs were exclusively used for hosting Nuclear Exploit subdomains, with no other sites sharing the IPs. These IPs were reserved in small ranges from OVH Canada and set up with identical services (nmap fingerprint). Consulting ARIN's referral whois database showed the reserved ranges and customer IDs. As an evasive measure, on Feb 7th, the bad actors moved to ***besthosting.ua***, a Ukrainian hosting provider. RIPE's whois service, which covers European IP space, does not always give details on reserved ranges and customers, but the Ukrainian IPs in this case were still set up with identical services. Therefore, we flagged them as prone to serve the same Nuclear campaign. On Feb 14th, the actors moved to a Russian provider, ***pinspb.ru,*** with a similar bulk IP range setup. On Feb 22nd, they moved back to OVH, notably changing their MO: the IPs being used have been allocated and used in the past for other content. This could be an evasion technique or resource recycling.

However, although bad actors have migrated between hosting providers to host the Nuclear Exploit serving domains, they still kept the name server's infrastructure (authoritative for the Nuclear domains) on ranges reserved on OVH by the same customers, which still allows us to track them. Thanks to the great collaboration with the non-profit security research group MalwareMustDie, a large number of Nuclear Exploit domains active at the time have been taken down [22].

Subsequently, bad actors have been circulating between OVH and other hosting providers. Lately, compromised domains, especially GoDaddy domains, are being used to host Nuclear and Angler Exploit kit domains (as we will cover in Section 5).

USE CASE 5: PREDICTING MALICIOUS DOMAINS IP INFRASTRUCTURE

As part of the study described in the previous section, we have been monitoring IP ranges reserved on OVH Canada by the suspicious customer(s) who reserved the ranges hosting Nuclear EK domains. Table 3 shows the number of reserved ranges, the total number of IPs they represent and the number of IPs effectively used for malicious purposes during the months of December 2013, January, February and early March 2014. These IPs were used to host Nuclear Exploit kit domains, Nuclear domains' name servers, and browlock domains.

Table 9. IP ranges reserved by suspicious customers.

Reservation dates	No ranges	No IPs	No IPs used
Dec 1st to 31st 2013	28	136	86
Jan 1st to 31st 2014	11	80	33
Feb 1st to 28th 2014	4	28	26
Mar 1st to 20th 2014 Mar 7th Mar 10th	43 40 3	364 352 12	215 208 7

Looking at the prefixes to which these malicious reserved sub-ranges belong, we notice that all 86 ranges described in Table 9 are concentrated in 4 large OVH prefixes as Table 10 shows.

Table 10. BGP prefixes of the rogue reserved ranges.

Nb IPs	BGP prefix
388	198.50.128.0/17
128	192.95.0.0/18
80	198.27.64.0/18
12	142.4.192.0/19

We used two investigative techniques to track rogue IP ranges: the first one is to monitor sub-allocated ranges reserved by suspicious customers. The second technique is to monitor the IP's service fingerprints. Below, we review a few examples of the IP ranges used to host Nuclear Exploit domains [21]:

1) For the IPs hosted on besthosting.ua, the live IPs in the range 31.41.221.131 to 31.41.221.143 all have the same server setup (nmap fingerprint)

```
22/tcp  open  ssh     OpenSSH 5.5p1 Debian 6+squeeze4 (protocol 2.0)
80/tcp  open  http    nginx web server 0.7.67
111/tcp open  rpcbind
```

2) For the IPs hosted on pinspb.ru, the IPs in the range 5.101.173.1 to 5.101.173.10 have the following fingerprint:

```
22/tcp  open  ssh     OpenSSH 6.0p1 Debian 4 (protocol 2.0)
80/tcp  open  http    nginx web server 1.2.1
111/tcp open  rpcbind
```

3) For the IPs hosted on OVH, the IPs in the range 198.50.143.64 to 198.50.143.79 have the following fingerprint:

```
22/tcp  open     ssh          OpenSSH 5.5p1 Debian 6+squeeze4 (protocol 2.0)
80/tcp  open     http         nginx web server 0.7.67
445/tcp filtered microsoft-ds
```

The IPs used to host the name servers also had the same fingerprints [21]. Notice that initially the malware IPs in a given range used to become active in bulk and sequential order, but later as an evasion method, the bad actors started bringing them up at random, one by one or a few at a time, right when they are about to deliver the Exploit kit attack.

The combination of the two investigative techniques made it possible to predict the next attack IPs with practically no false positives. As hosting providers become more aggressive in suspending rogue customers' accounts and swifter in taking down malware IPs, and as bad actors choose hosting providers on IP space where RIRs' whois service does not always provide full information about reserved ranges and customers (e.g RIPE), the first technique might not always work. The second technique of fingerprint tracking, however, still provides accurate results when combined with other intelligence.

USE CASE 6: DETECTING MALICIOUS SUBDOMAINS UNDER COMPROMISED DOMAINS

In this section, we discuss the results of a 5-month study we conducted between February and June 2014 that followed the study of Section 3. For this project, we designed a system to preemptively detect malicious subdomains injected under compromised domains (particularly GoDaddy domains) and track their IP infrastructure. The phenomena of compromised GoDaddy domains serving malware has been around for at least 2 years [23]. The compromise can happen through at least two methods: hacking GoDaddy accounts or injecting malicious redirection scripts into vulnerable GoDaddy websites. When the compromise is successful, subdomains (third level domains) are injected under the GoDaddy domains (second level domains), and these subdomains resolve to malicious sites.

Most abused ASNs

By monitoring this threat from February to the present day, we observed that the subdomains resolve to IPs serving Exploit kit attacks (typically Nuclear [24][25] and Angler [26][27]), and also browser-based ransomware. We recorded several hundred IPs hosting these malicious subdomains over the period of the study.

We see that the top 5 abused ASNs are:

- 16276 OVH SAS
- 24961 myLoc managed IT AG
- 15003 Nobis Technology Group, LLC
- 41853 LLC NTCOM
- 20473 Choopa, LLC

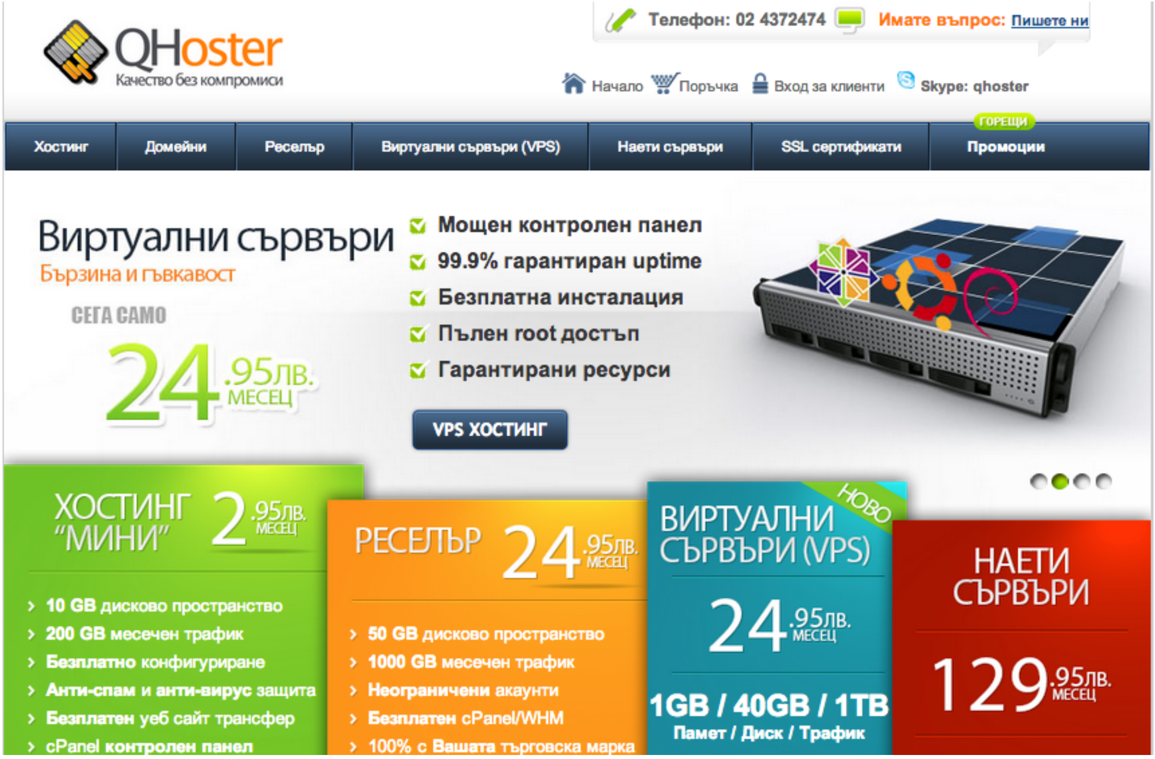
AS16276, which is OVH, hosted 18% of the total malicious IPs. In this specific case, as the abuse of OVH has been exposed through February 2014 (particularly for hosting Nuclear Exploit domains [21]), bad actors have changed their MO: they switched temporarily to other hosting providers, and started using recycled IPs (not reserved exclusively for Exploit domains). Additionally, OVH took action by suspending rogue accounts. However, by monitoring the compromised domains' campaigns, we observed that OVH was still being abused by bad actors to host malicious content. These were the general changes in bad actors' MO that we observed:

- From a domain perspective, for a while, bad actors had been abusing various ccTLDs (e.g. .pw, .in.net, .ru, etc.) facilitated by rogue or victim registrars and resellers. Then, they supplemented that approach with using compromised domains, particularly GoDaddy domains under which they inject subdomains to host Exploit kit landing urls and browlock (Notice that using compromised domains for attacks goes further back in the past for other different campaigns).
- From an IP perspective, bad actors used to bring the attack hosting IPs online in contiguous chunks, then they started bringing them up in randomized sets or one IP at a time.
- The other notable fact is that bad actors used to abuse OVH Canada (attached to ARIN) where rogue customers were reserving re-assigned small ranges (/27, /28, /29, etc.). By consulting the ARIN Rwhois database, it was possible to correlate the rogue customers with the IP ranges they reserve and therefore predict and block the IP infrastructures they set up for Exploit kit attacks. As the adversaries changed MO, this method became less effective in tracking them.
- The shift became clear when they started to more frequently use ranges on OVH's European IP space (which is attached to RIPE) as well as other European providers. Typically, we saw small gaming hosting providers being abused among other platforms.

Additionally, although the standard geolocation of OVH European IP space maps to France (FR), the attack IP ranges were reserved from OVH's server pools in various European countries (France, Belgium, Italy, UK, Ireland, Spain, Portugal, Germany, Netherlands, Finland, Czech Republic, and Russia). This clearly shows that the adversaries are diversifying their hosting assets, which provides them redundancy and evasive capabilities. Notice also that RIPE has stricter data protection laws so it would be more difficult to obtain information about customers, and that could explain the shift in hosting infrastructures by the bad actors.

More generally, we list a few of the small-scale hosting providers involved in hosting the attack subdomains. These hosting providers could either be abused, complicit with the bad actors, or simply lax about the maliciousness of the content they host. Notice the rogue providers among these will often switch prefixes by dropping dirty ones and reserving new ones from the backbone providers they are attached to.

- http://king-servers.com/en/ This hoster has been observed to host Exploit kit domains (Angler, Styx), porn, dating sites, pharma sites [28][29]. It was also described by a comment on Web Of Trust as "Offers bulletproof hosting for Russian-Ukrainian criminals (malware distributors, etc.)" [30].
- http://evrohoster.ru/en/ hosted browlock through redirections from porn sites [31].
- http://www.xlhost.com/ hosted Angler EK domains [32]
- https://www.ubiquityhosting.com/ hosted browlock.
- http://www.qhoster.bg/ hosted Nuclear EK domains.



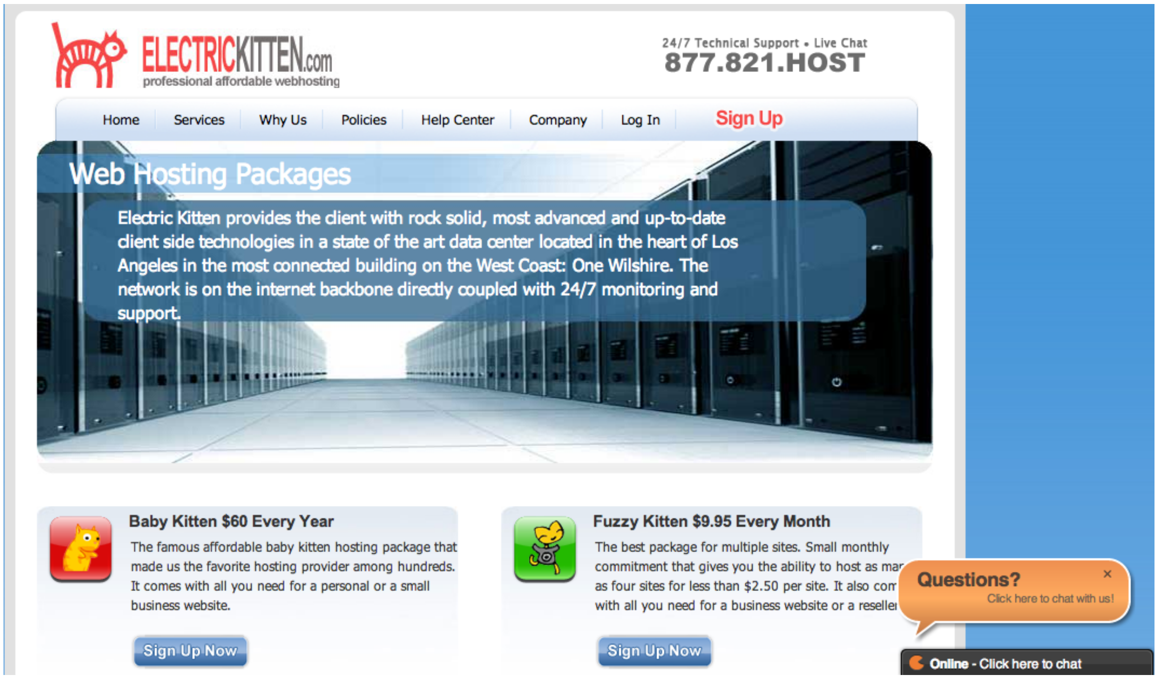
- ☑ Мощен контролен панел
- ☑ 99.9% гарантиран uptime
- ☑ Безплатна инсталация
- ☑ Пълен root достъп
- ☑ Гарантирани ресурси

- 10 GB дисково пространство
- 200 GB месечен трафик
- Безплатно конфигуриране
- Анти-спам и анти-вирус защита
- Безплатен уеб сайт трансфер
- cPanel контролен панел

- 50 GB дисково пространство
- 1000 GB месечен трафик
- Неограничени акаунти
- Безплатен cPanel/WHM
- 100% с Вашата търговска марка

Figure 22. Qhoster.bg main website.

- http://www.codero.com/
- http://www.electrickitten.com/web-hosting/


ELECTRICKITTEN.com
professional affordable webhosting
24/7 Technical Support • Live Chat
877.821.HOST
Home
Services
Why Us
Policies
Help Center
Company
Log In
Sign Up
Sign Up Now
Sign Up Now
Questions?
Click here to chat with us!
Online - Click here to chat

Web Hosting Packages

Electric Kitten provides the client with rock solid, most advanced and up-to-date client side technologies in a state of the art data center located in the heart of Los Angeles in the most connected building on the West Coast: One Wilshire. The network is on the internet backbone directly coupled with 24/7 monitoring and support.

Baby Kitten $60 Every Year

The famous affordable baby kitten hosting package that made us the favorite hosting provider among hundreds. It comes with all you need for a personal or a small business website.

Fuzzy Kitten $9.95 Every Month

The best package for multiple sites. Small monthly commitment that gives you the ability to host as mar as four sites for less than $2.50 per site. It also com with all you need for a business website or a reseller

Figure 23. electrickitten.com main website.

- http://hostink.ru/

String Analysis of Domain Names

During this study, we recorded 19,000+ malicious subdomains injected under 4200+ compromised GoDaddy 2LDs. By analyzing the strings used for the subdomains, we recorded 12,000+ different labels. We show the list of top 5 labels used; police, alertpolice, css, windowsmoviemaker, solidfileslzsr. police and alertpolice were the most common labels for hostnames serving browlock. The remaining labels were used for hostnames serving mainly Exploit kit attacks. In the Figure below, we show the frequency of number of occurrences for all used labels.

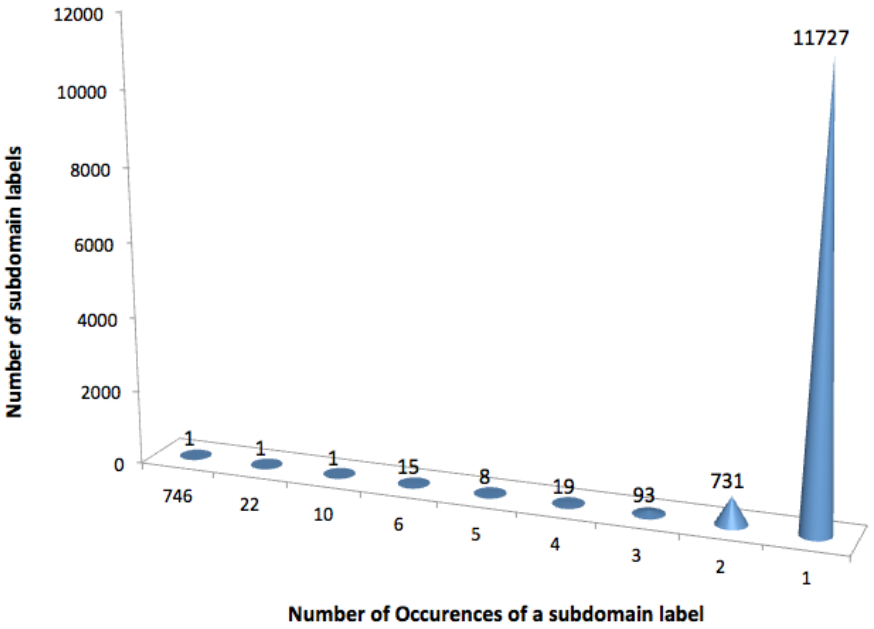


Figure 24. Frequency of number of occurrences of subdomains labels.

One label occurred 746 times (police), 1 label occurred 22 times (alertpolice), 1 label occurred 10 times (css), 15 labels occurred 6 times (windowsmoviemaker, solidfileslzsr are among them), and 11,727 distinct labels occurred a single time.

Part 3: 3D Visualization Engine

When it comes to graph visualization, there are multiple approaches to the problem however the main purpose of the engine is to analyze the *topology* of our knowledge base, therefore we need to orientate toward visualization techniques that will let the data drive the layout and not the opposite. For that very special kind of visuals, the state-of-the art generally revolves around *force-directed layouts* [33].

The general concept is fairly simple : A force system is created using the entities of the dataset. The system is then simulated inside a physics engine for a certain number of iterations and the result is an multi-dimensional arrangement (usually 2D or 3D) completely defined by the shape of the relational structure therefore highlighting hidden clusters or topological patterns that may have gone completely invisible before then.

The Fruchterman and Reingold algorithm

Discovered in 1991, the Fruchterman and Reingold layout is one of the classic force-directed layouts. The main idea is to treat vertices in the graph as *"atomic particles or celestial bodies, extering attractive and repulsive forces from one another".*

The force system operates as described in the diagram below :

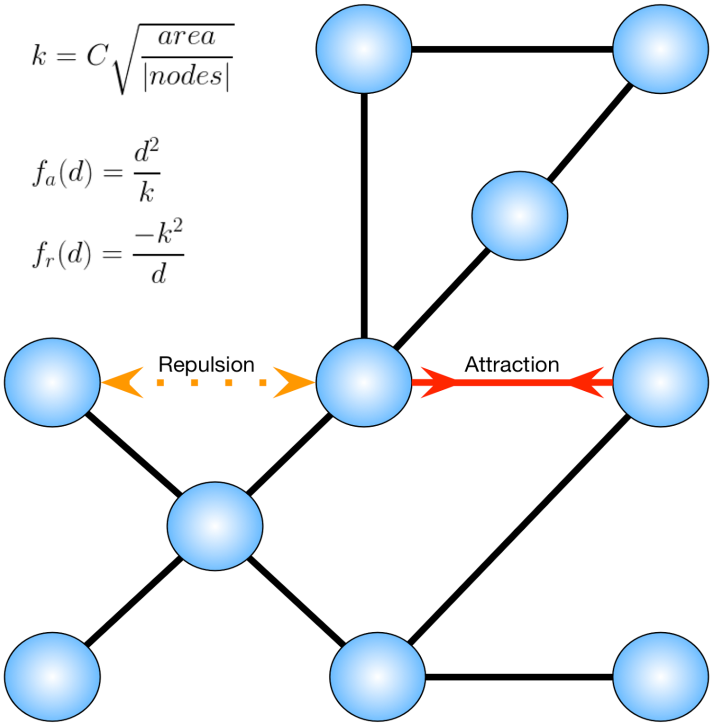


Figure 25. Force-directed system.

Without entering into too many technical details about the math supporting the model, the principle is elementary : Connected nodes attract each other and non-connected nodes repulse each other.
The attractive force ***fa(d)*** and the repulsive force ***fr(d)*** both depend on the distance between the nodes and a constant ***k*** controlling the density of the layout.
The algorithm also adds a notion of ***temperature*** which controls the displacement of the vertices. The higher the temperature, the faster the movement.
The physics represent a system inspired by electrical or celestial forces associated with a general technique called *"simulated annealing"* [34] where increasing/decreasing the temperature affects the particles thermodynamics vibration, helping them to progressively reach an equilibrium state where all the node forces become even. That state usually looks like a visually pleasing molecule-shaped layout where relational clusters will aggregate in the same areas.
This is only one of many variations of the force-directed layouts. Many other versions can be found on various papers. They will be integrated and documented in this white paper as they are implemented in the data visualization engine.

Dealing with large graphs

Being able to visualize a graph with a dozen of nodes and edges is absolutely not enough for modern day requirements. Most modern databases include millions or billions of entries. All 3D engines and particle systems have their physical limitations and force-directed layout algorithms usually increase in complexity as the size of the graph grows. Knowing those factors, how do we work around these issues ?

a. Entity grouping
One way to decrease the amount of information to process is the look at it from a higher level. Instead of dealing with entities, we can create nodes representing groups. The possibilities are endless depending on the subject we want to visualize. For instance : If we wanted to visualize the whole known universe with its planets and stars, it would make sense to structure our representation by a *fractal approach* where we would look first at galaxies then stars, planets,

continents, countries, cities (etc.) to reduce the size of the point cloud. We could then interactively decide to add more details on the fly as we move closer from a certain city. This would give access to the whole information without having to deal with all of it at once.

b. Sampling

Another interesting way to limit the size of a dataset without completely losing the fine details is to use sampling methods. We would take a certain certain percentage or a random sample of the complete dataset. The random subset could be built using a uniform or normal distribution (or any other user-defined distribution) and then more easily processed. Using the same previous universe analogy, for example we would randomly remove half of the galaxies, half of the planets/stars, half of the cities (etc.) and process the result. The data scientist has to adjust his hypotheses or assumptions based on the way the data was pruned.

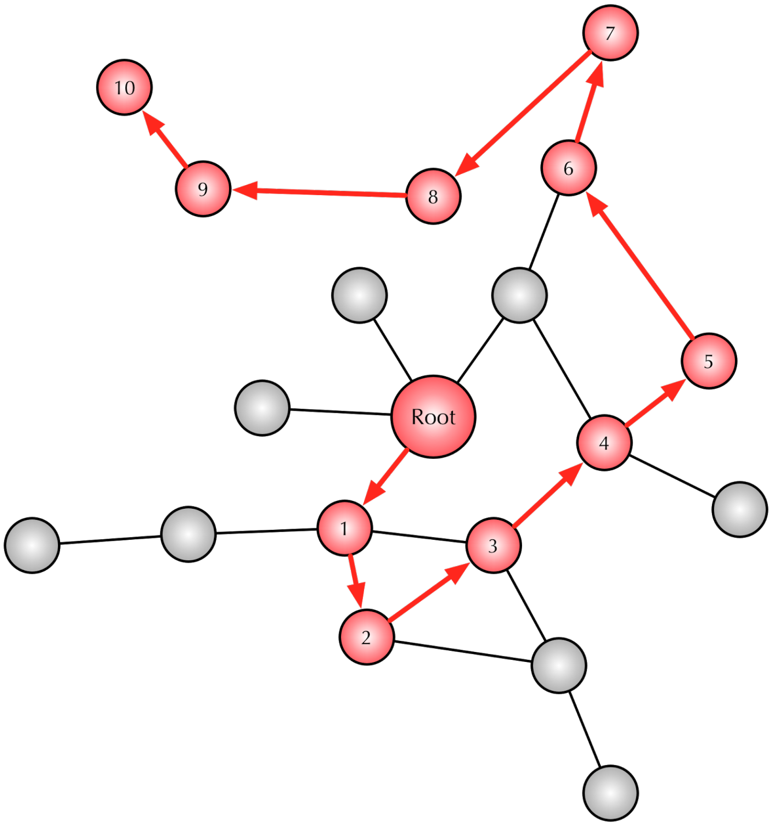


Figure 26. Random walk example.

When dealing with graphs, an easy way to take random sample of a large graph is to use a "***Random Walk"*** approach [35]. One would select random entry points in the graph and trace a random path in the graph starting from those points.

There are many ways to tweak such an exploration technique. It highly depends on the user modifications and the biases involved in the selection of the random candidates but in general a random walk helps understanding the general structure of a very large graph.

c. Parallelization

When every other pruning technique has been used, the last answer is *parallelization*. We can effectively add more computing power to a system by distributing the calculation. This can happen remotely using *"Grid Computing"* technologies or localy using the performance of multiple threads / cores / processes [36]. However, the processing algorithm needs to be rewritten to work in a parallel fashion, which is unfortunately not always completely possible.

Using the most recent graphic cards we can take advantage of efficient GPUs and distribute the calculation on their always-increasing number of cores and threads. GPUs have become insanely good at working with geometrical data (such as vectors, colors, matrices, textures or any kind of computation involving a combination of these).
Learning how to leverage GPUs (Or any parallel platform) with technologies such as OpenGL, GLSL and OpenCL (among many others) is definitely one key to unlock our theoretical barrier.

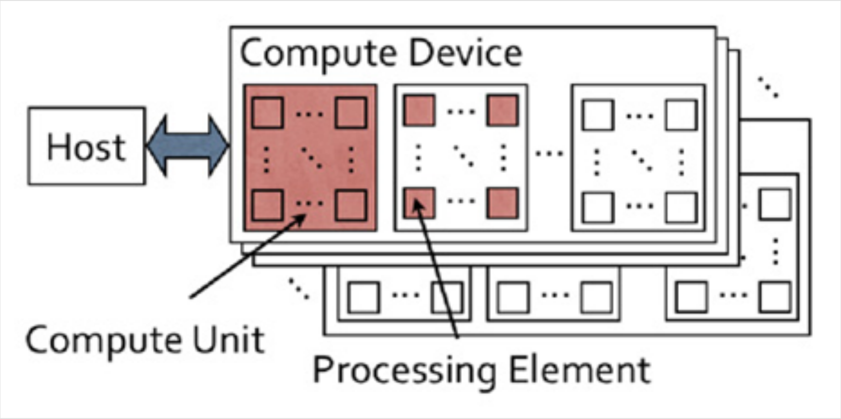


Figure 27. OpenCL architecture.

With OpenCL for example, a task can be fully or even partially distributed over several compute units. The efficiency of the whole system has then to be maximized by optimizing the different parts of the algorithm (Memory access, Instructions, Concurrency...).

CONCLUSION

In this paper, we covered a two-pronged strategy to catch malware at the DNS and IP level. First, we discussed methods to track fast flux botnets and presented a study on the zbot fast flux proxy network. Second, we proposed new methods to explore malicious IP space that enrich current reputation techniques. Known techniques assign maliciousness scores to IPs, prefixes, and ASNs based on counting volume of hosted content. In this work, we proposed to consider the topology of the AS graph, look at a granularity smaller than the BGP prefix and look at overlapping outages. In the first case, we showed cases of rogue sibling peripheral ASNs that are delivering common suspicious payloads. In the second case, we studied sub-allocated IP ranges and shed light on the MO of bad actors to abuse these allocations from providers and avoid detection. Our system provides actionable intelligence and helps preemptively detect, quarantine, and monitor or block specific rogue IP space. Finally, we presented our novel 3D visualization engine that adequately displays the use cases and offers a graph navigation and investigation tool. We demonstrated the process which transforms raw data into stunning 3D visuals. The engine features different techniques used to build and render large graph datasets: Force Directed algorithms accelerated on the GPU using OpenCL, 3D rendering and navigation using OpenGL ES, and GLSL Shaders.

REFERENCES

[1] Distributed Malware Proxy Networks, B. Porter, N. Summerlin, BotConf 2013
[2] http://labs.opendns.com/2013/12/18/operation-kelihos-presented-botconf-2013/
[3] https://zeustracker.abuse.ch/
[4] http://www.malware-traffic-analysis.net/
[5] B. Stone-Gross, C. Kruegel, K. Almeroth, A. Moser, E. Kirda, “Finding rogue networks”, Annual Comp. Security Applications Conference, ACSAC ‘09.
[6] F. Roveta, L. Di Mario, F. Maggi, G. Caviglia, S. Zanero, P. Ciuccarelli, “BURN: Baring Unknown Rogue Networks”, 8th Intl. Symposium on Visualization for Cyber Security, VizSec ‘11.
[7] T. Yu, R. Lippmann, J. Riordan, S. Boyer, “Ember: a global perspective on extreme malicious behavior”, 7th Intl. Symposium on Visualization for Cyber Security, VizSec ‘10.
[8] C. Wagner, J. Francois, R. State, A. Dulaunoy, T. Engel, G. Massen, “ASMATRA: Ranking ASs Providing Transit Service to Malware Hosters”, IEEE International Symposium on Integrated Network Management (IM 2013), 2013.
[9] A. Broido, K. Claffy, “Analysis of RouteViews BGP data: policy atoms”, Network Resource Data Management Workshop, May 2001.
[10] http://archive.routeviews.org/bgpdata/
[11] http://www.cidr-report.org/as2.0
[12] http://bgp.he.net
[13] http://en.wikipedia.org/wiki/Vertex_(graph_theory)
[14] http://www.team-cymru.org/Services/ip-to-asn.html
[15] http://dev.maxmind.com/geoip/legacy/geolite/
[16] https://www.virustotal.com/en/ip-address/5.254.120.124/information/
[17] http://pastebin.com/X83gkPY4
[18] http://telussecuritylabs.com/threats/show/TSL20130715-08
[19] http://www.f-secure.com/v-descs/trojan_html_browlock.shtml
[20] http://www.ovh.com/
[21] D. Mahjoub, “When IPs go Nuclear”, http://labs.opendns.com/2014/02/14/when-ips-go-nuclear/
[22] http://blog.malwaremustdie.org/2014/02/tango-down-of-nuclear-packs-174.html
[23] http://nakedsecurity.sophos.com/2012/11/23/hacked-go-daddy-ransomware/
[24] http://www.malware-traffic-analysis.net/2014/05/08/index.html
[25] http://www.malware-traffic-analysis.net/2014/05/13/index.html
[26] http://www.malware-traffic-analysis.net/2014/05/25/index.html
[27] http://www.malware-traffic-analysis.net/2014/06/03/index.html
[28] http://urlquery.net/report.php?id=1397035856786
[29] https://www.virustotal.com/en/ip-address/184.105.139.31/information/
[30] https://www.mywot.com/en/scorecard/king-servers.com/comment-15984778#comment-15984778
[31] https://www.virustotal.com/en/ip-address/62.75.195.244/information/
[32] http://urlquery.net/report.php?id=1399060473120
[33] http://cs.brown.edu/~rt/gdhandbook/chapters/force-directed.pdf
[34] http://en.wikipedia.org/wiki/Simulated_annealing
[35] http://en.wikipedia.org/wiki/Random_walk
[36] http://en.wikipedia.org/wiki/Grid_computing